EVALUATING SAFETY PROTOCOLS FOR MANNED-UNMANNED ENVIRONMENTS THROUGH AGENT-BASED SIMULATION

by

Jason C. Ryan

B.S. in Aerospace Engineering, University of Alabama (2007)
S.M. in Aeronautics and Astronautics, Massachusetts Institute of Technology (2011)

Submitted to the Engineering Systems Division
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Human Systems Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2014

Author ....
Engineering Systems Division
August , 2014

Certified by..
Mary L. Cummings
Visiting Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by..
R. John Hansman
T. Wilson Professor of Aeronautics and Astronautics and Engineering Systems

Certified by..
Julie A. Shah
Assistant Professor of Aeronautics and Astronautics

Accepted by ....
Munther A. Dahleh
William A. Coolidge Professor of Electrical Engineering and Computer Science
Acting Director of MIT Engineering Systems Division

EVALUATING SAFETY PROTOCOLS FOR MANNED-UNMANNED ENVIRONMENTS THROUGH AGENT-BASED SIMULATION

by

Jason C. Ryan

Submitted to the Engineering Systems Division
on August , 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Human Systems Engineering

Abstract

Recent advances in autonomous system capabilities have improved their performance sufficiently to make the integration of unmanned and autonomous vehicles systems into human-centered civilian environments a realistic near-term goal. In these systems, such as the national highway system, mining operations, and manufacturing operations, unmanned and autonomous systems will be required to interact with large numbers of other unmanned vehicle systems as well as with manned vehicles and other human collaborators. While prior research provides information on the different methods of controlling unmanned vehicles and the effects of these methods on individual vehicle behavior, it has typically focused on only a small number of unmanned systems acting in isolation. The qualities that provide the desired behavior of an autonomous system behavior in isolation may not be the same as the characteristics that lead to desirable performance while interacting with multiple heterogeneous actors. Additionally, the integration of autonomous systems may include constraints on operations that limit interactions between manned and unmanned agents. It is not clear which constraints might be most effective in promoting safe operations and how these constraints may interact with unmanned system control architectures.

Examining the effects of unmanned systems in these large, complex systems in reality would require significant capital investment and the physical construction and implementation of the unmanned vehicles of interest. Both of these aspects make empirical testing either difficult or impossible to perform and may also limit the ability of testing to fully examine all the parameters of interest in a safe and efficient manner. The objective of this thesis is the creation of a simulation environment that can replicate the behavior of the unmanned vehicle systems, manned vehicles, and human collaborators in the environment in order to enable an exploration of how parameters related to individual actor behavior and actor interactions affect performance. The aircraft carrier flight deck is chosen as an example domain, given that current operations require significant interactions between human collaborators and manned vehicles and current research addresses the development of unmanned vehicle systems for flight deck operations. Because the complexity of interactions between actors makes the creation of closed-form solutions of system behavior difficult, an agent-based modeling approach is taken. First, a list of actors and their characteristic tasks, decision-making processes, states, and parameters for current aircraft carrier flight deck operations was generated. Next, these models were implemented in an object-oriented programming language, enabling the definition of independent tasks, actors, parameters, and states. These models were later extended to incorporate features of unmanned vehicle control architectures by making minor modifications to the state, logic functions, or parameters of current agents (or tasks). This same tactic can be applied by future researchers to further pursue examinations of other influential aspects of system performance or to adapt the model to other domains.

This model, the Multi-Agent Safety and Control Simulation (MASCS), was then compared to data for current flight deck operations to calibrate and partially validate simulation outputs, first addressing an individual vehicle task before proceeding to mission tasks utilizing many vehicles at once. The MASCS model was extended to incorporate models of different unmanned vehicle control architectures and different safety protocols that affect vehicle integration. These features were then tested at different densities of mission operations on the flight deck and compositions (unmanned vs. manned) of missions in order to fully explore the interactions between variables. These results suggest that productivity on the flight deck is more heavily influenced by the safety protocols that influence vehicle integration as opposed to the types of unmanned vehicle control architecture employed. Vehicle safety is improved by increasing the number of high-level constraints on operations (e.g. separating unmanned and manned aircraft spatially or temporally), but these high-level constraints may conflict with implicit constraints that are part of crew-vehicle interactions. Additional testing explored the use of MASCS in understanding the effects of changes to the operating environment, independent of changes to unmanned vehicle control architectures and safety protocols, as well as how the simulation can be used to explore the vehicle design space. These results indicate that, at faster operational tempos, latencies in vehicle operations drive significant differences in productivity that are exacerbated by the safety protocols applied to operations. In terms of safety, a tradeoff between slightly increased vehicle safety and significant increases in the risk rate of crew activity is created at faster tempos in this environment. Lastly, the limitations and generalizability of the MASCS model for use in other Heterogeneous Manned-Unmanned Environments (HMUEs) was discussed, along with potential future work to expand the models.

Thesis Supervisor: Mary L. Cummings
Title: Visiting Professor of Aeronautics and Astronautics

Acknowledgments

I owe thanks to many different people not only for their help and support during my dissertation work, but for all the steps that got me to MIT and to this point.

First, to Missy Cummings, my dissertation advisor, thank you for pushing me every step of the way and making sure that I was doing the best possible work. I know working with me has been difficult at times, but I am incredibly lucky to have made it to MIT in time to be your student. You have taught me more about this field of research than I ever expected I could learn, and I will be forever indebted to you for allowing me to work with you. You have been a great teacher of the discipline, a great mentor for all the other aspects of my personal and professional life, and a great friend over these past five years.

To John Hansman, thank you for stepping a bit outside your typical area of research to serve on my dissertation committee. Having the outside perspective that you bring has been valuable in making me question my own assumptions and perspectives about what I am doing, what is important, and how it should be interpreted. To Julie Shah, providing the perspective from the automation's point of view has helped me avoid getting too stuck in the humanist perspective on things. Your enthusiasm in our discussions has also been wonderful to see.

To Sally, for helping me manage all the craziness surrounding my work, the work of the lab, my travels back and forth around the country, you have been a terrific help. Most especially, your assistance in finding and locating Missy and John was always of great help.

To Beth Milnes, our intrepid ESD academic administrator, your advice and encouragement in helping me navigate the muddy waters of ESD was always appreciated. You have also been a great influence and help to ESS; even though I was never an officer in the organization, your help in directing ESS, planning the retreat, and making sure that we are all still sane is more than valuable.

To ESS, for giving me a forum to think and talk about my research, for helping with Generals preparation, and helping me get out of my bubble every now and then to relax a bit, thank you.

To the labmates, the whole crazy lot of you over the past few years, I love you all. To Alex, Mark, and Kathleen, we were all here at the end of HAL, and having the two of you to keep me company these past two semesters has been great. To all the rest: Andrew, Paul, Jackie, Yves, Luca, Kris, Erin, Jamie, and Armen, thanks for all the great times, the laughs, the interesting discussions, and the support.

To my parents, Steve and Jackie, thank you for all the love and support over these many years. You've always encouraged me to succeed to the best of my abilities, even though most of the time we were never really sure what that meant and where it would take me. I know this has been a really interesting and novel experience for the both of you, but I know I could never have done it without you backing me all the way. I love you both dearly, and this all is dedicated to you.

To my loving girlfriend Kim, I'm sorry for all the stressful nights, and days, and lost weekends, and lost evenings, and preventing you from watching *Game of Thrones* and *House of Cards* and all those other things. You've been working a real job while I was still hanging out in school, but you've always been behind me 110% of the way. I love you and I always will, and I cannot tell you how much you have meant to me during this process. I cannot wait for the adventures that await us in the years to come.

Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms

A Area Separation

ABS Agent-Based Simulation

ANOVA ANalysis of VAriance

AUVSI Association of Unmanned Vehicle Systems International

CA Control Architecture

CDF Cumulative Density Function

CNA Center for Naval Analyses

DCAP Deck operations Course of Action Planner

DPHI Duration of Primary Halo Incursions

DS Dynamic Separation

DTHIA Duration of Tertiary Halo Incursions due to Aircraft

FAA Federal Aviation Administration

FD Duration of Vehicles in Fantail

GC Gestural Control

GCS Ground Control Station

GUI Graphical User Interface

HMUE Heterogeneous Manned-Unmanned Environment

HSC Human Supervisory Control

K-S Kolmogorov-Smirnov

LD Launch event Duration

LOS Line-Of-Sight

LP Launch Preparation

LSFT Landing Strip Foul Time

LT Local Teleoperation

MASCS Multi-Agent Safety and Control Simulation

MC Manual Control

ME Mission Effectiveness

NF Number assigned to Fantail

NHTSA National Highway Traffic Safety Administration

NTSB National Transportation Safety Board

NS Number assigned to Street

PDF Probability Density Function

PHI Primary Halo Incursion

PC Primary Collision warnings

RT Remote Teleoperation

SA Situational Awareness

SC Secondary Collision warnings

SD Duration of Vehicles in Street

SBSC System-Based Human Supervisory Control

SHI Secondary Halo Incursions

SME Subject Matter Expert

SP Safety Protocol

T Temporal Separation

T+A Temporal+Area Separation

TAAT Total Aircraft Active Time

TAXT Total Aircraft taXi Time

TC Tertiary Collision warnings

THI Tertiary Halo Incursions

THIA Tertiary Halo Incursions due to Aircraft

TTD Total Taxi Distance

UAV Unmanned Aerial Vehicle

UGV Unmanned Ground Vehicle

USV Unmanned Surface Vehicle

UxV Unmanned Vehicle

VBSC Vehicle-Based Human Supervisory Control

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

While unmanned and autonomous vehicle systems have existed for many years, it is only recently that these technologies have reached a sufficient maturity level for full-scale integration into human-centered work environments to be feasible. Already, integration of these vehicles into military (Office of the Secretary of Defense, 2013; Naval Studies Board, 2005) and commercial environments (Scanlon, 2009; Rio Tinto, 2014) is beginning to occur. Most previous implementations of automated and unmanned vehicles, however, have been predicated on a complete separation of unmanned vehicles from the rest of the environment (Federal Aviation Administration, 2014a). For these heterogeneous manned-unmanned systems, which include mining, warehouse operations, and aircraft carrier flight decks, this strategy of complete separation will not be feasible: these domains will require close physical coordination and interaction between the humans, manned vehicles, and unmanned or autonomous vehicles in the world. Increasing the safety of operations will require a different tactic: the implementation of safety protocols that dictate when and where human-vehicle interactions occur and tasks can be performed.

The scale and complexity of these systems implies that conducting empirically would be both time-consuming and resource intensive, assuming that the unmanned or autonomous vehicles of interest to be integrated have been constructed and are available for testing. This dissertation describes a simulation and assessment methodology constructed specifically for evaluating system design tradeoffs in these dynamic human/autonomous vehicle work environments. In particular, it explores the relationship between the safety protocols, the human-machine interface, and system performance in terms of both safety and productivity. The simulation is constructed for a single representative environment — the aircraft carrier flight deck — but the methodology can be used to construct models of other domains, allowing system designers and other stakeholders to estimate the likely outcomes of unmanned vehicle integration in their own domains.

1.1 Autonomous Robotics in Complex Sociotechnical Systems

The terms “unmanned” and “autonomous” systems are often used interchangeably to describe a variety of complex vehicle systems, which must further be differentiated from “automated” systems. The most recent Department of Defense roadmap for autonomy defines an “unmanned” aircraft as one “that does not carry a human operator and is capable of flight under remote control or autonomous programming” (Office of the Secretary of Defense, 2013). When remotely-controlled by a human operator, other applicable terms include “teleoperated” (Sheridan, 1992) and “Remotely-Piloted” or “Remotely-Operated” aircraft (Federal Aviation Administration, 2014b). In these remotely-controlled systems, a human operator is still the primary source of lower-level control inputs to the vehicle system. Alternatively, “autonomous” and semi-autonomous systems offload some of these activities to the vehicle system, allowing it “to be automatic or, within programmed boundaries, ‘self-governing’ ” (Defense Science Board, 2012). This promotes the human operator to a role of “supervisor” in which they provides instructions as to where the vehicle should travel (waypoints) rather than directly controlling steering, acceleration, and braking. The term “automated” then refers to the assignment of a singular, often repetitive, task to an individual machine.

While various *automated* technologies are common in manufacturing enterprises, autonomous or semi-autonomous robots are not commonly seen traveling through the halls of a hospital, rolling along an urban sidewalk, or flying through the skies loaded with cargo. However, the success of DARPA’s Grand Challenge (Seetharaman, Lakhotia, & Blasch, 2006) and Urban Challenge (Buehler2009, 2009) as well as the development of Google’s autonomous car (Markoff, 2010; Thrun, 2010) have demonstrated that fully-autonomous road vehicles are not far away. Unmanned aerial vehicles, using only a limited form of autonomy to communicate with and execute commands from a remote pilot, have been used for years in both U.S. military operations; in Japan, remotely piloted helicopter have been used in large numbers for years for agricultural crop dusting (Sato, 2003). The Association of Unmanned Vehicle Systems International (AUVSI) forecasts massive growth potential for unmanned systems in U.S. agriculture and public safety (Jenkins & Vasigh, 2013), and the FAA’s recent awarding of six Unmanned Aerial Vehicle (UAV) test sites should aid in the commercial deployment of such vehicles within the United States. In a very different context, a variety of work is also currently addressing the personal service robotics field, where a major focus is the development of robots to care for the growing elderly population (Roy et al., 2000), as well as a new generation of manufacturing robots that can better understand the actions of their human collaborators, improving their ability to understand human intentions in the context of work tasks (Nikolaidis & Shah, 2012).

Two primary concerns exist for the integration of unmanned and autonomous systems into these human-centered environments: safety and productivity. Autonomous vehicles must be capable of safely moving near to and interacting not only with the humans that can control or influence their behavior, but also with other bystanders (both human and manually controlled vehicles) that have no influence over them otherwise. They must also be able to do so without significantly reducing the efficiency of operations: for instance, in highway driving, autonomous cars must navigate effectively in traffic, merging in seamlessly with both human-controlled vehicles and pedestrians on or near the roadway. Aerial vehicles in public domains must adhere to the same sense-and-avoid actions that human pilots take, similar to the "rules of the road" that autonomous surface vessels must adhere to in navigating the seas amid other ships. Personal robots must attend to whatever tasks are required of them without running into, or over, their human companions or other items within the household. These environments can all be characterized as Heterogeneous Manned-Unmanned Environments (HMUEs), highlighting the fact that different types of both manned and unmanned vehicles (including humans) operate in close proximity within a given area. Figure 1-1 provides a notional representation of these domains.

While a great deal of research currently addresses the development of individual unmanned or autonomous systems, or small teams of them working together, very little research currently examines their integration into these larger HMUEs. These large-scale systems, in both size and number of active agents, are problematic to test for several reasons. First, the number of actors in

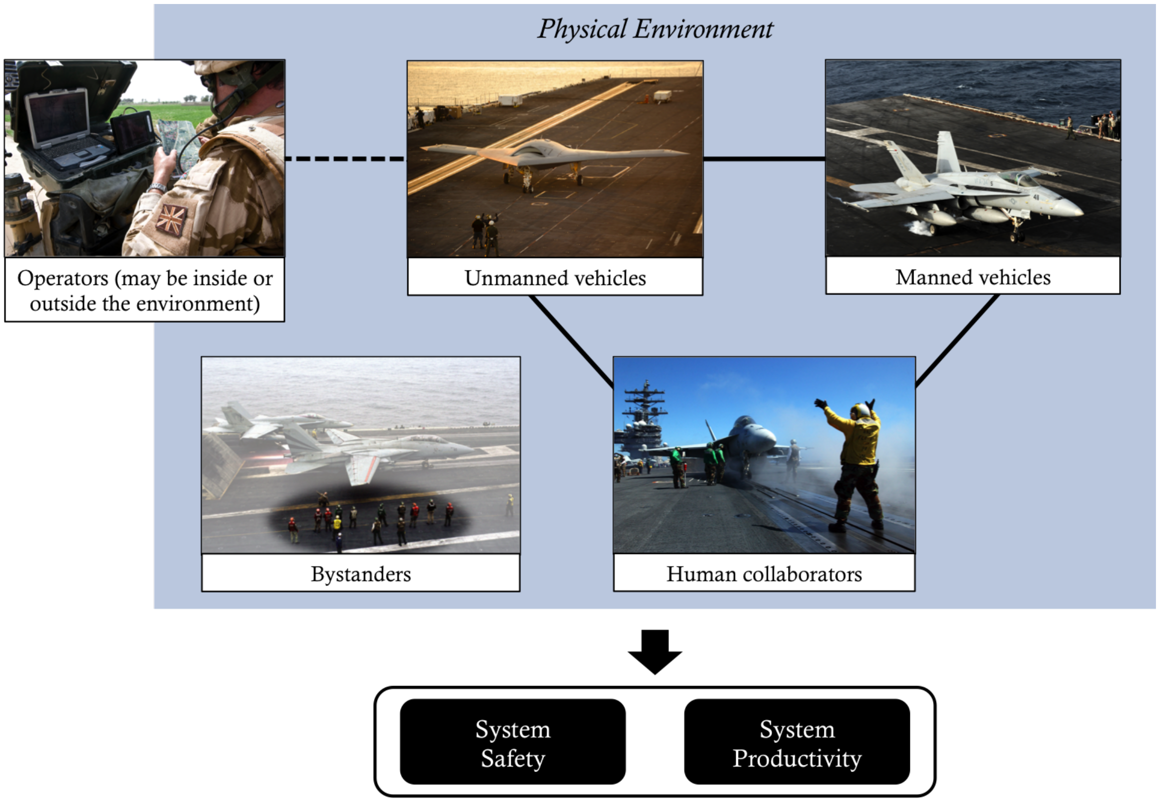


Figure 1-1: Notional diagram of Heterogeneous Manned-Unmanned Environments (HMUEs).

systems like the national highway system is so large that real-world testing at appropriate scales requires the participation of many individuals (in and of itself a challenge to organize), would incur significant expenses in terms of time and money, and would require substantial infrastructure to support operations. Additionally, because the autonomous systems are unproven, there may be substantially increased risk levied on the human participants in the system or to bystanders in the area. While small-scale tests could occur, for many of these complex sociotechnical systems, properties at the smaller scale (one or only a few vehicles) do not typically scale to larger systems (several dozen systems) due to non-linear relationships and trends in the system. Furthermore, the systems of interest may only be in the prototype (or earlier) stage and not actually be able for physical testing in the world, even if the other obstacles could be overcome.

One potential method of addressing these limitations is the creation of realistic simulations of HMUEs that replicate the logic and behavior of unmanned vehicles, manned vehicles, and human operators and bystanders in the world. A valid simulation of such environments would allow designers to explore the effects of design choices in both the unmanned vehicles and the rules that govern their implementation and integration into the system to be tested more quickly, less expensively, and over more possible scenarios than empirical, real-world testing would allow. This dissertation examines the use of agent-based modeling, a bottom-up approach that attempts to model the individual motion and decision-making of actors in the environment, to create simulations of these complex sociotechnical systems for the purposes of exploring the effects of safety protocols (the methods of integrating unmanned vehicles into operations) and methods of robotic vehicle control on system safety and productivity.

1.2 Research Statement

The goal of this dissertation is to develop a simulation of aircraft carrier flight deck operations as a case study in exploring the effects of implementing unmanned vehicles into Heterogeneous Manned-Unmanned Environments (HMUEs). In this exploration, the key interest is in how unmanned vehicles affect the safety and productivity of the system as a whole, rather than focusing on individual vehicles. Variations in these measures amongst different types of unmanned vehicle control architectures enables the identification of the key design parameters that drive performance along metrics of safety and productivity.

1.2.1 Research Approach

The differences between unmanned vehicles integrated into HMUEs lie in the low-level interactions between vehicles with the world, with other vehicles, and with other human collaborators in the world. This thesis utilizes agent-based model as its approach, as agent-based modeling is constructed in a bottom-up fashion by modeling low-level behaviors and interactions of actors in the environment.

When the simulation model is executed, it begins with a set of initial conditions for the agents and the world; the behavior rules and decision-making procedures of the agents, all working in a decentralized fashion, then dictate how the simulation evolves from that point forward. If properly constructed, when given a set of initial conditions, an agent-based model should evolve in the same fashion as the real system it is intended to replicate.

In addition to the behavioral and decision-making rules, defining agents in the system also requires defining the key states and parameters that characterize tasks that are performed in the world and how agents observe the world. Agent states define their current location in the world, speed, orientation, as well as other features such as current goal and health. Parameters define the range of available speeds that at which agents may travel or the range of times a task may take when executed. The behavioral rules then define, given the current state of the agent and observable states of the environment and other agents, (1) what its current goal should be, (2) the tasks it should perform to achieve that goal, and (3) whether or not it can execute those tasks. The defined agent parameters then describe how agents perform the prescribed tasks. Agent models may include not only the actors in the system (people, vehicles, etc.) but machinery, equipment, and other resources within the environment as well as the environment itself. Together, the definitions of agents, tasks, and states should characterize not only individual behavior, but the interactions between individuals in the environment.

Agent-based modeling has been used to explore a variety of systems, ranging from pure decision-making models in economics to models of human and vehicle motion in the world. In one study of human motion, researchers explored ways to improve fire safety evacuations of crowds. The non-intuitive result from this work was that placing a column directly in front of the exit door improved egress performance: agents self-organized into a more efficient and orderly exit pattern than observed in simulations without the column (Helbing, Farkas, & Vicsek, 2000). Agent-based models have also been used extensively in highway traffic safety modeling (Gettman & Head, 2003; Archer, 2004; Yuhara & Tajima, 2006) in order to better understand accident prevention without risking the safety of potential test subjects or bystanders.

A difficulty in modeling futuristic HMUE environments is that, by definition, futuristic environments do not exist and that models of futuristic environments cannot be validated. However, if a current operational environment is similar to the futuristic one, then a valid model of current operations can be used as a template for future operations and lend validity to these futuristic models. One such example of this is aircraft carrier flight deck operations, which this dissertation takes as its point of investigation. The modern U.S. aircraft carrier flight deck is an example of a heterogeneous *manned* environment, where humans and manned vehicles interact and coordinate within close physical spaces. The United States Navy is currently investigating the inclusion of unmanned aerial vehicles into flight deck operations, where the vehicles will be required to interact with the flight deck crew and other manned aircraft in a similar fashion as manned aircraft currently do. The modeling of futuristic aircraft carrier flight deck operations begins with the modeling and calibration of an agent-based model of current aircraft carrier flight operations before proceeding to the development and integration of unmanned vehicles into the simulation. This final simulation is then used as an experimental testbed, comparing the performance of various forms of UAV control and safety protocols in flight deck operations.

1.2.2 Research Questions

This thesis addresses several questions concerning the simulation of heterogeneous manned-unmanned environments and the examination of safety protocols and control architectures therein:

1. What are the specific agents and their related states, parameters, and decision-making rules required to construct an agent-based model of carrier flight deck operations?

2. What are the key parameters that describe variations in unmanned vehicle control architectures and their interactions with the world, in the context of an agent-based model of flight deck operations?

3. In the context of the modeled control architectures, how do these key parameters drive performance on the aircraft carrier flight deck? How do these parameters interact with the structure of the environment, and how do the effects scale when using larger numbers of aircraft in the system?

4. If high-level safety protocols are applied to operations, what effects do they have on the safety of operations? Do the safety protocols leads to tradeoffs between safety and productivity in the environment?

1.3 Thesis Organization

This dissertation is organized as follows:

- Chapter 1, *Introduction*, describes the motivation, objectives, and research questions.

- Chapter 2, *Literature Review*, describes prior research into unmanned vehicle control, safety protocols applied to unmanned vehicles, and prior simulation methods for both human interaction with unmanned systems and safety.

- Chapter 3, *Model Development*, provides background on current aircraft carrier flight deck operations and the construction of the agent-based model of this environment. This includes descriptions of the empirical data used to describe vehicle behavior, the stochastic variables used to model vehicle performance, and the logic functions used to replicate decision-making by humans and vehicles in the environment.

- Chapter 4, *Model Validation*, describes the process of validating aspects of this model, including the empirical data used in testing, the series of tests executed, and key results that support that the model is a reasonable simulation of flight deck operations.
- Chapter 5, *Multi-Agent Safety and Control Simulation (MASCS) Experimental Program*, describes the implementation of new unmanned vehicle and safety protocols into the simulation model, the experimental matrix used in the testing process (the independent variables varied throughout testing), performance metrics used in the analysis, and the results of testing.
- Chapter 6, *Exploring Design Options*, describes other potential applications of the MASCS model in examining the interactions between Control Architectures, Safety Protocols, and mission settings in the context of a design space exploration.
- Chapter 7, *Conclusions*, describes the important conclusions from this work. This includes both key results regarding the integration of unmanned vehicles into aircraft carrier flight deck operations, how these results inform unmanned system integration into other environments, and lessons learned regarding the use of agent-based modeling methods in this work.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Literature Review

> "... To cast the problem in terms of humans versus robots or automatons is simplistic... We should be concerned with how humans and automatic machines can cooperate."
>
> *Telerobotics, Automation, and Human Supervisory Control (1992)*
> Thomas B. Sheridan

A Heterogeneous Manned-Unmanned Environment (HMUE) requires close physical interaction between humans, manned vehicles, and unmanned, autonomous, or robotic vehicles. Currently, HMUE environments are rare, but there exist several heterogeneous *manned* environments (human collaborators and manned vehicles) that are seeking to integrate unmanned vehicles into their operations. This ranges from airborne operations such as aircraft carrier flight decks and other military air operations, ground operations such as autonomous "driverless" cars and automated forklifts, and human-centered environments, such as interactive manufacturing robots and assistive robotics for personal and medical use. In each case, unmanned vehicles bring with them new capabilities and limitations, and the rules that traditionally govern interaction with manned vehicles may no longer be appropriate. Ultimately, the effectiveness of the unmanned robotic systems are contingent upon their capabilities to perform tasks and interact with the environment and the safety with which these operations and interaction occur. This chapter reviews prior research in these two areas. The first section examines prior research on the safety of unmanned vehicle operations in terms of the rules that govern their interaction with the environment — the "Safety Protocols" that dictate when, where, and how vehicle interact with the world. The second section addresses the types of "Control Architectures" used to control unmanned vehicle systems, which dictate their capabilities to move, observe, and interact with the world. The third section then describes how the nature of Safety Protocols and Control Architectures relates to the use of agent-based simulation models.

2.1 Safety Protocols for Heterogeneous Manned-Unmanned Environments

Over the past few decades, a variety of competing ideas regarding the causes of accidents in complex systems have been developed. Some have argued that complex systems always eventually fail (Perrow, 1984), while others have demonstrated that this is not necessarily true for certain systems (Rochlin, La Porte, & Roberts, 1987; Weick & Roberts, 1993; LaPorte & Consolini, 1991). Others have explored the distinctly human contributions to error, including problems that develop over time due to both individual and organizational factors (Reason, 1990; Reason, Parker, & Lawton, 1998; Dekker, 2005). More recently, a systems perspective has been applied to safety (Leveson, 2011, 2004). The common theme within this research is that preserving safety — preventing accidents — is a behavioral concern. Errors (incorrect actions taken by individual), hazards, and accidents ultimately occur because certain aspects of the system — human or otherwise — did not behave as expected and the constraints and processes put in place to prevent accidents all have failed. Additionally, failure to execute the processes, as well as human behavioral factors, cause the system to degrade over time. Reason referred to this as the generation of "latent" failures in the system that lie in wait for the right set of circumstance to become active (Reason, 1990). This is encapsulated in Fig. 2-1.

To prevent the occurrence of an accident, a behavioral constraint at some level must intervene. These can be applied at the organizational and supervisory levels, dealing with topics such as

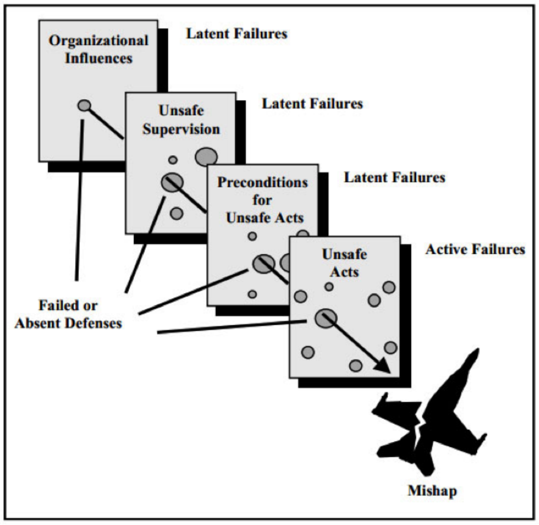


Figure 2-1: Adaptation of Reason's "Swiss Cheese" model of failures (Reason, 1990).

"just culture" and proper training of staff. In the context of human interaction with unmanned vehicles, significant concerns also exist at the lower levels, in that operators and vehicles must avoid unsafe acts and preconditions to those unsafe acts. Unsafe acts in these contexts most often correspond to failures of vehicles to stop in time to prevent a collision (be it not receiving the order to stop, attempting to stop and not being able to, or incorrectly interpreting the stop command) and failures of the crew and supervisors to command that certain actions not occur in parallel (failures of communication, simple human error, and similar cases). While these are known issues, there is again little past literature on how unsafe control actions (and other hazards and accidents) are avoided or otherwise mitigated. The following sections review research into unmanned vehicle safety as well as examples of safety protocols used currently within heterogeneous manned environments. Within the context of the latent failure framework, these sections attempt to characterize common methods of increasing safety in unmanned and robotic operations.

2.1.1 Safety Protocols for Unmanned Vehicles

Research concerning safe autonomous robotic systems addresses a variety of cases, but the primary focus is on a single vehicle interacting with the world. Various studies have addressed automated manipulator robots (Heinzmann & Zelinsky, 1999; Kulic & Croft, 2005; Tonietti, Schiavi, & Bicchi, 2005; Alami et al., 2006; Kulic & Croft, 2006; Nikolaidis & Shah, 2012), Unmanned Aerial Vehicles (UAVs) (Schouwenaars, How, & Feron, 2004; Teo, Jang, & Tomlin, 2004; Sislak, Volf, Komenda, Samek, & Pechoucek, 2007), Unmanned Ground Vehicles (UGVs) (Johnson, Naffin, Puhalia, Sanchez, & Wellington, 2009; Yoon, Choe, Park, & Kim, 2007; Burdick et al., 2007; Campbell et al., 2007), Unmanned Surface Vehicles (USVs) (Larson, Bruch, Halterman, Rogers, & Webster, 2007; Bandyophadyay, Sarcione, & Hover, 2010; Elkins, Sellers, & Monach, 2010; Huntsberger, Aghazarian, Howard, & Trotz, 2011), and personal robots (Jolly, Kumar, & Vijaykumar, 2009). In the context of safety, the major concern is the motion of the unmanned systems near to human collaborators and other manned and unmanned vehicles. As such, the focus of safety research lies on path planning and navigation systems of the vehicles — the primary decision-making action of vehicles in the aforementioned studies. For collisions between unmanned robotic systems and other humans, vehicles, and objects, the system attempts to avoid the hazardous states that could lead to collisions. Examples of hazards might include (1) being too close and not having sufficient stopping distance, (2) turning the wrong direction (applying an incorrect action), or (3) not seeing an obstacle (and thus unable to apply the correct control action). Safety protocols can act to avoid this hazardous state entirely (by maintaining an appropriate distance between vehicles), move out of a hazardous state if it exists (by moving away or past the obstacle), or at worst, attempt to minimize the damage caused if an accident actually happens (by decreasing speed).

Three general methods of Unmanned Vehicle (UxV) collision avoidance are observed in the literature: a) path planning prior to the start of movement, b) reactive collision avoidance during movement, and c) damage/injury mitigation at contact if collision occurs. Each applies some form of constraint to vehicle behavior, either restricting their possible paths or speeds, in order to prevent the occurrence of the hazardous state of being too close to other objects. Ideally, option a) is taken, and the motions of all vehicles in the environment are planned such that no collisions are ever likely to occur (Kulic & Croft, 2005; Schouwenaars et al., 2004; Teo et al., 2004; Johnson et al., 2009). To do so requires knowledge of the current position and trajectory of all nearby agents in the world and accurate predictions of how they will move over time. Given this information, paths for the vehicles are constructed simultaneously with the minimum safe "buffer" distance applied through all paths at all times. However, these plans will likely not evolve as intended given uncertainty in sensor data and stochasticity in both the environment and task execution. As such, many autonomous vehicles will also utilize software that employs method b) — a reactive collision avoidance system working in real time, continually monitoring the motion of nearby obstacles and adjusting the vehicle's path to avoid them (again using the declared buffer distance), otherwise referred to as "detect, sense, and avoid" (Kulic & Croft, 2006; Sislak et al., 2007; Yoon et al., 2007; Burdick et al., 2007; Campbell et al., 2007; Larson et al., 2007; Bandyophadyay et al., 2010). In cases when contact is inevitable, such as a robotic manipulator arm working closely with human colleagues, the system ensures that collisions occur at low kinetic energy levels in order to minimize injury or damage (Heinzmann & Zelinsky, 1999; Tonietti et al., 2005; Alami et al., 2006; Jolly et al., 2009). In general, however, these protocols attempt to do two things: maximize the distance between vehicles at all times and minimize the speed of impact if a collision cannot be avoided. However, these two concepts are fairly limited in scope, applying only to collisions between vehicles. Complex systems such as the HMUEs of interest in this work can have a large host of other hazardous states that lie outside of simple vehicle collisions and may require additional protocols. Some examples of these are reviewed in the next section.

2.1.2 Safety Protocols in Related Domains

While HMUEs are currently rather rare in practice, there are a variety of similar domains that can provide some guidance as to common hazards of human-machine interaction and the protocols that are used to prevent them. A first example is the manufacturing domain, which has utilized automated systems for several years. These systems lack the intelligence and path planning capabilities of the systems in the previous section, and thus a different tactic has been utilized in order to ensure safe operation near human beings — physical isolation (Marvel & Bostelman, 2013). One common practice is to place these systems within a physical cage where no human coworkers are allowed, preventing any possible collisions between human and machine (Fryman, 2014). However, the International Organization for Standardization (ISO) is currently developing standards for more integrated human-robot interaction that address both manipulator arms and mobile robotic systems (ISO/DST15066 (ISO, unpublished)). In these draft standards, four potential safe operating modes are identified: (1) stopping in a predetermined position when a human enters the area, (2) manual guiding of robot activities by the human, (3) speed and separation monitoring, and (4) power and force limiting (Fryman, 2014). The first option is, in essence, still a form of isolation: the robot and human are not operating in the same space at the same time. The second is a form of teleoperation, with the human in full control the robot's activity from a safe distance away. In the third and fourth, the robot is allowed to operate autonomously near the human collaborator, with key behavioral rules encouraging safety. In the third option, attempts are made to ensure no contact ever takes place; in the fourth, contact is allowed, but only at low impact forces.

The general rule for manufacturing systems, then, has been to keep human operators separate from the robotic systems when at all possible. Enabling greater collaboration means allowing collaboration, but ensuring that collisions either do not occur or are of low danger if they do occur. These same principles of separation have also been used in more advanced mobile robotic systems like the Kiva "mobile-robotic fulfillment system" used in some Amazon.com and Zappos warehouses (Scanlon, 2009). In this warehouse management system, rather than requiring human workers to move about the facility locating and loading packages onto carts or forklifts, Kiva's robots retrieve small shelving units from the shop floor and deliver them to human packers. These packers remain in a fixed location in the warehouse; the robots include of sophisticated path planning algorithms that allow them to safely move around each other, but they cannot account for the presence of a human coworker. In these systems, it is the human who is kept isolated from the robots, staying within a specific area while shelves and packages are brought to them.

Two final domains — naval aviation and commercial mining — both involve the use of very large, mobile vehicles that directly require human interaction to perform tasks in the world. On the aircraft carrier deck, upwards of forty aircraft interact with over one hundred human crewmembers and a dozen support vehicles to fuel, load, launch, and land aircraft in rapid fashion. In open pit mining, 300-ton "haul trucks" work in close spaces with front loaders, dumping chutes, and human supervisors traversing the site in smaller vehicles (typically small pickup trucks and sport utility vehicles) to move loose ore from blast pits to processing sites. These tasks typically require close interaction and maneuvering of vehicles, as well as navigation of roadways and intersections similar to that on more typical civilian road systems. Additionally, each of these two domains is actively investigating the introduction of UxV systems into operations. The United States Navy is currently engaged in developing an unmanned, autonomous combat vehicle for use on the aircraft carrier deck (McKinney, 2014; Hennigan, 2013) while the mining industry (Rio Tinto, 2014; Topf, 2011) is exploring the development and use of autonomous hauling trucks for use in open pit mining. Given the dangerous nature of operations in these environments, a high priority is placed on safety in operations, resulting in the development of numerous safety protocols to safeguard system operations.

Published standards were not located for the mining and aircraft carrier domains, but a variety of information on operations was obtained from training manuals, interviews with subject matter experts, and observations of operations. These sources of information were used to compile a set of safety protocols that work within the system. In reviewing this documentation, a framework was defined to describe an accident in terms of the potential hazards that generated the accident that the safety protocols (rules of behavior) put in place to avoid the hazards. This resulted in 33 different combinations for aircraft carrier operations (10 different accident types, 25 different hazards, and 18 safety protocols) and 35 for mining operations (7 general accident types, 15 different hazards, and 11 different safety protocols). Examples of these appear in Table 2.1 below, with a full list appearing in Appendix A. Together with the examples from prior sections, three general areas of focus appear to emerge, based on the types of hazards being mitigated in each case. These three major safety "principles" of Collision Prevention, Exclusive Areas, and Task Scheduling are explained in more detail in the next section.

2.1.3 Defining Classes of Safety Principles

The types of safety protocols that act within manufacturing, aircraft carrier, and mining environments appear to fall within three general categories, listed within Table 2.2. These three safety principles provide a framework both for classifying current safety protocols observed in heterogeneous vehicle environments and for creating new protocols that could be applied. The first principle, Collision Prevention, includes safety protocols that govern the spacing of vehicles when in motion.

Table 2.1: Examples of accidents, hazards, and safety protocols from the mining and aircraft carrier domains.

Hazard	Accident	Aircraft Carrier Flight Deck	Mining
Vehicle collision while in transit	Collision between vehicles	(Observation) During transit on deck, Aircraft Directors maintain a large spacing between vehicles.	In travel mode, automated trucks in the mine site are not allowed within 50 m of human workers.
Interference with a dangerous task	Collision, fire, human injury and death	Do not enter catapult area if aircraft is on catapult.	Automated trucks may never enter within 5 m of a stationary human worker.
Interference between dangerous tasks	Collision, fire, human injury and death	Crew are instructed to never fuel and load weapons onto an aircraft at the same time.	Excavators only load one truck at a time.
Accidental vehicle motion	Collision between vehicles, human death	Tie down aircraft if inactive (prevent accidental motion).	Lockdown vehicles that are in use (prevent accidental operation).

This category would include the "buffer" areas included in the unmanned vehicle motion protocols covered in Section 2.1.1 and the "speed and separation monitoring" operating mode in the draft ISO (ISO, unpublished) standards discussed in Section 2.1.2.

The second safety principle is Exclusive Areas, in which certain areas should not be accessed while certain systems are operating. One example from aircraft carrier operations is that the area immediately surrounding a catapult is off-limits while an aircraft is being attached and launched; this decreases the risk that a crewmember or other vehicle will be in the launch path and will be struck during a launch. From the ISO standards, the first operating mode — which requires that robotic systems not be operation while a human collaborator is near — would also fit within this classification.

Task Scheduling protocols, the third class, describes constraints on task execution – that two actions either should not be performed at the same time, should not occur in a certain order, or should not occur within a certain period of time of one another. In the aircraft carrier domain, one example is that aircraft fueling and weapons loading do not occur at the same time; a fuel accident and fire occurring while weapons are being loaded could be catastrophic. This can also include cases in which a task is inserted solely to reduce risk in operations. For instance, an example from the aircraft carrier environment concerns waiting for appropriate crew to escort a vehicle across the deck. While an aircraft could begin a taxi motion without the full escort group, doing so may significantly increase the likelihood of an accident occurring on the deck. The first and second modes of operation in the draft ISO standard might also fit in this category, as the robotic systems would not be allowed to work simultaneously with humans in the environment. The primary difference between Task Scheduling and Exclusive Areas is in how many different regions exist within the environment: if multiple areas exist, Exclusive Areas would allow parallel operations in each individual region. Otherwise, Task Scheduling would describe the ordering of tasks within the single region.

Table 2.2: The three general classes of safety principles elicited from HMUE source materials (Rio Tinto, 2014; Topf, 2011) and observations.

Principle Class	**Constraints**
Collision Prevention	Safety is enforced through sensing of nearby vehicles and obstacles. Vehicles in motion must maintain a minimum safe separation distance from other vehicles at all times.
Exclusive Areas	Safety is enforced through physical segregation. The working environment is divided into separate regions for manned and unmanned operations. Vehicles must operate within their prescribed regions at all times and should not exit their prescribed regions.
Task Scheduling	Safety is enforced through scheduling. Tasks are performed only in a specified order to reduce risk in operations. This may apply to multiple tasks for a single vehicle or for separating entire groups of vehicles/humans.

The three safety principles listed above all share a common feature: each describes a type of constraint on individual behavior (human and vehicle, manned and autonomous) in the environment, addressing when, where, and how agents perform tasks in the world. Dynamic Separation protocols require additional logic and decision making about the paths vehicles are allowed to travel and requires them to monitor the world around them, looking for conditions where they should adjust their speed or path. Exclusive Area protocols require additional logical functions to know when vehicles can enter a given area, if ever. Task Scheduling protocols address the logic behind task execution, describing when and under what conditions tasks should or should not be performed. The common feature of these constraints is that *behavioral rules* describe what the vehicle should do: this is a feature of the software of the unmanned and autonomous vehicles and are not direct properties of the physical hardware. As such, modeling the effects of safety protocols in HMUEs requires modeling the logic behind these actions, in the context of the motion of agents in the world and their interactions with others.

These safety principles, when applied to a specific domain and specific set of operations, form the set of safety protocols (Table 2.2) that apply to specific vehicle behaviors and tasks in the world. In this work, these principles are applied to aircraft carrier flight deck operations (which will be described further in Chapter 3). In applying these principles, protocols take the form provided in Table 2.3, described in terms of vehicle separation. The Collision Prevention principle describes the “Dynamic Separation” of vehicles in motion, where aircraft should maintain a minimum safe distance between themselves and other aircraft in front of them. This protocol would be used in both manned-only and heterogeneous manned-unmanned environments, although the minimum safe distance may vary between vehicle systems and operating conditions. Exclusive Areas translates into an “Area Separation” protocol in which the flight deck is divided into forward and aft sections. Manned and unmanned aircraft begin in separate areas and are only allowed to operate within those areas. Task Scheduling becomes a “Temporal” separation, with operations organized such that manned aircraft begin operations first with unmanned aircraft remaining inactive until manned operations complete. Additional details as to how these rules are encoded for specific actors and agents will be provided in Chapter 5.

Testing the effects of these Safety Protocols on productivity and safety of HMUE operations requires generating models of the manned and unmanned aircraft that operate within them. Models of manned, or manually controlled, vehicles can be created through empirical observations and measurements that characterize their decisions and movements within the world. However, because by definition futuristic unmanned vehicle systems do not yet exist, their models must be constructed through different means. Many of these systems exist only with laboratory environments or as basic simulation designed to explore new graphical user interfaces, both far from being at sufficient fidelity for observation and modeling. However, this research does provide sufficient background to describe

Table 2.3: The three general classes of safety protocols that will be applied to unmanned carrier flight deck operations.

Safety Protocol	**Definition**
Dynamic Separation	Aircraft in motion should maintain a minimum safe distance of one vehicle length forward of the aircraft while in motion on the flight deck.
Area Separation	Operations will be planned such that manned and unmanned aircraft begin parked in separate regions of the flight deck. Operations are planned such that aircraft are only assigned to and taxi within their specified areas of operation.
Temporal Separation	Operations are planned such that manned aircraft are the first to operate on the flight deck. Unmanned aircraft are not allowed to begin taxi operations until manned aircraft complete their taxi operations.

how these systems might be utilized in the future and provide some guidance as to their capabilities. The next section describes several different classes of unmanned vehicle systems, classified according to how a human operator interacts with them — their "Control Architectures."

2.2 Human Control of Unmanned Vehicle Systems

The key element of HMUEs is the presence of unmanned vehicles in the system. However, the term "unmanned" can describe a variety of different systems with varying types and degrees of capabilities. Some systems have highly advanced autonomous capabilities and can navigate the world largely on their own; other unmanned vehicles are simply remote-controlled systems, with little onboard intelligence and requiring the same inputs from human operators that manned aircraft do. The changes in these systems can be characterized along three different axes: the location of the human operator, what control inputs they provide, and how they provide them. Changes along these axes have significant effects on the performance of the human-vehicle team in terms of how they acquire information on the state of their surroundings, determine what tasks to perform, and execute those tasks in the world.

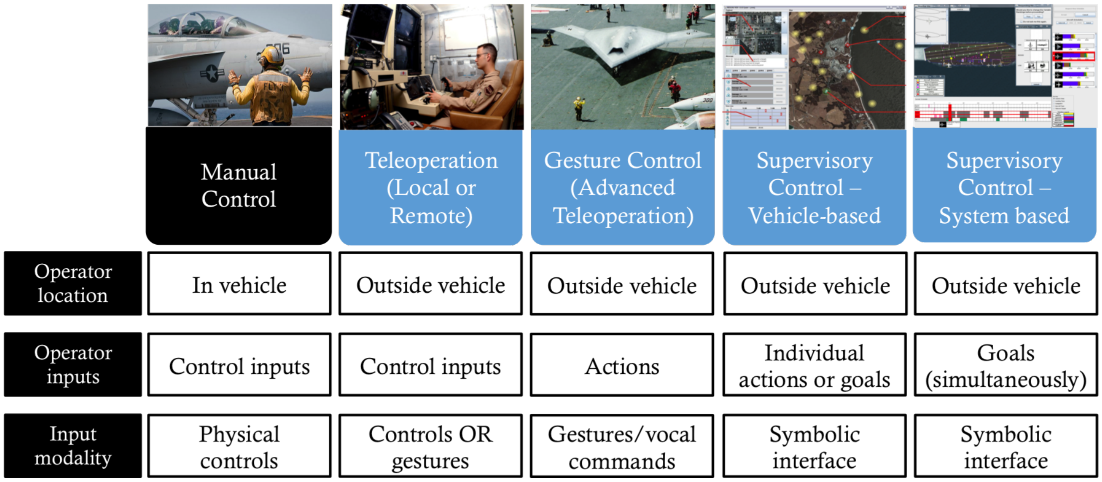
This section covers five main types of control architectures: Local Teleoperation (LT), Remote Teleoperation (RT), Gestural Control (GC), Vehicle-Based Human Supervisory Control (VBSC), and System-Based Human Supervisory Control (SBSC). These are compared to a baseline concept of Manual Control (MC), where a human provides control inputs directly to the vehicle while seated within it. An overview of the differences between the systems appears in Figure 2-2. Because local and remote teleoperation are identical (except for the distance from operator to vehicle), they are combined in a single column at this time. Beginning with the first row (operator location), all unmanned vehicle systems involve the relocation of the human operator from the vehicle to some location outside of the vehicle. For the remaining two rows (control inputs and input modality), moving from left to right moves from physical input of low-level control inputs (acceleration, steering, etc.) to more abstract definitions of tasks ("go to this area and monitor for activity") input through a Graphical User Interfaces (GUIs) using symbolic interactions. Subsequent sections explain how these differences in control architecture definition affect performance of these systems. Together, these changes shift the operator from providing highly-detailed control inputs to a single vehicle to providing highly abstract instructions to multiple vehicles, moving from managing the behavior of one vehicle to optimizing the performance and interaction of many.

2.2.1 Manual Control

In this work, Manual Control (MC) is defined as a system where the operator is seated within the vehicle and provides low level control inputs directly to the vehicle through physical devices. Consider a basic general aviation aircraft: the pilot controls pitch, yaw, and roll by directly manipulating the yoke, throttle, and rudder pedals. The pilot thus dictates the direction and speed of the vehicle, performing all tasks related to guidance and navigation, and is responsible for reacting to the world and all external inputs and closing all control loops[1]. These actions can be characterized in terms of three main functions: *sensing* the world around them, *selecting a response* given those sensor inputs, and *executing* the selected response. The general pattern of behavior of the human operator is characterized by Wicken's information processing loop (Wickens & Hollands, 2000), shown in Figure 2-3.

The operator acquires information on the world through their senses, which is then translated into general perceptions about the state of the world and its relationship to the operator's intended goals. Operators then move through a process of response selection, supported by both short-term (storing current information) and long-term (recalling similar experiences) memory. Once the action

[1] While, in certain systems, automatic control systems and autopilots may also be used, they are not the default state of the system and are not necessarily active throughout the entirety of vehicle activity.



	Manual Control	Teleoperation (Local or Remote)	Gesture Control (Advanced Teleoperation)	Supervisory Control – Vehicle-based	Supervisory Control – System based
Operator location	In vehicle	Outside vehicle	Outside vehicle	Outside vehicle	Outside vehicle
Operator inputs	Control inputs	Control inputs	Actions	Individual actions or goals	Goals (simultaneously)
Input modality	Physical controls	Controls OR gestures	Gestures/vocal commands	Symbolic interface	Symbolic interface

Figure 2-2: An overview of the variations between the five different control architectures described in this section.

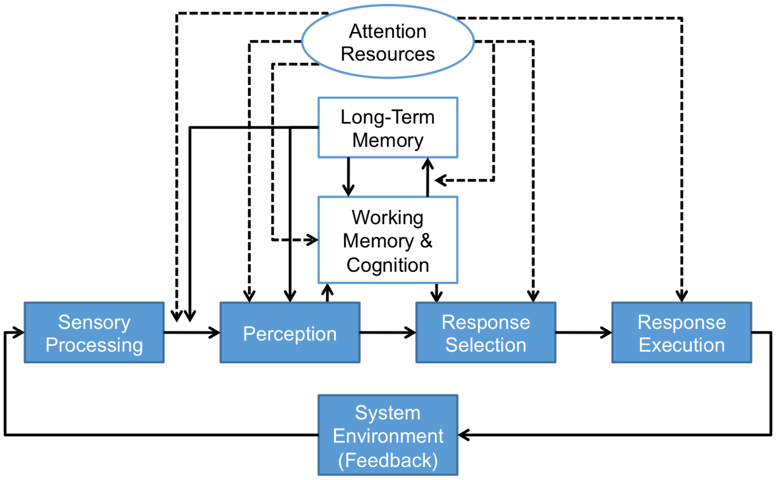


Figure 2-3: Human Information Processing loop (Wickens & Hollands, 2000).

is selected, it is implemented physically, providing control inputs to the vehicle, which then responds to the control inputs. While depicted as a staged process, the operator is actually in a continual state of monitoring the vehicle and all relevant inputs, often monitoring multiple sources of information at once.

This processing loop is also accurate for unmanned vehicles, with the key change being that the implementation of actions is no longer done through physical artifacts attached to the vehicle. The processing loop thus extends to include the vehicle as a separate entity and various intermediaries connecting operator to vehicle. The most commonly cited model of this is Sheridan's Human Supervisory Control (HSC) loop, pictured in Figure 2-4 (Sheridan & Verplank, 1978; Sheridan, 1992). A variety of systems are architected as shown in the figure, with the human operator providing input to a computer or other mediator than transmits commands to the vehicle platform (which itself may include a computer). This can take a variety of forms; in some cases, it is simply a Ground Control Station (GCS) with a communications link, while in others, it is a more complex computer system with intelligent planning algorithm. These changes in communications patterns change what information the operator provides, what they receive, and what is required of the vehicles in the world. The remaining sections describe different types of these control systems, beginning with the ones most similar to manual control. Teleoperated systems require the operator to provide the same control inputs in the same manner as MC, but does so from a GCS that may be either nearby or beyond line-of-sight communications.

2.2.2 Teleoperation of Unmanned Vehicle Platforms

A teleoperated system "extends a person's sensing and/or manipulating capability to a location remote from that person" (Sheridan, 1992). This might refer to teleoperated manipulator arms or two mobile vehicles. In current military operations, most unmanned aircraft are teleoperated, with

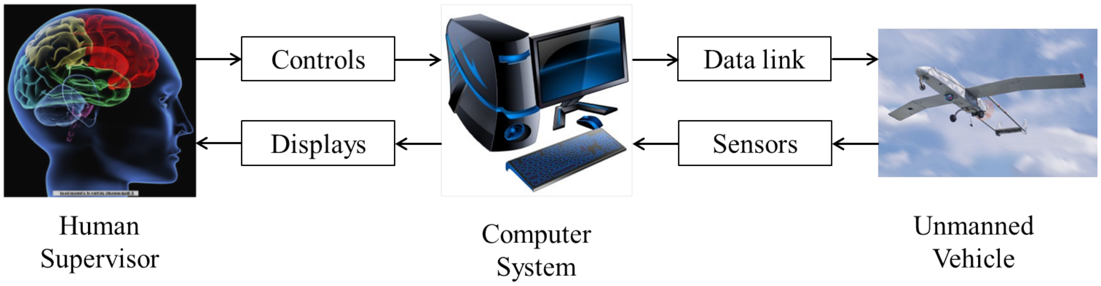


Figure 2-4: Human Supervisory Control Diagram, adapted from Sheridan and Verplank (Sheridan & Verplank, 1978).

the human operator providing largely the same control inputs to the vehicle, using the same physical devices, as they would in manual control. As such, these are often referred to as "remotely piloted aircraft, or RPAs. For these systems, the Ground Control Station (GCS) used by the human operator includes a throttle, yoke, and rudder pedals similar to those of the actual aircraft; which are used in the same fashion as before. In this case, however, the control inputs are then translated by the GCS into computer data, transmitted to the vehicle through some fashion, and then reinterpreted into control actuator commands. This same transmission mechanism is also used to provide information on the state of the vehicle back to the operator, which is then displayed on the GCS. All information on the state and orientation of the vehicle, including views of the surrounding world as conveyed by onboard cameras, is received through the GCS. The pilot's perception of the world is thus limited in three ways: their visual field is limited by both the onboard cameras and the size of the displays on the GCS, while their physical location removes any haptic and proprioceptive information about accelerations and orientation.

Just as in Manual Control, the human operator has the responsibility of closing all control loops on the vehicle; anything that happens in the environment requires the operator to determine and implement the appropriate response. The earliest teleoperation systems, and the majority of early research, concerned nuclear materials handling, exploration of the moon, and underwater exploration (Sheridan & Ferrell, 1963) – tasks not easily or safely performed by a human being. The purpose was not to increase the performance of the system, but to remove the human operator from physical danger. Given this goal, the costs of implementation — a reduced ability to observe the system and, in many cases, the presence of substantial latency in data transmission — were acceptable. Both of these elements (visibility and signal lag) have been shown to be detrimental to the ability of the human-vehicle combination to perform tasks in the world in studies involving telemanipulation systems (McLean & Prescott, 1991; McLean, Prescott, & Podhorodeski, 1994) and ground vehicles (telerovers) (Spain & Hughes, 1991; Scribner & Gombash, 1998; Oving & Erp, 2001; Smyth, Gombash, & Burcham, 2001). These tasks are similar to what will occur with teleoperated aircraft on the flight deck and are chosen as the focal point for modeling. In each of these studies, the lone change in the system was a change from human operator “direct viewing” (seated within the vehicle) to some alternative form of visual display; these ranged from helmet-mounted systems linked to cameras on top of the vehicle (such that the operator was still seated within) to the mounting of computer displays within the vehicle to the use of remote ground stations.

In general, the experimental tasks for telerovers involved navigating a road course of some form, with performance measured by time to complete the task and error measured by the number of course deviations or markers hit by the vehicle. Similar measures were used for the telemanipulation tasks. Results from those studies (McLean & Prescott, 1991; McLean et al., 1994; Spain & Hughes, 1991; Scribner & Gombash, 1998; Oving & Erp, 2001; Smyth et al., 2001) demonstrated a range of results in both measures. The increase in task completion time for teleoperated systems ranged from 0% to 126.18%, with a mean of 30.27% and a median of 22.12%. The tasks used in these experiments typically required drivers to navigate a marked course that required several turning maneuvers. In these tasks, drivers showed a significantly higher error rate (defined as the number of markers or cones struck during the test) as compared to non-teleoperated driving. Percent increases in errors ranged from 0% to 3650.0% with a median of 137.86% and a mean of 60%. While these elements are difficult to translate to other types of tasks, conceptually, they signify a difficulty in high-precision alignment tasks — a skill required of several tasks on an aircraft carrier flight deck.

The studies described above were all instances of “local” teleoperation, with delays in communications between vehicle and operator being much less than one second; the movement of the operator to a location beyond Line-Of-Sight (LOS) communication — such as in the U.S. military’s current use of “Remote-Split” operations with Predator UAVs — introduces a non-trivial amount of latency into signal transmission. This delay has two significant repercussions on human control of teleoperated vehicles. The first is simply the existence of the delay in communications: transmitting data from the vehicle to the remote operating station requires a period of time (X) to travel between locations. Transmitting commands back from the remote station to the vehicle requires a second period of time (Y), with $X + Y$ being equal to the round-trip delay in communications from time of data transmission by the vehicle to the vehicle’s receipt of a new command from the operator.

The more interesting ramification is in how operators respond to the presence of lag in the system and change their strategies to compensate. A naive teleoperator inputs tasks assuming there is no lag, continually overshooting their target and requiring multiple corrective actions to finally reach the target (and much greater amounts of time). Sheridan and Ferrell were the first to observe that more experienced teleoperators used an “open-loop” control pattern, inputting tasks anticipating the lag, then waiting for the tasks to complete before issuing new commands (Sheridan & Ferrell, 1963). While this strategy is not as efficient as operations with zero latency, it avoids exacerbating the problem by continually inputting (erroneous) corrective actions. In both cases, the time required to complete tasks increases at a rate higher than the amount of lag adding to operations (it is a multiplicative function) — one second of latency translates into much more than one second of additional time. Fifteen different papers, totaling 56 different experimental tests, show that the percentage change in task completion time *per second of lag* ranges from 16% to 330%, with a mean of 125% and a median of 97% (Kalmus, Fry, & Denes, 1960; Sheridan & Ferrell, 1963; Hill, 1976; Starr, 1979; Mar, 1985; Hashimoto, Sheridan, & Noyes, 1986; Bejczy & Kim, 1990; Conway, Volz, & Walker, 1990; Arthur, Booth, & Ware, 1993; Carr, Hasegawa, Lemmon, & Plaisant, 1993; MacKenzie & Ware, 1993; Hu, Thompson, Ren, & Sheridan, 2000; Lane et al., 2002; Sheik-Nainar, Kaber, & Chow, 2005; Lum et al., 2009). Thus, at the median, one second of lag results in a near doubling of the time to complete a task (a 100% increase); two seconds of lag equals a 200% increase. These tests addressed both cases where operators where directly observing the system they were controlling, or while receiving information via a video or computer display.

These studies imply that the nature of human teleoperation of unmanned vehicles involves two distinct systems that must be modeled. In the first, termed Local Teleoperation (LT), the main effects are that the operator's view of the world is artificially constrained by the cameras onboard the vehicle and the screens on the operator's GCS, which the leads to increases in the time require to complete tasks (essentially, a decrease in vehicle speed). In the second, termed Remote Teleoperation (RT), additional latency is introduced into the signal, further degrading performance while also forcing operators to adopt new strategies of interactions. For these systems, the preservation of the safety of the human pilot, now a remote operator, comes at a significant cost of vehicle performance. These changes, however, do not guarantee any benefits to the vehicles themselves, or the people that interact with them within the HMUEs. It is quite possible that the effects of latency on operations lead these vehicles to be less safe than before, or greatly reduce the productivity of operations. Safety protocols applied to these vehicle must adequately address these changes in operator capabilities but are ultimately limited by the properties of the vehicles. Otherwise, improving the safety and efficiency of the unmanned vehicles in the system requires changing the form of operator interaction that occurs. One potential upgrade from teleoperation involves the inclusion of limited autonomy onboard the unmanned vehicle that negates the need for a ground control station, instead allowing individuals within the environment to command actions directly. These advanced forms of teleoperation are discussed in the next section.

2.2.3 Advanced Forms of Teleoperation

The more traditional teleoperation systems described in the previous section still rely heavily on a single human operator to close the control loop on the unmanned vehicle under control; this forces a variety of constraints into the design of the system, requiring a communications link to a GCS and the transmission of video and vehicle state data to the operator for review. These architectural constraints may lead to significant performance degradations because of the limitations inherent in the system. More advanced forms of these systems might require no pilot involvement at all, instead possessing sufficient autonomous capabilities to recognize task instructions from local human collaborators, no longer requiring a human remote pilot to be part of the information processing loop. Two forms of such systems are gesture- and voice-based control, in which the human operator communicates *commands* rather than control inputs (”move forward” instead of “throttle up”). In these systems, the computers onboard the vehicle are responsible for decomposing the verbal or gestural command inputs into lower-level control inputs from a pre-defined list of possibilities. In the context of Figure 2-2, these changes occur in the second and third rows, moving from control inputs to commands and the modality from physical interfaces to “natural” gesture and voice commands.

Voice-based command of robotic systems has been utilized in a variety of fields, including surgical robotics (Mettler, Ibrahim, & Jonat, 1998), unmanned vehicles (Sugeno, Hirano, Nakamura, & Kotsu, 1995; Doherty et al., 2000; Perzanowski et al., 2000; Kennedy et al., 2007; Bourakov & Bordetsky, 2009; Teller et al., 2010), and motorized wheelchairs (Simpson & Levine, 2002). However, these studies mostly focus on discussing the system as a proof of concept without reporting comparison to other control architectures, or even reporting the accuracy of the voice recognition system itself. What results are reported demonstrate error rates ranging from as small as 2 to 4% to upwards of 45% (Atrash et al., 2009; Wachter et al., 2007; Sha & Saul, 2007; Fiscus, Ajot, & Garofolo, 2008; Furui, 2010; Kurniawati, Celetto, Capovilla, & George, 2012). However, these studies also deal with large sets of vocabulary words and are not directly focused on command language. A few of these studies are intended for precisely this function, with several studies demonstrating speech recognition rates above 95% (Ceballos, Gómez, Prieto, & Redarce, 2009; Gomez, Ceballos, Prieto, & Redarce, 2009; Matsusaka, Fujii, Okano, & Hara, 2009; Park, Lee, Jung, & Lee, 2009; Atrash et al., 2009). While perhaps promising in some applications, these system require low levels of background noise and are unlikely to be useful for aircraft carrier flight deck operations, the system of interest in this work.

Gestural control systems utilize stereo-vision cameras and sophisticated computing algorithms to first detect human operators, then track them within the local environment. Their actions are then converted into data points that are compared with databases of previously-recorded motions to determine the command being issued. These systems rely heavily on machine learning techniques and training on previous data sets in order to address both between- and within-person variability in motions. As such, the systems should get better over time, but may struggle with motions that are not exaggerated. However, these systems may be particularly useful in flight deck operations, as gestural communication is already the primary method in which crew communicate instructions to pilots. Effective gesture control technologies would mean that the integration of UAVs would have a minimal effect on the deck crew in terms of training and adaptation. Currently, most research involves the use of human motion data sets — large databases of videos of people performing various movements. The datasets most often observed in the literature search are the KTH (Schuldt, Laptev, & Caputo, 2004), Weizmann (Blank, Gorelick, Shechtman, Irani, & Basri, 2005), and Keck ("Gesture") datasets (Lin, Jiang, & Davis, 2009). However, these data sets are typically based around motions that are interesting to the researchers (for instance, running, boxing, and hand waving) rather than any concepts of direct commands to a vehicle system. The data sets also typically use a very small set of gestures (6, 9, and 14 for the sets), reflective of the current immaturity of the technology.

Even so, research on these sets provide a good picture of the state-of-the-art. A set of papers, all from the 2005-2010 time frame and each referencing many other studies as comparison points were reviewed (Blank et al., 2005; Lin et al., 2009; Jhuang, Serre, Wolf, & Poggio, 2007; Scovanner, Ali, & Shah, 2007; Laptev, Marszalek, Schimd, & Rozenfeld, 2008; Schindler & Gool, 2008; Yuan, Liu, & Wu, 2009; Song, Demirdjian, & Davis, 2012) and their accuracy ratings were compiled into the histogram in Fig. 2-5. Of these 146 studies reported, 91 report better than 90% accuracy, 37 report results better than 95%, and 11 report results of 100% accuracy in classifying human gestures according to their definition. These results suggest that, at least in terms of accuracy, these systems are quickly approaching the point at which they would be viable for field use. However, as noted above, most of these are for random, common human gestures — not command type gestures that would be used in flight deck operations. Song et al. are building a gesture-recognition system specifically for this purpose, which requires tracking not only the human, but tracking the arms and hands independently in order to understand different hand position (thumbs up/down, hand open/closed) (Song et al., 2012). This is a significantly more difficult task, and many of the motions are quite similar to one another, at least for the algorithms that track and describe motion. Currently, their system reports accuracies near 90%, but at a three-second computational time. Results for other systems using simpler data sets in the literature vary significantly, ranging from sub-second to up to ten seconds. Given that these systems are also using relatively "clean" data — stable background environments, little to no visual occlusion of the human being observed, no sun glare — for these simple and small gesture sets, these fast computational times and high accuracies are not likely to occur in flight deck operations.

While substantial work is still needed for these systems, they may provide a more suitable replacement for manual control vehicles than remotely teleoperated systems. Additionally, further increases in vehicle autonomy might enable further increases in performance while also removing some of the potential failure modes of gesture control systems. Centrally networking vehicles together, relying on shared sensor data rather than visual perception, enables new capabilities and efficiencies in behavior unavailable for other types of vehicles. These are referred to as "human supervisory control" systems and grant a single operator the ability to control many vehicles at once. They are discussed in the next section.

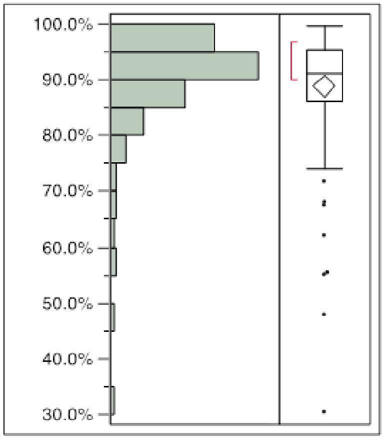


Figure 2-5: Histogram of accuracies of gesture control systems.

2.2.4 Human Supervisory Control of Unmanned Vehicles

While the term "supervisory control" may be applied to any of the systems described in this chapter, for the remainder of this thesis, the term will refer solely to systems where a single human operator controls one or more highly-autonomous vehicles through the use of a computerized display and GUI. In such systems, operators issue abstract task commands, such as "go to this area and look for a vehicle" to the unmanned platform; the platform is then capable of determining where it currently is, how to navigate to the area of interest, and how to execute the monitoring task on its own, possibly to the point of alerting the operator when something of interest has occurred. Typically, this requires substantial intelligence onboard the vehicle, including a variety of high-fidelity sensors to observe the world (GPS, LIDAR, and other range and sensing systems), cameras, and high-accuracy inertial measurement devices, along with high-powered computer systems running intelligent planning algorithms to handle vehicle motion and translate operator commands into low level control inputs. Because of the capabilities of these vehicles, operators are no longer constrained to operating one vehicle at a time and can issue commands to multiple vehicles for simultaneous execution. For these systems, the human operator is more of a "supervisor" of the actions of the largely-independent vehicles, rather than a "controller" or "operator." The tasks performed by this supervisor are dependent on exactly how capable the vehicles are: how many tasks can be assigned to them at a given time, how reliability they execute those tasks, and how often they require additional instructions.

Typically, the GUIs for these systems moves utilizes a system-centric, map-based representation (see (Trouvain, Schlick, & Mevert, 2003; Roth, Hanson, Hopkins, Mancuso, & Zacharias, 2004; Malasky, Forest, Khan, & Key, 2005; Parasuraman, Galster, Squire, Furukawa, & Miller, 2005; Forest, Kahn, Thomer, & Shapiro, 2007; Rovira, McGarry, & Parasuraman, 2007) for examples), rather than a vehicle-centric one, to provide supervisors with sufficient Situational Awareness (SA) of the vehicles and states of the world. In many cases, a representation of the system schedule is also utilized, providing SA on the current state of vehicle task execution and when they may require operator intervention (see (Howe, Whitley, Barbulescu, & Watson, 2000; Kramer & Smith, 2002; Cummings & Mitchell, 2008; Calhoun, Draper, & Ruff, 2009; Cummings, How, Whitten, & Toupet, 2012) for examples). Beyond these features, there are two broad ways of classifying how the operators provide tasks to the vehicles. The first form is referred to as Vehicle-Based Human Supervisory Control (VBSC), in which supervisors supply commands to a single vehicle at a time, working iteratively through the queue of vehicles waiting for instruction. These systems require operators to define tasks (Malasky et al., 2005; Ruff, Narayanan, & Draper, 2002; Ruff, Calhoun, Draper, Fontejon, & Guilfoos, 2004; Cummings & Mitchell, 2005; Ganapathy, 2006; Cummings & Guerlain, 2007) or destinations (Trouvain et al., 2003; Calhoun et al., 2009; Dixon & Wickens, 2003, 2004; Dixon, Wickens, & Chang, 2005; Olsen & Wood, 2004; Crandall, Goodrich, Olsen, & Nielsen, 2005; Cummings et al., 2012) for vehicles, possibly in terms of a preprogrammed travel route. As the autonomy onboard the vehicles now handles low level control, operators might also attend to additional tasks, such as error and conflict resolution (Cummings & Mitchell, 2005) or target assignment and identification (Rovira et al., 2007; Calhoun et al., 2009; Ruff et al., 2002; Cummings et al., 2012; Wright, 2002; Kaber, Endsley, & Endsley, 2004). The performance of the system as a whole is dependent on the supervisor's service rate and whether or not they can keep up with the incoming rate of tasks from vehicles, which in turn is dependent on the aforementioned vehicle capabilities. Past research has shown that operators can control up to eight ground vehicles or twelve aircraft at a time (Cummings & Mitchell, 2008), suitable for a flight deck where no more than 8-12 vehicles are taxiing at once.

Controlling more than eight to twelve vehicles can be more effectively achieved through the inclusion of additional autonomy in the system, although not on the vehicles themselves. System-Based Human Supervisory Control (SBSC) systems utilize intelligent planning algorithms to assist the human supervisor in replanning tasks for many vehicles simultaneously (see (Roth et al., 2004; Forest et al., 2007; Howe et al., 2000; Kramer & Smith, 2002; Cummings et al., 2012; Ryan, Cummings, Roy, Banerjee, & Schulte, 2011; Anderson et al., 1999; Parasuraman, Galster, & Miller, 2003; Bruni & Cummings, 2005) for examples). In these systems, the supervisor is interacting with a planning algorithm, providing guidance in terms of the desired performance of *the schedule* in the form of both performance goals and constraints on activities. The algorithm then attempts to optimize all tasks for all vehicles in accordance with these inputs. Replanning typically occurs in three different instances – pre-mission planning (Howe et al., 2000; Kramer & Smith, 2002; Kaber, Endsley, & Onal, 2000; Scott, Lesh, & Klau, 2002) that creates a schedule prior to any vehicle activity, event-based replanning (Roth et al., 2004; Ruff et al., 2002; Cummings & Mitchell, 2005; Ganapathy, 2006; Kaber et al., 2000) that requires new schedules after certain events (like failures) occur, and dynamic replanning (Cummings et al., 2012; Ryan et al., 2011; Cummings, Nehme, & Crandall, 2007; Ryan, 2011; Maere, Clare, & Cummings, 2010), where supervisors create new schedule periodically to increase the overall performance of the system. The latter case is needed because the information and uncertainties in information used to create the original schedule evolve over time; the expected performance of the schedule degrades over time as new uncertainties emerge and the environment as a whole changes.

Very few instances of field testing of human supervisory control systems have occurred because, in most cases, the level of autonomy required of such systems is not yet feasible for real world conditions. In the few cases that have been observed, they have not included any empirical comparison to other control architectures. The papers referenced within this section provide a rich set of examples of how these systems can be architected, along with an understanding of the level of accuracy required for these systems to be functional, but there is no empirical data with which to ground a definition of vehicle task performance in the world. The data provided in this section can be used to create a general model that describes how these various unmanned vehicle architectures are defined within the world. This framework can then serve as a guide for simulation development and help to clarify what the differences in performance among systems might be. Such a model is described and constructed in the next section, followed by a discussion about how such a model might be translated into a simulation environment.

2.2.5 A Model of Control Architectures

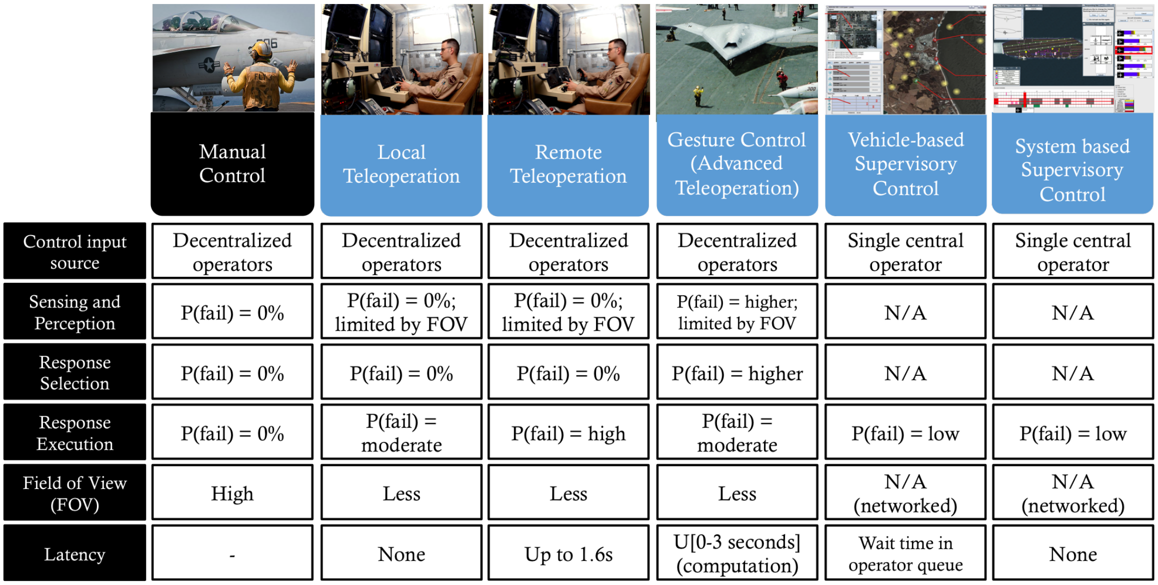
This section began with an overview of several different forms of “control architectures” defining how human operators and supervisors control Unmanned Vehicles (UxVs), describing changes in the architectures in terms of operator location, operator control inputs, and operator input modality (Fig. 2-2). Those sections provided details on the different types of architectures, describing variations in performance and implementation that result from the changes listed in Fig. 2-2. Where possible, the expected performance of the architecture was described quantitatively in terms of prior research results. However, a significant missing piece in prior research is an empirical comparison of the different control architectures to one another. These comparisons have typically been pairwise in nature, with no research located that provides comparisons of the full spectrum of unmanned vehicle control architectures. The near complete lack of empirical tests of HSC systems in field trials also leaves a substantial gap in comparing system performance. However, we can still describe relative comparisons between the different control architectures given a basic understanding of the tasks required of HMUE domains.

In this dissertation, five specific unmanned vehicle control architectures are defined for the aircraft carrier flight deck based on the overview provided in the previous sections. In addition to a baseline of Manual Control (MC), models of Local Teleoperation (LT), Remote Teleoperation (RT), Gestural Control (GC), Vehicle-Based Human Supervisory Control (VBSC), and System-Based Human Supervisory Control (SBSC) are created and described in terms of their differences from MC. LT systems describe systems that are teleoperated, but where communications latencies are very small and not noticeable by the operator. These operations are largely the same as Manual Control, but will have restrictions on the operator's ability to observe the world through the vehicle's cameras. RT systems do include a noticeable latency in communications, which should result in higher error rates in task execution as well as other changes in operator strategy in executing tasks. GC is utilized as the advanced form of teleoperation as opposed to voice-recognition, as the significant noise level of aircraft carrier operations makes any vocal communication extremely difficult. GC replaces the human operator entirely, but should result in increased chances of failing to sense the environment, of failing to select the correct response given certain inputs, and in executing tasks in the world.

For each of these three unmanned vehicles control architectures (LT, RT, GC), control inputs are provided by individual, decentralized operators as opposed to the single, centralized controller that provides control inputs to all vehicles in Vehicle-Based Human Supervisory Control (VBSC) and System-Based Human Supervisory Control (SBSC) systems. VBSC operators assigning tasks to individual vehicles in series and SBSC operators utilizing a planning and scheduling system to replace all tasks for all vehicles simultaneously. For VBSC systems, a model of operator interaction times for defining tasks must be included, as well as a queuing policy for prioritizing aircraft.

Figure 2-6 provides an overview of the six key variables (rows) that define the differences between the manual control and the five unmanned vehicle control architectures (columns in the figure). Chapter 5 will re-examine these variables and provide additional detail on their definitions for models of flight deck operations. The source of control inputs appears in the first row in the figure, defining whether commands are provided by a central operator (the two supervisory control systems) or from decentralized, individual operators. The ability of a vehicle to sense and perceive the world appears in the second line, given as a probability of failing to recognize specific elements in the world. For MC, LT, RT, and GC systems, this is a primarily visual process requiring a human pilot and/or a set of cameras on the vehicle. The third row in the figure describes how, based on that visual information, the operator or vehicle's ability to select the appropriate response based on those inputs. The two supervisory control systems would be networked to the central system and acquire information on the world through other means, thus making the sense/perceive and response selection elements not applicable. The fourth line describes the ability of the vehicle in the world to execute a task once selected for execution, in terms of a probability of failing to properly complete that task.

The final two rows are physical constraints based on the design of the system. The Field of View variable relates to systems relying on human eyesight (MC) and/or using cameras onboard the



	Manual Control	Local Teleoperation	Remote Teleoperation	Gesture Control (Advanced Teleoperation)	Vehicle-based Supervisory Control	System based Supervisory Control
Control input source	Decentralized operators	Decentralized operators	Decentralized operators	Decentralized operators	Single central operator	Single central operator
Sensing and Perception	P(fail) = 0%	P(fail) = 0%; limited by FOV	P(fail) = 0%; limited by FOV	P(fail) = higher; limited by FOV	N/A	N/A
Response Selection	P(fail) = 0%	P(fail) = 0%	P(fail) = 0%	P(fail) = higher	N/A	N/A
Response Execution	P(fail) = 0%	P(fail) = moderate	P(fail) = high	P(fail) = moderate	P(fail) = low	P(fail) = low
Field of View (FOV)	High	Less	Less	Less	N/A (networked)	N/A (networked)
Latency	-	None	Up to 1.6s	U[0-3 seconds] (computation)	Wait time in operator queue	None

Figure 2-6: Variations in effects of control architectures, categorized along axes of control input source, sensing and perception failure rate, response selection failure rate, response execution failure rate, field of view, and sources of latency.

vehicle (LT, RT, and GC), describing how much of the environment is visible at any given time. This parameter is also not applicable for VBSC and SBSC systems, given their reliance on the central network. The final variable corresponds to the types of latencies present in vehicles operations. RT systems include the latency in signal communications, GC systems include computational delays as well as the potential for repeated time penalties due to failures, and VBSC include delays due to waiting in the operator queue.

The remaining knowledge required to model these various control architectures is a representation of the Heterogeneous Manned-Unmanned Environment (HMUE) and its safety protocols, linking different agents and actors together through the passage of information between agents and elements of the system. In this case, the transmitted "information" includes direct commands from the operator to the interface then to the vehicles, interactions between humans and vehicles while collaborating on tasks, and the effects of tasks on the world at large. Figure 2-7 contains this aggregated model describing Heterogeneous Manned-Unmanned Environments (HMUEs), based on the descriptions of the control architectures in the previous sections and an understanding of the environments in which they work. The intent of this framework was not to abstract away from the details of these environments; rather, the goal was to include all details together into a single cohesive framework that can capture any of the control architectures described in this chapter.

The core of the diagram is the Human Supervisor block in the upper center of the figure. The Human Supervisor communicates with the Supervisory System block and its potential components in a process of Activity Selection, defining tasks and actions for vehicles to take. In this, the operator

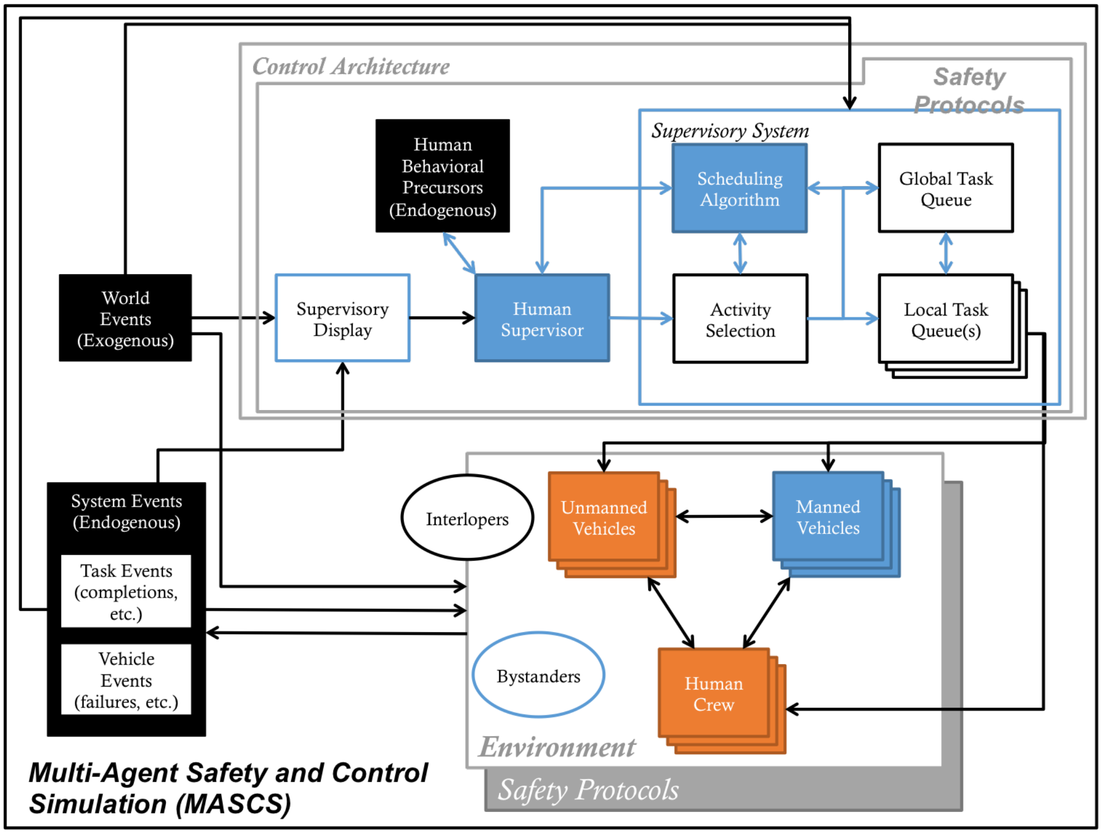


Figure 2-7: Aggregate architecture of manned-unmanned environments.

performs the same processes described in Fig. 2-3 and Fig. 2-4, acquiring information on the current state of the world and using it to determine tasks that agents under the supervisor's control should accomplish. The contents of this supervisory control block may vary substantially, given the form of control architecture involved. All, broadly, involve the definition of "Activities" for agent to execute, an assignment of these activities to individual Task Queues, which the agents then independently execute in the world. For Local and Remote Teleoperation, individual UAV operators determine activities for their vehicles, hold the tasks within their own mental task queue, and are responsible for executing the tasks in the queue. In Supervisory Control systems, this process might instead involve the operator defining Activities for the system, which are stored in a Global task queue before being assigned to individual vehicles by a planning algorithm. In a VBSC system, the operator may perform a similar process without the aid of the planning algorithm.

Once activities and tasks are assigned, the task queues are executed by Human Crew, Manned Vehicles, and Unmanned Vehicles in the world (this is a replica of Fig. 1-1 in Section 1.1). This occurs collaboratively within one another within the Environment, under the auspices of the Safety Protocols that govern their behavior. Additionally, though not discussed previously, are two additional groups of agents whose are not actively assigned tasks: "Bystanders" that are expected to be involved in operations and “Interlopers” that are not. It is up to the humans and vehicles in the system to observe and account for these other classes of agents. As agents complete tasks in the Environment, they generate System Events that have ramifications in the world — tasks are completed, resulting in changes in the world and in the system itself — and are affected by Vehicle Events, such as failures, that affect a vehicle’s ability to perform tasks. World Events, exogenous to the system and outside the control of the Supervisor or the Agents in the Environment (such as weather, changes in operational priorities, etc.) might also occur and lead to changes in the system. Both System and World Events must also be displayed to the Supervisor so that they might employ their best judgment in guiding the system.

This framework should be considered as a roadmap for modeling a number of HSC systems within a simulation environment — to appropriately examine a control architecture’s performance, the appropriate elements from the framework diagram in Fig. 2-7 should all be modeled within the simulation environment. In general, this diagram is a recharacterization of Sheridan’s original HSC loop shown in Fig. 2-4, with extensions made to include various possible pathways for assigning tasks to vehicles and including the interaction of the vehicle with the world, both necessary for correctly modeling the effectiveness of unmanned vehicle systems in the real world. Key to this diagram is the idea that control of unmanned vehicles is a process of transferring *information* in terms of commands, activities, states, and performance data. Given information on the state of the world, supervisors assign tasks to vehicles, which require other environment on the state of the world in order to execute their tasks appropriately. These vehicles work independently, each making its own observations of the world and executing tasks individually based on these observations. Human collaborators and manned vehicles work similarly, simultaneously, and in a decentralized fashion. Because these actions occur in the same shared space, the actions of one vehicle affects the actions of all others. Safety protocols are also put in to place, affecting the behavior of all vehicles in the system and forming an additional constraint in the logic and actions of agents. A proper simulation of these environments should include all of these features, adequately replicating how agents in the world independently observe, decide, and act. In total, these suggest the use of an Agent-Based Simulation (ABS) environment, whose properties and past work are discussed in the next section.

2.3 Modeling Heterogeneous Manned-unmanned Environments

The previous sections have outlined how both safety protocols and control architectures act to affect the performance of actors in a Heterogeneous Manned-Unmanned Environment (HMUE). Safety protocols serve to constrain behavior in order to preserve a safe system state, while control architectures change the level of performance of the vehicles within the environment, with or without the presence of the safety protocols. The combination of these two factors contributes to the performance of the entire system. Because each of these elements affects vehicle behavior, the type of simulation environment used in modeling HMUEs must also address individual vehicle behavior. This section discusses the applicability of Agent-Based Simulation (ABS) modeling to model human and unmanned vehicle behavior in Heterogeneous Manned-Unmanned Environment (HMUE).

ABS modeling is a relatively new modeling technique that focuses the on the actions of the individual, specifically on decision making, motion, and situational awareness in the world (Bonabeau, 2002). ABS models are based on the assumption that if the behavior of actors and their interactions with the world and with one another are accurately modeled, the simulation should evolve in the same fashion as in the real world. ABS methods are most appropriate when the actions of individuals have significant repercussions within the system, either in terms of motion in the world or in terms of strategic decision-making. In such systems, developing closed-form solutions that describe the interarrival rates at processing servers (as required for discrete-event simulations) may be difficult, if at all possible. Furthermore, in such systems, the interarrival rates may not be independent and identically distributed, a primary assumption in the development of the model.

Similar issues are faced with systems dynamics models; mapping the complex interactions between agents physically in the world to the stocks and flows required to define the model may be incredibly difficult. Even if these could be accomplished, it is difficult to know how changes to the parameters that describe unmanned vehicle control architectures alter these characteristic, closed-form equations for different systems. Agent-based models offer an alternative approach and have seen wide use in the biology, economics, and sociology domains due to their ability to capture the emergent behaviors that occur in these systems (Epstein, 1999).

The nature of the safety protocols (which affect individual decision-making, motion, and behavior) and control architectures (which affect individual behavior and ability to follow safety protocols) in HMUEs makes them candidates for modeling in an agent-based simulation. The remainder of this section reviews a variety of studies that involved modeling aspects of HMUEs in order to demonstrate the utility of ABS models for this dissertation. This is not intended to be a comprehensive review of the content of agent-based models nor of techniques in their development; for the latter, the reader is directed to Heath et al., who provide a review of 279 ABS over the decade ending in 2008 (Heath, Hill, & Ciarallo, 2009). The intent, rather, is to provide a review of studies that demonstrate similar characteristic to the HMUEs of concern in this work; the first section reviews agent-based models of both human and unmanned vehicle behavior, followed by a review of agent-based models for testing the effectiveness of safety protocols.

2.3.1 Models of Human and Unmanned Vehicle Behavior

ABS models have been used extensively in the biology, economics, and sociology domains because of their ability to independently model human behavior and demonstrate the nonlinear and emergent properties expected from these domains (Bonabeau, 2002). Agent-based models have also appeared within the health (Kanagarajah, Lindsay, Miller, & Parker, 2008; Laskowski & Mukhi, 2008; Laskowski, McLeod, Friesen, Podaima, & Alfa, 2009), automotive (traffic) (Peeta, Zhang, & Zhou, 2005; Eichler, Ostermaier, Schroth, & Kosch, 2005; Kumar & Mitra, 2006; Lee, Pritchett, & Corker, 2007), and military fields (Gonzales, Moore, Pernin, Matonick, & Dreyer, 2001; Bonabeau, Hunt, & Gaudiano, 2003; Lauren, 2002) for the testing of different tactics, behaviors, and system capabilities. General models of human crowd motion have been developed (Hu & Sun, 2007; Carley et al., 2003; Patvivatsiri, 2006; Helbing et al., 2000) in addition to models of human decision-making (Sycara & Lewis, 2008; Edmonds, 1995; Macy & Willer, 2002; Tesfatsion, 2003; Edmonds, 2001).

Models examining human behavior have typically used two different frameworks: explicitly changing the behavior of individual actors or changing the environment in which actors behave (or both). The latter may include physical changes to the environment, or changes in the number of agents, resources, of activities that are available in the system. Studies in the health field for emergency room performance (Kanagarajah et al., 2008; Laskowski & Mukhi, 2008; Laskowski et al., 2009) have addressed how different staffing protocols affect the ability of the ER to process patients. Studies in the automotive (traffic) and aerospace (air traffic control) domains (Peeta et al., 2005; Eichler et al., 2005; Kumar & Mitra, 2006; Lee et al., 2007) have examined how changes in both the behavior and the awareness of human operators affect accident rates. Models of airport terminals (Schultz, Schulz, & Fricke, 2008) and emergencies in buildings and aircraft (Zarboutis & Marmaras, 2004; Sharma, Singh, & Prakash, 2008; Pan, Han, Dauber, & Law, 2007; Batty, Desyllas, & Duxbury, 2003; Chen, Meaker, & Zhan, 2006; Hu & Sun, 2007; Carley et al., 2003; Patvivatsiri, 2006) have examined how changing the physical environment affects the ability of individuals to move through the environment quickly and efficiently.

Military research has addressed similar elements, examining decision-making and performance in novel command and control methods (Gonzales et al., 2001; Bonabeau et al., 2003), military operations (including operations-other-than-war, OOTW; see (Milton, 2004; Hakola, 2004; Efimba, 2003; Erlenbruch, 2002) for examples), and team performance (Sycara & Lewis, 2008). These studies have all addressed understanding the behavior of actors in the world under circumstances not easily observable in the world. A similar task is done in the work in this dissertation — modeling HMUE environments for the purposes of investigating the effects of varying safety protocols and control architectures.

These studies have not, however, addressed any combination of UAV control architecture or safety protocols, although each of these has been modeled individually. Agent-based modeling of UAVs has been used to validate implementations of novel control or collaborative sensing algorithms (Lundell, Tang, Hogan, & Nygard, 2006; Rasmussen & Chandler, 2002; Liang, 2005; Chen & Chang, 2008; Mancini et al., 2008; Cicirelli, Furfaro, & Nigro, 2009; Rucker, 2006). These studies have primarily addressed systems involving the collaboration of multiple vehicles in performing system tasks, such as searching for and tracking targets. Another common, and related, area of interest is in the modeling and control of UAV "swarms" (large, coordinated groups of robots coordinating on a specific task) (Ho, 2006; Corner & Lamont, 2004; Munoz, 2011; Vaughn, 2008). Both swarms and cooperative search require agents to sense the world and make individual decisions, making them candidates for agent-based modeling. In cooperative search, agents make decisions on how best to achieve the overall system goal via the partitioning of tasks among the vehicles in the system (for example, tracking multiple targets individually). Swarms typically consider how best to collaborate on a single task, such as tracking a single target over many hours. Agent-based models have also been built to test unmanned vehicle survivability (McMindes, 2005; Weibel & Hansman Jr., 2004; Clothier & Walker, 2006), single UAV search (Schumann, Scanlan, & Takeda, 2011; Steele, 2004), and the performance of intelligent munitions (Altenburg, Schlecht, & Nygard, 2002), each of which involves modeling the effectiveness of individual awareness and behavior.

Human interaction with unmanned vehicles is an additional domain whose characteristics make it a candidate for ABS modeling. However, little work has been done on this topic overall. The studies that have been observed to include human models of reasonable fidelity fall into two categories – discrete event models for cognitive processing or mathematical estimates of how many vehicles a human operator can manage (the "fan-out" problem). The first topic attempts to build models of how human operators cognitively interact with unmanned vehicle control systems, using data from experiments to build representative models of behavior that can be repeatedly tested (Nehme, 2009; Mkrtchyan, 2011; Mekdeci & Cummings, 2009). The latter topic attempts to build definitive equations that describe the maximum number of vehicles a human operator can attend to in the environments (Olsen & Wood, 2004; Cummings et al., 2007; Cummings & Mitchell, 2008). However, none of the references listed here include any concern for safety protocols active in the environment, nor do they address alternate vehicle control methods or human interaction outside of control. Agent-based models addressing safety protocol performance are discussed in the next section.

2.3.2 Agent-Based Simulations of Safety Protocols

The action of safety protocols to modify actor behavior is especially appropriate for agent-based models, but few simulations have been built for the exploration of the effectiveness of safety protocols. These studies come primarily from two environments: traffic safety (Archer, 2004; Wahle & Schreckenberg, 2001; Paruchuri, Pullalarevu, & Karlapalem, 2002; Yuhara & Tajima, 2006; Gettman & Head, 2003) and airspace safety (Pritchett et al., 2000; Xie, Shortle, & Donahue, 2004; Blom, Stroeve, & Jong, 2006; Stroeve, Bakker, & Blom, 2007). Traffic safety models have focused both on driver interactions at intersections (Archer, 2004) and highway traffic (Wahle & Schreckenberg, 2001; Paruchuri et al., 2002). Airspace safety analyses have dealt both the national airspace system in general (Pritchett et al., 2000) as well as more localized models of runway interactions at airports (Xie et al., 2004; Blom et al., 2006; Stroeve et al., 2007).

As with the references in the earlier section, these environments are all characterized by agents acting independently in terms of certain objectives. In these studies, human drivers are left to act independently, with researching examining how changes in the behavior of drivers affects accidents, near-accidents, and efficiency on the roadway (Archer, 2004; Wahle & Schreckenberg, 2001; Paruchuri et al., 2002; Yuhara & Tajima, 2006). Airspace safety simulations have also investigated how changes in behavior affect the system. In these cases, however, the behavior in question is not only that of the pilots in the aircraft (Blom et al., 2006; Stroeve et al., 2007), but also that of the air traffic controllers that provide structure to the environment in terms of scheduling and routing (Xie et al., 2004).

Interestingly, no studies have considered the testing of safety protocols for unmanned vehicles in heterogeneous manned-unmanned environments, even though the prior work referenced in this section provides successful examples of human, unmanned vehicle, and safety protocol modeling. The likely reason for the absence of literature on the modeling of safety protocols in HMUEs is the relative immaturity of these systems; few examples of these environments exist in the real world. Additionally, substantial hurdles also exist in terms of implementing these environments in public domains. In most cases, researching human control of unmanned vehicles or the performance of the autonomous control systems takes precedence over modeling the implementation of the vehicle in the system. However, modeling the behavior of the vehicle in the system operating under a variety safety protocols may better inform the design of the human interaction with and control algorithms embedded within the unmanned vehicles. If increased autonomy on the vehicle provides no tangible benefits in terms of safety or productivity, it may not be worth the investment on behalf of the institution funding the project.

2.4 Chapter Summary

This chapter first reviewed literature regarding the nature of safety protocols and how they are applied to unmanned vehicles systems. In this section, a set of four primary classes of safety “principles” were described, providing general guidance in the implementation of safety protocols for future systems. The second section reviewed different forms of human control of unmanned vehicles, moving from teleoperated to more advanced systems and comparing them to the baseline of manual human control. From these, a general model of performance was created, as well as a general framework describing the relationships and flow of information between elements of Heterogeneous Manned-Unmanned Environments (HMUEs). The last section reviewed how agent-based modeling can be used in creating simulation models of these HMUEs and prior work in modeling human/unmanned vehicle behavior and safety protocols usage and performance.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Model Development

This chapter describes the process by which aircraft carrier flight deck operations, a representative Heterogeneous Manned-Unmanned Environment (HMUE), is modeled through Agent-Based Simulation (ABS) techniques. The aircraft carrier flight deck is selected for two primary reasons. First, it is a form of heterogeneous *manned* environment, where human collaborators work in close physical proximity to manned aircraft and other manned vehicles. More specifically, aircraft on the flight deck are fueled, equipped, and given taxi instructions by crew traversing the flight deck on foot. These crew communicate with pilots, supervisors, and each other to coordinate activities on the flight deck. Second, the Navy is actively researching the development of unmanned fighter aircraft (and other unmanned vehicles) that would be integrated directly into deck operations. In fact, the Navy recently conducted shipboard flight testing of one such vehicle (Hennigan, 2013).

As described by Eric Bonabeau, agent-based modeling is as much a perspective as it is a methodology (Bonabeau, 2002). Whereas the use of discrete-event or systems dynamics models elicits specific representations of structure and constituent components, agent-based models may vary widely in content and in methods of application. Chapter 2 previously provided a list of examples of agent-based simulation models, ranging from transportation systems involving both ground and aerial vehicles to human decision-making to medical systems involving human motion and scheduling. What matters is that, in each case, the agent-based modeling paradigm views the world "from the perspective of its constituent units" (Bonabeau, 2002)— that the agents retain the same independence and employ similar decision-making strategies as the real-world entities they are intended to replicate. The agents are then placed into the simulated environment, given a set of goals (which may be self-generated during the simulation), and allowed to make decisions independently from that point forward. A canonical example is that of Epstein and Axtell's work on artificial ant colonies, the topic of their book "Growing Artificial Societies" (Epstein & Axtell, 1996). In Chapter 1 of their book, they describe that their models include: (1) "agents," the active entities in the system, comprised of a sets of internal states and behavioral rules; (2) the environment they act within; and (3) an object-oriented implementation, which preserves the independence of the agents and their behaviors. Rules defined for each individual artificial "ant" describe under what conditions they move within the world, based on the amount of sugar they "see" in nearby locations in the grid world they exist within. The world is initialized with both a population of virtual ants (which may appear in random locations) and a topography of virtual sugar that the ants seek out. From this initial configuration, the simulation is allowed evolve based solely on the rules defined for agents (the "ants") and the environment.

This general template — independent agents modeled in an object-oriented fashion, containing sets of states, parameters, behavioral rules, initialized into a world and allowed to evolve over time — is applicable to all ABS and forms the basis of the Multi-Agent Safety and Control Simulation (MASCS) model described in this work. The remainder of this chapter describes how the different elements of MASCS are modeled, addressing the environment, agents, states, parameters, and rules that populate the system. This modeling focuses on aircraft carrier launch operations, in which upwards of 20 aircraft are taxied from parking spaces to launch catapults with the assistance of multiple crew, and whose assignments are planned by supervisors on the flight deck. The next section provides a description of flight deck launch operations, characterizing the specific agents required, the tasks they are responsible for planning and executing, and important relationships and interactions amongst agents in the system.

3.1 Modern Aircraft Carrier Flight Deck Operations

The modern aircraft carrier flight deck is sometimes described as "organized chaos" (Roberts, 1990); it is a complex environment that requires a variety of simultaneous interactions between crew and vehicles within a shared and densely populated physical space, making it difficult to know how the actions of one individual may impact future operations. Multiple vehicles are active simultaneously, their actions governed by a network of human crew attempting to execute a flight plan developed by their supervisor, the Deck Handler. The pilots manning the aircraft receive instructions from crew on the flight deck via hand gestures; these crew dictate all tasks given to the pilot — move forward, stop, turn, unfold wings, open fuel tank, and many others — and work together as a team to move aircraft around the flight deck. The crew work in a form of "zone coverage," with a single Aircraft Director guiding an aircraft through his assigned area before passing it on to another Director in an adjacent zone on the way to the aircraft's assigned catapult. Four different launch catapults are available for use (two forward and two aft), but do not work completely in parallel and may be periodically inaccessible due to aircraft currently parked or transiting through the vicinity. Once an aircraft arrives at a catapult, the pilot and crew complete preparatory tasks for launch after which the aircraft is accelerated down the catapult to flight speed and into the air. Launch preparation requires that several manual tasks be performed in parallel by multiple crewmembers. The aircraft must be physically connected to the catapult, the catapult properly set for the aircraft's weight, the catapult then advanced to place the attachment under tension (to prevent the assembly from breaking when the aircraft is accelerating), and control surfaces on the vehicle tested.

Building an agent-based model of flight deck operations requires modeling each of these aspects — the movement of crew and vehicles on the flight deck, the execution of tasks (time to complete, failure rates, etc.), and decision-making on the part of both pilots and other crew. Table 3.1 provides a list of the primary actors involved in flight deck launch operations and their roles in executing operations, each of which must be properly modeled in order to replicate flight deck operations. Table 3.2 provides a list of the primary resources used by those actors, which may require rules describing when and how then can be used, or may be referenced by other decision-making rules. The remainder of this section describes these roles more fully and provides additional information regarding the important parameters and decision-making functions required by each.

A digitized map of the flight deck appears in Figure 3-1. Labels in the figure highlight the features of the flight deck relevant to launch operations. Four launch catapults (orange lines) are used to accelerate aircraft from a stationary position to flight speed and into the air. The catapults are numbered one to four, from right to left (starboard to port) and forward to back (fore to aft). A single landing strip (parallel green lines) is the lone runway for recovering aircraft. During landing ("recovery") operations, four steel cables, called arresting wires, are laid over the aft area of the landing strip; aircraft must catch one of these wires with their tailhook in order to land. Note that the landing strip occupies the same space as the aft catapults, Catapults 3 and 4. As such, only one form of operations can occur in this area at any time — the deck is either landing aircraft or launching them from this area, and switching between operations introduces a time cost due to the

Table 3.1: List of actors involved in flight deck operations and descriptions of their primary roles in deck operations.

Agent type	Role
Aircraft/Pilot	Primary agent in launch operations; goal of flight deck system is to launch all aircraft in minimum possible time
Aircraft Director	Relays instructions on task execution (primarily taxi) to Aircraft/Pilot
Deck Handler	Responsible for scheduling launch assignments and arranging aircraft on flight deck

Table 3.2: List of resources involved in flight deck operations.

Resource type	Role
Catapult	Resource used by aircraft in launching from the deck
Parking Spaces	Parking locations on the flight deck allocated to aircraft at the start of operations.

need to prepare the related equipment.

Each catapult is capable of launching a single aircraft at a time, but the four catapults are not able to operate in parallel for two reasons. The first is that a single set of crew operates each pair of catapults and can only manage one catapult at a time. The second, and more important, is that the catapults are angled towards one another — simultaneous launches would result in a collision and loss of both aircraft. This necessitates that the catapults alternate launches within each pair. The separate pairs can process in parallel, although they still try to avoid simultaneous launches (but a spacing of a few seconds is allowable).

Each catapult can queue two aircraft at a time in the immediate area — one on the catapult preparing for launch and a second queued safely behind a jet blast deflector that raises from the deck prior to launch. This process of queuing is an important feature in the planning and implementation of the schedule of launches. Queuing aircraft at catapults minimizes the time between launches, as the next aircraft is already at the catapult ready to begin its launch preparations. Queuing also forms a major constraint in terms of planning — once a catapult has two aircraft assigned to it, all available slots are filled. Aircraft needing to launch must be sent elsewhere, or must wait for an opening at this, or another, catapult.

The process of launching an aircraft requires that several tasks be done in parallel. Physically, the aircraft must taxi forward on to the catapult, aligning its front wheel (nose gear) with its centerline. The nose wheel is then physically attached to the catapult mechanism (the “pendant”); while this is occurring, other crew are confirming the aircraft’s current fuel level and total weight so as to properly calibrate the catapult for launch. The pilot will also complete a set of control surface checks to ensure that the ailerons, elevators, and rudder are all functioning. Once all of

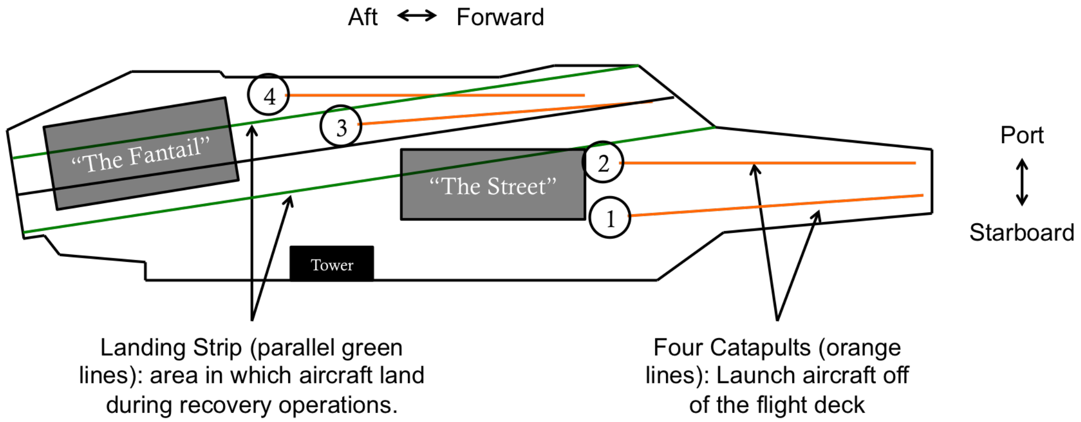

Aft ↔ Forward
4
3
2
1
“The Fantail”
“The Street”
Tower
Port
Starboard
Landing Strip (parallel green lines): area in which aircraft land during recovery operations.
Four Catapults (orange lines): Launch aircraft off of the flight deck

Figure 3-1: Digital map of the aircraft carrier flight deck. Labels indicate important features for launch operations. The terms “The Fantail” and “The Street” are used by the crew to denote two different areas of the flight deck. The Fantail is the region aft of the tower. The Street is the area forward of the tower, between catapults 1 and 2 and starboard of catapults 3 and 4.

these actions are complete, the catapult is triggered and the pendant begins pushing the nose gear and the aircraft forward up to flight speed. Once the launch is complete, the jet blast deflector is lowered, the next aircraft is taxied forward onto the catapult, and another aircraft is taxied behind the jet blast deflector. Within MASCS, catapults will be modeled as independent agents, just as the crew and aircraft. The nature of catapult queuing, constraints on launches between adjacent catapults, and the nature of the preparatory tasks and action of accelerating down the catapult are all important features that must be modeled within the MASCS environment.

The goal of the crew in managing the allocation of aircraft to catapults, as well as the management of the traffic patterns that result, is to make sure that the catapult queues are always fully populated. At minimum, a second aircraft should always arrive prior to the launch of the aircraft currently at the catapult. So long as this happens, the deck maintains a high level of efficiency, but ensuring that this occurs requires both planning of the layout of the aircraft prior to operations and proper management and assignment of aircraft once operations begin. Figure 3-2 shows one potential layout of aircraft on the flight deck prior to the start of launch operations; Subject Matter Experts (SMEs) report the general arrangement of aircraft in the picture to be fairly standard across the carrier fleet. Importantly, the Deck Handler attempts to make taxi tasks as easy as possible by relocating as many aircraft as possible from the interior of the flight deck, parking them on the edge of the deck, and parking them such that they are pointed towards the interior of the deck. However, parking spaces on the edge of the deck are limited, and some aircraft will be parked in places that obstruct taxi paths for other aircraft; the five aircraft parked in the interior of the deck in Fig. 3-2 are one example, as well as the aircraft at the top left of the image. This latter group is parked on top of one of the launch catapults (Catapult 1), preventing it from being used in operations. While this is not ideal, these aircraft will also be some of the first sent to launch, eventually freeing the catapult for use and increasing the flexibility of operations. The aircraft parked in the interior of the deck (in “The Street”) may also be launched early in the process.

Deck Handlers plan the parking of aircraft on the flight deck with the knowledge of the general priority in which aircraft should launch. Aircraft highest in the priority list are parked so that they are the first to depart their parking spaces with as clear a taxi path as possible. The locations and priority of aircraft, then, form the initial conditions of launch operations. A set of operational heuristics guides the manner in which areas of the flight deck are cleared during the launch mission. These heuristics were developed as part of previous work (Ryan, 2011; Ryan et al., 2014), and Figure 3-3 provides a general guide as to the order in which areas of the flight deck are cleared under the heuristics. As noted above, aircraft parked in obstructing areas are moved first (labeled as “1’s” in the figure). Clearing these areas early on clears more taxi paths on the flight deck and increases the efficiency of operations. At next highest priority is the region near the front of catapult 1 and an area in the aft of the flight deck called “The Fantail” (Fig. 3-1). This area sits directly on top

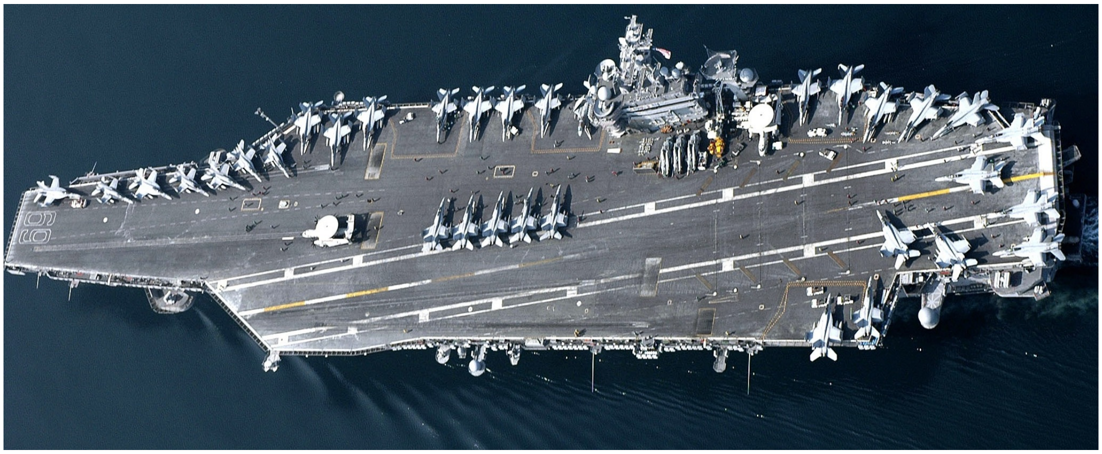


Figure 3-2: Picture of the USS Eisenhower with aircraft parked prior to launch operations. Thirty-four aircraft and one helicopter are parked on the flight deck (Photo by U.S. Navy photo by Mass Communication Specialist Seaman Patrick W. Mullen III, obtained from Wikimedia Commons).

of the landing strip. While the current focus of the MASCS model is not on landing ("recovery") operations, clearing this area is a priority for the crew. Clearing this area as early as possible allows any airborne aircraft that require an emergency landing to return to the carrier.

While the figure denotes the general order of clearance of these areas, it is not a strict ordering. Areas of priority 1 will be cleared first, based on the available catapults and the current geometry of the deck. Pending the current allocation of aircraft to catapults and where aircraft are currently parked, some assignments of aircraft to catapults may not be feasible. For instance, if aircraft are

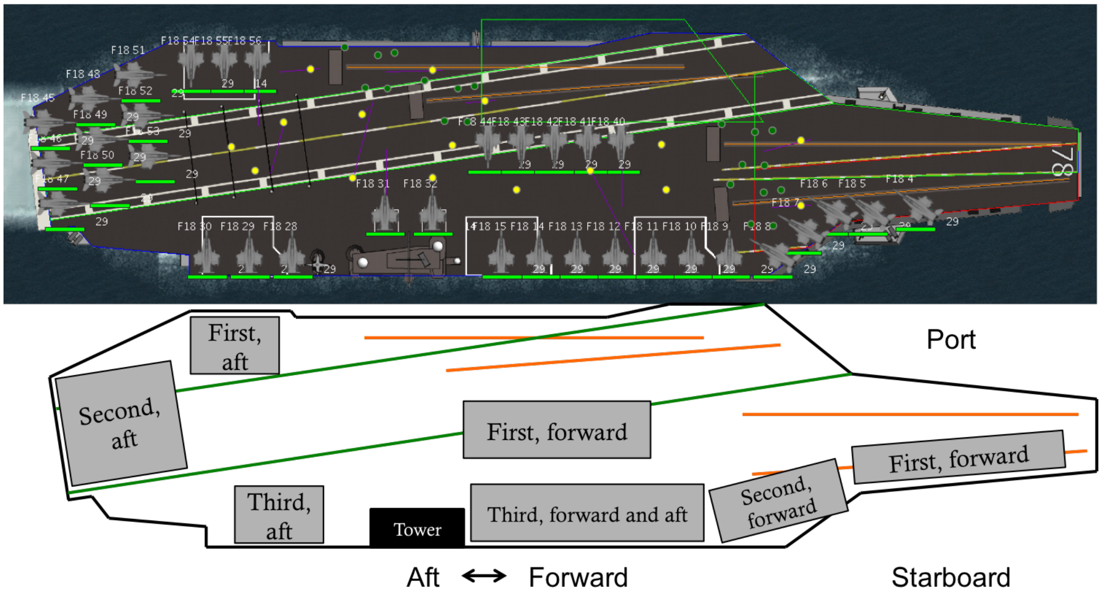


Figure 3-3: (top) Picture of the MASCS flight deck with 34 aircraft parked, ready for launch operations. (Bottom) General order in which areas of the deck are cleared by the Handler's assignments.

parked in The Street (Fig. 3-1), aircraft parked forward cannot be assigned to aft catapults. As such, the Handler would continue to assign aft aircraft from lower priority areas to the aft catapults to keep launches happening. If a catapult is open, the highest priority aircraft able to access that catapult is assigned. The Handler and his staff execute those heuristics dynamically as the mission evolves, reacting to conditions in real time so as to best compensate for the stochasticity and dynamics of operations. As such, these heuristics are one of the most important features of the MASCS simulation and its Deck Handler models. The locations of aircraft prior to launch operations, although a task of the Deck Handler, are initial conditions and could be modeled as static templates to provide the same initial conditions across multiple different scenarios.

While the Deck Handler is responsible for determining the parking locations of aircraft prior to launch operations and in managing the assignment of aircraft to catapults during operations, the Handler is not an active, moving entity on the flight deck. Rather, he remains within the Tower at deck level, utilizing a tabletop map of the flight deck and aircraft-shaped metal cut-outs to determine catapult assignments for aircraft. Given the Handler's assignments, it is the responsibility of members of the crew to facilitate the taxiing of aircraft from their parking locations and to their assigned catapults. The most important members of the crew for these tasks are the yellow-jerseyed Aircraft Directors, who provide all instructions (taxi commands and otherwise) to pilots during deck operations. These instructions are provided entirely through hand gestures, using a codified set of over one hundred individual gestures to describe all possible actions that can be taken (e.g. move forward, stop, turn left, or unfold wings). To provide some flexibility into operations, Directors work in a form of "zone coverage" in which the individual Directors each manage a specific area of the flight deck. Directors taxi an aircraft through their zone before handing off to a nearby Director in an adjacent zone. This minimizes the amount of travel that an individual Director must do, while also ensuring that Aircraft always have crew available nearby to provide guidance (that is, they are not waiting for a Director to return from the far end of the ship).

Pilots are not allowed to take any action independently, except for stopping to avoid an accident. If a pilot cannot see a Director, they are instructed to stop immediately. To maintain appropriate lines of sight, Directors align themselves to one of the outside wingtips of the aircraft some distance away from the aircraft, so that they are visible past the nose of the vehicle. This means that there is a defined area in the front of the vehicle within which a pilot is looking for his current Director. If the Director has not gained the attention of the pilot and is not within this area, the Director will move into this area as quickly as possible. While vehicles are turning, or while moving forward, Directors may also move in the same direction to maintain their alignment and spacing. Directors must also coordinate amongst the team to route aircraft through the deck, recognizing where aircraft are heading, which Directors are adjacent and able to accept incoming aircraft, and how to properly pass vehicles between one another. They must also monitor nearby aircraft and foot traffic to ensure that their aircraft is not taxiing toward a collision, or taxiing to a position in which they block another aircraft's taxi path. Along with the Deck Handler's heuristics, modeling this "zone coverage" routing is an extremely important feature of the MASCS model, requiring both an adequate modeling of how aircraft are routed in zone coverage, as well as how crew and pilots interact during the routing.

Each of the elements described in the previous section — aircraft, crew, and supervisory agents, their decision-making and coordination rules, the tasks that each perform both individually and while interacting with others — must be modeled in order to construct an appropriate agent-based model of flight deck operations. The next section describes this process of generating **Agents** that represent the human crew, aircraft (and pilots), the physical flight deck and its equipment, and the supervisory Deck Handler, **Tasks** that these agents perform, and **Decision-making** behaviors that each work through in the course of the mission. The section begins by describing the nature and modeling of an "agent" within MASCS.

3.2 Model Construction

As discussed in Chapter 2, agent-based simulation was chosen as the modeling methodology based on its focus in modeling decision-making and movement of individual actors and information flow between them, each of which is important in characterizing Unmanned Vehicle (UxV) control architectures and safety protocol application. These elements also play key roles in modeling crew and pilot interactions in the unstructured and stochastic environment that is the aircraft carrier flight deck. As noted at the beginning of this chapter, this requires adequately capturing the decision-making rules of the agents in the system, as well as the states and parameters required to execute the decision rules. For modern aircraft carrier flight deck operations, this modeling approach focuses on the launch phase of operations as described in the previous section, where sorties of 18-34 aircraft are taxied from their parking spots to one of four launch catapults. While there are other aspects of aircraft carrier operations that are potentially of interest, none require the same substantial degree of human interaction with unmanned vehicles and manned vehicles. Only launch operations provide the largest number of simultaneously active agents and subsequently a high level of interaction between agents, making it desirable for the forms of safety analyses that are a focus of this work.

Section 3.1 provided a description of the type of agents involved in launch operations, their roles in planning and executing the mission, and some of the general behavioral rules that govern the tasks they perform. Each of these aspects must also be captured within MASCS, with a key focus being on replicating the heuristics of agent behavior and their movement on the flight deck. To capture these aspects in an object-oriented architecture, MASCS was constructed using the Java programming language and the Golden T Game Engine (GTGE) Java library[1], which controls the updating of

[1] http://goldenstudios.or.id/products/GTGE/

agent and animates their motions on-screen. The object-oriented nature of Java allows agents to be defined independently, allowing rules and states to be defined individually different types of agents. Agent states may either be defined as static variables (numeric or Boolean), or may be randomly sampled from continuous distributions. These states are updated as the simulation executes, driven by the GTGE package; state updates may be based on elapsed time and a rate of change, or by logical rules that examine the states of the world or other nearby agents. For example, a catapult cannot operate if aircraft are in a specified caution area around the catapult; the catapult's update function continually checks for the presence of any aircraft in this area, halting operations if one appears. This is a key safety policy on the flight deck, but one that would not likely change so long as crew are involved in flight deck operations.

The previous section also provided descriptions about the actions taken by crew and vehicles on the flight deck. These exist alongside the states, parameters, and decision-making processes described earlier and form the third primary component of the MASCS model. While these elements are unique to each type of agent in the model, the tasks they perform are independent of any one agent. Actions are defined as independent objects that can be assigned to any agent in MASCS; rules within each Action determine, given the assigned agent, their current state, the current state of the world, and a goal state, what series of tasks must be accomplished to reach the goal state. Actions exist to define aircraft taxi tasks, refueling tasks, launch (both preparation at and acceleration down the catapult), landing, and various other events. Agents maintain a list of Actions that they must accomplish; as agents are updated by the main MASCS program, they in turn update their own Actions. When one Action completes, the next one begins. The small increments of time applied during each update and the rate at which updates occurs means that agent actions happen in parallel.

Many of the actions used to describe flight deck operations (taxi, launching, etc.) are categorized as "Abstract" actions, describing complex tasks occurring on deck. These Abstract actions can in turn be broken down into combinations of other elemental actions, termed "Basic" actions. These include behaviors such as "Move Forward," "Turn," or "Accelerate:" the smallest decomposition of physical tasks, barring the modeling internal cognitive processes of humans or individual parts and components within an aircraft. The relationship between these Abstract and Basic actions can be explained in the context of Figure 3-4, in which an aircraft parked on the starboard side of the deck is asked to taxi to catapult 3 and launch. The aircraft would be assigned a Taxi action, a Launch Preparation action, and a Takeoff action. The Taxi action would declare a goal state of reaching catapult 3; rules within the action would determine that, given the location of the aircraft, nearby crew, and the state of the deck, reaching catapult 3 requires four "Move Forward" tasks and three "Rotate" tasks in the sequence shown. These would be added into the queue of subtasks tin the Taxi action and executed in series. At the end of the last Basic action, the higher-level Taxi action checks to see if the vehicle is at the desired goal state: if so, the Taxi action ends. Otherwise, additional rules add in new Basic actions in order to reach the goal. The rules within the Abstract actions are adaptable, capable of automatically solving the motions required for any aircraft at any location on the flight deck to reach any of the four catapults.

As Basic actions are completed, they are removed from the Abstract action's queue and the next Basic action is begun. When the queue is empty, the Abstract action should be complete and, if the goal state has been achieved, the Abstract action is removed from the agent's queue and the next Abstract action (if any) is begun. In the example above, once the Taxi action completes, the Launch Preparation task would begin, with its rules defining the set of Basic actions that must occur to satisfy its goal. When an Agent's action queue is empty, it may either wait for further instructions to be assigned or its own internal rule base may generate a new action (such as returning to its original location). This process of defining and executing Actions occurs independently for each Agent in MASCS, and rules within each agent and task dictate when and how tasks are executed. These include how vehicles avoid collisions with other vehicles, how Aircraft Directors align to aircraft in order to be visible, and other aspects that prevent tasks from being executed as designed. A description of these rules, along with pseudocode for these rules and for several other elements, can be found in Appendix B.

The individual Basic actions are all based on time, and the execution of a Basic action completes when the agent has reached the expected execution time. For example, a motion-based task like driving forward in a straight line can be decomposed into a total time given the agent's current position, movement speed, and final destination. The launch acceleration task that models the motion of an aircraft down a catapult to flight speed can be solved similarly. For these motions tasks, updating the agent's position in the world is simply a function of applying elapsed time to the rate of travel and decomposing the movement in the x- and y-axes along the vehicle's current orientation angle. The vehicle's position is updated incrementally at very small time steps until the total elapsed time of the motion has elapsed. Other tasks are purely time-based and require no physical movement form vehicles; these are modeled as randomized parametric distributions, with

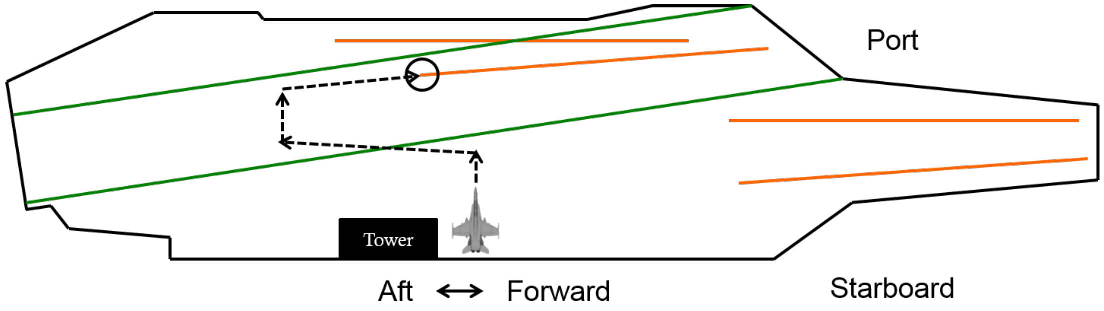


Figure 3-4: Example: path, broken into individual actions, of an aircraft moving from a starboard parking location to launch at catapult 3.

execution times independently sampled at the start of each task. The most important of these are the launch preparation task (tasks performed at catapult prior to launch) and the launch acceleration task (physically accelerating the vehicle down the catapult) described previously. Table 3.3 notes the parametric distributions for these two tasks, and Appendix C contains the observed times for completing these actions used to build these distributions (Tables C.2 and C.3).

In summary, the MASCS model of deck operations is comprised of a combination of independent agents and modularized action classes. Each relies on the definition of internal rules and states that govern their behavior. These may also reference states within other agents during execution in order to mimic information gathering and sensing of actions on the flight deck. Actions are assigned to vehicles, which are updated by the primary MASCS update routine and which in turn update action execution. A variety of rules dictate when and how actions can be executed, and higher-level rule bases dictate which of several lower-level decision-making processes should be utilized. The preceding section has given a broad overview of how MASCS functions but has not discussed the agent models in any significant detail. The next section describes the relevant rules and states required for each class of agent models individually, as well as more detailed descriptions of certain Actions used within the simulation.

3.2.1 Aircraft/Pilot Models

Section 3.1 described the basics of launch operations on the aircraft carrier flight deck. In the context of flight deck operations, much of the description centered on the interaction between pilots (seated within their aircraft) and the Aircraft Director crew that provided instructions to them. In order to model this interaction in the form of the Actions described in the previous section, this must be broken down into three components: (1) pilots' visual sensing and perception of Aircraft

Table 3.3: List of primary Actions used in the MASCS mode of launch operations.

Action	Description	Inputs
MoveForward (Basic)	Moves aircraft forward in a straight line between two points	Initial point, destination point, speed of motion
Rotate (Basic)	Rotates aircraft in place across a given angle	Initial angle, final angle, speed of rotation
Taxi (Abstract)	Navigates aircraft across flight deck. Iteratively defines motions and Director assignments	Initial point, destination point, assigned Director, speed of motion
Launch Preparation (Basic)	Replicates series of tasks required to prepare aircraft for launch (catapult settings, control surface checks, physical attachment to catapult)	Preparation Time ($N(109.648, 57.803^2)$); see Appendix C
Acceleration (Basic)	Replicates acceleration and motion of aircraft down catapult to flight speed	Acceleration Time (Lognormal, $\mu = 1.005, \sigma = 0.0962$); see Appendix C

Directors, (2) pilot comprehension of information and instructions and correct response selection, and (3) pilots' execution of the commanded task. Modeling these requires defining a series of states that relate to these motions (location, heading) and how instructions are being relayed (current Director assigned to aircraft), parameters related to the motion of the aircraft (such as the field of view of the pilot and range of speeds), and probabilities regarding failures. Additionally, other task parameters regarding the execution of the launch preparation and launch acceleration times must be defined, as well as logic regarding the avoidance of collisions on the flight deck under the Dynamic Separation safety protocol. A list of these features appears in Table 3.4, and details for each are provided in this section.

The ability to observe a Director is determined by two things; first, whether the Director is in an area visible to the pilot, and secondly, whether the pilot actually observes and recognizes the Director once in that region. The visible area, referred to as the field-of-view of the pilot, is defined as a state variable describing the angular area in front of the vehicle visible to the pilot. This is stored within the aircraft class and depicted graphically in Figure 3-5. While a pilot can turn their head to see a great deal of the deck area (close to 300° of radial area) or to signal agreement or disagreement with instructions, operationally, Directors will attempt to move into and align to an area in front of aircraft in order to be visible to the pilot within the aircraft. If they are not in this area, the pilot may not be able to observe their instructions.

Initially, field of view values were at a value of 180°, but animations of Aircraft Director alignments were clearly inappropriate for deck operations. The field-of-view variable for manual control was judged to perform appropriately at a value of 85°. As such, Aircraft Directors must be in front of the aircraft within 42.5° of either side of its centerline. This requires specific behaviors on the part of the Aircraft Directors (an alignment task)[2]. Once in this area, there is a chance as to whether or not the pilot visually acquires and recognizes the correct Director. To model such an event, a static variable relating to the chance of failure is defined for the aircraft agent and referenced when Director alignment tasks occur. However, while the chance of a failure of Director acquisition may exist, SMEs report that the likelihood of failing to observe a Director is very low and does not have a noticeable effect on operations. At this time, it is left at a 0% failure probability.

The ability of pilots to execute tasks involves both defining the time required to complete a task as well as the probability of failing to execute that task. Just as with the other failure rates, SMEs judged the likelihood of a pilot failing to properly align to the catapult as a very rare occurrence: less than once in every 200 launches. When asked, SMEs were comfortable with leaving this as a 0% failure rate. In addition to failure rates, models describing the time required to complete a task are also required. Two different types of tasks exist, relating to the Basic actions described earlier. Motion-based actions address vehicle movement on the flight deck — taxiing, turning, accelerating,

[2] Pseudocode describing both pilot behavior and Aircraft Director alignment appears in Appendix B

Table 3.4: List of important parameters and logic functions for Aircraft/Pilot modeling.

Feature	Description	Value
Location (state)	Current (x,y) position on deck	-
Heading (state)	Current orientation angle on deck	-
Current speed (state)	Current assigned speed	sampled from parametric distribution
Director assignment status (state)	Is a Director currently assigned to the aircraft	True/false
Catapult assignment status (input)	Is a catapult currently assigned to the aircraft	True/false
Catapult assignment (input)	To which catapult the aircraft should be taxied	1-4
Aircraft Director assignment	Which Director is currently responsible for giving taxi instructions	-
Current taxi command (input)	Current task prescribed by Director (turn, move forward, etc.)	
Speed (parameter)	Vehicle movement speed on the flight deck (miles per hour)	$N(3.5, 1.0^2)$
Turn rate (parameter)	Rate at which vehicles rotate during taxi operations	18^o per second
Field of View (parameter)	Area visible to vehicle and to which Directors align	$\pm 42.5^o$ from centerline
Probability of missing director command (parameter)	Likelihood of pilot failing to understand a task	0%
Probability of failing to execute task (parameter)	Likelihood of pilot failing to properly execute a task	0%
Director visibility (logic)	Is assigned Director currently in field of view; if not, stop immediately	-
Dynamic separation (logic)	Replicates actions to stop aircraft from making physical contact	One aircraft length forward, one aircraft width laterally
Forward collision threshold (parameter)	Minimum safe distance forward	One aircraft length
Lateral collision threshold (parameter)	Minimum safe distance laterally	One aircraft width

and stopping. Time-based tasks require no motion, but still require a specific amount of time to complete; this includes the launch preparation time and the launch acceleration time. Motion-based tasks require a definition of vehicle speeds, sampled from random distributions to reflect the variability in operator behavior on the flight deck.

Aircraft taxi speeds were reported by subject matter experts to be equivalent to brisk human walking speed; this is captured in MASCS as a Normal distribution with mean 3.5 mph and a standard deviation of 1.0 mph ($N(3.5, 1.0^2)$) within a specified range of [3.0, 4.0]. An aircraft's speed is sampled randomly from this distribution and assigned at the start of its motion task, then referenced by the Basic actions to update and animate the motion of the aircraft through the deck.

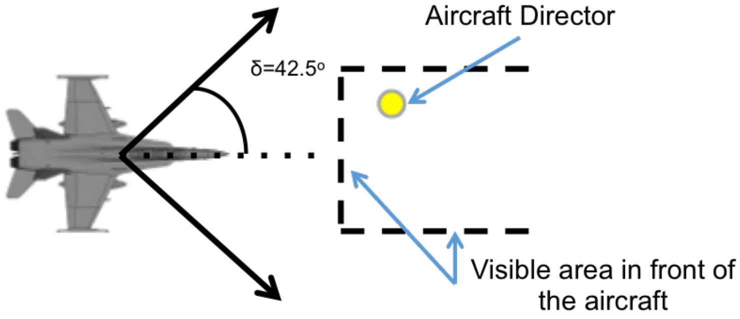


Figure 3-5: Diagram of the Director alignment process

The modeled speeds were compared to observations of flight deck operations to verify that they were relatively accurate; a more formal examination of taxi motion occurs in Chapter 4. Aircraft turning rates were defined as a static variable at 18 degrees per second (10 seconds to turn 180°) and were compared to empirical observations in a similar fashion; a table of the estimated turn rates for a set of observations can be found in Appendix C.

A final important aspect of the aircraft models is the inclusion of a collision prevention routine for the Dynamic Separation safety protocol, which models a pilot's actions to stop his aircraft from running into another vehicle. This action is dictated by a rule base contained within the vehicle model and referenced during Taxi action execution. A flow chart of the decision rules relating to this appears in Fig. 3-6. On execution, the rule base compares the position of the aircraft agent to nearby aircraft forward or adjacent to it. The distance between the two aircraft is calculated and decomposed into forward (through the aircraft's nose) and lateral (through the wing) components. If either distance is below a certain threshold (set to be one aircraft length forward or one aircraft width laterally), the vehicle is paused until the other vehicle moves out of the way.

Limiting the system to aircraft nearby and forward of the aircraft replicates the role of both the pilot in taking active control of the vehicle, as well as commands from the Directors to stop and wait for other vehicles. MASCS does not, at this time, make any differentiation between the two as both are reported by SMEs to be highly reliable — collisions during taxi are quite rare. However, the behavior of these collision avoidance routines, even if working effectively, does have repercussions on the flight deck. It is possible that chains of near-collisions occur: vehicle A might stop because of the location of vehicle B, which is in turn stopped due to vehicle C, which is waiting for another aircraft to launch from a catapult. While it effectively prevents collisions, it may have other, downstream effects on operations that are difficult to predict. It should also be noted that the collision prevention routine is supplemented by the dynamic routing of traffic on the flight deck by the Aircraft Director agents; directors will attempt to route aircraft in ways that do not conflict with one another. This decision-making process behind this is described in the following section, along with other aspects of the Director models.

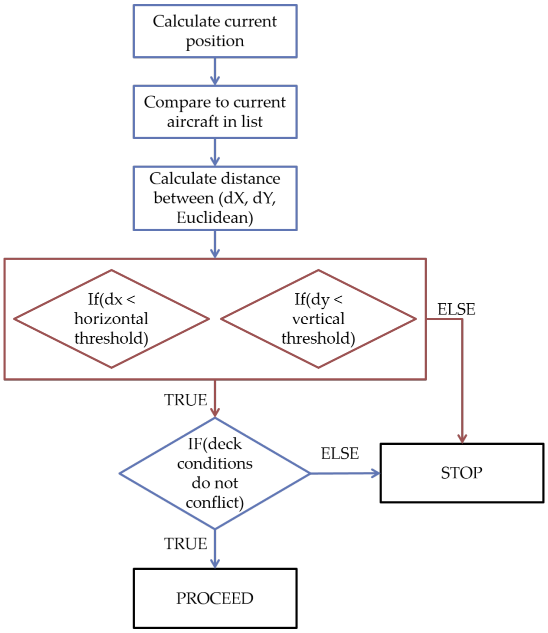

Calculate current position
Compare to current aircraft in list
Calculate distance between (dX, dY, Euclidean)
If(dx < horizontal threshold)
If(dy < vertical threshold)
ELSE
TRUE
IF(deck conditions do not conflict)
ELSE
STOP
TRUE
PROCEED

Figure 3-6: Flow diagram of the collision avoidance process.

3.2.2 Aircraft Director Models

Aircraft Directors are the primary members of the crew implemented on the flight deck, as they are the most active during launch operations. While there is a significant number of other crew involved in pre-launch preparations, they do not participate in launch activities and typically move to safe locations that do not affect taxi operations. States and inputs for Aircraft Directors relate to how they route aircraft through the flight deck, describing their current assignments, if they are available for aircraft to be handed off, and to what other Directors they can route aircraft. Directors also contain variables describing their speed of motion, identical to that of the aircraft models described in the previous section. A summary of these features appears within Table 3.5.

As aircraft receive their catapult assignments, they are assigned to Directors; this relationship is stored by both the aircraft and the Director (”Current aircraft assignment”). Once a Director takes assignment of an aircraft, they must move to a location in which they are visible to the pilot (Fig. 3-5). For the manual control operations modeled here, this is done by staying ahead and to the side of the vehicle's center line — SMEs note that Directors attempt to stay even with one wingtip of the aircraft, sufficiently ahead of the aircraft that they are not hidden from the pilot by the aircraft's nose. Observations of flight deck operations showed that Directors do follow this general rule for the majority of operations. A codified version of this rule based on geometry and the destination of the aircraft dictate where Directors should move is stored in the Director model. Additional rules determine where Directors should move if they are initially not visible by the pilots and how to stay in alignment while the aircraft turn and move forward. These additional rules were also based on

Table 3.5: List of important parameters and logic functions for Aircraft Director modeling.

Feature	Description	Value
Location (state)	Current (x,y) position on deck	-
Current speed (state)	Current assigned speed	sampled from parametric distribution
Assignment status (state)	Is Director currently assigned to an aircraft	True/false
Current aircraft assignment (input)	Which aircraft the Director is responsible for.	-
Zone/catapult assignment (input); current Aircraft assigned; current Aircraft destination (catapult)	-	
Crew network topology (input)	To which other Directors can an aircraft be passed	-
Speed (parameter)	Rate at which crew move on the flight deck.	$N(3.5, 1.0^2)$
Aircraft routing (logic)	Determines which Director an aircraft should be passed to.	-
Alignment to aircraft (logic)	Determine where Director should move to become visible to pilot.	-

observations of operations and discussions with SMEs.

While instructing pilots, Directors also monitor nearby traffic on the flight deck and attempt to appropriately manage multiple conflicting vehicle paths. Because pilots of manned aircraft have little knowledge of the catapult assignments of other aircraft on the flight deck (and may not even know their own), it is left to the Directors, monitoring nearby traffic and communicating with each other through hand gestures, to deconflict taxi actions across the flight deck. As described by SMEs, if an aircraft is already in motion, it has priority in moving on the flight deck and any new taxi commands should not interfere with those actions. However, if the two aircraft are sufficiently far apart that the second can be taxied clear of the area before the first arrives, Directors will do so in order to maximize the efficiency of operations.

Traffic deconflictions are replicated with MASCS through the use of a priority list of aircraft assigned to taxi through a specific area, with a set of rules governing when aircraft are added to the list, when they are removed, and in what cases no true conflict exists. The traffic management rules rely on a series of lists that maintain the order in which aircraft request to taxi through the area. The first aircraft in this list is by default allowed to taxi through the area; all aircraft not first in the list must wait for this aircraft to exit the area before continuing.

There are a variety of exceptions to this rule that can occur, however, and the decision rules examine each of these to see if no true conflicts exist. For example, if the first aircraft in the list is taxiing to a forward catapult, and it has already driven by the second aircraft in the list, that aircraft should be allowed to move. If it is heading forward, it will not catch up to the first aircraft, and if heading aft, cannot cause any problems. Otherwise, the vehicle waits until it either becomes the first aircraft in the priority list or satisfies one of the exceptions. A detailed discussion of these rules appears in Appendix B.

Once the Director taxis the aircraft complete through his zone, the aircraft must be handed off to a new Director in a nearby zone. Each Director maintains an understanding of how their zone connects to other zones on the flight deck (the input of the crew network topology). Figure 3-7 shows the topology of possible connections within the Director network on the flight deck. Rules dictate, given the available connections and the current destination of the aircraft, to which Director the aircraft should be sent next. If no Directors are available, the aircraft waits until one of the connected Directors finishes with the aircraft they are currently instructing and becomes available. A more detailed description of the logic for this model appears in Appendix B.

Together, this combination of rules — Director handoffs, alignment, and traffic management — govern the behavior of Directors in executing flight deck operations and facilitate the movement of aircraft from parking spaces to launch catapults. However, implicit in each of these actions is that the aircraft has already received an assignment to a launch catapult. These assignments are provided by the Deck Handler agent and its decision-making rules, described in the next section.

3.2.3 Deck Handler Model

The most important supervisory-level agent in the simulation is the Deck Handler that determines the assignments of aircraft to launch catapults on the flight deck. The codified rules that govern these decisions were generated in earlier work, based on interviews with SMEs regarding the typical strategies for allocating aircraft tasks (Ryan, 2011; Ryan et al., 2014). In this earlier work, SMEs with many years of deck experience were interviewed to understand how Deck Handlers allocate aircraft

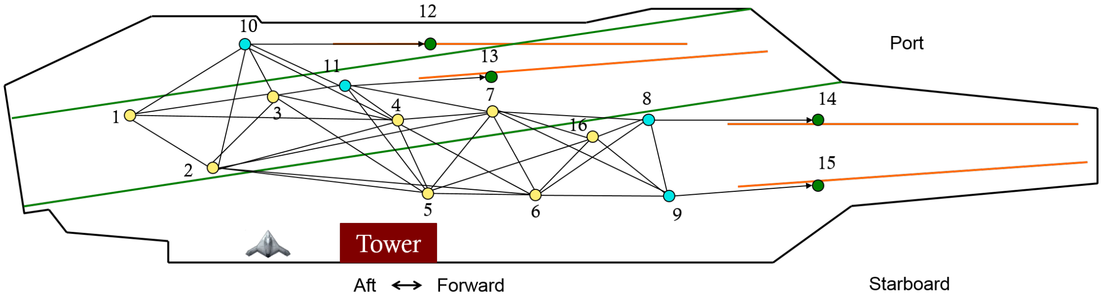


Figure 3-7: Map of the network of connections between Directors on the flight deck. Green dots denote Directors responsible for launching aircraft at the catapults ("Shooters"). Blue dots denote Directors that control the queuing position behind the jet blast deflectors ("Entry" crew). Yellow denotes denote other Directors responsible for taxi operations through the rest of the flight deck. Shooters and Entry Directors may also route aircraft through the rest of the deck if not currently occupied with aircraft in queue at their assigned catapult.

both at the start of a mission and in response to failures during a mission. In these interviews, SMEs were presented with different arrangements of the deck and different potential failure conditions (e.g., a catapult becomes unusable) and asked to describe how a new plan would be designed. From iterative explanations of different scenarios, a set of general rules of planning were created, describing the “rules of thumb” used by Handlers in planning operations under a variety of circumstances. These rules appear in Table 3.6 and when applied to a given scenario on the flight describe specific tactics for parking aircraft, assigning them to catapults, and routing them through the deck.

Rules 5 (make as many resources available as possible), 6 (maintain an orderly traffic flow), and 8 (park for maximum availability) combine to govern where aircraft should be parked prior to operations. As much as possible, aircraft are parked on the edges of the flight deck facing inwards in order to make the taxi process easier. Aircraft are parked in the interior of the deck (the Street) and on top of catapults 1 and 2 only when all other options have been utilized. Aircraft parked on catapults and in the interior regions will also be the first to be assigned in order to open up taxi paths on the deck and to make more catapults available. Because these areas are cleared first based on Rule 5, higher-priority aircraft in the mission will be parked in these areas based on Rule 8.

Rules 1 (translated as “simplify operations”) and 6 result in Handlers assigning aircraft to the nearest available catapults, ensuring the simplest and fastest taxi operations and minimizing any risk of aircraft interfering with each other’s taxi routes. However, the goal of Rule 4 (distributing workload) eventually means that aircraft must be assigned to taxi to catapults that are farther away than desired; it is better to use all available catapults than to minimize taxi path distances. Additionally, Rule 4 also means distributing priority aircraft around the deck to maximize their likelihood of being assigned to catapults. If the highest priority aircraft is parked forward, second-highest might then be parked aft, and so forth, to balance priorities across the areas. Rules 7 and 9 are not applicable to the current MASCS modeling as major failures and landing operations are not included in the simulation scenarios; however, for future extensions of the model, these heuristics should be included. Rule 3 is included, however, as it relates to the Dynamic Separation protocols described in the previous chapter.

Table 3.6: Heuristics of Deck Handler planning and re-planning developed as part of previous work (Ryan, 2011; Ryan et al., 2014)

General	Deck	Airborne
(1) Minimize Changes	(4) Evenly distribute workload on deck	(7) Populate landing order by fuel burn rate, fuel level, then misc.
(2) Cycle quickly, but maintain safety	(5) Make as many resources available as possible	(8) Park aircraft for maximum availability next cycle
(3) Halt operations if crew or pilot safety is compromised	(6) Maintain orderly an orderly traffic flow	(9) ”True” vs ”Urgent” emergencies in Marshal Stack

These rules were translated into sets of initial conditions for where aircraft should be parked and a set of logical rules that describe, given the current aircraft on the deck, their priorities, and their locations, where they should be assigned. The MASCS Deck handler also requires an understanding of what catapults are operational and whether a catapult has an open spot in its queue. The Handler routine always attempts to assign an aircraft to the nearest available catapult, given the current geometric constraints on the flight deck. The routine first attempts to assign aircraft to empty catapults (those without a prior assignment), before proceeding to catapults with only one prior assignment. A set of rules determines whether or not the vehicle has an accessible taxi path to the catapult (whether or not aircraft are parked in the interior of the flight deck, blocking the path).

The priority listing of aircraft included in the initial conditions is organized in accordance with Rules 5, 6, and 8, and when the Handler rules assign aircraft to the nearest catapults, it results in areas of the deck being cleared in a specific order. This is shown in Figure 3-8. The "Street" is typically among the first cleared. In a mission using 34 aircraft, its aircraft will largely be sent forward. Aircraft parked on the forward, starboard side of the deck will also be sent forward so as to remove them from the catapult they are blocking. Once these areas are cleared, aircraft parked in the forward are of the deck are cleared in a front to back fashion. In the aft area of the deck, the few aircraft parked near catapult four might be cleared first, followed by all of the aircraft parked in the Fantail area. Aircraft parked on the starboard side of the deck would be allocated last from this area. As operations progress, however, the Handler heuristics will assign aircraft whenever and wherever possible, regardless of region. These heuristics prioritize assigning an aircraft to a nearby catapult over taxiing an aircraft the length of the deck to reach an opening: it takes longer to taxi the length of the deck than to complete a launch at a catapult. Additionally, taxiing the length of the deck is disruptive to taxi operations along the entire deck and is avoided as much as possible.

A rigid definition of these rules breaks down when the highest-priority aircraft has no feasible taxi path to the available catapults. If this exists, it is because aircraft in the Street are blocking taxi routes and have divided the deck into forward and aft halves. Because this potential exists,

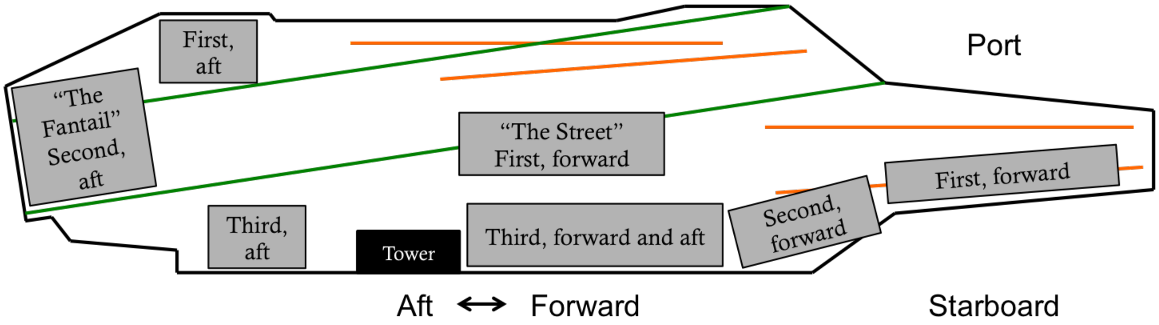


Figure 3-8: Map of the flight deck noting the general parking areas of aircraft (gray), the order in which they are cleared, and to which catapults (forward or aft) they are allocated.

the Handler rules are allowed to examine the first n aircraft in the list, beginning with the first (highest-priority) aircraft. The value n defines the lowest-ranked aircraft allowed to be assigned, and a value of five was used in the system. This accounts for full assignment of aircraft to a forward or aft catapult pair (four) plus making an additional assignment in the system (plus one). The arrangement of priority aircraft in the initial parking locations should ensure that there is a balance between the forward and aft sections of the flight deck and that some assignments are always possible. Table 3.7 contains a list of the key inputs to the Deck Handler model, which are explained below. A more detailed discussion of the logical rules comprising the Handler Allocation model can be found in Appendix B.

Once provided with a catapult assignment, the aircraft begins moving under the command of the Aircraft Directors. As described previously, the aircraft completes its taxi action when it arrives at its launch catapult queue, where it waits until being loaded onto the catapult. At that time, it begins launch preparation before completing the launch acceleration action. Parts of each of these aspects are handled by models for the deck and various deck equipment, described in the next section.

3.2.4 Deck and Equipment Models

The equipment agents of primary concern in launch operations are the four launch catapults and the landing strip. Additional constructs were created for the parking locations on the flight deck (gray areas in Fig. 3-8) to facilitate bookkeeping of data. For each of these elements, the most important state variable concerns their availability (whether or not they are currently in use) and

Table 3.7: List of important parameters and logic functions for Deck Handler modeling.

Feature	Description	Value
States	-	- Current list of unassigned aircraft (input)
Aircraft awaiting catapult assignments	-	
Current priority of each unassigned aircraft (input)	Position of aircraft in priority list	-
Current functional status of each catapult (input)	Is the catapult operational	True/false
Current availability of each catapult (input)	Number of aircraft in catapult queue; if < 2 can accept a new aircraft	True/false
Status of parking spaces 8/9 (input)	Is parking space 8 or 9 occupied by a parked aircraft	True/false
Lowest priority assignment allowed (input)	Lowest ranked candidate for assignment to catapult	5
Aircraft assignment (logic)	Attempts to assign highest priority aircraft to nearest available catapult based on where aircraft can taxi (status of parking spaces 8/9)	-

their operational status (whether or not they can be used). For parking spaces, this translates to tracking whether or not a vehicle currently occupies the parking space. Rules track and update these statuses according to the location of nearby vehicles. For catapults, the area in the immediate vicinity of each, referred to as the "foul area," should not be entered by any unauthorized crew or vehicles during launch processes; doing so requires an immediate halt of launch operations. These are noted by markings on the flight deck in the real world, converted into the simulated environment.

Other rules for catapults dictate how aircraft queue within the system and how launches are executed. Each catapult can queue up to two aircraft: one on the catapult and another parked behind. Each pair of catapults (one forward, one aft) alternates launches within the pair, but the two pairs operate in parallel. The first aircraft to arrive at a catapult within a given pair begins preparing for launch; any aircraft that arrive on the adjacent catapult do not begin preparing for launch until the first aircraft launches. States in the models define whether or not a catapult is actively "launching" and how many aircraft are in queue, as well as whether or not an active aircraft is stationed on ("occupying") a given catapult. An aircraft only begins preparing for launch if it currently occupies a catapult launch position and the adjacent catapult is not actively preparing to launch its own aircraft. Rules for the queuing process describe where aircraft should stop taxiing near the catapult; aircraft advance only if the catapult is not currently occupied. Table 3.8 provides an overview of the important inputs, states, and parameters of the catapult agents.

These rules, as with other rules and behavior models within MASCS, are all executed independently by agents in the system. The execution of a simulation run begins with a set of initial conditions, with agent decisions and reactions dictating how the simulation evolves from that point. The simulation execution process is described in the next section.

Table 3.8: List of important parameters and logic functions for Catapult modeling.

Feature	Description	Value
Operating (state)	Catapult is functional and can be used	True/false
Occupied (state)	Aircraft is currently stationed on catapult	True/false
Launching (state)	Aircraft occupying catapult is actively preparing for launch	True/false
Number of aircraft in queue (state)	Current number of vehicles in queue	[0,2]
Launching status of adjacent catapult (input)	Catapult can only be launching if adjacent catapult is not launching	
Queueing rules (logic)	Aircraft only allowed to taxi to catapult if queue size is less than two; aircraft not allowed to begin launch preparation unless adjacent catapult is not launching and not occupied	-

3.3 Simulation Execution

The previous sections have described how the individual agents within the MASCS model of flight deck operations make decisions within the world; all that is required is an initial configuration of the world, the agents active within it, their starting locations, and their initial goal states (if any). Also included are their order in the priority list, and crew assignments (to taxi zones or catapults). Appendix D provides example text from the configuration files, while Figure 3-3 previously showed the layout of the deck with 34 aircraft (that maximum number used). Figure 3-9, top shows another possible configuration of the flight deck with only 22 aircraft.

From the initial configuration and priority listing submitted in the initial configuration files, the Deck Handler's rule base begins to assign aircraft based on their priority in the launch order based on the current allocations in the system, the current locations of aircraft, and accessibility and geometry constraints on the flight deck. Doing so both assigns the aircraft to a Director and a Taxi action to the aircraft. The Director commands a motion action to the Aircraft, replicated within the Taxi action being executed. When the aircraft completes the taxi action, the Director's logic for aircraft handoffs is called and the aircraft is assigned to a new Director, who creates a new Taxi, and the process repeats. While the Taxi actions are executed, aircraft pilots employ their collision prevention routine to avoid collisions while Directors attempt to deconflict traffic around the deck.

When aircraft reach their launch catapults, they complete the Taxi action and begin executing

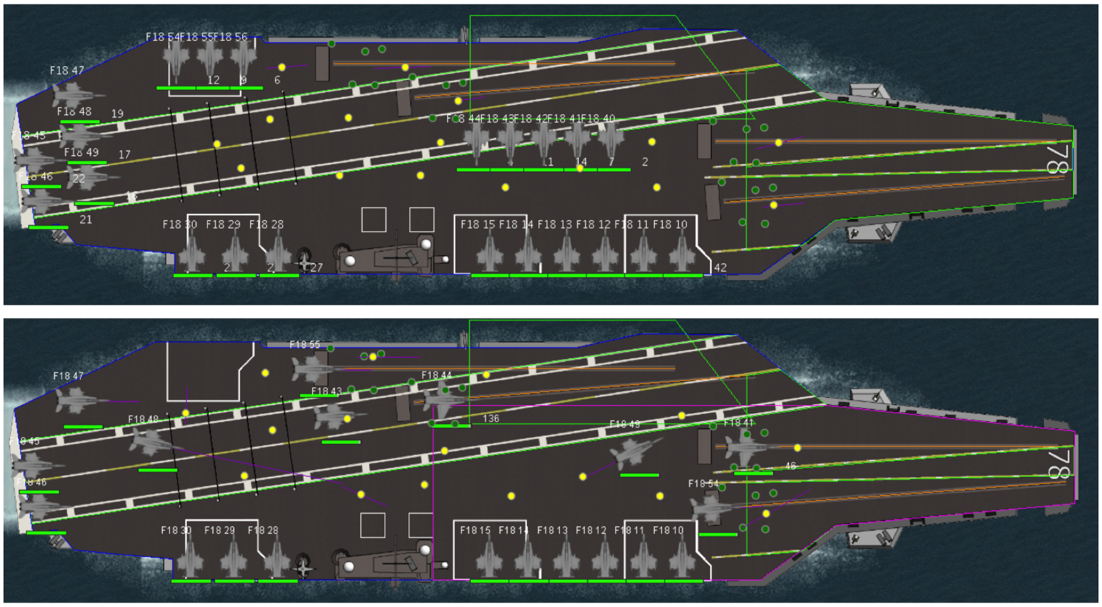


Figure 3-9: Top: Picture of the MASCS flight deck with 22 aircraft parked, ready for launch operations. Bottom: Picture of the MASCS flight deck with 22 aircraft parked, in the midst of operations. Three aircraft have already launched, four are currently at their designated catapults, four more are taxiing to or in queue at their catapult, and eleven are currently awaiting their assignments.

the Launch Preparation task, randomly sampling the execution time from the provided distribution. Once the required time has elapsed, the aircraft begins the Takeoff task, accelerating down the catapult in a time randomly sampled from its relevant distribution. All aircraft operate simultaneously, executing their own actions and employing their own rule sets individually. Figure 3-9, bottom shows one example of the 22 aircraft case in the middle of executing a run. Currently, aircraft at catapults 2 and 3 are executing their launch preparation tasks, while aircraft are taxiing on to catapults 1 and 4. Additional aircraft are queued at catapults 2 and 3, and two aircraft in the aft area of deck are taxiing forward to queue at catapults 1 and 4. The remaining aircraft parked around the edge of the deck will begin taxiing as soon as the Handler routine assigns them to catapults; the priority designations are such that the remaining aircraft should be assigned from left to right. The simulation completes when the last aircraft completes its Takeoff task at its assigned catapult.

3.4 Chapter Summary

This chapter has provided details on the nature of aircraft carrier flight deck operations and how these operations are transformed into rules and states for the constituent agent models. The chapter began with a discussion of the nature of agent-based modeling and its reliance on rules and states. This was followed by a discussion of flight deck operations that attempted to highlight the aspects that relied heavily on rule bases to define agent behavior. The architecture of MASCS, noting its uses of Action classes to organize and define interactions between agents, was then discussed. The final sections of the chapter described how the features of aircraft carrier flight deck operations were translated into specific sets of rules and states for different agent types, including details as the specific rule processes and how certain state variables were defined. The next chapter describes the process of validating this baseline manual control model described in this chapter. The manual control model described in this chapter then serves as the template for the development and integration of the unmanned vehicle control architectures as defined earlier in Chapter 2. A discussion of these models and their formal definition will occur in Chapter 5.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Model Validation

> "Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful."
>
> *Empirical Model-Building and Response Surfaces (1987)*
> George E. P. Box

The previous chapter provided an overview of modern U.S. aircraft carrier flight deck operations before describing how these operations could be modeled in an agent-based simulation. Chapter 3 outlined the different agents required for the simulation, their corresponding states, parameters, and behavioral rules, and the models of agent tasks that were required. These models were based on empirical data where possible, supplemented by information from Subject Matter Experts (SMEs) with experience in flight deck operations. This chapter describes how this simulation model, with its constituent agents, is tested against empirical data on flight deck operations. Demonstrating that the simulation produces output similar to that of the modeled system provides confidence that the simulation is a valid representation of that system: that the use of agent-based models is reasonable and that those agent models and their interactions have been appropriately defined. Demonstrating that the current Multi-Agent Safety and Control Simulation (MASCS) model described in Chapter 3 suggests that the modeling strategies are suitable for this environment and that the models are applicable to modeling unmanned vehicles in future tests. A valid simulation of current operations also serves as a future comparison point for other extensions of the simulation model, as the same output variables can be encoded into each.

The testing process for the MASCS simulation model was conducted in two different phases. In the first phase, the performance of a single aircraft proceeding from an initial parking location to a launch catapult was compared to empirical observations of the same task. The second phase of testing examined the performance of the simulation for missions using multiple aircraft, testing launch events using 18 to 34 aircraft with each taxiing from an initial parking location to a launch catapult. Tests within each of these phases addressed both the internal and the external validity of the simulation: how well it replicates empirical data on operations (external validity) as well as its response to changes in internal parameters (internal validity). These tests aim to provide confidence that the MASCS simulation mode defined in Chapter 3 is able to accurately replicate the behavior of flight deck operations. An additional review of the simulation with experienced Naval personnel provided additional confidence that the simulation was a reasonable model of operations, judging whether the animated evolution of the simulation environment was qualitatively similar to that of the real system.

However, limitations in the availability of data for flight deck operations means that fully validating all aspects of the model remains unachievable at this time. Several elements of the model as described in Chapter 3, such as the Deck Handler planning heuristics and other human behaviors, have no data available for comparisons and thus cannot be guaranteed to be valid. As such, MASCS can only be partially validated, with its outputs calibrated to the obtained performance data. The tests that were performed suggest that the simulation is a reasonable model of flight deck operations. This chapter discusses the validation tests that did occur, beginning first with a discussion of how agent-based (and other) simulation models are validated and how this applies to the MASCS model.

4.1 Validating Agent-Based Models

Many authors describe the process of simulation construction in terms of several phases, beginning with data gathering and proceeding through model construction, model verification, and model validation (Sargent, 2009; Carley, 1996; Law, 2007). The tasks of data gathering and model construction were discussed in the previous chapter, while this chapter focuses on the processes of model verification and validation. These two aspects describe similar processes at two levels of a simulation environment. Generally, a simulation can be viewed as an aggregation of submodels and function that interact to produce a single coherent simulation of a system. Verification is the process of ensuring that the submodels and subfunctions are properly defined and encoded; it is often described as "making sure you've coded the X right." This typically means ensuring that the mathematical equations governing behavior are correct and that any logical decision structures behave correctly. Model validation, then, is the process of validation is "making sure you've coded the right X," demonstrating that the output of the verified submodels matches that of the aggregate system. Correctly defining and implementing all submodels might still produce an invalid simulation if the inputs to and interactions between those models were incorrect.

All models are simplifications of reality; to properly model all elements of a real world system at their true level of complexity would require substantial time and resources. They are all "wrong" in some sense, and the question is not how accurate *can* they be, but how *inaccurate* they are allowed to be before they are considered inappropriate or not useful. This is often couched in terms of the external validity of the simulation — the accuracy of its outputs as compared to the outputs of the real system. Tests of validity can examine these outputs in a variety of ways, from point-to-point comparisons to comparisons of distributions of data. Which is chosen depends on the available data, the stochasticity of the real and simulated systems, and the precision with which data is collected. Additionally, validation can also address the response of the system to changes in input and internal parameters. This may either test the internal validity of the simulation — that small changes in parameters should not generate extreme responses — as well as external validity, if the response of the real system to the same changes in parameters is known.

Most texts that discuss simulation construction and validation focus on the use of systems dynamics and discrete event simulation, in which the movement from submodels and subfunctions to the full system simulation occurs in one step: the submodels are created and verified, then directly assembled into a model of the overall system. Agent-based models contain an additional intermediate level: submodels and subfunctions define *agent* behavior, with multiple agents interacting to create system behavior (Figure 4-1 provides notional diagrams for the three model types). It is useful in agent-based modeling to validate the individual agents' behaviors before testing that of the aggregate system. This demonstrates that the agents themselves are externally valid representations of real world actors and ensures that any future errors in the part of the simulation data comes from agent interactions. While accurate simulation results might be achievable even with faulty agent models, significant questions regarding the true external (or structural) validity of the simulation would exist. Both aspects of model validation — individual agents and the aggregate system model — for the MASCS simulation environment are described in this chapter, beginning in the next section with single aircraft agent validation.

4.2 Single Aircraft Calibration

Chapter 3 defined the list of agents required in the MASCS model and characterized the states, parameters, and decision-making rules required for each. These agent models must adequately replicate not only the decision-making of agents on the flight deck, but their motion on the deck and their interaction with other agents. Of primary concern is ensuring that models of pilot/aircraft behavior are correct, as these are the primary focus of launch operations and will be the primary comparison point in future Unmanned Aerial Vehicle (UAV) testing. This first section describes the calibration of the behavior of an individual aircraft executing a launch mission on the flight deck, using the minimum necessary number of crew. The ability of the MASCS simulation to replicate this task provides confidence that the models of individual vehicle behaviors and the characteristic distributions describing their speed of travel and the tasks they perform are all accurate. After

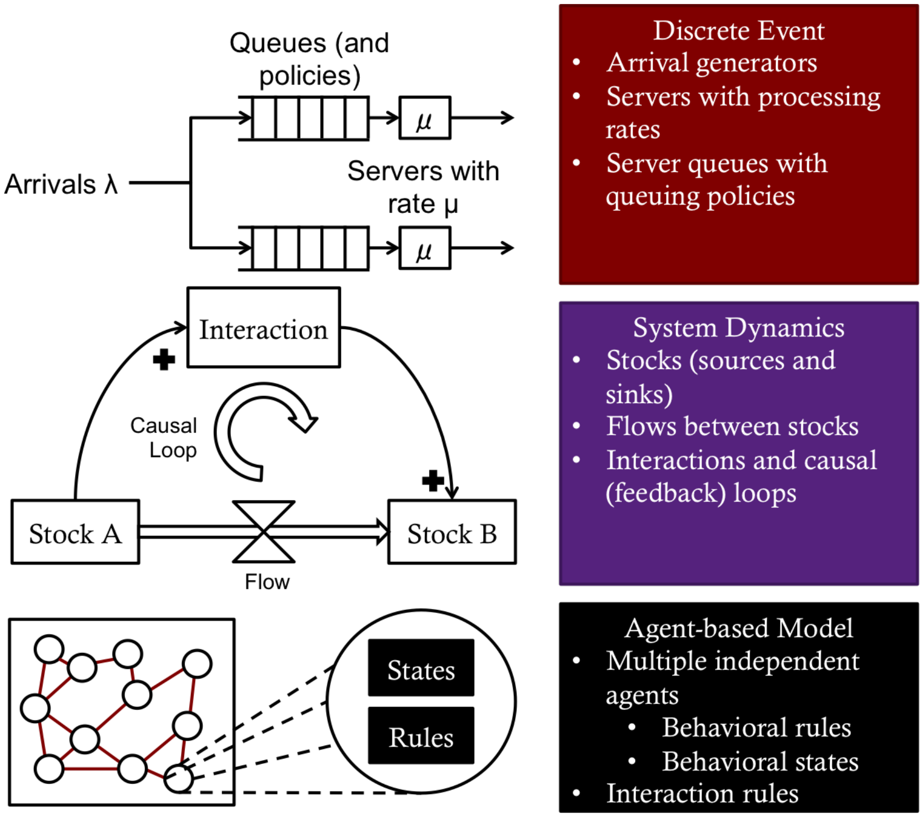


Figure 4-1: Simple notional diagrams for Discrete Event, Systems Dynamics, and Agent-Based Models.

these models are correctly calibrated, tests can then proceed to the tests at the mission level using multiple aircraft.

The tasks described in this chapter were made difficult by the fact that the United States Navy does not make flight deck performance data publicly available in large quantities. From discussions with SMEs, it appears that what data is collected on flight operations is done at the behest of officers and senior enlisted on each ship and is not forwarded on to the Navy or any of its research affiliates. As such, the data used in the tests in this chapter have been amalgamated from a variety of both government, military, and civilian sources.

For the single aircraft calibration testing, only a single data point for comparison was obtained, from a video uploaded to Youtube.com (Lockergnome, 2009). Even though a large number of videos of flight deck operations are available overall, only a small number provide full footage following an aircraft for more than the final takeoff task, and of those, most involve multiple aircraft taxiing simultaneously and interacting with one another. The video used in this testing is the only one discovered that follows a single aircraft from its initial parking place through final takeoff without interacting with other aircraft. No published performance data was located that provides any information on individual aircraft performance for a characteristic launch task. While this is not ideal, these observations can still be used in calibrating the Aircraft agent to ensure that it is not unreasonable.

In the recorded observation, the aircraft begins parked in the central aft area of the deck, then taxis to Catapult 3 (the interior of the two aft catapults) and launches. This first requires a *Taxi* task to move from parking to catapult (consisting of several individual Move Forward and Rotate actions), a *Launch Preparation* task, and a *Takeoff* task. Figure 4-2 shows the starting location and transit path of the vehicle. Times for each of these subtasks were calculated by reviewing the video and manually coding event times; they are referred to as the Taxi Duration, Launch Preparation Duration, and Takeoff Duration, respectively. Their values are annotated in the figure: 78 seconds to Taxi, 157 seconds in Launch Preparation, 3 seconds Takeoff by accelerating down the catapult, for a total of 238 seconds to complete the launch event. Note that the observed time to complete the launch preparation task was not included in the original observations used to generate the launch preparation time model (Appendix C, Table C.2). The observed value of launch preparation was also not included, but several observations of 3.0 second accelerations did occur.

If the MASCS model of operations is accurate, then the Taxi Duration, Launch Preparation Duration, and Takeoff Duration from the empirical observation should be reasonably similar to the average times produced by the simulation. At the same time, too closely matching this data suggests overfitting. Generally, being within 1 standard deviation of the mean value suggests that a data point comes from that distribution, given that 68% of all data points lie within one standard deviation of the mean. Higher quality data was available at the mission level (Section 4.3), and tests of MASCS mission performance will provide a more substantial validation of the simulation as a whole.

To test the accuracy of the individual aircraft models within MASCS, an initial configuration file was created that placed the test aircraft at the approximate starting location of the aircraft

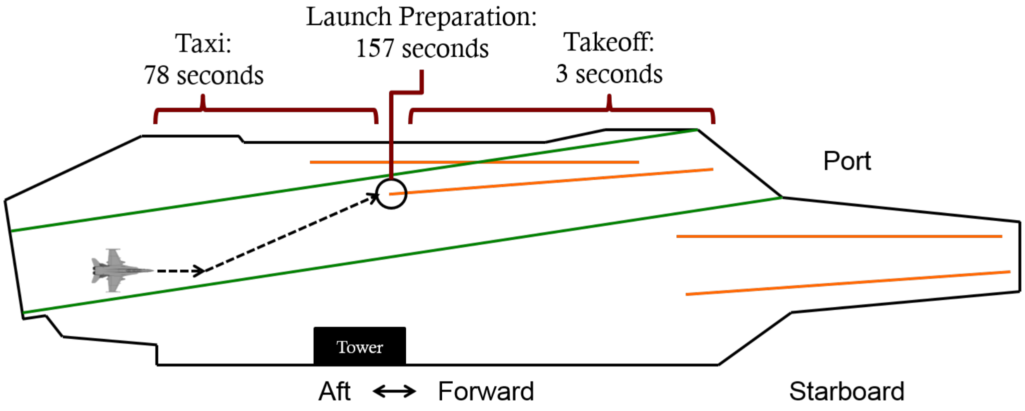


Figure 4-2: Diagram showing the path taken by the aircraft in the observed mission scenario. The figure also details the breakdown of elements in the mission: Taxi, Launch Preparation, and Takeoff.

from the video observation and directing the aircraft to launch from same catapult (#3, the interior aft catapult). This scenario was then replicated 300 times in the MASCS environment to ensure a full exploration of the stochastic variables in the simulation (Taxi speed, Launch Preparation time, and Takeoff acceleration time are all random variables). Data on these metrics were logged and compiled in order to compare their values to the empirical observations. The results of these comparisons appear in Figure 4-3. Simulation results appear in blue, with whiskers denoting ±1 standard deviation in the simulation results. The empirical data from the video recording appears in green.

The results in Figure 4-3 indicate that the empirical observations for Launch event Duration (LD), Taxi Duration, Launch Preparation Duration, and Takeoff Duration are all within one standard deviation of the simulation means. Specifically, they are 0.56, -0.99, 0.75, and 0.97 s.d. away from their respective means. Statistical tests also demonstrate that the resulting data are all Normally distributed (Appendix F). The mean Taxi Duration for the simulation results (blue) is 6 seconds greater than the empirical observation (green), while the mean Takeoff Duration for the simulation results is 0.25 seconds smaller than the empirical observation. However, these two differences are far smaller than the standard deviation observed in the Launch event Duration values (49.63 seconds) and the Launch Preparation Duration (48.84 seconds in the simulation results). These data also suggest that, because of its large mean and standard deviation, the Launch Preparation process is the largest driver of variation in the total Launch event Duration (LD) for these tasks.

The two additional data sets in Fig. 4-3 (yellow and gray) examine combat "Surge" operations that occur only rarely and for which no empirical data could be located. These data are included only as comparison points of what might be possible in terms of improving operations. SMEs describe the

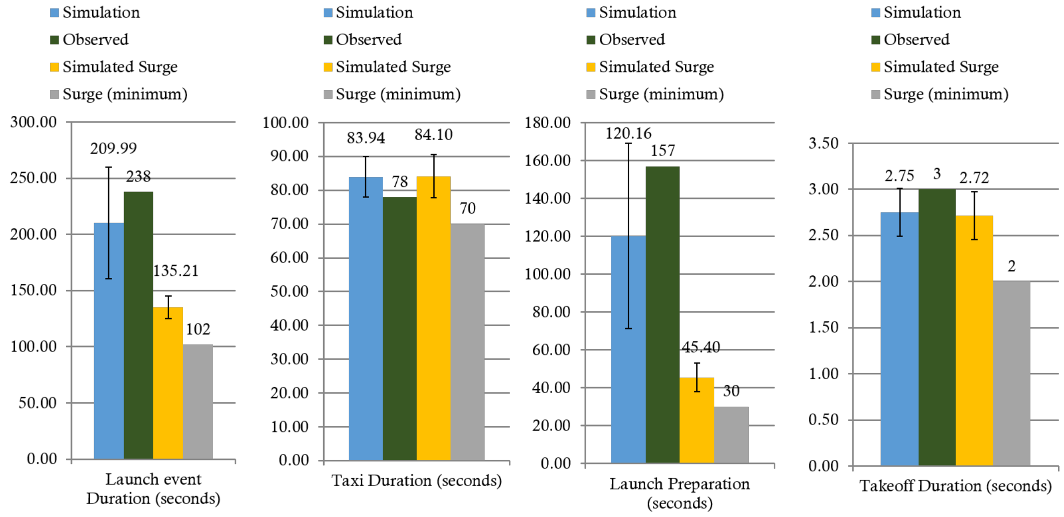


Figure 4-3: Results of preliminary single aircraft calibration test. Simulation data appear in blue with error bars depicting ±1 s.d. Empirical observations are in red.

major change in Surge operations as being a streamlining of the launch preparation process, reducing the time required to complete the process. Accordingly, simulation runs for "Surge" operations (yellow) modified the launch preparation time model to be normal with a mean of 45 seconds, a standard deviation of 10 seconds, and a range of 30-60 seconds (described by SMEs as being the typical range of launch preparation times in Surge operations). Otherwise, all other parameters and initial conditions were left unchanged. Modifying just this launch preparation process results in a substantial improvement in LD values (55% decrease in mean). The fourth column in each chart (gray) depicts the observed (for taxi and acceleration) or theoretical minimum values (for launch preparation) for each measure, providing a view of the best possible results under the provided distributions.

An additional test of the accuracy of the aircraft models can be done through the use of a sensitivity analysis, which examine whether the model puts changes in unexpected ways given a plausible range of uncertainties (Sterman, 2000). In such tests, if the changes in output are not extremely different from what would be expected, they provide confidence that the model is robust to any errors in its provided parameters (Sterman, 2000). For individual aircraft within MASCS, the two most important stochastic parameters are the launch preparation time and aircraft speed. Increasing the mean launch preparation time should increase the total launch event duration while increasing vehicle speed (a rate measure) should decrease it, and vice versa for each. The goal of the sensitivity testing is to demonstrate that changes to these stochastic parameters do not generate extreme changes in outputs: for this data, this concerns the changes in LD values. Since the Launch event Duration is a simple summation of two stochastic parameters (taxi time and launch preparation time) that are independently sampled, changes in LD values should be at or less than the percentage changes to those parameters. That is, a 25% increase in taxi speed should produce a $\leq$25% change in LD.

A set of sensitivity analyses individually varied the mean values of the taxi speed and launch preparation time models, running 300 replications for each case. Both the mean launch preparation time and the mean taxi speed were individually varied at ±25%; a 25% change corresponds to roughly half the standard deviation of the launch preparation time model (the larger of the two means), with 25% changes applied to the taxi model for consistency. Results for these tests, indicating the resulting percent change in launch event duration mean, appear in Figure 4-4.

For both launch preparation time and taxi speed, the relative change in LD values is less than the relative change (25%) in each parameter, as desired. However, the results also show that the relative increases and decreases in LD for a given parameter were not equivalent in magnitude, which is unexpected. Reviewing the output data from the simulation (Figure 4-5), it can be seen that the changes in the means of the parameters did not produce symmetrical changes in the sampled values during simulation runs: increasing the launch preparation mean by 25% resulted in samples that

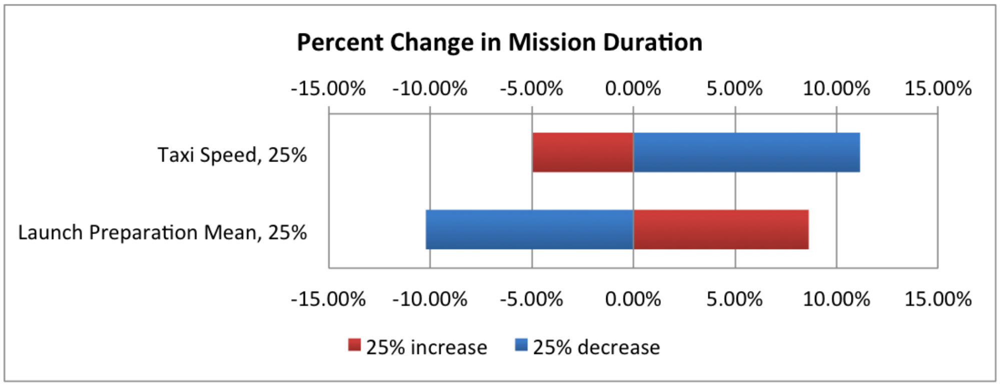


Figure 4-4: Sensitivity analysis of single aircraft Manual Control Case.

averaged a 30% increase from the baseline, while decreasing the mean by 25% resulted in only an 8% decrease in values. This is a result of the boundaries placed on the parameter values (launch preparation time has a minimum of 30 seconds, taxi speeds bounds of $[3.0, 4.0]$ miles per hour). The imbalance in sampling partially explains the low percent changes in LD in Fig. 4-4 and part of the imbalance in the results.

The direction of changes in LD in Fig. 4-4 was consistent with expectations, however, and the relative changes in LD remained lower than 25%. Taxi speed values increased by nearly 30% but resulted in only an 11.15% decrease in LD; launch preparation mean values were decreased by nearly 30% and resulted in LD decreases of only 8.63%. For changes that would decrease LD values (decreasing launch preparation times or increasing taxi speed), changes were near or less than the resulting parameter changes. Taxi speeds were decreased by roughly 17% and increased LD values by only 5%. Launch preparation values were increased by roughly 8.5%, with LD values increasing by 10.22%. This is a relatively minor inconsistency that can be attributed to the taxi speeds sampled (which were roughly 1% below average) and the interactions of random sampling of these two stochastic processes.

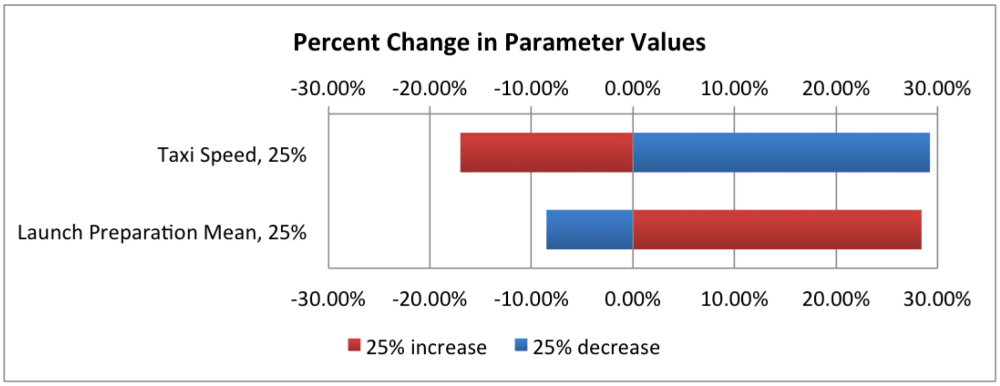


Figure 4-5: Changes in parameters during sensitivity analyses.

Together, the sensitivity analyses and the comparison to empirical data both provide evidence that the simulation is a reasonable model of individual aircraft behavior. The data points from the single aircraft empirical observations all lie within one standard deviation of the simulation data, suggesting that the empirical observation is a reasonably likely occurrence within the simulation. Sensitivity analyses on the taxi and speed parameters (varying their means by ±25%) providing changes in launch event duration of less than 25% and occurring in the correct directions. At worst case, resulting changes in LD should have been equal to 25%; that they are lower demonstrates there is some robustness in the simulation model to these errors. This, in turn, suggests that the models of individual aircraft and their interactions with crew, developed in Chapter 3, are accurate for flight deck operations and should be acceptable for use in mission simulations. This testing is described in the next section, in which mission ranging from eighteen to thirty-four aircraft are executed in order to examine the ability of MASCS to replicate real-world mission performance.

4.3 Mission Validation Testing

The previous section described the calibration of individual agent motion on the flight deck, providing evidence that the individual aircraft models can adequately replicate empirical observations in the world. However, the more important testing addresses the ability of the full MASCS simulation to replicate mission scenarios using multiple different aircraft. In these simulations, eighteen or more aircraft proceed from parking to launch catapults over a period of time, executing taxi tasks simultaneously while interacting with multiple crew.

Validating these operations requires the acquisition of data that also describes flight deck launch events. The primary source of information for this data is two reports from the Center for Naval Analyses (CNA) (Jewell, 1998; Jewell et al., 1998). The most important data from these reports concerns the number of aircraft used in a single mission sortie (mission Density) and a Cumulative Density Function (CDF) of the interdeparture times of launches from the flight deck. The former helps define the test conditions within the simulation (the number of aircraft that should be included in test scenarios) while the latter provides a highly-detailed measure of system behavior. Figure 4-6 provides a histogram of the number of aircraft used in sorties during the CNA study. While it is notable that the data can be fit by a Normal distribution ($\mu = 27.014$, $\sigma = 4.2614$, Kolmogorov-Smirnov statistic = 0.09866, p = 0.47351), knowledge of the distribution is not necessary for current testing. Rather, that the bulk of operations utilized between 18 and 34 aircraft serves as an important input parameter for MASCS testing.

Figure 4-7 shows the second important data point from these CNA reports: a CDF of the interdeparture times of deck operations over the course of one week of missions. An interdeparture time is defined as the time between two successive launches (departures) from the flight deck across any two catapults; if launches happen from catapult 1 at $t = 25$ and from catapult 3 at $t = 40$, the

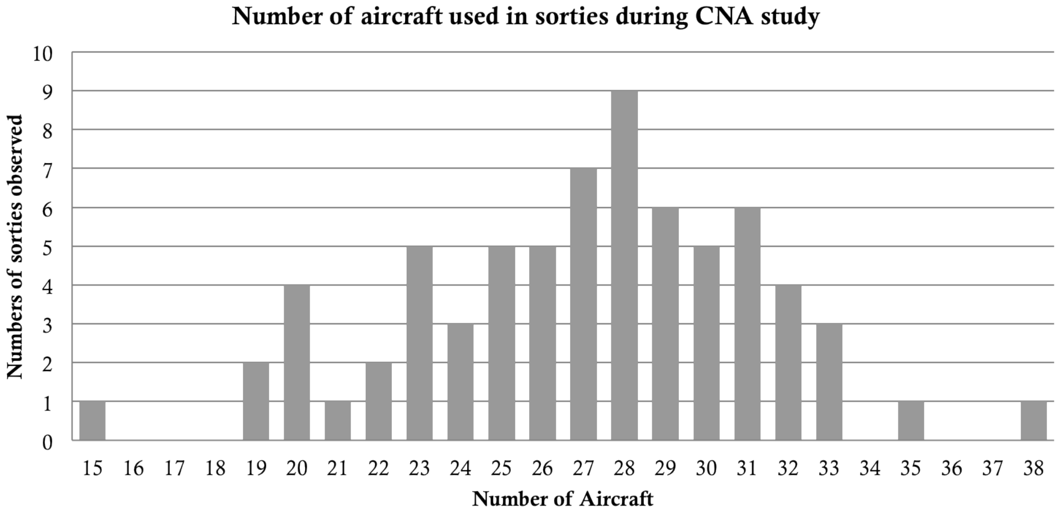


Figure 4-6: Histogram of the number of aircraft observed in sorties during the CNA study (Jewell, 1998).

interdeparture time is 15 seconds. In queuing theory, interarrival and interdeparture times are key inputs to system models and serve as a powerful metric of a system's behavior. For this work, this empirical distribution serves as an important comparison point for the MASCS model; if the results of simulation testing for the mission profiles shown in Fig. 4-6 provides the same interdeparture times as shown in Fig. 4-7, it provides evidence that MASCS is not an invalid model of operations.

The usefulness of the CNA distribution is further enhanced by the fact that it comes from a variety of different mission profiles aggregated together, allowing it to be applied to a variety of launch event profiles (primarily, the number of aircraft). Unfortunately, the authors of the CNA reports do not clearly identify the types of operating conditions observed (day, night) nor the specific distribution mission sizes utilized. However, the authors did regard the data as representative of fleet performance. The data contained in Figure 4-7 can be converted first into Probability Density Function (PDF) form, then fitted with a negative exponential fit (the standard form of an interdeparture rate) in the form $\lambda e^{-\lambda * t}$, with λ=0.01557; this provides an average interdeparture rate of 64.22 seconds. Details on the process of converting the CDF data into PDF and the statistical tests for goodness of model fit can be found in Appendix E. If the MASCS simulation is a valid replication of activity on the flight deck, the interdeparture rates of the MASCS model should align closely with the reported rate from the CNA data.

The mission profiles listed in Fig. 4-6 were used as the template for mission cases for MASCS validation. Table 4.1 contains the list of missions tested in this phase. Each mission scenario was replicated thirty times in the simulation, with data on Launch event Duration (LD) and launch interdeparture times tracked for each. A total of eighteen different mission profiles were tested,

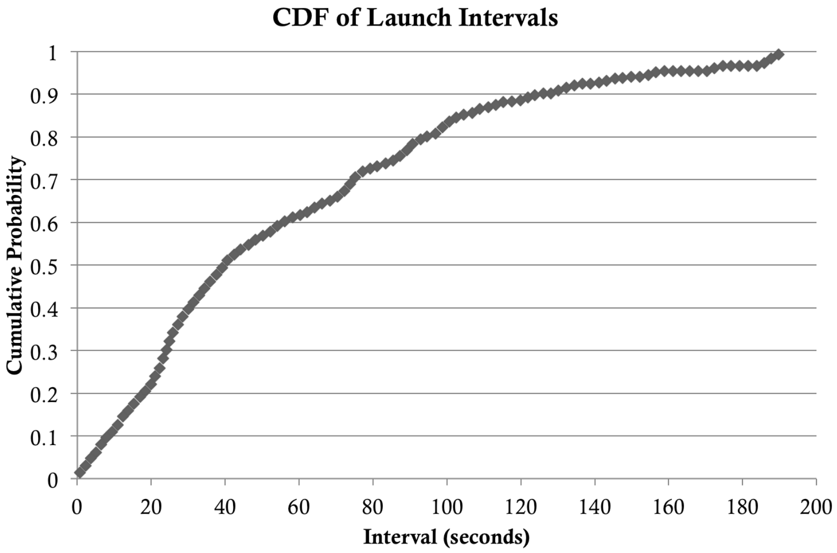


Figure 4-7: Cumulative Density Function (CDF) of launch interdepartures from (Jewell, 1998). This CDF can be converted into a Probability Density Function (PDF) in the form of a negative exponential of the form $\lambda e^{-\lambda * t}$ with λ=0.01557

varying from 18 to 34 aircraft and using either three or four catapults. The numbers of aircraft used in the test program reflect the significant changes in deck organization that occur from 20 to 24 aircraft; once the mission includes 22 or more aircraft, vehicles must be parked in the interior of the deck. As this occurs, taxi lanes on the flight deck being to close off, and operations become more complex in terms of allocating aircraft. While this is not expected to cause significant effects in terms of mission duration (the planning heuristics of the handler and the structure of operations should compensate for these effects), multiple runs were taken in this area to ensure that no adverse interactions occurred.

Thirty replications were run for each of the missions listed in Table 4.1. Output variables included the launch times for each aircraft (to generate interdeparture times) as well as total launch event duration values. Launch interdeparture times were calculated for each mission independently and fitted with negative exponential distributions in the EasyFit program. The corresponding λ values and Kolmogorov-Smirnov (K-S) goodness of fit p-values were recorded. Distribution fitting was also performed in the JMP PRO 10 software to obtain confidence intervals on the λ values, which

Table 4.1: List of scenarios tested in mission validation (Number of aircraft-number of catapults).

Mission Scenarios				
18-3	21-3	23-3	26-3	34-3
18-4	21-4	23-4	26-4	34-4
20-3	22-3	24-3	30-3	
20-4	22-4	24-4	30-4	

EasyFit does not provide, with values of λ verified to match between the two programs. Full results for these tests appear in Appendix F. Figure 4-8 shows the final λ values and confidence intervals of the negative exponential fits of the data. The orange line denotes the λ value of the original fit from the CNA reports.

As the figure shows, λ values for all simulation results lie relatively close to the original CNA value, which falls well within the confidence intervals on the simulation λ fits. Values at the four catapult missions tend to provide higher λ values (slightly faster interdeparture rates), which would be expected given the parallel nature of operations that occur between catapult pairs: adding an additional active catapult provides additional opportunities for launches to occur nearly parallel between the forward and aft catapults. For many aircraft Density levels (number of aircraft used in the mission), the λ values at the 3 and 4 aircraft cases bracket the original CNA λ value. In general, these results support that the interdeparture times generated by MASCS are accurate with respect to the empirical data obtained from the CNA reports. More specifically, it suggests that the selection of the Launch Preparation time distribution and the modeling of catapult launches are accurate with respect to the real system. However, this is only one aspect of operations in an environment in which the launch task has significant influence. The next section examines the robustness of the simulation with respect to changes in internal parameters, as well as the ability of

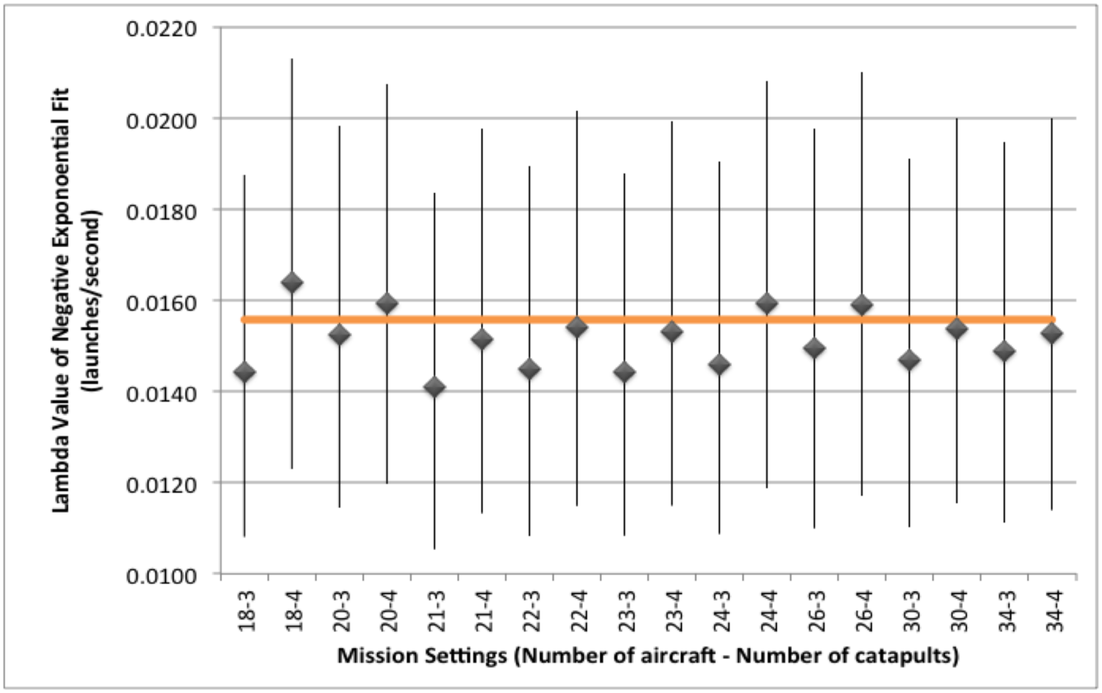


Figure 4-8: Graphical depiction of launch interdeparture negative exponential fits for the final round of mission validation testing. Triangles denote λ values. Vertical lines denote 95% confidence intervals. Note that the mean value of launch interdepartures across all scenarios and replications is 66.5 seconds.

the simulation to adequately replicate launch event durations over a variety of input parameters.

4.4 Mission Sensitivity Analyses

Sensitivity analyses examine the response of the simulation's output to changes in the parameters of the system. The earlier sensitivity analysis of the single aircraft models examined changes to parameters internal to the MASCS simulation model. Additionally, tests can also address changes to the input parameters that define the initial conditions of the simulation. The earlier single aircraft sensitivity analyses simply sought to determine whether or not the response of MASCS was robust (non-extreme) to changes in these parameters. Tests can also examine how well the response of the simulation agrees with the response of the real system, if this data is available (Carley, 1996). The tests of robustness are a check of the internal validity of the simulation model, while tests against empirical data are checks of the external validity of the model. Both forms of sensitivity analyses were performed for mission-level testing in the MASCS environment and are discussed in this section. The first section addresses the response of the simulation to changes in input parameters, comparing its response to known expectations.

4.4.1 Sensitivity to Input Parameters

A sensitivity test utilizing the simulation's input parameters examines the accuracy of the simulation over various conditions. For MASCS, the primary input parameters of interest are the number of aircraft included in the mission and the number of available catapults. Within these tests, parameters regarding agent motion and agent decision-making (including the Deck Handler planning routines) are left unchanged. Performing this test requires an understanding of the performance of the real flight deck under these conditions.

Information on flight deck performance was obtained from two sources. First, the CNA reports referenced earlier also include some general "rules of thumb" concerning how often aircraft launch and land, providing general guidelines as to the expected time to complete a launch event, the general range of times that are possible, and how flight deck performance responds to changes in the number of aircraft used in the mission. These reports indicate that launches on four-catapult carriers occur every 30 to 60 seconds (Jewell, 1998), although it is unclear as to whether this figure includes both "Surge" and non-combat operations. This value compares favorably to the mean launch interdeparture data reported in Fig. 4-7 (a mean of roughly 56 seconds). Given a specified number of aircraft on the flight deck, and assuming that the catapults maintain uninterrupted queues throughout the course of the mission, these values can be used to estimate the total time required of a mission. This suggests that a 20-aircraft mission should range from 10 to 20 minutes in duration, plus some additional time for taxi operations at the start of the event.

A third perspective views the flight deck as a queueing system in which launches are allocated evenly between the forward and aft catapults and each pair of catapults works as an independent server with independent queues. Because each catapult pair alternates launches within themselves, they have as a single serial server; the time to complete a mission is then just a linear function of the average number of launches between the catapult pairs. The launch preparation time was defined in Chapter 3, based on previous data, as $N(109.6, 57.8^2)$. If aircraft are evenly allocated across catapult pairs, then the estimated time to complete a mission for a given number of aircraft $N_{aircraft}$ is $N_{aircraft} * \mu/2$ (number of aircraft times average departure rate of 54.8 seconds). This model assumes that the queues of aircraft at catapults exist through the mission and does not account for taxi operations prior to the first launch. It is useful, however, for understanding how the system should respond to changes in the number of aircraft within the mission: if the allocation of aircraft remains balanced between catapult pairs, then the time to complete the mission should respond linearly.

When using fewer than four catapults, the ability to balance assignments on the flight deck is disrupted, as one catapult "server" can now hold only two aircraft in queue at a time (as opposed to four). This disrupts the process of allocations in the system and should have an effect on operations. The end result is that, for a mission utilizing only three catapults, aircraft are overallocated to the catapult pair and underallocated to the lone catapult. Assuming that all launches require the same amount of time, this results in requiring up to two extra launches from the pair while the lone catapult sits idle. Appendix G provides a more detailed discussion of the general process of queuing aircraft at catapults over the course of a mission, addressing how aircraft are assigned to and queue at catapults; a summary diagram of the resulting allocation appears in Figure 4-9. Additionally, barring any significant flaws in Deck Handler assignment logic, a three catapult case should not occur in less time than a four catapult case; doing so would require a significant lack of balance in assignments between catapult pairs.

Figure 4-10 plots the mean mission duration values, ±1 standard deviation, for the mission scenarios listed in Table 4.1 by number of catapults used. As described above, all reported data suggest that the response of the simulation to changes in the number of aircraft should be linear with slops similar to the average interdeparture time and average launch preparation time, and that a small but consistent difference should exist between the three- and four catapult missions. The

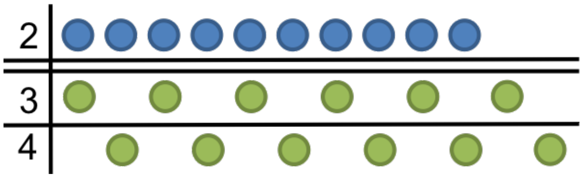


Figure 4-9: Notional diagram of allocations of aircraft to catapults in a 22 aircraft mission. Dots represent assigned aircraft launches; in the image, catapults 3 and 4 are paired and launches alternate between the two. Catapult 1 is inoperable, and catapult 2 process all assigns in that section in series.

data reported in the figure comes from the same runs used to generate the launch interdeparture data presented in Section 4.3.

Within the figure, linear fits of the data are overlaid and the corresponding R^2 values reported. Statistical tests on each linear model (F-test) showed statistically significant results ($F(1,7) = 1743.53, p = 0.0001$ and $F(1,7) = 773.90, p = 0.0001$, respectively). Model coefficients (66.6 and 68.4 seconds for the three- and four-catapult missions, respectively) are also significantly different from zero (full results for these tests appear in Appendix H) and are close to the expected value of mean interdeparture value from the simulation test data (average = 66.5 seconds) but slightly higher than the values reported in the CNA data (heuristic of up to 60 seconds, reported interdeparture of 64.1 seconds) and that predicted by the simple queueing model (half the launch preparation time, 55 seconds). However, while there are some distances, they are relatively small.

A relatively consistent difference between launch event duration values for the three- and four-catapult missions was also expected. The results show a mean different of 100.2 seconds, slightly less than the mean launch preparation time of 109.8 seconds. Viewing results on a per-mission basis, differences between three and four catapult ranged from 50 seconds (slightly less than half of one launch) to 142 seconds (1.29 launches), well within the expected difference of zero to two launches. Additionally, the average duration of three catapult missions at a given mission Density (e.g., 20 or 24 aircraft) is never smaller than that of the four-catapult mission. However, there does appear to be some interesting variations in launch event duration between 20 and 24 aircraft, likely due to how aircraft are placed on the flight deck. Using more than 20 aircraft requires that aircraft be parked in the center of deck, eliminating some of the available taxi routes on the flight deck. These routes continue to be eliminated as additional aircraft are added, requiring changes in how aircraft are prioritized and missions are planned and executed. Past 26 aircraft, the changes in patterns stabilize and return to a more linear trend.

The results presented in this section provide confidence that the simulation responds reasonably with regards to changes in the input parameters of number of aircraft and number of available catapults. The behavior of the simulation in regards to both is in line with prior data and expectations of SMEs, suggesting that the simulation reacts in a similar fashion to the real world system. Although the differences between three and four catapult missions did vary, at no time was a three catapult case observed to require less time than a four catapult case, nor were any substantial and unexpected changes in average launch event duration observed. More specifically, these results suggest that the modeling of the launch preparation process, the interactions between aircraft and catapults, the taxi routing process, and the catapult queueing processes that were all defined in Chapter 3 are accurate and are reasonable with respect to the output variables tested within this section. However, as with the single aircraft calibration, sensitivity analyses of internal parameters should also be performed in order to determine the robustness of mission performance with regards to those modeled parameters.

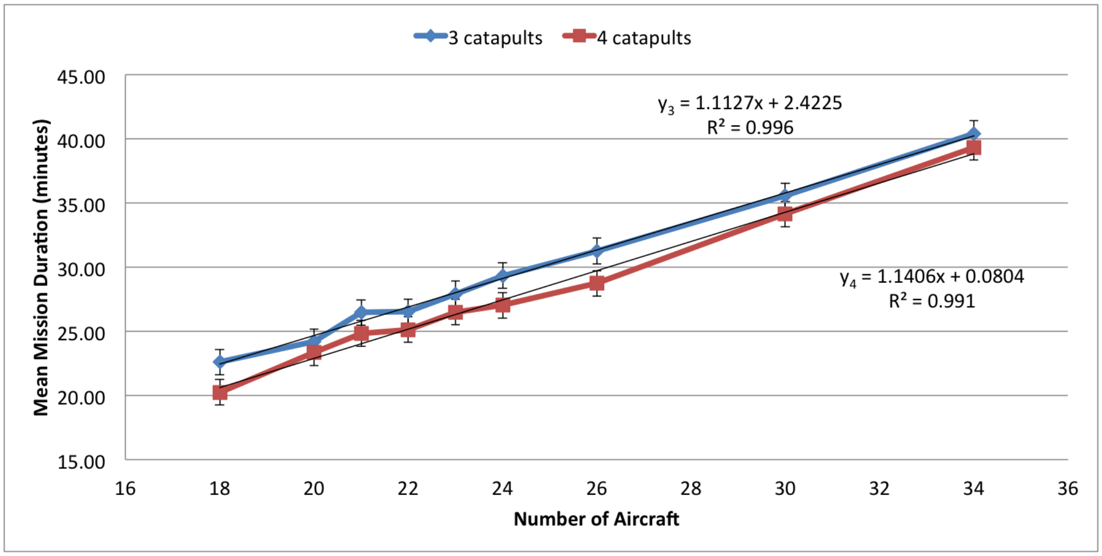


Figure 4-10: Plot of event durations versus number of aircraft, sorted by three- (blue) vs four-catapult (red) scenarios.

This is described in the next section.

4.4.2 Sensitivity to Internal Parameters

Sensitivity analyses of internal parameters within the model examine the internal validity and robustness of the simulation to errors in the parameter models. The model should not be overly sensitivity to errors in the definition of its parameters, given that the estimates of those parameters, based on real data, are likely to be imperfect. For the MASCS model of flight deck operations, the most important internal parameters include the launch preparation time (defined in Chapter 3, Section 3.2.1), vehicle speeds, and the collision prevention parameters. The first two of these were also tested for single aircraft in Section 4.2; collision prevention parameters could not be examined because they require at least two vehicles to be present.

For launch preparation time, mean values are adjusted by ± 10%, while taxi speeds and collision prevention thresholds are adjusted by ± 25%. As demonstrated in the previous section, the response of mission event durations is linear with respect to the number of aircraft and the mean launch preparation time. As such, a 10% change in the launch preparation time mean should lead to a 10% change in mission behavior and be detectable through statistical tests. However, the structure of queueing on the flight deck should limit the effects of taxi speed and collision prevention parameters. Increasing the speed of transit only makes vehicles arrive at queues earlier and increases their wait time at the catapult; it will have little meaningful effect on operations. Reducing speeds should have some effect, but must be substantial enough to interrupt the queueing process at the catapults. Similar effects should occur for the collision prevention parameters, with reductions in their values removing constraints on vehicle motion and speeding their transit to catapults. Initial tests using only 10% changes produced almost no detectable differences, whereupon the value with increased to ±25%.

Table 4.2 provides a list of the mission scenarios tested in this sensitivity analysis. For this phase of testing, only values of 22 and 34 aircraft were utilized; SMEs report that 22 aircraft is a typical mission profile, while 34 aircraft is the largest value seen in prior CNA reports (Jewell, 1998; Jewell et al., 1998). The table lists scenarios in the format (Number of Aircraft)-(Available Catapults) while also denoting the parameter changes involved in each scenario; a notation of $X\%$ denotes that the parameter was varied by both $+X\%$ and $-X\%$ in two separate scenarios. Each scenario was replicated 30 times at both the $+X$ and $-X$ settings. Note that at no time was more than one parameter varied during a run. Given the relative certainty in how launch preparation changes would affect operations, these were only conducted for two mission profiles (22-3 and 34-4).

Fig. 4-11 provides the results of these tests in the form of a "tornado" chart. Each set of bars reflect the plus and minus deviations of a specific parameter for a specific mission, with results ranked in terms of greatest difference between plus and minus values. From this chart, it can be seen that the changes in launch preparation distribution are near ±7%, smaller in magnitude than the changes to the corresponding internal parameters, demonstrating a slight dampening effect. A review of the output of launch preparation time values demonstrated that, due to the bounding of values in random number generator, output values were roughly ±7% rather than ±10%. This means that even though the stochastic launch preparation time generator did not provide the expected values, the simulation responded appropriately to the values that were generated.

ANOVA tests reveal that significant differences exists for launch preparation time tests at the 22-3 ($F(2,87) = 29.1344, p = 0.0001$) and 34-4 ($F(2,87) = 34.4374, p = 0.0001$) cases, reflecting the significant effects of the launch preparation time in driving flight deck performance. Tukey post-hoc tests show that significant pairwise differences exist across all three cases for each mission settings at $p = 0.0025$ or less (full results are available in Appendix H. From the figure, the effects of launch preparation on operations are clearly larger than the effects of changing the taxi and collision prevention parameters, even though the changes to launch preparation time were smaller (in terms of percent differences) than the change to speed and collision prevention parameters (10% vs. 25%).

Table 4.2: Matrix of scenarios tested during internal parameter sensitivity testing.

	3 catapults	4 catapults
22 aircraft	Taxi±25% CA±25% Launch±10%	Taxi±25% CA±25%
34 aircraft	Taxi±25% CA±25%	Taxi±25% CA±25% Launch±10%

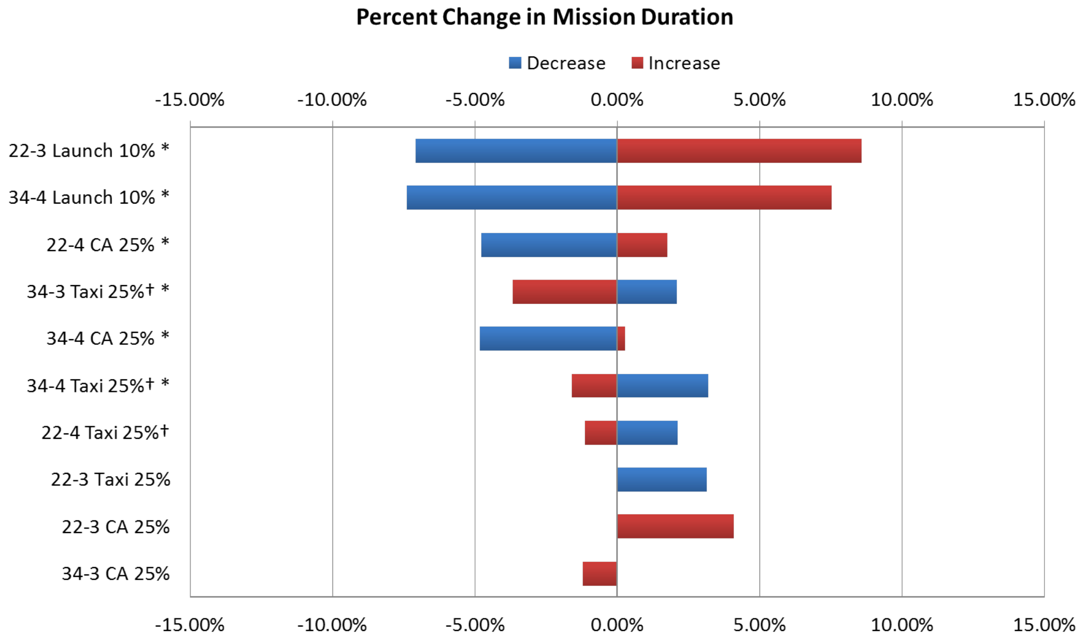


Figure 4-11: Results of final sensitivity analysis. Labels indicate number of aircraft-number of catapults, the parameter varied, and the percentage variation of the parameters. Dagger symbols (†) indicate an inverse response to changes in parameter values is expected. Asterisks (*) indicate significant differences in ANOVA tests.

Significant results also appear for changes to aircraft taxi speed at the 34-3 ($F(2,87) = 4.9753, p = 0.009$) and 34-4 settings ($F(2,87) = 3.8031, p = 0.0261$). Tukey post-hoc tests reveal that, for both of these cases, reductions in taxi speed provide significantly different results from increases in taxi speed ($p = 0.022$, $p = 0.007$, respectively), but neither is significantly different from the mean in either case; full results again appear in Appendix H). These results just how weak the effect of taxi speed is on effecting on operations: even at changes of ±25% to the parameter, significant differences only exist between the extremes. As described previously, this is likely an effect of the launch preparation process regulating the flow of traffic on the flight deck. Interestingly, because taxi speeds only had effects at the 34 aircraft case, it shows that the effects of speed may accrue over increasing numbers of aircraft. Alternatively, it may be indicative of how the changes in how missions evolve at the 34 aircraft level (more aircraft, fewer available taxi paths, and a difference in the order of how areas are cleared) are more sensitive to vehicle transit speeds.

Tests for the effects of changing the collision prevention parameters show significant results only at the 22-4 ($F(2,87) = 4.6278, p = 0.0123$) and 34-4 ($F(2,87) = 5.4986, p = 0.0056$). Tukey post-hoc tests (full results in Appendix H) show that for both cases, decreasing the collision prevention parameters significantly decreases mission completion times. For the 22-4 results, the collision prevention -25% case is significantly different from the +25% case ($p = 0.0116$) and marginally significant from the baseline ($p = 0.0855$), while for the 34-4 mission, the -25% collision prevention case is different from both the +25% case ($p = 0.0014$) and the baseline ($p = 0.0176$). This suggests that when using four catapults, in which more aircraft are moving on the flight deck, the current collision prevention settings may slightly overconstrain operations. Reducing these constraints and providing aircraft mode freedom of motion in operations provides slight benefits to operations. However, increasing these constraints appears to provide no further penalty to operations and indicates some robustness in the system. However, because the results of the earlier input sensitivity and launch interdeparture tests were accurate, the baseline collision parameter settings were left unchanged.

The results described within this chapter thus far provide confidence that the MASCS model of aircraft carrier flight deck operations is a reasonable representation of operations, partially validated in terms of the output variables of launch interdepartures and launch event durations values and its responses to changes in internal and input parameters. The system has been shown to respond reasonably to changes in taxi speed, collision prevention parameters, and changes in the launch preparation. The latter produces an almost linear response to changes in the launch preparation time distribution, as expected. The effects of the other parameters are limited due to the structure of operations: parameters that increase the rate of taxi operations have no effect, as they are bounded by the final launch tasks at catapults. Tasks that decrease the efficiency of operations have limited effects due to specific interactions with system conditions, but whose effects can be easily explained.

These results provide evidence that the agent and task models that comprise MASCS are accurate and respond appropriately to system inputs, but as noted at the start of this chapter, models for the Deck Handler and other human elements in the system cannot be fully validated at this time due to a lack of data. Additional data collection at a much higher level of detail than currently performed must occur to capture details about individual behaviors and decision-making on the flight deck to ensure that agent behaviors are accurately captured. In light of these limitations, the accuracy of these behaviors can also be reviewed by SMEs familiar with deck operations to critique their representations at a high level. The next section describes the SME review that occurred during the MASCS validation process.

4.5 Subject Matter Expert Review

After concluding the sensitivity analyses, the simulation environment and test results were presented to a set of Subject Matter Experts (SMEs) for review. SMEs attending this meeting had a range of experience on the flight deck, each over multiple years (two pilots each with over 800 flight hours in fighter aircraft; another individual with over 3,000 hours as a Naval Flight Officer and 2 years as a carrier-based launch and recovery officer) All were employed by Navy research organizations at the time of the meeting. During the meeting, the simulation (which displays a live animation of the evolution of the flight deck environment) was presented first to SMEs for their critiques. SMEs were shown live executions of the MASCS environment using missions of 22 and 34 aircraft and using either three or four catapults, allowing the simulation to execute completely.

SMEs were specifically asked to review all aspects of the simulation as it played out — the accuracy of movement of the crew, their alignment to aircraft, the motion of aircraft on the flight deck and how their paths are deconflicted, and the assignment of aircraft to catapults. They were also asked to provide comments on any other aspects of the simulation outside of those factors. The simulations were often paused or restarted in order to the SMEs to provide comments on what was currently happening or to get a closer look at what had just occurred. This continued until participants had no remaining comments, upon which the discussion shifted to discussing the results of validation testing described in Section 4.3. These results were presented in a series of slides to the SMEs.

The SMEs agreed that, qualitatively, the animations of the simulation were accurate overall. The SMEs expressed concern over what they considered to be large launch event duration values, but were not overly worried given observations of the rest of the simulation. The SMEs also admitted that they did not have a good understanding of how mission durations varied between combat "Surge" operations, which occurs at a higher tempo due to faster launch preparations, and the majority of slower tempo, more typical operations. They also commented that they were largely biased to thinking in terms of Surge operations, but agreed that given the accuracy of the motion of crew, aircraft, and the assignment of aircraft to catapults, the model should replicate Surge operations with the correct (faster) launch preparation model.

In terms of sensitivities, SMEs explained that they had relatively poor mental models of these effects. Aircraft only operate at one taxi speed, and they have no real understanding of how taxi speeds might affect mission operations. In terms of the collision prevention parameters, aircraft directors are all typically very risky when navigating aircraft. They described Directors as having very little variation in how conservatively they navigated aircraft on the flight deck, but did agree with the explanations of the sensitivity analyses when presented with that data. Additionally, they also realized they had no real understanding of how the number of catapults affected the time to complete the launch event: what is available is used, and extra catapults are thought to help clear more space on deck.

That SMEs do not notice a difference is supported by the results of tests in Section 4.4. The difference in performance between the three and four catapult missions averaged less than two minutes in those tests, a difference not likely strong enough to be noticed by crew. Crew also noted that they do not often consider the effects of including more aircraft in the mission, as this number is not really an option for the deck crew. Even so, conceptually, the SMEs did not disagree with the results and approved the simulation as an accurate replication of flight deck operations.

4.6 Discussion

The testing described in this chapter has demonstrated that the MASCS model was able to adequately replicate launch interdepartures on the flight deck while also demonstrating responses to changes in parameters in line with expectations, providing confidence that the MASCS model is a reasonable model of flight deck operations. As such, it suggests that the modeling techniques used in creating MASCS are also reasonable and are suitable for use as templates for generating models of the unmanned vehicle control architectures and their integration into flight deck operations. The results of the testing discussed in this chapter also provide insight into the important parameters concerning UAV implementation, as well as the use of SME in analyzing system performance. However, as noted at the beginning of this chapter, insufficient data is available to suggest that the simulation model is fully validated. The data used in this testing process has been used to successfully calibrate the model, but a variety of aspects of the environment have not yet been addressed.

Further validation will not be possible until a much greater amount of detailed data is available for operations. The most pressing need is data on the activity of individual crew on the flight deck, including the Deck Handler and his planning strategies, as well as in individual interactions between Aircraft Directors and aircraft in the routing topology they use on the deck. While the heuristics that govern these processes were based on discussions with subject matter experts, training manuals, and observations of operations, they can only be considered as representative models of operations. Substantial variability may exist within these behaviors that have not been captured in the MASCS model at this time. Even so, the results of calibrating this model suggest that it is useful in exploring the sensitivity of the flight deck environment to vehicle behaviors as modeled in MASCS.

In particular, the sensitivity tests presented in Section 4.4 suggest that the main drivers of performance, in terms of mean event duration, lay in the launch process and not in any parameters relating to aircraft motion on the flight deck. However, changes to movement speed and collision prevention (vehicle's freedom of motion) may still have effect if they can disrupt the queuing process at the catapults. This is one of several potential tradeoffs that exist within the system. Increasing taxi speeds in fully autonomous environments may be possible, but will be of little utility if launch preparations are not changed. Likewise, the benefits of greatly reducing launch preparation times may be lost if vehicles cannot arrive at catapults at the proper times. Given the Navy's goal to put UAVs on carriers, it is unclear how UAVs will perform in this current environment, or what changes will be needed to incorporate their inclusion. Given the sensitivity of the launch time preparation parameter, these results suggest that launch preparation capabilities should be a primary concern for UAV developers in terms of aircraft carrier integration.

Interviews with SMEs revealed important limitations on their knowledge of the system and their mental heuristics. Their bias towards combat operations was already discussed and is not surprising, given human nature and the impactful nature of those memories. However, the limitations of their knowledge of system sensitivity perhaps reveal some misunderstanding of flight deck dynamics. In conversations, a variety of SMEs noted that extra catapults are always used in order to clear space on the flight deck, will little understanding of how this actually affects flight deck performance. The MASCS environment demonstrates that there is little practical benefit in terms of completing launch operations at a cost of increasing the congestion on the flight deck, which should also be reflected in significant changes in the safety metrics that will be examined later. The results of the testing at this time suggest that it may be more beneficial, and perhaps safer, to commit resources to speeding the launch process rather than to include an additional catapult. This is an interesting, and unexpected, result of the simulation validation process that will be examined further in later chapters. This result also demonstrates that the MASCS simulation may be useful in further examining the failures of SME heuristics in understanding both current operations, and future ones involving UAVs

4.7 Chapter Summary

This chapter has described the validation testing process for the Multi-Agent Safety and Control Simulation (MASCS) simulation environment. This chapter began with a discussion of the process of validation and how it occurs for agent-based models. The chapter then described the phases of simulation validation testing, beginning with the calibration of the aircraft agent model before proceeding to validation testing of full missions utilizing 18 to 34 aircraft. In each section, the results of both internal and external validation testing were discussed, describing the robustness of the simulation to changes in different parameters and its ability to accurately replicate empirical data from real-world testing. A final section described a review of the simulation by Subject Matter Experts (SMEs) that deemed the behaviors of individual crew, aircraft, and the Deck Handler planning to be accurate.

The results of this section all provide confidence that the MASCS simulation model is an accurate representation of aircraft carrier flight deck operations. However, because data is not available for all agents and for all aspects of flight deck operations in sufficient detail to validate all aspects of the model, it cannot be claimed that MASCS is fully validated. At this time, it can only claim to be a partially-validated model calibrated based on limited data on flight deck operations. However, performance in the validation testing was sufficient to suggest that the MASCS model is ready for the investigation of adding futuristic unmanned vehicles, new safety protocols, and changes to the organization structure of operations. Chapter 5 describes the experimental test program defined for the MASCS simulation environment, including the unmanned vehicle control architectures and safety protocols of interest, how they are modeled, and their effects on the safety and productivity of flight deck operations.

Chapter 5

MASCS Experimental Program

> "The Three Laws of Robotics: 1: A robot may not injure a human being or, through inaction, allow a human being to come to harm; 2: A robot must obey the orders given it by human beings except where such orders would conflict with the First Law; 3: A robot must protect its own existence as long as such protection does not conflict with the First or Second Law."
>
> *I, Robot (1950)*
>
> Isaac Asimov

The previous chapter described the calibration and validation testing processes of the Multi-Agent Safety and Control Simulation (MASCS) model of aircraft carrier flight deck operations, demonstrating that it is a reasonable replication of the real world system. The MASCS model was assembled by modeling aircraft, crew, and supervisory staff as independent decision-making entities, initialized and allowed to interact with the environment on their own through independently-modeled tasks executed by these agents. The process of creating these agent models was described in Chapter 3, which also provided background on flight deck operations. This chapter describes how, within the partially-validated MASCS environment, future unmanned vehicle control architectures and their integration into flight deck operations can be modeled and evaluated over a variety of operational conditions. This chapter also describes different forms of safety protocols that are applicable to flight deck operations, how they can be integrated into the simulation environment independent of the unmanned vehicle models, and what their effects on operations might be.

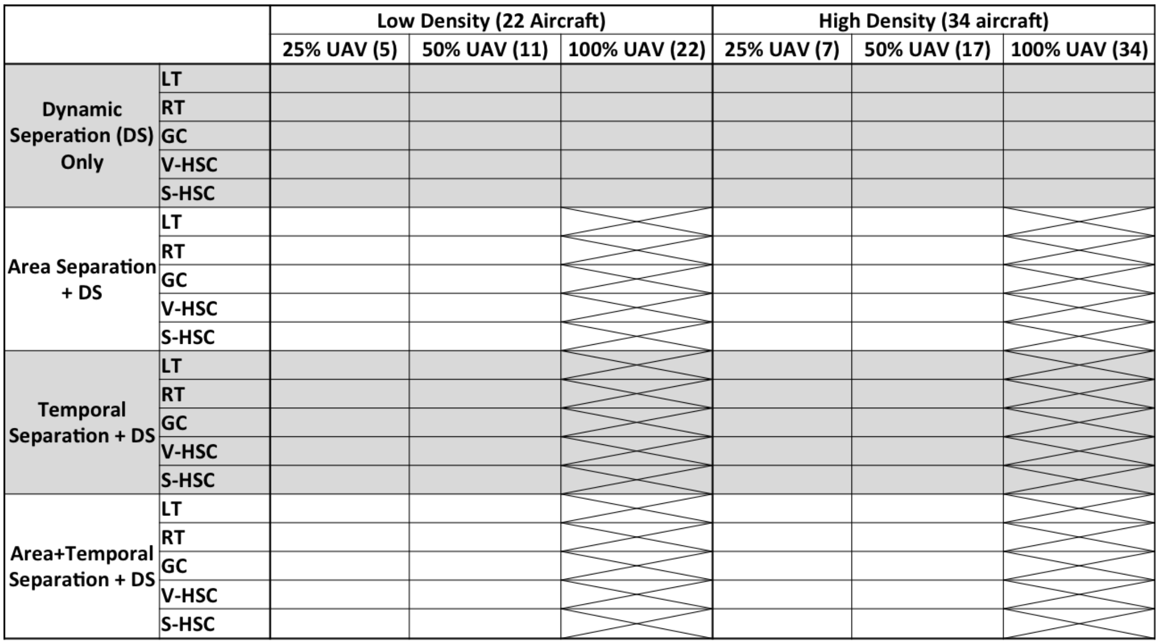
A test matrix is presented, which details the experimental runs that will be used to explore the effects of different Control Architectures and Safety Protocols on flight deck performance and how these are influenced by both the total number of aircraft included within a mission and the number of Unmanned Aerial Vehicles (UAVs) within that total. This description of the experimental protocol also includes the definition of dependent variables that describe both safety and productivity on the flight deck. This chapter concludes with a review of the results of this experimental program. The first section of this chapter begins with a description of the experimental matrix to provide an overview of the test program; the elements of this matrix are described in subsequent sections.

5.1 Experimental Plan

The experimental plan is designed to understand the interactions between UAV **Control Architectures (CAs)**, operational **Safety Protocols (SPs)**, the **Density** of mission operations (number of aircraft utilized), and the **Composition** (percentage of mission that is unmanned or autonomous) affect the safety and productivity of aircraft carrier flight deck operations. These four variables are crossed in order to explore all possible interactions between these conditions. Control Architectures and Safety Protocols are the most important independent variables of interest and are the main focus of the test program. The Control Architecture factor includes six levels, modeling the control architectures of Manual Control (MC) (partially validated in the previous chapter), Local Teleoperation (LT), Remote Teleoperation (RT), Gestural Control (GC), Vehicle-Based Human Supervisory Control (VBSC), and System-Based Human Supervisory Control (SBSC).

Each control architectures involves changes both to individual vehicle behavior (ability to execute tasks and observe crew) as well as changes to how operations are conducted (how plans are generated and how taxi commands are issued), both of which may affect operational performance. Four different Safety Protocols are tested, moving from unconstrained (Dynamic Separation) to moderately constrained Temporal or Area Separation) to highly constrained (Temporal+Area Separation). These Safety Protocols force adjustments to the planning and routing of aircraft on the flight deck, altering when and how many unmanned vehicles are active at any given time.

The mission conditions will also affect how Safety Protocols are employed and cause further changes to mission operations. However, as described in Chapter 3, taxi patterns become more constricted and one or more catapults may not be accessible at increased mission Density. This may have interesting interactions with the CAs and SPs defined in the previous section. Density is tested at two levels: 22 (an average mission) or 34 (the largest typical mission). Lastly, the percentage of UAVs within the mission has operational implementation on how the Safety Protocols are defined for operations and also causes fundamental changes in how the mission evolves. Composition is tested at three levels: 100% (all UAV, examining futuristic operations), 50% (even mix), and 25% (an early, conservative integration of vehicles). Figure 5-1 presents the experimental matrix that results from this cross of variables.



		Low Density (22 Aircraft)			High Density (34 aircraft)		
		25% UAV (5)	50% UAV (11)	100% UAV (22)	25% UAV (7)	50% UAV (17)	100% UAV (34)
Dynamic Seperation (DS) Only	LT						
	RT						
	GC						
	V-HSC						
	S-HSC						
Area Separation + DS	LT			☒			☒
	RT			☒			☒
	GC			☒			☒
	V-HSC			☒			☒
	S-HSC			☒			☒
Temporal Separation + DS	LT			☒			☒
	RT			☒			☒
	GC			☒			☒
	V-HSC			☒			☒
	S-HSC			☒			☒
Area+Temporal Separation + DS	LT			☒			☒
	RT			☒			☒
	GC			☒			☒
	V-HSC			☒			☒
	S-HSC			☒			☒

Figure 5-1: Experimental Matrix for MASCS evaluation.

The matrix is not a full cross, as the Area, Temporal, and Temporal+Area protocols are not applicable at the 100% Composition level (these protocols require heterogeneous vehicle types within a mission; this is explained in the next section). In total, the matrix includes 90 runs, each of which will be replicated thirty times in order to explore the stochastic distributions in the system. Two additional runs examining Manual Control vehicles at the 22 and 34 aircraft levels are also part of the data set but were already conducted in Chapter 4.

5.1.1 Safety Protocols

As discussed in Chapter 2, Safety Protocols (SPs) describe explicit changes to rules and procedures within an environment in order to minimize the number of unsafe interactions between agents in the system. For the Heterogeneous Manned-Unmanned Environments (HMUEs) that are the focus of MASCS, this means changes that limit the physical interaction of unmanned aircraft, manned aircraft, and crew on the flight deck. Chapter 2 also provided three main classes of safety protocols that are applicable to unmanned aircraft on the flight deck environment. They are described in the following sections.

Dynamic Separation Only

Dynamic Separation (DS) protocols attempt to maintain separation between vehicles during transit operations; while vehicles are moving, these protocols act to ensure that they do not become too close to one another by maintaining a minimum safe distance x between one another. Dynamic Separation protocols are utilized within both manned and unmanned vehicles systems, and a version was instantiated for the original MASCS model (see Chapter 3, Section 3.2.1). This original collision prevention routine worked as follows: for a vehicle V_i that is in motion, stop if any vehicle is within a distance d_f forward or distance d_l laterally, where d_f = one body length and d_l = one body width.

This same logic is used for each of the UAV control architectures described later, including utilizing the same thresholds defined earlier for manual control operations. Implicit in this choice is that crew and pilots on deck have the same level of trust in the UAVs that they do in other vehicles; otherwise, it would be expected that the minimum safe distance for UAVs should be increased by some amount. This assumption of equal trust in the UAVs is the basis of many assumptions in the Control Architecture models discussed later — it allows the testing to differentiate between the actual structural features of the architectures and to minimize or eliminate the effects due to environmental or crew social factors. The Dynamic Separation protocol is always active in the system, regardless of whether any other safety protocols are applied, and is applicable to all vehicles models. Additionally, because it only affects an individual vehicle performing a specific task (and does not affect planning or traffic routing on deck), it provides the fewest constraints on operations.

Area Separation

Chapter 2 previously defined the Area Separation protocol as enforcing the constraint that agents of a certain class must remain within their designated area on the flight deck. The intent of such Area Separation (A) protocols is to isolate the believed “unsafe” unmanned vehicles from manned vehicles. In a more mathematical form, the working environment is divided into N regions, where members of a group G_k must remain in region j, where $j = 1 : N$. This corresponds to set of constraints in planning: when allocating members of group G_k to resources in the environment, only assignments to resources within the corresponding region j are allowable. Operations within each region are independent and can occur in parallel. However, because the intent of the protocol is to separate groups of vehicles, there must be at least one vehicle from two or more groups present in order for the protocol to be meaningful. As such, once the population of vehicles in the environment are all from the same group, the constraints on operations can be relaxed.

In the strictest form, this means that members of every vehicle group G_k should remain within their defined region j at all times. In flight deck operations, this would require dividing the carrier flight deck into regions, altering initial parking configurations to place aircraft in the correct regions, and ensuring that each region has access to at least one launch catapult. The experimental plan will only address missions using two classes of vehicles (manned and one type of unmanned), this requires only two regions on the deck. Given that there are two sets of launch catapults (one forward, one aft), segregating the deck into forward and aft halves in a sensible approach. Figure 5-2 demonstrates how this might occur and how operations might evolve.

Manned aircraft (blue areas) will begin parked in the forward section of the flight deck, as crew and the Handler would likely have higher trust in their ability to navigate the high-density Street area of the deck. Unmanned aircraft (yellow areas) would then be placed in the more open Fantail area in the aft. Initial parking configurations are altered to ensure aircraft start in their correct areas, with priority orderings properly balanced to ensure that both groups are assigned in parallel and that taxi paths within regions do not conflict with one another. In making these changes, as few alterations as possible were made from the original MC validation scenarios in order to maintain as much consistency as possible across initial conditions. The Deck Handler planning routine is altered to include the constraints on assignments described above: unmanned aircraft are only assigned to aft catapults (numbers 3 and 4) and manned only forward (catapults 1 and 2) so long as at least one of each vehicle exists on the flight deck.

Temporal Separation

Chapter 2 previously defined the Temporal Separation protocol as enforcing the constraint that only one group of agents is allowed to be active at any given time during operations. The intent is to isolate groups of vehicles from one another, just as in the Area Separation protocol, but to do so while retaining access to the full environment. For this protocol, vehicle groups G_k are scheduled in a pre-defined order, beginning with G_1, then G_2, and so forth to G_n where $n = 1 : k$. At any given time, aircraft from only a single group G_k are allowed to taxi on the flight deck. Once all members of the current group complete taxi operations, the next group is allowed to begin taxi operations.

Figure 5-3 diagrams how this might be applied to flight deck operations. A logical implementation of this protocol would involve operating manned aircraft (blue) first in the mission, followed by unmanned aircraft (yellow) for the same reasons as in the Area Separation protocol. Although crew should trust the UAV systems, given the choice, they would likely prefer that manned aircraft are active first due to the high density of aircraft on the flight deck. Crew would also likely again prefer that manned aircraft begin the mission parked in the congested, high-traffic Street area in the forward section of deck.

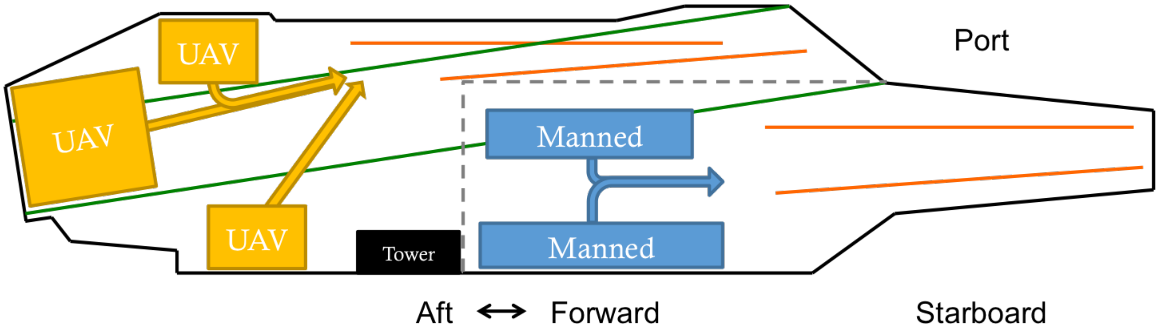


Figure 5-2: Example of the Area Separation protocol assuming identical numbers of UAVs and aircraft. In this configuration, all manned aircraft parked forward are only assigned to the forward aircraft pair, while all unmanned aircraft parked aft are only sent to the aft catapults. When one group finishes all its launches, these restrictions are removed.

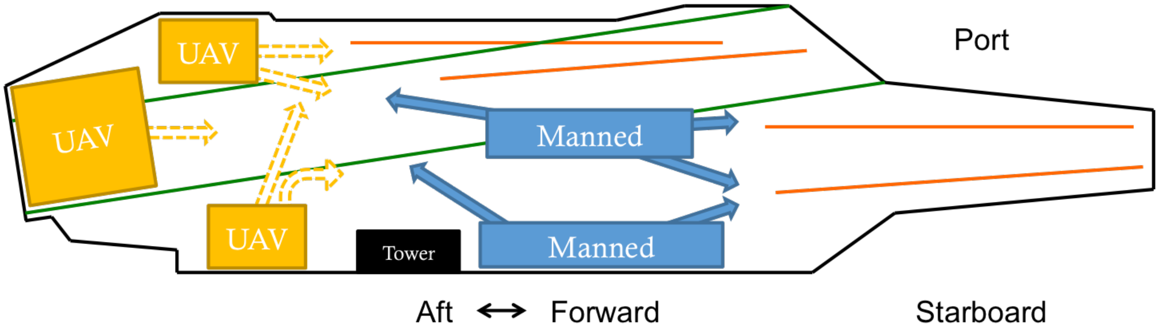


Figure 5-3: Example of the Temporal Separation protocol assuming identical numbers of UAVs and aircraft. In this configuration, all manned aircraft parked forward are assigned to catapults first but are allowed to access the entire deck. After all manned aircraft reach their assigned catapults, the unmanned vehicles are allowed to taxi and are also given access to the entire deck.

Integrating this into the MASCS simulation environment also requires modifications to the initial conditions of test scenarios and the Handler planning routine. In the initial conditions, manned aircraft must be parked such that they can easily taxi to catapults without requiring any of the UAVs to be relocated. This moves the manned aircraft into the interior parking spaces on the flight deck and places the UAVs on the edges and aft areas. Changes to the Deck Handler planning heuristics enforce the constraint that manned aircraft should be assigned first; when the number of manned aircraft without assignment reaches zero and all manned aircraft have arrived at their assigned catapults, unmanned aircraft may then be assigned.

Combining Protocols

It is also possible to combine the Area and Temporal Separation protocols, although this is likely to overconstrain activity on the flight deck. Employing these simultaneously means that one group of vehicles will launch first from one catapult pair, after which the remaining group will have full access to the flight deck. During the initial phase of these combined Temporal+Area missions, the fully-functioning aft catapults do not receive aircraft assignments; manned aircraft are only routed to the forward pair of catapults. This means that during manned operations, the flight deck is operating below its full capacity, as half the available catapults are not used. This should then increase the total launch event duration as compared to the other safety protocols.

A reasonable method of implementing this follows from the previous descriptions of Area and Temporal Separation: manned aircraft are parked forward, with UAVs parked aft in the last areas to be cleared (Figure 5-4). Manned aircraft are operated first, being sent only to the forward catapults; once the last manned aircraft is close to his destination catapult, the UAVs would be allowed to begin operations and have access to the entire flight deck. This should minimize interactions amongst aircraft and increase overall safety, but it will likely cause significant increases in total launch event duration as opposed to the other safety protocol options. Additionally, the combined T+A should also be highly sensitive to the number of manned aircraft included in the mission: the larger this complement, the longer the deck remains in an inefficient state (as no assignments are made to the aft catapult pair).

Given this, there are four distinct Safety Protocol cases that exist: Dynamic Separation (DS), Area Separation (A), Temporal Separation (T), and Temporal+Area Separation (T+A), each of which are applicable to any type of UAVs system. However, it should be noted that the A, T, and T+A protocols are only meaningful when the mission includes a heterogeneous set of vehicles. In the MASCS testing program, the effectiveness of these protocols will be examined in terms of different mission profiles and in terms of their interactions with the Control Architectures for the UAV systems. As described in Chapter 2, five primary types of UAV Control Architecture exist, ranging from simple teleoperation systems to more advanced supervisory control systems. The next section characterizes how each of these five architectures are modeled within the MASCS environment as modifications of the original Manual Control model presented in Chapter 3.

5.1.2 Control Architectures

Control Architecture (CA) is the second major factor in the experimental matrix in Figure 5-1, nested within Safety Protocols on the vertical axis. As described in Chapter 2, the Control Architecture (CA) of an unmanned vehicle system refers to the way(s) in which human operators provide control inputs to those vehicles. This was previously characterized in terms of operator location, operator control inputs, and the modality of operator input, as shown in Figure 5-5. The figure is architected such that, from left to right, the types of control inputs provided by the operator become more abstract. For manual control and teleoperated systems, operators provide inputs physically to control linkages in the system, manipulating the control yoke and rudder pedals to direct the vehicle or moving a throttle to control speed. For gestural systems, operators command an *action* to be executed (to turn, or slow down, or stop) through physical gestures), which the

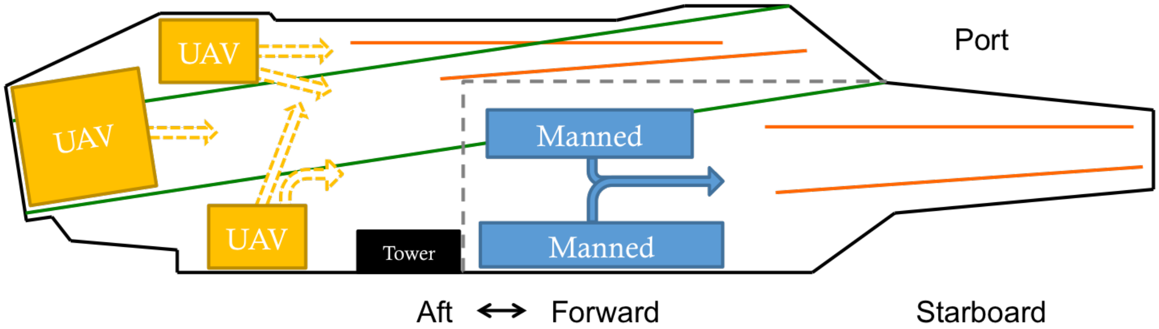


Figure 5-4: Example of the Temporal Separation protocol assuming identical numbers of UAVs and aircraft. In this configuration, all manned aircraft parked forward are assigned to catapults first but are only allowed to taxi to forward catapults. After all manned aircraft reach their assigned catapults, the unmanned vehicles are allowed to taxi and are given access to the entire deck.

vehicle's onboard autonomy translates into control inputs. For supervisory control systems, operators command more abstract goals (such as "taxi to this location") through symbolic interactions in a Graphical User Interface (GUI). These symbolic interaction are then decomposed into a set of more discrete actions, which are then further decomposed into control inputs.

In the context of flight deck operations and models of agent behavior, the important activities of a pilot that must be replicated by an unmanned vehicle control architecture can be summarized in terms of three key aspects: visual detection of an Aircraft Director that is attempting to provide instructions, correctly interpreting task instructions, and correctly executing the commanded tasks. Each of these corresponds to a related variable: the probability of observing the Director if within the vehicle's field of view, the probability of correctly interpreting the command, and the probability of correctly executing a task. For manual control, the likelihoods of these failures are so low that subject matter experts would consider them nonexistent (0% likelihoods). A fourth difference in behavior comes from the fact that the Director is providing instructions in this "zone coverage" routing; this may not occur for some forms of UAV control, and thus how vehicles are routed forms a fourth behavioral variable.

Two other variables also affect vehicle behavior, relating to the physical characteristics of the vehicle: the field of view of the pilot/aircraft, which describes where the Director should be in order to provide commands, and the latency in processing commands. The field of view for manual operations is related to human sight range and the constraints of being in the cockpit; for unmanned vehicles, field of view is related to the cameras used on the vehicle. The latency in processing commands could come from a number of sources: delays in the communication of signals between operator and vehicle, the time required for the vehicle's computer system to process, or time required for the human operator to interact with the system. Each of these can also be modeled as variables

	Manual Control	Teleoperation (Local or Remote)	Gesture Control (Advanced Teleoperation)	Supervisory Control – Vehicle-based	Supervisory Control – System based
Operator location	In vehicle	Outside vehicle	Outside vehicle	Outside vehicle	Outside vehicle
Operator inputs	Control inputs	Control inputs	Actions	Individual actions or goals	Goals (simultaneously)
Input modality	Physical controls	Controls OR gestures	Gestures/vocal commands	Symbolic interface	Symbolic interface

Figure 5-5: Variations between control architectures, categorized along axes of operator location, operator control inputs, and operator input modality.

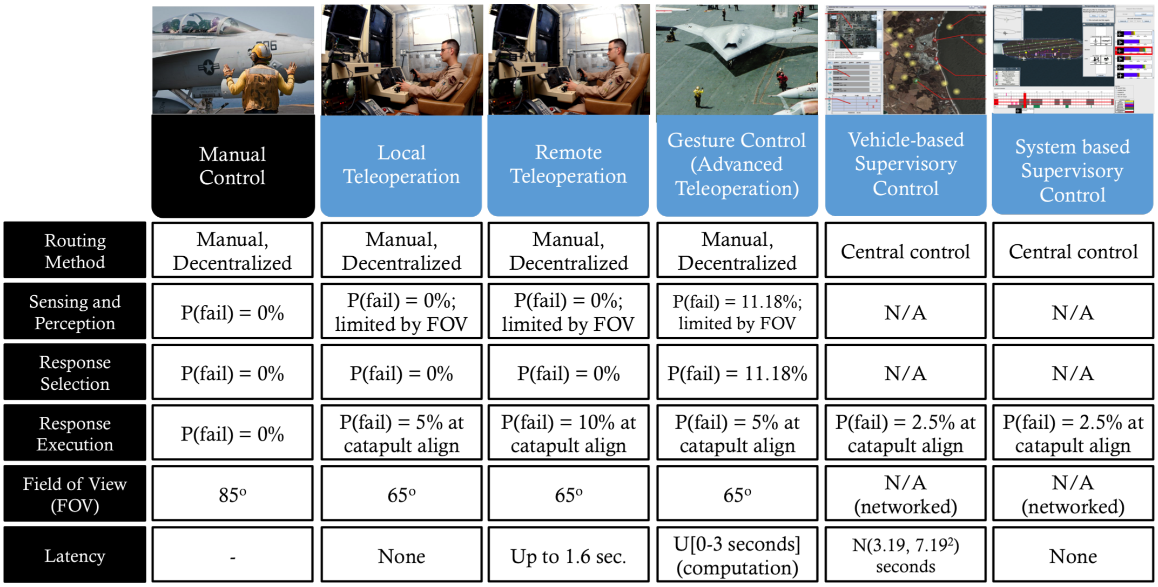
within the MASCS simulation environment, and a full list of these six design variables appears below.

1. Vehicle Behavior variables
   (a) **Routing method**: the topology of the flight deck and how aircraft are routed. Uses either crew "zone coverage" methodology or central operator control.
   (b) **Visual sensing and perception**: the probability of failing to detect an operator when they are within the field of view. Applicable to MC, LT, RT, GC architectures. Bounded $[0, 1.0]$; applied to any task commanded by an Aircraft Director (move forward, turn, etc.).
   (c) **Response selection**: the probability of failing to select the correct response given an issued command. Applicable to MC, LT, RT, GC architectures. Bounded $[0, 1.0]$; applied to any task commanded by an Aircraft Director (move forward, turn, etc.).
   (d) **Response execution (catapult alignment)**: the probability of failing to properly align to a catapult. Applicable to all architectures. Bounded $[0, 1.0]$; applied to final motion task to travel on to catapult.
2. System Design variables
   (a) **Field of View**: visual field of the pilot and/or cameras onboard the vehicle. Applicable to MC, LT, RT, GC architectures. Bounded $[0, 360]^o$.
   (b) **Latencies and delays**: delays that exist in operations. Applicable to RT (communications latency), GC (processing delays and time penalties due to failures), and VBSC architectures (operator interaction times).

Comparing the unmanned vehicle control architectures to the baseline manual control model, Local Teleoperation (LT), Remote Teleoperation (RT), and Gestural Control (GC) systems are the most similar to current operations: each still receives assignments from the Handler, who passes these instructions to crew that relay taxi instructions to the vehicles through physical gestures. LT and RT both utilized a remote human pilot to recognize and interpret those commands, while GC replaces the pilot with a gestural-recognition system capable of identifying directors and interpreting their commands. Modeling these systems only requires creating new vehicle models to replace the original manually controlled F-18 models from Chapter 3. The two remaining Human Supervisory Control (HSC) systems, VBSC and SBSC, require new vehicle models as well, but they also require architectural changes to flight deck operations. Both of these systems are centrally controlled and neither requires interactions with the crew for taxi instructions. Commands are instead provided by a single operator through the central system.

For the VBSC system, assignments and taxi commands would be issued by the Deck Handler through a GUI that allows them to issue catapult assignments and provide taxi waypoints to vehicles individually. In such a system, the Handler behaves as a server, processing requests for instructions from vehicles serially. Past research suggests that operator performance degrades as the number of vehicles controlled in a VBSC system increases; as such, it would be likely that the VBSC operator would only be given control over the VBSC vehicles, and manned aircraft would operate with crew as they always have. For an SBSC system, a central planner would generate assignments for all aircraft at once and issue them in parallel, no longer requiring the Handler to service vehicles individually. However, optimization routines that work in this fashion work best when given control over all elements of the system. For the SBSC model, manned aircraft would be included in the planning system and would no longer be under the control of the Aircraft Directors on the flight deck. These structural changes for the VBSC and SBSC systems have other repercussions as well, as will be discussed later.

Each of these UAV control architectures can be modeled through changes to the variables listed above, an overview of which appears in Figure 5-6 and will be elaborated on through the remainder of this section. The first four lines of the figure describe the four major behavioral variables related to human pilot operations; the last two lines describe the physical limitations of the control architectures when work in the world. In generating parameter values and logic routines, as much as possible, the definitions were based on prior research work in unmanned vehicle development; however, the novel nature of the aircraft carrier domain means that a variety of assumptions must be made in order to provide specific values to vehicle parameters. As described earlier, many of these assumptions assume a "reasonable best-case scenario" for the unmanned systems, assuming that they are mature technologies being used by expert users that are proficient in their operation and whose colleagues trust their capabilities. These assumptions attempt to ensure a fair evaluation of the physical and technological *systems* and attempt to avoid penalizing systems for perceived human social factors such as distrust, dislike, or other issues.



	Manual Control	Local Teleoperation	Remote Teleoperation	Gesture Control (Advanced Teleoperation)	Vehicle-based Supervisory Control	System based Supervisory Control
Routing Method	Manual, Decentralized	Manual, Decentralized	Manual, Decentralized	Manual, Decentralized	Central control	Central control
Sensing and Perception	P(fail) = 0%	P(fail) = 0%; limited by FOV	P(fail) = 0%; limited by FOV	P(fail) = 11.18%; limited by FOV	N/A	N/A
Response Selection	P(fail) = 0%	P(fail) = 0%	P(fail) = 0%	P(fail) = 11.18%	N/A	N/A
Response Execution	P(fail) = 0%	P(fail) = 5% at catapult align	P(fail) = 10% at catapult align	P(fail) = 5% at catapult align	P(fail) = 2.5% at catapult align	P(fail) = 2.5% at catapult align
Field of View (FOV)	85°	65°	65°	65°	N/A (networked)	N/A (networked)
Latency	-	None	Up to 1.6 sec.	U[0-3 seconds] (computation)	$N(3.19, 7.19^2)$ seconds	None

Figure 5-6: Variations in effects of control architectures, categorized along axes of routing method, sensing and perception failure rate, response selection failure rate, response execution failure rate, field of view, and sources of latency.

Visual Acquisition Parameters

One of the key requirements of pilots in flight deck operations it to quickly visually identify the Aircraft Director who is currently providing instructions then sense and perceive nearby Directors, comprehend the instructions they provide through physical gestures and select the appropriate response, then execute that response. For the non-HSC UAV models (LT, RT, and GC), each aspect of this process of Sensing and Perception, Response Selection, and Response Execution may be affected by the properties of the control architecture as defined. Each action may have a higher error rate than that observed for manual control. Each error may also have an associated time penalty. Additionally, visual identification may not occur as quickly for computer systems or remote pilots as for pilots seated within an aircraft.

Local Teleoperation (LT), RT, and GC systems all rely heavily on the use of cameras to observe the outside world, with previous studies reporting that LT and RT operators experience a restricted view of the environment that is detrimental to performance. This constraint on field-of-view should also be included within the MASCS system. The research reported previously in Chapter 2 utilized cameras and displays ranging from 40° to 257° in field of view, while the stereo camera used by Song et al. has models ranging from 43° to 100° (Song et al., 2012). Given that the manual control case was calibrated to 85° in earlier testing, a smaller value will be used for the UAV systems; the median stereo camera field-of-view of 65° is chosen here. Operationally, this means that Aircraft Directors will have to move slightly closer to the centerline of LT, RT, and GC aircraft to be visible during operations. However, the same alignment logic used previously is still applicable; all that is required is an adjustment of the aircraft model's field-of-view parameter. Vehicle-Based Human Supervisory Control (VBSC) and System-Based Human Supervisory Control (SBSC) systems do not utilize cameras for processing tasks or relaying information to a remote operator, implying that this limitation in camera field-of-view has no effect on their operations and is not necessary in the modeling.

Once the Director is within the field of view, LT, RT, and GC systems must then visually identify the Director and recognize that (1) this is their current Director that should be instructing them and (2) the Director is commanding a specific task for them to perform. LT and RT systems both continue to use a human pilot to perform this task, and prior research suggests that once an object of interest is in the remote pilots camera view, they have a very high likelihood of detecting it. As such, the probability of failing to notice a Director and failing to select the correct response a task are both left at 0%. For VBSC and SBSC systems, these variables are again not applicable as these systems do not rely on optical systems for task acquisition and these parameters are again not necessary.

GC systems, however, require substantial changes to the model. Gesture recognition technologies are still maturing, and the ability to recognize the complicated physical gestures performed by

Directors in real-world settings is still an area of investigation. As noted earlier in Chapter 2, the mean failure rate reported in the research, which includes laboratory settings and far less complicated gestures, is currently 11.16%. While performance will certainly improve in laboratory settings, a variety of environmental conditions (sun glare, obstructions, large numbers of moving vehicles) ultimately limit performance in the real world. The median failure rate in the research is quite likely to be close to the real-world failure rates, even with enhanced processing and vision capabilities. As such, the median failure rate of 11.16% observed in the research is used as the failure rate of the baseline GC system; the effects of this parameter on GC behavior will be explored later on in Chapter 6.

There are also technical differences between identifying a Director and selecting a response given their commands, with the latter being much more difficult. Failing to recognize either a Director or a Task would involve some form of time penalty, either for a Director to issue an override to obtain control over the vehicle (perhaps through a handheld wireless device) or to issue a “stop” command to the vehicle and attempt to issue new instructions, respectively. A logical GC implementation in this form would imply that a failure to recognize a Director might only happen once, after which the Director would have the authority to force control of the vehicle. A failure to recognize tasks does not have such a solution, due to the significant number of tasks that might be assigned. It thus may take several attempts for the vehicle to register the correct command, with each failure accruing a time penalty. For the baseline MASCS model of Gestural Control, the penalty for failing to acquire a new Director is set at 5 seconds, representing the time required to realize the vehicle has not recognized the Director and for the Director to issue an override. The time penalty for failing to acquire task is modeled as a uniform distribution between 5 and 10 seconds, representing the time for the operator to stop, issue a cancel command, and wait for the system to become ready to accept new instructions (these times are also influenced by the expected latency in signal processing, described later in Section 5.1.2). The number of failures in command comprehension and response selection is determined at the start of each action commanded by a Director by a Bernoulli trial generator, after which the total time penalty is solved by randomly sampling the uniform interval $[5, 10]$ once for each failure incurred.

Task Execution

The last of the original performance parameters involves task execution, in terms of alignment to the catapult for launch. Chapter 3 noted that the probability of pilots failing to align to catapults was very low, less than once per more than 200 launches. This was incorporated in the original Manual Control model partially validated in Chapter 4. In transitioning from manual control to remotely operated systems, previous research has reported that, even with large field-of-view cameras, the remote pilots of LT and RT systems suffer degraded performance in their tasks. The common theme in the literature reviewed in Chapter 2, although utilizing a variety of vehicles and tasks, was that tasks that required accurate alignment of a vehicle (or robotic arm) were most difficult to perform and were even more so when in the presence of signal lag. This suggests that both LT and RT systems should suffer a higher failure rate than manual control, but the information available from prior research does not provide a true estimate of failure probability (most often reporting numbers of errors). An appropriate model may be to raise the LT failure rate to once per mission (5%, once per 20 launches) and RT to roughly twice per mission (10%, once per 10 launches).

For the more advanced GC, SBSC, and VBSC systems, the limitations of the remote pilot to sense his or her location in the world and align to a target are removed, but these systems instead require some sort of technological intervention to ensure accurate alignment. It would be reasonable to assume, for a future system being allowed full operational status on a carrier, that the performance of these systems would be equal to or better than LT performance. Also, because VBSC and SBSC systems would rely on greater autonomy and likely include enhanced sensor capabilities as opposed to the rest, they would likely have the best non-MC failure rates. However, given the difficulties of navigating on the flight deck, it is difficult to argue that the failure rate would be zero. These ideas then present a relationship of failure rates between systems: MC < VBSC&SBSC < LT&GC < RT (MC has a lower failure rate than VBSC and SBSC systems, which is lower than LT and GC systems, which are better than RT systems). Given that the chance of catapult alignment failure for manual control is once per many missions (100+ launches), and that LT was once per mission, GC is also set to once per mission (5%) while VBSC and SBSC are set to once per two missions (2.5%). This failure, like the visual recognition failures, also includes a time penalty, which Directors report as being in the range of 45-60 seconds.

Latencies

Outside of the original parameters, certain UAV systems may introduce latency into the process of response selection and task execution. This is expected to occur for RT, GC, and VBSC systems, albeit for different reason in each. VBSC systems incur a form of latency related to the Handler's ability to process tasks through the supplied GUI. This is described in the next section and will be accompanied by a description of the VBSC system latencies. The source of latency in GC systems comes from software processing as the system attempts to translate the hand gestures of the Director into understandable action commands. Prior research shows this requires anywhere <<1 to 3 seconds. This is modeled as a uniform distribution over the range $[0, 3]$, randomly sampled at the start of every action commanded by a Director. Remote Teleoperation (RT) system latencies arise from the transmission of signals from the vehicle to the remote operator, who issues commands that take additional time to travel back. This means that there is a defined lag from the time the Director commands a task to the time that the operator begins execution, as well as a potential lag at the end of the task when the Director commands the next task. Prior reports have placed the round-trip latency at 1.6 seconds (Vries, 2005), meaning that that the start of each action is delayed by 1.6 seconds.

The effects at the end of the action are not as straightforward, however. Prior work demonstrated that experienced operators adapt their strategies to the system, commanding tasks that attempt to anticipate the lag, then wait for the tasks to complete execution before issuing new commands (Sheridan & Ferrell, 1963; Sheridan & Verplank, 1978). Such a strategy would likely be used by experienced remote pilots in flight deck operations in relation to when they expect the Director to issue a command. This provides a continuum of possibilities ranging from ending the task too early to ending the task too late; Figure 5-7 shows how this might occur for the "worst case" late completion of a task. Two timelines are depicted, one for the actions of the Director and Pilot (top), the second for the response of the vehicle (bottom). The vehicle beings its motion task on the left at $T = 0$, and it should complete at $T = 5$. In the very worst case scenario, the pilot does not anticipate the end of the task, waiting for the Director to issue the "STOP" command. If this occurs at $T = 5$, when the moving aircraft has reached the desired end point, the round-trip latency in the system would delay the end of the task by 1.6 seconds: the time required for the "STOP" command to travel to the remote pilot and for this "STOP" signal to reach the vehicle. What was supposed to be a 5-second task completes in 6.6 seconds.

In order to execute the task precisely as defined, preemptive action must be taken either by the Director or the Pilot to compensate for the latency in the system; these actions appear in purple towards the right side of Figure 5-8. The Director must provide a "STOP" command at least 1.6 seconds early to account for the round trip latency of 1.6 seconds. The remote Pilot must act 0.8 seconds early as they must only handle latency in one direction. Pilots might also overcompensate for the lag in the system if they do not have an accurate mental model of the current latencies in communications of if they are unsure of the final taxi destination. For experienced operators working with a mature system, it is doubtful that they would initiate actions any earlier than double the

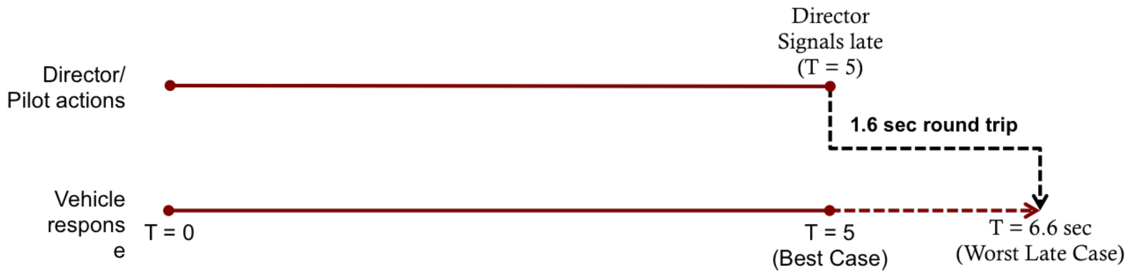


Figure 5-7: Diagram of worst-case scenario for RT latency effects for a five-second task. Upper timeline shows Director and Pilot actions, bottom timeline shows vehicle response to Pilot commands. If Director provides command late (at the 5 second mark) and the Pilot does not preemptively stop the aircraft, the round-trip 1.6 second latency in communicating the "STOP" command to the vehicle, the pilot issuing the command, and the command being executed causes the vehicle to stop late. The task completes at 6.6 seconds.

lag (3.2 seconds before the action should stop). Given that the latency in a one-way signal is 0.8 seconds, this places a minimum bound at tasks ending 2.4 seconds early. In Fig. 5-8, this shows that the 5 second task desired would end at T =2.6 seconds. While this is a significant shift in terms of a 5 second task, most taxi tasks on the flight deck take much greater than 5 seconds to complete and the relative inaccuracy is not nearly as severe.

Given the assumptions in the previous two paragraphs, we have a continuum of possibilities ranging from tasks being complete 2.4 seconds early to 1.6 seconds late. This can be modeled in MASCS as an adjustment to the expected completion time of the task itself. To reflect the variability in behavior, this takes the form of a uniform distribution over the range $[-2.4, 1.6]$ sampled randomly for each task and added to the expected end time.

UAV Routing Methods

The final major difference between UAV Control Architectures involves how they are issued taxi commands and routed through the flight deck. As described earlier, LT, RT, and GC systems still operate under zone coverage, although their performance is affected by various factors described in the previous sections. VBSC systems would rely on commands issued from the Handler through a Graphical User Interface, which requires modeling the behavior of the Handler in operating such a system. Prior work has demonstrated that operators of these system behave as queuing servers, processing vehicle tasks individually according to some form of queuing policy, with the time required to issue tasks being modeled by a randomized distribution. For flight deck operations, experienced Handlers would likely operate in a “highest priority first” fashion, prioritizing aircraft nearest to catapults in order to keep traffic flowing on the deck.

The time required to process tasks is taken from prior research by Nehme, whose research involved operators controlling a team of unmanned vehicles through a VBSC-style interface, requiring them to define waypoints that defined vehicle transit paths, among other task assignments (Nehme, 2009). Nehme characterized the “idle” task, in which operators assigned a new transit task to a waiting

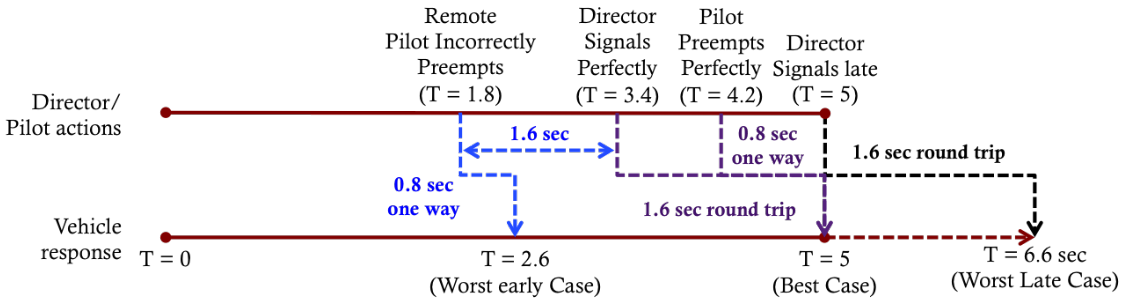


Figure 5-8: Diagram of optimal and worst-case early scenarios for RT latency effects for a five-second task. Upper timeline shows Director and Pilot actions, bottom timeline shows vehicle response to Pilot commands. Perfect execution requires preemptive commands by either the Director or the Pilot. If the Pilot further pre-empts the Director's actions, the task may complete early.

vehicle, as normally distributed with a mean of 3.19 seconds and standard deviation of 7.19 seconds. The idle task described by Nehme is analogous to the action of assigning taxi tasks to idling aircraft on the flight deck, so this same distribution will be used within MASCS for the baseline model of VBSC systems. The interaction of the Handler's interaction time with each vehicle and the current number of vehicles in the queue generates latencies for VBSC systems: not only must a VBSC vehicle wait $N(3.19, 7.19^2)$ seconds at the start of each of its own tasks, it must wait for all other vehicles ahead of it in the queue to be processed. If 4 vehicles are ahead in the queue, the fifth vehicle must wait (one average) $3.19 * 5 = 15.95$ seconds before starting its next task. Thus, there is a significant penalty for including additional vehicles in the VBSC queuing system, which has been characterized in many previous studies (see (Cummings & Guerlain, 2007; Cummings & Mitchell, 2008)).

Additionally, since this style of routing does not rely on the network of crew on the flight deck, the Handler can also assign longer taxi tasks on the flight deck — taxiing from aft to forward, barring other constraints, could be done in a single motion rather than requiring multiple handoffs between Directors. This is essentially a change in the topology of the flight deck; as described in Chapter 3, the crew "zone coverage" routing system defines how aircraft can move throughout the flight deck. The network of possible routes is altered for the SBSC and VBSC models, with longer taxi movements now possible. However, the location of nodes in the network remains the same, with the two supervisory control models allowed to make longer taxi motions. The general flow of traffic, however, remains the same: Handlers have experience with flight deck operations and would likely still employ the same general rules and routes of taxi actions as used today.

SBSC systems are also planned from a centralized systems but instead use a centralized planning algorithm to define tasks for all vehicles simultaneously. All assignments are made in parallel, and the Handler no longer functions as a queuing server in processing tasks. Once assignments are made, just as with VBSC vehicles, tasks are carried out without the assistance of the Aircraft Directors in taxi operations. What types of planning algorithms are useful in the aircraft carrier environment is still an area of ongoing research, and prior work in this area demonstrated that even complex planning algorithms have trouble decomposing the geometric constraints on the flight deck and provide little benefit over human planning (Ryan et al., 2014). The appropriate development and testing of planning algorithms for the flight deck lies outside the scope of this research; for SBSC systems, the same Handler planning heuristics applied to the other CAs, which performed as well or better than other planning algorithms in prior work, are applied here. As such, all five UAV control architectures utilize the same Deck Handler assignment heuristics for assigning aircraft to launch catapults.

A final change for the VBSC and SBSC systems involves the use of crew on the flight deck. It was previously noted that the Aircraft Directors are not needed for issuing taxi tasks to VBSC and SBSC vehicles on the flight deck. However, given the Navy's emphasis on safety in deck operations, UAVs still would not likely be allowed to taxi entirely on their own. It is likely that white-shirted Safety Officers, already used to observe operations on the current flight deck, might be transitioned into roles as aircraft Escorts, accompanying VBSC and SBSC vehicles as they taxi along to ensure that they maintain sufficient clearance to avoid hitting other aircraft. These Escort crew might also be supplied with a “kill switch” to stop the vehicle temporarily if a collision is imminent and the vehicle appears to not recognize the danger. These crew are modeled as new Agents within the MASCS system, requiring a modeling of assignment to escort specific vehicles as needed and using similar alignment logic as the Aircraft Directors. At the start of a vehicle’s taxi operations, an Escort is assigned that follows them through the duration of taxi operations; once the vehicle launches, the Escort returns to a “home” position to await a new task. A total of ten Escorts are placed in the system, with clusters of five appearing forward and aft. This provides the capability to taxi eight vehicles to catapults while two others are returning from just-completed taxi operations and is a sufficiently large number to not constrain operations.

Other Effects

Additionally, for GC, VBSC, and SBSC systems, the shape of the vehicle is changed from an F-18 planform to a X-47B Pegasus unmanned vehicles. Qualitatively, F-18s are longer and have a smaller wingspan than the X-47s. However, the thresholds of collision avoidance were adjusted to maintain the same distances for each model to maintain a consistency of behavior. Additionally, the flight crew is assumed to have enough experience with the UAVs to be comfortable with working with them (they trust the vehicles to perform safely) and taxiing them at the same speeds as manual control aircraft. The distribution of speeds for the UAV models is thus identical to that used for Manual Control, which also removes any potential confounds due to speed differences.

Control Architectures Summary

Together, the changes in parameters described in the previous sections define the differences between Manual Control aircraft and each of the five UAV Control Architectures defined for MASCS, as detailed in Figure 5-6. If properly modeled, these changes to parameters and models should generate significant differences in performance for individual vehicles taxiing on the flight deck. The movement from Manual to Local Teleoperation introduces a minor constraint on field of view on the pilots and would be expected to worsen performance slightly. The move from Local Teleoperation to Remote Teleoperation introduces (in this model) a small lag into communications that delays the start of any task and affects the end of task execution, both of which should potentially degrade performance. The move from Remote Teleoperation to Gestural Control replaces the pilot with a less-accurate computerized system with a larger processing time at the start of each task and the chance to fail multiple times. This worsens performance further and should provide the worst performance across all systems.

Moving from Gestural Control to Vehicle-based Human Supervisory Control eliminates the inaccurate gesture-processing system but replaces it with a single supervisor processing tasks for all vehicles in series. For a single aircraft, the delay in supervisor interaction may be less than the delays due to the gesture recognition system; if this occurs, the GC system should produce better performance. At higher numbers of aircraft, however, more vehicles are placed into the VBSC supervisor queue, greatly increasing the supervisor's workload. Under these conditions, the wait times for vehicles may increase significantly, leading VBSC systems to provide poorer performance than GC systems at the mission level. Lastly, because System-based Human Supervisory Control provides the ability to assign all tasks simultaneously, the limitations of the supervisory in servicing vehicles in series is removed and should greatly improve the performance of SBSC systems. Together, these differences in performance should appear as an inverted-U for individual aircraft models. This may not be true at the mission level, where interactions with the crew routing methods, catapult queuing strategies, and the Deck Handler planning heuristics may either mitigate or exacerbate these qualities. Section 5.2 tests each of the control architectures using a single aircraft mission to explore whether or not the six variables described here drive variations in performance.

5.1.3 Mission Settings: Composition and Density

The remaining two independent variables address how the context of the mission itself affects performance; these appear on the horizontal axis in Figure 5-1. The first variable is mission Density, set to either 22 or 34 aircraft, which examines how increases in the number of vehicles affects operations. Subject Matter Experts describe a typical mission as including 22 aircraft, with 34 being a maximum value for operations. At this 34 aircraft setting, the number of available taxi routes is reduced due to the number of aircraft parked on the flight deck. Additionally, one of the forward catapults (catapult 1) is covered with parked aircraft and not usable. The Street area also contains more aircraft than the 22 aircraft setting, leading to an increase of the number of vehicles performing close proximity interactions. Due to these factors, the assignment of aircraft also evolves in a different fashion; aircraft parked on catapult 1 must be taxied early in the mission, which means that the Street area remains blocked for a significant amount of time, limiting the flexibility of aircraft assignments.

The Composition variable addresses the fact that the Area, Temporal, and Temporal+Area Separation protocols described in Section 5.1.1 can only be applied while at least one manned and one unmanned aircraft are present on the flight deck. The Composition variable describes the relative numbers of unmanned aircraft involved in operations as a percentage of the Density value. At the 22 aircraft Density level, a 25% Composition setting uses 5 unmanned aircraft and 17 Manual Control aircraft. Differences in the values of the Composition variable affects the implementation of the different safety protocols and how they affect the evolution of a mission. These effects are summarized in Table 5.1. For example, under the Area Separation protocol, the two groups of manned and unmanned aircraft work in parallel in specific, segregated areas of the flight deck and have equal access to catapults, regardless of the Composition level. Under the 50% Composition protocol using Area Separation, these two groups are equal in number and should launch at similar rates; the Area Separation protocol should remain in effect for the entire mission. At the 25% Composition level under Area Separation, the small set of unmanned aircraft will launch more quickly than the larger set of manned aircraft; once the unmanned aircraft have launched, the Area Separation constraint can be relaxed as the only aircraft that remain are manned. The same constraints are applied to both the 25% and 50% Composition cases, but differences in the number of unmanned aircraft lead to differences in how the missions evolve.

Under the Temporal Separation protocol, the differences between Composition values are only in terms of when unmanned aircraft become active during a mission. In both cases, manned aircraft operate first, starting from their positions in the forward area of the flight deck. The larger number of manned aircraft in the 25% Composition level means that unmanned aircraft will become active later in the mission than would occur under the 50% Composition level. This is also true for the Temporal+Area protocol, with the only difference being that T+A cases send all manned aircraft forward. Missions can also be run at a 100% Composition level, utilizing only a single type of vehicle during a mission (only manned or only unmanned), but the Area, Temporal, and Temporal+Area protocols will not be feasible as only one type of vehicle is present. Dynamic Separation is the only applicable protocol for the 100% Composition cases, with all other entries crossed out in Fig. 5-1.

Before proceeding to the execution of the matrix, however, additional testing should be performed in order to verify that the UAV control architecture models and safety protocol logic routines function as desired. The next section replicates the testing done in Chapter 4 in single aircraft calibration with the new UAV models. While the results cannot be directly validated against empirical data, the results should demonstrate the expected inverted-U shape of performance.

Table 5.1: Description of effects of Safety Protocol-Composition interactions on mission evolution.

	Area Separation	**Temporal Separation**
Deck Access (same for all Composition values)	Restricted to specified area	Unrestricted
Operations (same for all Composition values)	Parallel (across areas)	Series (manned first, then unmanned)
Composition 50%	Aircraft groups launch at similar rates. Constraints likely to remain in place throughout mission	Unmanned aircraft become active halfway through mission
Composition 25%	Smaller group of unmanned aircraft launches quickly. Constraints are relaxed afterwards	Unmanned aircraft become active three-quarters through mission

5.2 Unmanned Aerial Vehicle Model Verification

The changes defined in section were implemented into the MASCS model of flight deck operations, defining new unmanned aircraft object classes and adjusting parameters for those agent models, as well as making adjustments to existing logical structures within the system. Once these tasks were completed, individual submodels within the code (e.g., the Bernoulli trial generator replicating GC response selection failures) were verified for accuracy. After this was completed, a series of verification tests of the single aircraft models were conducted using the same mission profile as the single aircraft calibration testing described in Chapter 4, Section 4.2, again conducting 300 replications in each trial. Testing of the UAVs proceeded incrementally, introducing a single new feature at a time in order to gauge its effects; these results are reported in Appendix J. The results of the final model tests appear in Figure 5-9, which provides the mean Launch Duration value for each vehicle class with whiskers indicating ±1 standard deviation.

As can be seen in the figure, the performance across unmanned vehicle control architectures is that of the expected inverted-U shape. A Kruskal-Wallis non-parametric one-way analysis of variance shows that significant differences exist among the data at the $\alpha = 0.05$ level ($\chi^2(5) = 197.1712, p < 0.0001$). A Steel-Dwass simultaneous comparisons test[1] reveals significant differences between multiple pairs of CAs. A total of nine significant pairwise differences exist at the $\alpha = 0.05$ level, comprising all possible combinations between two sets of three CAs: the three "delayed" cases of RT (mean rank= 981.15), GC (mean rank= 1153.39), and VBSC (mean rank= 1046.74), where vehicles experience some sort of latency or lag in operations, and the three "non-delayed" cases of MC (mean rank= 786.26), LT (mean rank= 771.84), and SBSC (mean rank= 663.63).

[1] Steel-Dwass is a nonparametric version of the Tukey test; see (Conover, 1980).

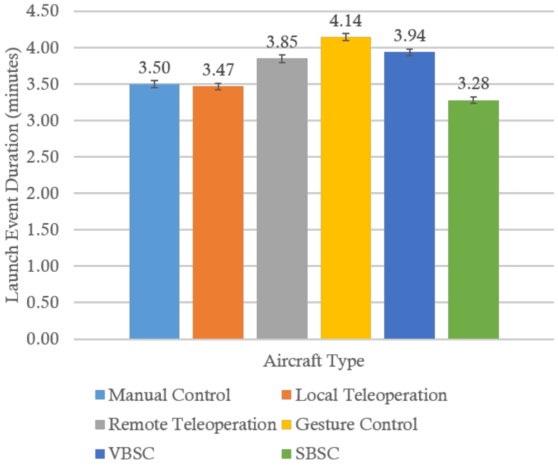


Figure 5-9: Column chart comparing time to complete single aircraft mission across all five UAV types. Original results from Manual Control modeling are also presented.

These results for individual aircraft tasks suggest that latencies and lags experienced in task execution by these vehicles have significant effects on performance, although this was expected. However, as observed in the sensitivity tests of the manual control model in Chapter 4, Section 4.4, changes that affect taxi operations "upstream" of operations may not actually have effect at the mission level. The performance of the different UAV control architectures might also be influenced by interactions with the Safety Protocols defined in this chapter in Section 5.1.1. Determining the drivers of variation in safety and productivity on the flight deck is the focus of the MASCS experimental testing program, described in the next section.

5.3 Dependent Variables and Measures

The focus of the MASCS experimental program is to examine the changes in safety and productivity in aircraft carrier flight deck operations across different Control Architectures and Safety Protocols. As noted in Chapter 4, very little information on flight deck operations is publicly available. As such, there is little guidance in terms of what metrics are of importance for flight deck operations; even if this guidance were available, it is not clear if these metrics would be appropriate for assessing futuristic unmanned vehicle operations. Rather, inspiration for the performance metrics for use in MASCS testing is taken from prior work on unmanned vehicles and driving research.

In prior work, Pina et al. defined five major classes of metrics for human supervisory control systems that examine not only the performance of the entire human-unmanned vehicle system, but that also examines humans and unmanned systems individually, as well as their efficiency in collaboration and communication (Pina, Cummings, Crandall, & Penna, 2008; Pina, Donmez, & Cummings, 2008). While all five classes are applicable to assessing the UAVs used in MASCS, four of the five classes correspond to *settings* within MASCS and were defined as part of the various control architectures; only the category of Mission Effectiveness (ME) metrics would exhibit significant variation in MASCS testing.

This class of Mission Efficiency metrics describes measures of productivity within the aggregate system. Pina et al. subdivided these measures into categories of *Expediency* (time to complete tasks), *Efficiency* (resources consumed in completing tasks), and *Safety* (how much risk was incurred during task execution). For MASCS, Expediency measures address how quickly the launch event was completed, termed the Launch event Duration (LD). Efficiency measures address how well operations were conducted, tracking parameters related to individual vehicle task execution and helping to clarify differences in LD and the effects of the safety protocols on operations. Total Taxi Distance (TTD) tracks the total travel distance of aircraft, providing an understanding of how efficiently aircraft were routed on the flight deck, while Total Aircraft taXi Time (TAXT) and Total Aircraft Active Time (TAAT) examine how long aircraft were actively executing taxi tasks or all tasks (including launch preparation), respectively.

Safety measures address both collision and non-collision hazards. Traffic researchers have explored the use of “surrogate” safety measures as proxies for true accident values. These surrogate measures tracks events that occur more often than true accidents (making them easier to track) but remain highly correlated with them (Archer & Kosonen, 2000; Archer, 2004; Gettman & Head, 2003; Gettman, Pu, Sayed, & Shelby, 2008). These measures often address the distances between vehicles and the time-to-collision, both typically in regards to specified thresholds (e.g., one car length or within one second of collision). The first set of MASCS safety measures tracks the number of times another vehicle or human is within a specified distance of an aircraft; every time another agent moves within this distance of a vehicle, that vehicle’s counter increases by one. Three different distance values are utilized within MASCS, corresponding to Primary (PHI, half-wingspan), Secondary (SHI, full wingspan), and Tertiary (THI, two wingspans) Halo Incursions (Figure 5-10). The term “incursion” is borrowed from commercial aviation, in which the entry of an unapproved aircraft into a runway (or other restricted space) is known as an “incursion.” While these restricted areas are specified as an enforceable constraints, “halos” in MASCS are purely descriptive measures (but could be used as control features). The counts of incursions can also be subdivided by the type of agent crossing the halo radius (aircraft or crewmember). While the inner-most Primary Halo Incursion (PHI) measure is a small distance, it is common (although undesirable) for Aircraft Directors to be this close to aircraft during operations.

Companion measures to the Halo Incursions track the total amount of time that each “halo” is contains an incursion, as it is not clear which variable is more highly correlated with accidents on the flight deck: the number of incursions or the total time that incursions exist. A variation of the halo incursion measure takes guidance from traffic research and views collisions from a predictive stance, tracking how many times a vehicle is within one second of colliding with another actor on the

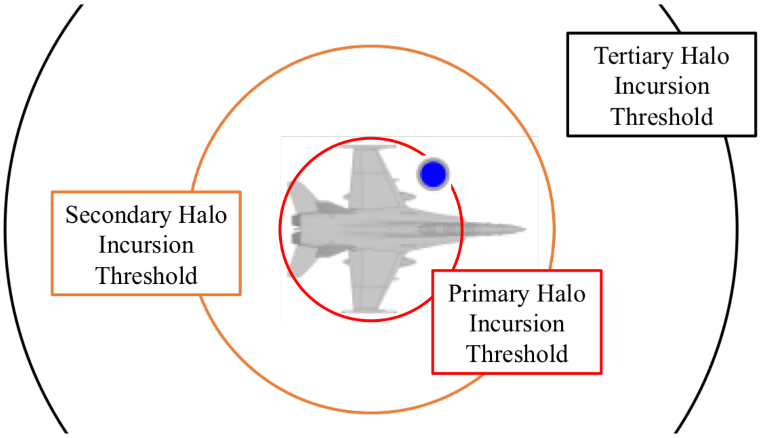


Figure 5-10: Diagram of “halo” radii surrounding an aircraft. Primary incursions are within one half wingspan of the vehicle. Secondary incursions are within one wingspan; tertiary are within two. In the image shown, the aircraft has its wings folded for taxi operations. A transparent image of the extended wings appears in the background.

flight deck. This prediction measure also utilizes the radii from the figure above, tracking predicted Primary (PC), Secondary (SC), and Tertiary (TC) collision warnings.

Non-collision safety measures address the factors that influence other hazards on the flight deck, taking inspiration from the idea of latent failures. The primary metric in this class is the total time that vehicles are active within the landing strip area, term the Landing Strip Foul Time (LSFT). Blocking the landing strip may potentially delay airborne aircraft low on fuel from landing, directly increasing their risk of accidents. Other metrics focus on congestion on the flight deck, tracking the total number of aircraft that transit through the Street and Fantail area (NS, NF), the maximum number of aircraft in those areas at one time (MS, MF), and the total amount of time that these areas are occupied (DS, DF). A list summarizing these primary performance measures appears in Table 5.2. The next section addresses the experimental hypotheses of performance for these dependent measures in terms of the experimental matrix shown previously in Fig. 5-1. Hypotheses are presented both in terms of safety protocol performance with a given control architecture and across control architectures individually (independent of safety protocols).

5.4 Experimental Hypotheses

The hypotheses for the experimental matrix in Figure 5-1 are organized into two different matrices in two figures below. Figure 5-11 shows the hypotheses across control architectures (independent of safety protocols), while Figure 5-12 examines hypotheses for safety protocols within each control architecture. For every dependent variable described in the previous section, lower scores are better indicating either lower risk of operations or faster/more efficient mission execution. Entries in the cells in the figure denote whether performance should provide smaller values (better) or larger values (worse), with blank cells denoting performance somewhere in between.

Within Figure 5-11, the hypotheses are that, regardless of SP settings, the detrimental effects of the delays and errors of the GC system will generate the worst performance in terms of Expediency

Table 5.2: List of primary performance metrics for MASCS testing.

Category	Metric	Abbreviation
Expediency	Launch event Duration	LD
	Average interdeparture time	ID
Efficiency	Total Aircraft taXi Time	TAXT
	Total Taxi Distance	TTD
	Catapult wait time	
Safety	Primary, Secondary, and Tertiary Halo Incursions	PHI, SHI, THI
	Duration of Halo Incursions	DPHI, DSHI, DTHI
	Predicted Collisions (Primary, Secondary, Tertiary)	PC, SC, TC
	Landing Zone Foul Time	LSFT
	Number of vehicles in Street (total, max at any time)	NS, MS
	Number of vehicles in Fantail (total, max at any time)	NF, MF
	Duration of time of vehicles in Street, Fantail	SD, FD

and Efficiency measures, while the lack of constraints afforded by the VBSC and SBSC systems should generate the best performance (smallest values) on these measures. GC should also generate the worst performance for Safety measures that deal with time (Duration of "Halo" incursions, etc.), as longer mission times should also correlate to larger values for those measures. For Safety metrics dealing with counts of near collisions and predicted collisions, the SBSC control architecture is expected to provide the highest values (worst performance) because it lacks the constrained taxi patterns and artificial spacing introduced by the "zone coverage" system. However, because removing these constraints is expected to increase the speed of operations, it is expected that safety measures involving durations of time would perform better (lower values) for SBSC. For all metrics considering the number of vehicles sent to the Street and the Fantail, the choice of CA should have no effect; greater effects for these should come from the Safety Protocols and are reviewed in Figure 5-12.

In terms of Figure 5-12, within individual Control Architectures, the Dynamic Separation only protocol should provide the best performance in terms of productivity measures as it provides the fewest constraints on operations. The Temporal Protocol should also perform well on these measures as it retains the flexibility to allocate aircraft across the entire deck. Conversely, the constraints put in place by the Area and Temporal+Area cases should be detrimental to productivity measures, most especially the Temporal+Area case.

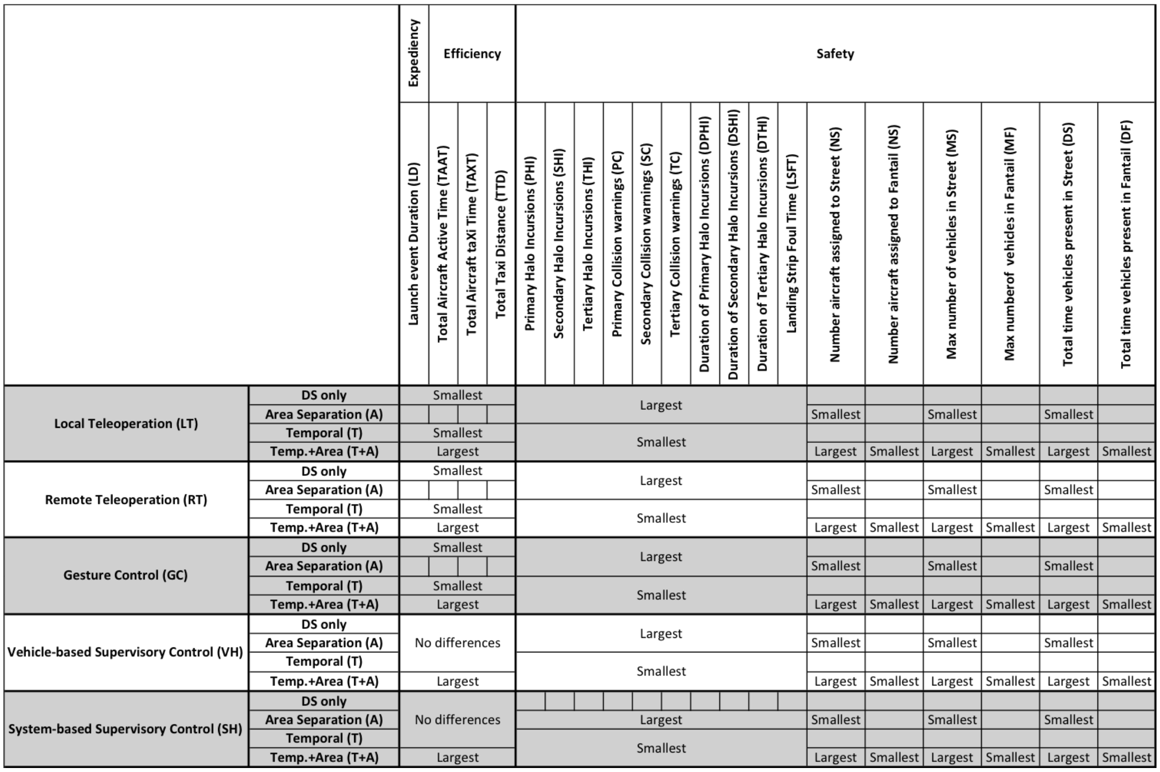
	Expediency	Efficiency			Safety															
	Launch event Duration (LD)	Total Aircraft Active Time (TAAT)	Total Aircraft taXi Time (TAXT)	Total Taxi Distance (TTD)	Primary Halo Incursions (PHI)	Secondary Halo Incursions (SHI)	Tertiary Halo Incursions (THI)	Primary Collision warnings (PC)	Secondary Collision warnings (SC)	Tertiary Collision warnings (TC)	Duration of Primary Halo Incursions (DPHI)	Duration of Secondary Halo Incursions (DSHI)	Duration of Tertiary Halo Incursions (DTHI)	Landing Strip Foul Time (LSFT)	Number aircraft assigned to Street (NS)	Number aircraft assigned to Fantail (NS)	Max number of vehicles in Street (MS)	Max numberof vehicles in Fantail (MF)	Total time vehicles present in Street (DS)	Total time vehicles present in Fantail (DF)
LT					No difference															
RT																				
GC	Larger										Largest								Largest	
V-HSC	Smaller										Smaller									
S-HSC	Smallest				Largest						Smallest								Smallest	

Figure 5-11: Hypotheses for performance across Control Architectures. E.g.: System-Based Human Supervisory Control (SBSC) is expected to perform best (smallest values) for all productivity measures and safety measures (smallest) related to durations. Gestural Control (GC) is expected to perform worst (largest values) across all measures related to time.

In terms of safety measures, however, these same constraints are expected to afford Temporal+Area the best performance, as it should minimize the interaction between vehicles of all types. The Temporal protocol is also expected to fare better in safety measures than the Area Separation protocol: because the Area Separation protocol confines aircraft into specific regions of the deck (increasing the density of operations in each area), the collision-related measures are all expected to perform worse in these cases. However, Area Separation is expected to provides the lowest use of the Street area of the deck; conversely, the T+A protocol should utilize the Street most heavily (worst performance) and the Fantail the least.

The only expected interaction between Safety Protocols and Control Architectures is that, because the VBSC and SBSC cases do not utilize the zone coverage routing methods, their productivity performance is not expected to differ between safety protocols. An interaction is also expected between Composition and the Temporal+Area Separation protocol. Because this protocol isolates manned aircraft in the forward section of the flight deck, neglecting the aft catapults, utilizing more manned aircraft in the 25% Composition case should lead to longer Launch event Duration values.

The dependent variables described in the previous paragraphs were integrated into the MASCS



		Expediency	Efficiency			Safety															
		Launch event Duration (LD)	Total Aircraft Active Time (TAAT)	Total Aircraft taXi Time (TAXT)	Total Taxi Distance (TTD)	Primary Halo Incursions (PHI)	Secondary Halo Incursions (SHI)	Tertiary Halo Incursions (THI)	Primary Collision warnings (PC)	Secondary Collision warnings (SC)	Tertiary Collision warnings (TC)	Duration of Primary Halo Incursions (DPHI)	Duration of Secondary Halo Incursions (DSHI)	Duration of Tertiary Halo Incursions (DTHI)	Landing Strip Foul Time (LSFT)	Number aircraft assigned to Street (NS)	Number aircraft assigned to Fantail (NS)	Max number of vehicles in Street (MS)	Max numberof vehicles in Fantail (MF)	Total time vehicles present in Street (DS)	Total time vehicles present in Fantail (DF)
Local Teleoperation (LT)	DS only	Smallest				Largest															
	Area Separation (A)															Smallest		Smallest		Smallest	
	Temporal (T)	Smallest				Smallest															
	Temp.+Area (T+A)	Largest														Largest	Smallest	Largest	Smallest	Largest	Smallest
Remote Teleoperation (RT)	DS only	Smallest				Largest															
	Area Separation (A)															Smallest		Smallest		Smallest	
	Temporal (T)	Smallest				Smallest															
	Temp.+Area (T+A)	Largest														Largest	Smallest	Largest	Smallest	Largest	Smallest
Gesture Control (GC)	DS only	Smallest				Largest															
	Area Separation (A)															Smallest		Smallest		Smallest	
	Temporal (T)	Smallest				Smallest															
	Temp.+Area (T+A)	Largest														Largest	Smallest	Largest	Smallest	Largest	Smallest
Vehicle-based Supervisory Control (VH)	DS only	No differences				Largest															
	Area Separation (A)															Smallest		Smallest		Smallest	
	Temporal (T)					Smallest															
	Temp.+Area (T+A)	Largest														Largest	Smallest	Largest	Smallest	Largest	Smallest
System-based Supervisory Control (SH)	DS only	No differences																			
	Area Separation (A)					Largest										Smallest		Smallest		Smallest	
	Temporal (T)					Smallest															
	Temp.+Area (T+A)	Largest														Largest	Smallest	Largest	Smallest	Largest	Smallest

Figure 5-12: Hypotheses for performance of Safety Protocols within each Control Architecture. Example: for Local Teleoperation (LT), the Temporal Separation safety protocol is expected to perform best (smallest values) across all measures of productivity and safety. Area+Temporal is expected to provide the worst productivity measures (largest values), while Area Separation is expected to be the worst for safety.

system and verified before beginning experimental runs, with the system set to output all measures to file (as well as provide logs of system behavior) at the end of each simulation iteration. Each entry in the experiment matrix was replicated thirty times, taking between one hour and 2.5 hours to complete each set of replications (running at 10x normal speed). These results were then compiled and examined, the results of which are presented and discussed in the next section.

5.5 Results

The experimental matrix shown in Figure 5-1 was executed using a team of Windows 7 machines using the MASCS Java codebase and the Eclipse Java IDE. Simulations were executed at identical screen resolutions (1536x864) across machines, with confirmation runs executed prior to the start of the test matrix to ensure that the results of the simulation were not affected by differences between computer systems. Thirty replications of each entry in the experimental matrix were performed (92 x 30 = 2760 runs in total). This section reviews the differences in performance in terms of these dependent variables for the MASCS simulation environment, examining how changes in Control Architecture (CA) and Safety Protocol (SP) interact both with one another and with the mission configuration (number of total aircraft and percentage of UAVs within the mission).

This section utilizes Pareto frontier graphs to describe the general trends in the data, with statistical tests formally determining whether any significant differences exist between cases. The variances were typically not equal across data sets, and the statistics employed in the analysis are primarily non-parametric Kruskal Wallis one-way analyses of variance and non-parametric Steel-Dwass simultaneous comparison tests (Conover, 1980). As the testing generated a very large amount of data, not all of it can be shown in this section. Appendices K through N include additional analyses on the collected data as well as details on the statistical tests performed on the data as well as additional data plots that not covered in this chapter. The section begins by reviewing differences in the Launch event Duration (LD) productivity measure before moving on to the performance of the remaining safety and productivity metrics.

5.5.1 Effects on Mission Expediency

The time required to complete a launch event served not only as an important validation metric for the baseline manual control MASCS simulation, it is also the most critical evaluation metric for Subject Matter Experts (SMEs) in carrier operations. In Section 5.4, it was stated that the Gestural Control (GC) system, because of its delays and errors, should perform the worst overall while the Vehicle-Based Human Supervisory Control (VBSC) and SBSC systems should perform the best and possibly be equivalent to Manual Control (MC) performance. The Local Teleoperation (LT) and Remote Teleoperation (RT) systems were expected to lie somewhere in between. In terms of Safety Protocols, it was expected that for all productivity measures (Expediency and Efficiency), Temporal Separation (T) Separation and Dynamic Separation (DS) would provide the best performance, Temporal+Area Separation (T+A) the worst, and Area Separation (A) in between.

Figure 5-13 first provides column charts for LD performance for both Safety Protocols (left) and Control Architectures (right) for the 22 aircraft Density level. This data does not differentiate between the different Composition values [2]. From the data on Safety Protocol performance (left), the detrimental effects of the T+A Safety Protocol can be observed, confirming the earlier hypothesis regarding its performance. LD values for the remaining settings are all fairly similar, however: Temporal, Area, and Dynamic Separation are not substantially different from one another. The data for control architectures (right) also indicates that there is little variation across the different cases.

Statistical tests show that there are significant differences amongst control architectures as well as between safety protocols. A Kruskal-Wallis nonparametric one-way analysis of variance for the Area, Temporal, and Dynamic separation safety protocols (ignoring the T+A case) reveals significant differences at both the 22 aircraft level ($\chi^2(2) = 34.2927, p < 0.0001$) and 34 aircraft levels ($\chi^2(2) = 100.6313, p < 0.0001$). Post-hoc Steel-Dwass multiple comparisons tests reveal that at the 22 aircraft level, the Temporal Separation protocol (mean rank = 629.62) is different from both Area (497.488) and Dynamic Separation (511.682) protocols, with the latter two not significantly different from one another. At the 34 aircraft level, significant differences exist for all pairwise comparisons between the three protocols, with Temporal Separation (mean rank = 689.25) providing the largest values (different from both Area and Dynamic Separation a $p < 0.0001$). Area Separation (mean rank =

[2] A table of results for each individual SP-Composition pair can be found in Appendix K

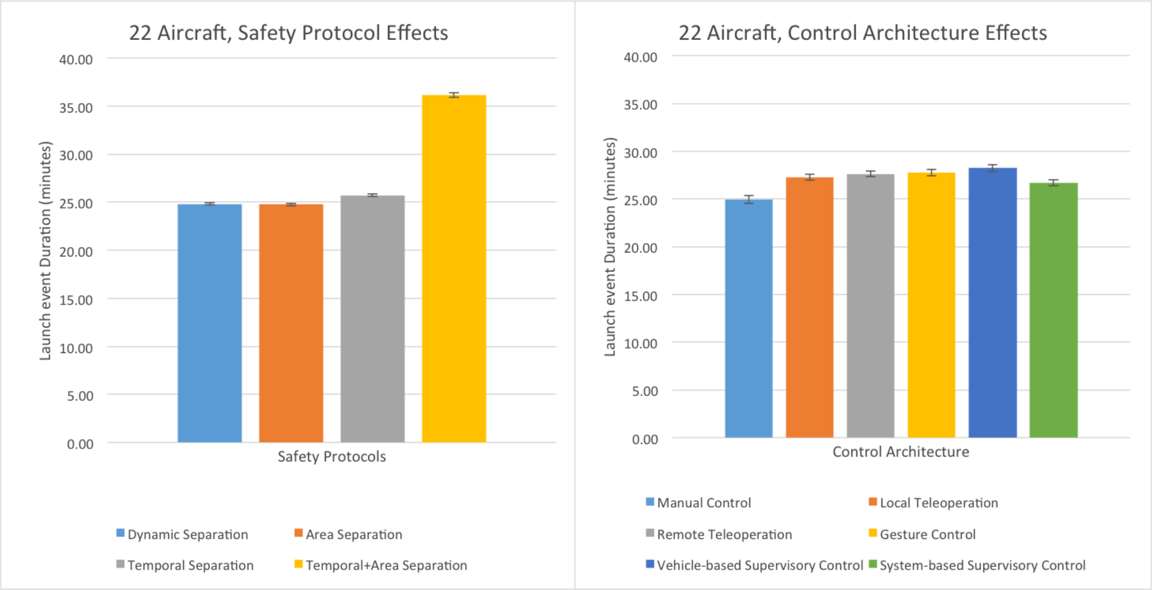


Figure 5-13: Column charts of effects of Safety Protocols (left) and Control Architectures (right) for 22 aircraft missions.

448.24) provides better performance than Dynamic Separation (mean rank = 505.195, $p = 0.0435$).

For control architectures, a Kruskal-Wallis test using CA as the grouping variable show that significant differences exist at the 22 aircraft level ($\chi^2(5) = 37.7272, p < 0.0001$) but not at the 34 aircraft level ($\chi^2(5) = 5.8157, p = 0.3246$). A post-hoc Steel-Dwass non-parametric simultaneous comparisons test for the 22 aircraft data reveals that significant pairwise differences exist between the SBSC (mean rank = 438.6) case and each of the RT (605.7), GC (582.2), and VBSC (562.0) cases. The LT (mean rank = 515.2) and RT cases are also shown to be significantly different from one another. These results, at least for 22 aircraft, suggest that the presence of latencies and delays in the control architectures — which occurs for the RT, GC, and VBSC systems — has some effect on mission productivity. However, while these results are statistically significant, the practical differences between control architectures are only on the order of a few minutes.

Figure 5-14 provides a column chart depicting the results for individual control architectures at the 22 aircraft Density level, grouped by Safety Protocol but excluding the T+A cases. The data in figure 5-14 demonstrates that not only is there relatively little variation across safety protocols, there is very little variation across control architectures within a single protocol. The most notable aspect of the data in the figure is that the SBSC architecture appears to provide better performance than the other architectures under the Dynamic Separation only and Area Separation safety protocols. Additionally, RT, GC, and VBSC appear to cluster together as worst performers under the Temporal Separation protocol. Statistical tests that compared control architecture performance separately within each Safety Protocol were performed to further examine these differences.

For the Dynamic Separation protocol, an ANOVA shows significant differences amongst con-

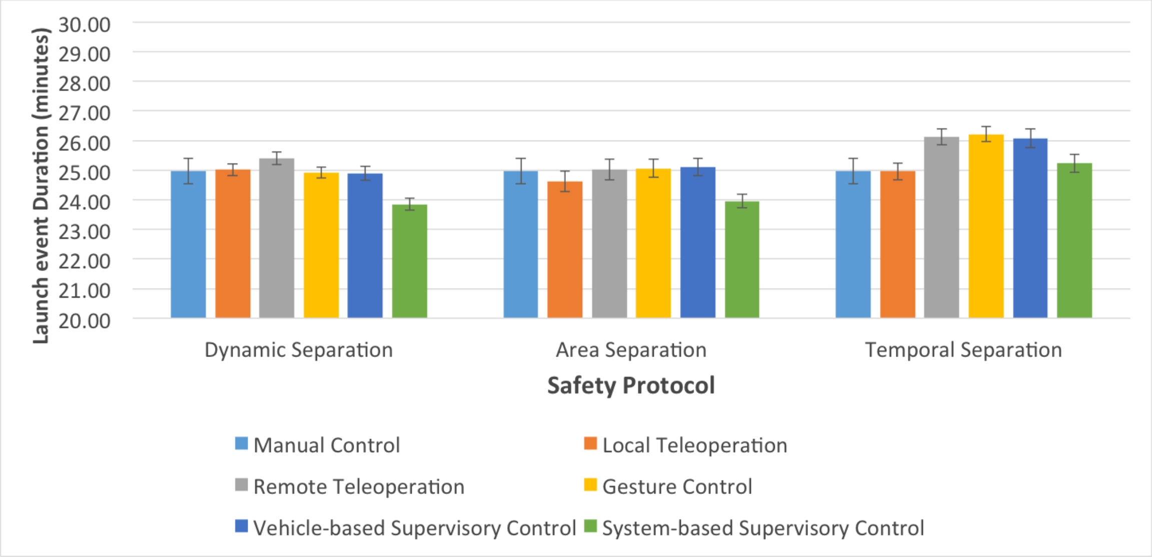


Figure 5-14: Column chart comparing Launch event Duration (LD) values across all Control Architectures (CAs) sorted by Safety Protocol (SP). Whiskers indicate ±1 standard error.

trol architectures ($F(5, 474) = 6.016, p < 0.0001$). A Tukey HSD test across all pairwise comparisons shows that significant differences exist only between the SBSC architecture and the LT, RT, GC, and VBSC architectures (all at $p \leq 0.0061$). Under Dynamic Separation, SBSC and MC are not significantly different from one another. For the Area Separation protocol, an ANOVA shows no significant differences between architectures ($F(5, 324) = 2.021, p < 0.0753$). For the Temporal Separation protocol, an ANOVA shows significant differences amongst control architectures ($F(5, 324) = 3.947, p = 0.0017$), with a Tukey test showing that only two significant differences exist: Local Teleoperation (LT) provides smaller LD values than both the RT ($p = 0.0477$) and GC ($p = 0.0250$) architectures. These combined results suggest that variations in control architecture performance are limited, but that some important interactions exists between safety protocols and control architectures.

An extended analysis of the LD results can be found in Appendix K, which performed a statistical comparison across all individual run cases, no longer grouping data according to Safety Protocol or Control Architecture. What these results demonstrated was that the significant main effects of CA and SP described previously are due to differences between only a few specific missions: the Temporal Separation applied to the RT, GC, and VBSC results in larger LD values, while the Dynamic and Area Separation protocols applied to the SBSC architecture results in smaller (better) LD values overall. The benefits of the SBSC architecture are likely due to change in the deck topology that occur with the removal of the Aircraft Director crew and their "zone coverage" routing system. Removing this crew removes constraints on when and where SBSC vehicles are allowed to taxi, as they are not forced into the routing network in Fig. 3-7 in Section 3.2.2. This leaves the SBSC architecture as the least-constrained architectures in terms of motion on the flight deck, which appears to provide slight benefits in terms of launch event duration values.

However, as noted, the results show relatively little practical variation in results. This could be due to an error may exist within one or more components of the MASCS model that serves to limit possible variations in system performance. One option is that the individual vehicle models are flawed. However, the results of single aircraft testing in Section 5.2 did show significant differences in performance between the architectures for individual aircraft. These differences are somehow being obscured at the mission level. Errors may exist in how the crew "zone coverage" routing architecture handles the simultaneous taxiing of aircraft, but the SMEs that reviewed the execution of the MASCS simulation judged that the models of Aircraft Directors and their routing of vehicles were accurate.

An alternative explanation involves the heuristics that govern the catapult queueing process and the current launch preparation time model. As described in Chapter 3, Deck Handlers can assign two aircraft to each catapult. Within a catapult pair, one aircraft prepares for launch while the other three wait nearby (see Figure 5-15, part a). After the first aircraft launches (part b) the aircraft on the adjacent catapult begins launching, the next aircraft at the original catapult moves forward, and the Handler assigns a new aircraft to this queue. As additional launches occur, the Handler continues to assign new aircraft to the catapult, often taxiing multiple aircraft in parallel towards the same catapult pair (Figure 5-15, part c).

It is an explicit goal of the Handler to keep these catapult queues filled: if queues are full, the chances of any catapult sitting idle while waiting for a task is minimized. The crew "zone coverage" routing architecture facilitates this by allowing decentralized and parallel routing of aircraft on the flight deck, allowing the queues to be refilled quickly during operations. If the queues are kept filled through the mission, then the time required to complete the mission is simply a matter of how quickly the queues are emptied. This is an effect of the queuing policies at catapults and of the launch preparation task, both of which were defined independently of the control architectures and safety protocols of interest here. Differences in performance will only arise if an architecture fails to provide aircraft to keep queues sufficiently filled.

Disrupting a queue requires that the new aircraft arrive at the catapults after the queue has been emptied. That is, the three aircraft already at the catapults (Fig. 5-15, part b) must complete their launches while the new aircraft is taxiing. Within MASCS, the time to complete the launch preparation task averages roughly two minutes; this means that the queue is emptied in six minutes on average. If the average taxi time for the new aircraft is less than six minutes, then aircraft will, on average, arrive in time to keep queues filled. Figure 5-16 provides a selection of output data from MASCS execution regarding the average taxi times and queue wait times for aircraft. Taxi times range from 3 to 4 minutes, clearly less than the six required to clear the queues, which would suggest that the queues remain in place throughout the mission. This is verified by the queue wait time data on the right side of the chart, which shows that the average wait time in queue for aircraft ranges

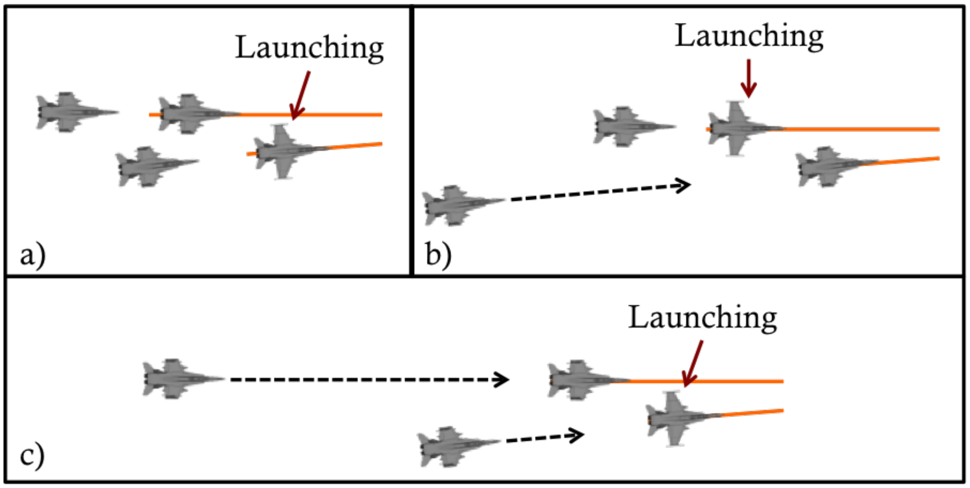


Figure 5-15: Diagrams of queuing process at catapults. a) start of operations with full queues. b) After first launch. c) after second launch.

from 2 to 3.5 minutes. This occurs across all safety protocols and across all control architectures. The differences exhibited by aircraft in the single aircraft tests in Section 5.2 all occur upstream of the launch preparation process and are largely mitigated by the queueing process.

The queueing pattern that arises at the catapult pairs is the result of human-generated heuristics that govern taxi routing and aircraft assignments, the launch preparation task (a task dominated by manual human tasks, as described in Section 3.1), and the inability to launch adjacent catapults in parallel (a physical constraint influenced by the heuristics governing the design of the deck). Until this queueing process is altered, little can be learned about the true effectiveness of manned or unmanned vehicles in flight deck operations. Chapter 6 will explore how the queueing and launch preparation processes might be improved in order to eliminate the queueing phenomenon observed here and how it affects variations in flight deck productivity. The next section continues the review of results from the experimental matrix, describing the effects of the safety protocols and control architectures on the safety of flight deck operations.

5.5.2 Effects on Mission Safety

Several different safety metrics were defined in Section 5.3, each addressing different ways of viewing the interactions between vehicles and crew on the flight deck. Additionally, each of these measures could be viewed either as an aggregate or be separated into independent measures for aircraft and crew. The settings of the collision prevention routines for aircraft prevent the occurrence of PHI and Secondary Halo Incursions (SHI) incursions by other aircraft; vehicle interactions only appear at the Tertiary Halo Incursions (THI) level, which is reviewed in this first section. Figure 5-17 presents a Pareto chart of THI plotted against LD for the 22 aircraft cases, with the points forming the Pareto frontier connected via a black line (the bottom left corner). This chart includes data points for

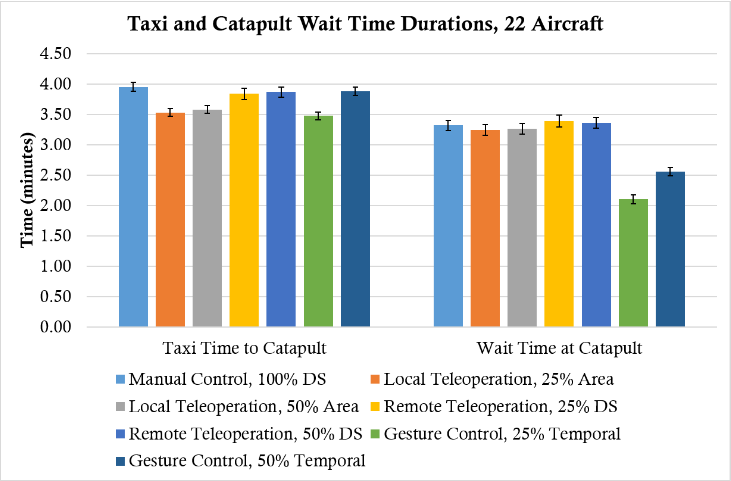


Figure 5-16: Average taxi times, average launch preparation time, and average time to clear a queue of three aircraft for selected 22 aircraft missions.

each replication of every run setting (Control Architecture+SafetyProtocol+Composition). Marker shapes are consistent for a specific CA (e.g., GC appears as boxes, VBSC as diamonds), while marker colors denote the Safety Protocol (SP) utilized (blue for Dynamic Separation, red for Temporal, etc.). Because results were largely consistent across different Composition values, no differentiation is made for this variable in order to improve the readability of the charts. Pareto charts using individual colors for each SP-Composition pairing can be found in Appendix N.

The general trend in the data in the figure is in terms of the different control architectures. VBSC provides the worst performance (largest values), followed by GC, then a cluster of results for MC, LT, and RT, then with SBSC providing the best results. This can be highlighted when looking at results for only the Dynamic Separation case in Figure 5-18. These results are in direct opposition to the hypotheses stated in Section 5.4, which expected that the SBSC case should produce the

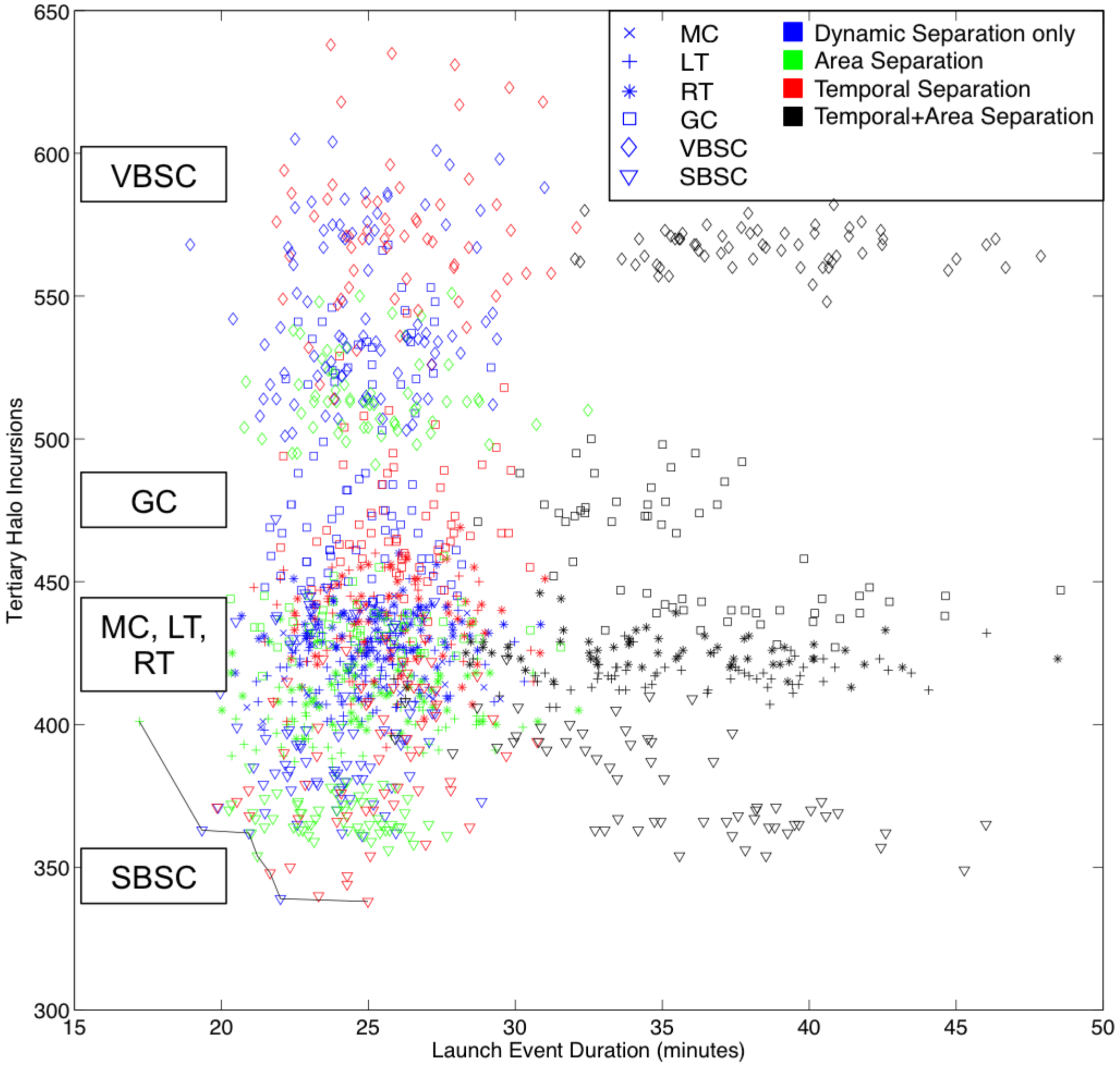


Figure 5-17: Pareto frontier plot of Tertiary Halo Incursions (THI) versus LD for 22 aircraft missions.

highest values in terms of Halo Incursion counts on the flight deck. Additionally, the wide disparity between the VBSC and SBSC cases was not expected. The only major difference between the two cases is the utilization of crew: under the SBSC protocol, only Escorts are utilized, while under the VBSC protocol, a mix of Escorts and Directors are utilized. This suggests that, for these two systems, the removal of the Aircraft Directors from the SBSC cases may have significant effects on the THI metric. Since THI is an aggregate metric including both crew and aircraft contributions, the contributions of one group may overshadow the other.

Figure 5-19 shows only the Tertiary Halo Incursions due to Aircraft (THIA) contribution to the THI metric, and the trends in the THIA data are significantly different than the trends in THI in Fig. 5-17. Most importantly, the only two main effects that appear to exist are that (1) the largest counts of THIA values occur under the Dynamic Separation protocol and (2) that the T+A safety protocols (bottom right corner, black markers) appear to generate the smallest values of THIA. Additionally, the high THIA value Dynamic Separation values only comes from a subset of the control architectures: GC (squares), VBSC (diamonds) and SBSC (triangles), each of which also shows a much range of THIA values than the MC, LT, and RT architectures. In fact, SBSC provides both some of the best (smallest) and some of the worst (largest) values in the results. Results for MC,

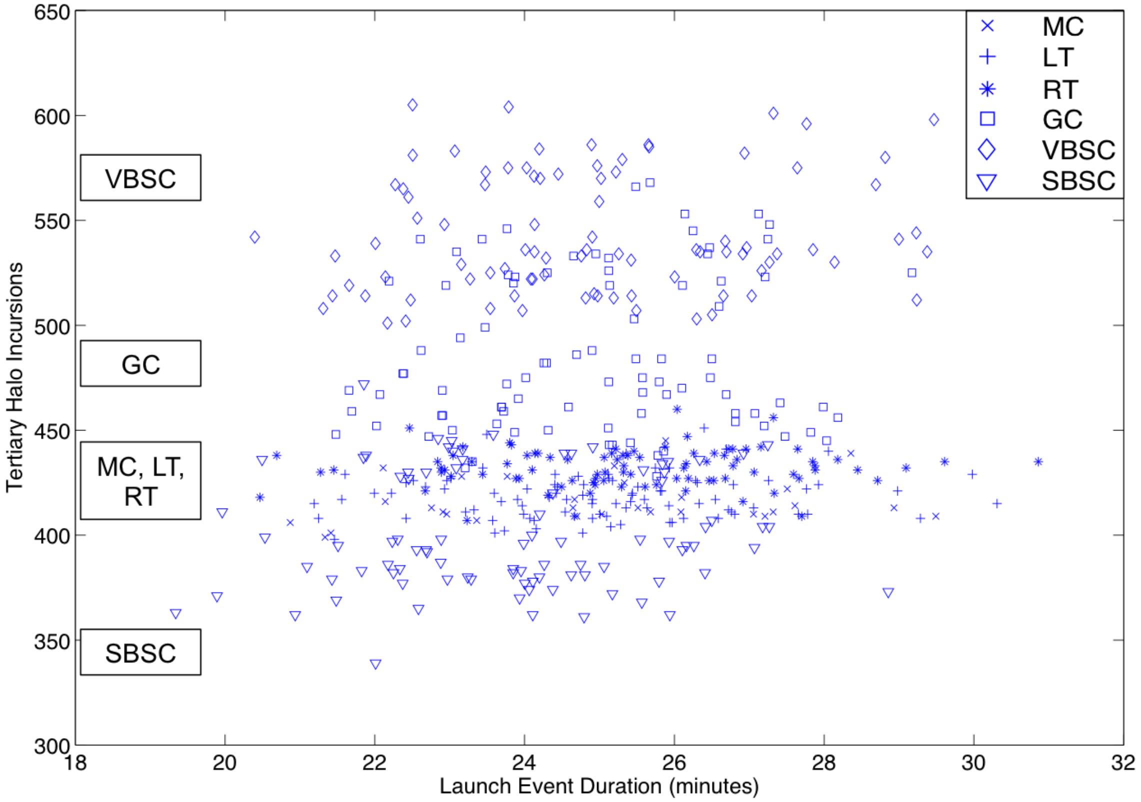


Figure 5-18: Pareto frontier plot of Tertiary Halo Incursions (THI) versus LD for 22 aircraft Dynamic Separation missions.

LT, and RT all appear to cluster in the lower half of the chart; Figure 5-20 provides a side-by-side comparison of the LT and SBSC architectures to highlight these differences more clearly.

A Kruskal-Wallis test of the data utilizing SP as a factor returns significant results ($\chi^2(3) = 422.86, p < 0.0001$) with the Dynamic Separation cases (mean rank = 922) returning the highest overall values (worst performance), T+A returning the lowest (321), and Area (718) and Temporal (660) in between. A Steel-Dwass multiple comparisons tests shows that the T+A and Dynamic Separation cases are significantly different from all other safety protocols (all at $p < 0.0001$), while the Temporal and Area cases are not significantly different from one another ($p = 0.1386$). These results partially support the hypotheses stated in Section 5.4: as expected, the T+A protocol provides the smallest overall values and DS the largest. However, these results fail to confirm the hypotheses

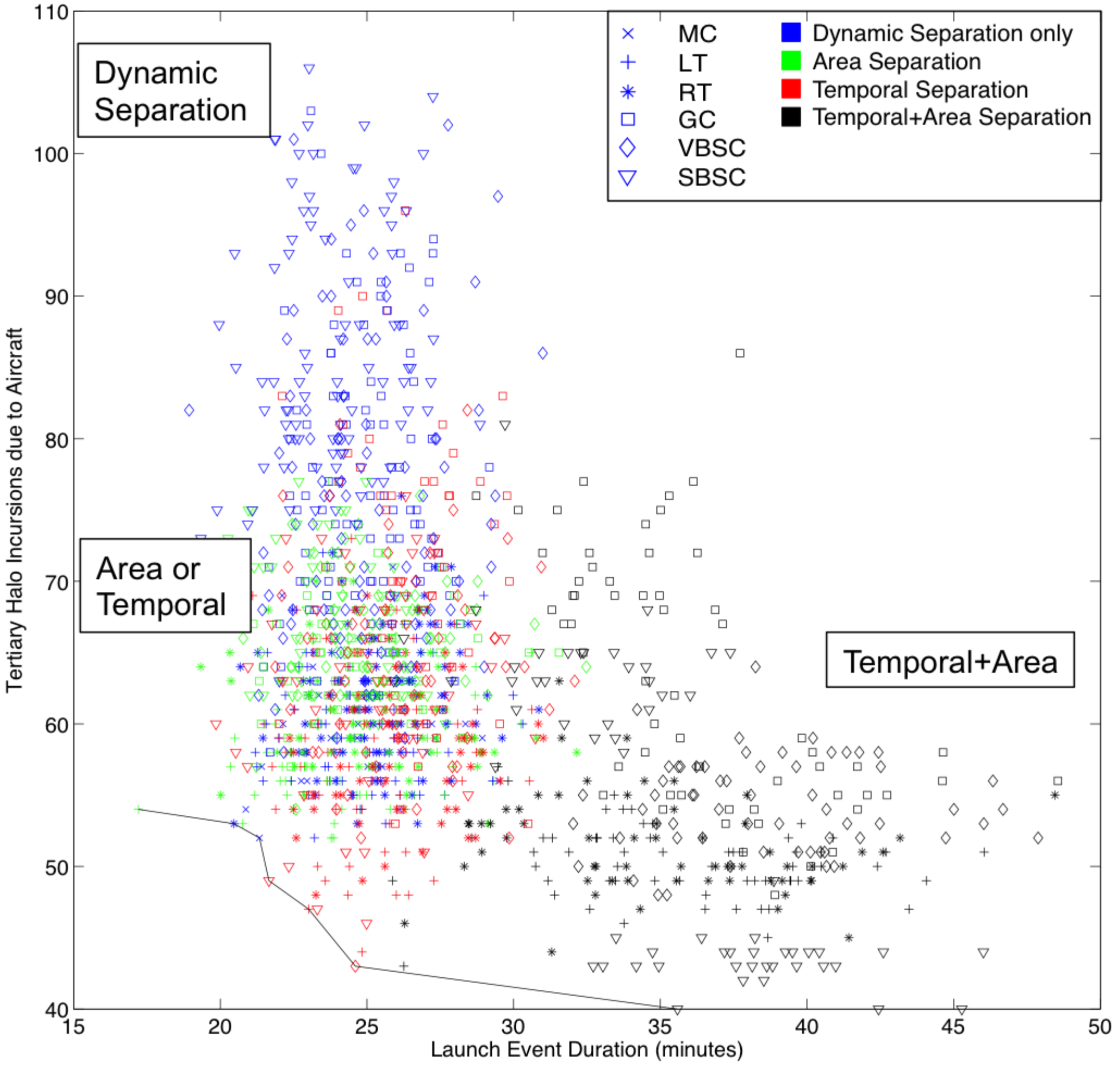


Figure 5-19: Pareto frontier plot of Tertiary Halo Incursions due to Aircraft (THIA) versus LD for 22 aircraft missions.

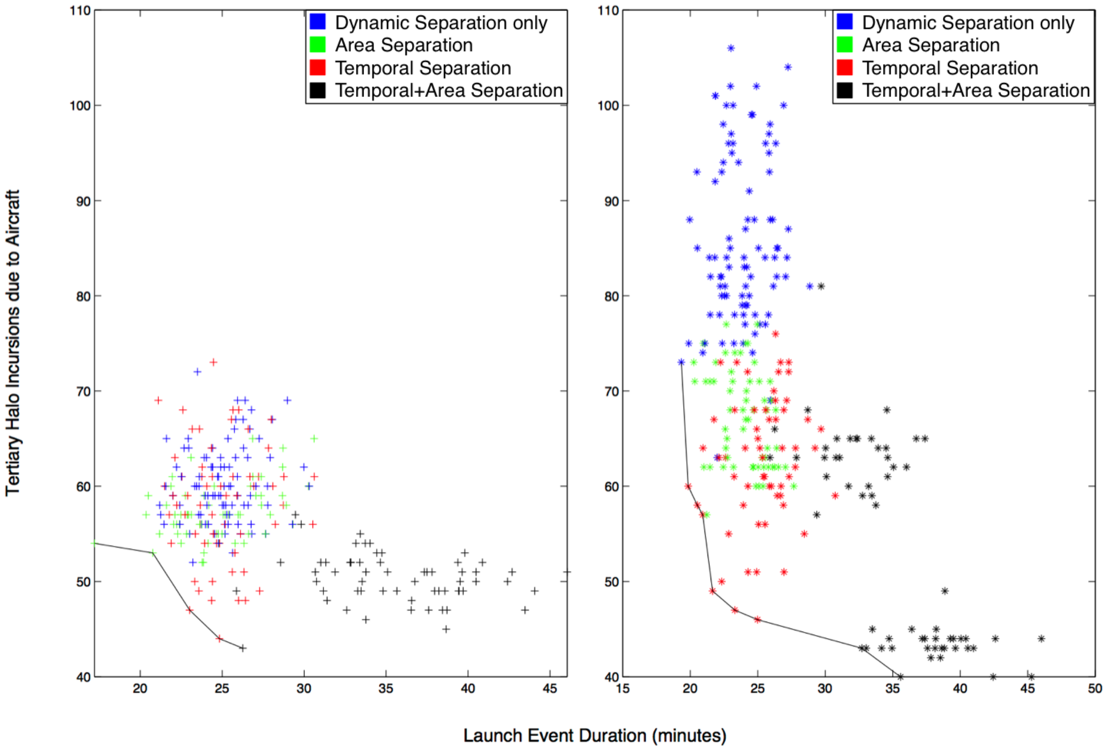


Figure 5-20: Pareto frontier plots of Tertiary Halo Incursions due to Aircraft (THIA) versus LD for 22 aircraft missions for Local Teleoperation (LT, left) and System-Based Supervisory Control (SBSC, right).

that Temporal would be equivalent to T+A and Area would be equivalent to Dynamic Separation.

In terms of mean THIA values, T+A (mean value = 54.54) provides a 24% improvement in average THIA scores compared to the Dynamic Separation protocol (mean value = 71.33) and a 13% improvement as compared to Temporal (mean = 62.54) or Area Separation (mean = 63.26). However, as pointed out in Section 5.5.1, this comes at a cost to LD: the average LD for the T+A cases is 36.17 minutes, a 44.3% increase over the average 25.06 minute mission duration for the combined Area, Dynamic Separation, and Temporal protocol results. Applying multiple high-level constraints to operations with the combined Temporal+Area protocol does appear to provide a benefit to vehicle safety (smaller THIA values), but this comes at a much larger relative cost to mission productivity (increased LD values).

Significant differences also exist across the different control architectures (Kruskal-Wallis test, $\chi^2(5) = 362.0433, p < 0.0001$), with Steel-Dwass multiple comparisons tests showing differences between two sets of vehicles: the three "autonomous" UAV architectures of GC (mean rank = 1050.8), VBSC (972.8), and SBSC (811.74) all perform significantly worse than the "manned" cases MC (mean rank = 260.27), LT (234.0), and RT (431.0), with GC also being significantly worse than VBSC (VBSC v. MC, $p = 0.0026$; all others, $p < 0.0001$; full results appear in Appendix N). These results are likely influenced by the poor performance of GC, VBSC, and SBSC under the Dynamic Separation protocol. These results also fail to confirm the hypotheses from Section 5.4: while the SBSC architecture does provide the largest values and worst performance (as expected), significant differences were not expected between the other architectures.

The original hypothesis was that the iterative interactions of the Deck Handler in VBSC and the use of the “zone coverage” routing structure in MC, LT, RT, and GC and would each generate sufficient spacing between vehicles to keep THIA values low. The statistical tests of the previous paragraph suggest that this does occur for the MC, LT, and RT architectures, as tests demonstrated that these three architectures were significantly different from, and provided smaller THIA values than, the remaining architectures. For the MC, LT, and RT architectures, the combined Temporal+Area case also further reduced THIA values, but only slightly. This suggests that the “zone coverage” mechanism used by these architectures may function as a different form of safety protocol during operations that is incompatible with the Area and Temporal separation protocols.

Interestingly, the GC architecture provides significantly worse performance than LT and RT although it also utilizes the crew’s zone coverage routing method. This poor performance is likely due to the failures and processing latencies of the GC system; as shown previously in Section 5.2, these failures and delays made GC the slowest of the UAV control architectures. This is partially due to the topology of the zone coverage routing network, which requires multiple small taxi tasks be performed as vehicles are passed between Aircraft Directors on the deck. When a GC vehicle is handed off from one Director to another, the vehicle has a chance first to fail to recognize the new Director, then multiple chances to fail to recognize his commands, as well as delays from processing latencies. While this is occurring, the first Director is providing instructions to a new vehicle. If this vehicle more readily finds the Director and accepts his commands, it will begin moving forward but will be blocked by the first vehicle. The second aircraft will taxi forward until its collision prevention routine commands that it stop, incurring a new THIA penalty. If this process continues repeatedly for all vehicles on the flight deck, this would generate a higher rate of THIA values than LT and RT.

The poor performance of the VBSC and SBSC architectures is also likely explained by the change in the topology of the deck due to their lack of use of the “zone coverage” crew routing mechanism. This change in topology means that these two architectures do not experience the same “natural spacing” as the other architectures. Instead, they are given greater freedom in taxiing on the flight deck, coming as close to other vehicles as the Dynamic Separation collision prevention routine will allow. Interestingly, the delays of the VBSC architecture do not exacerbate the effects of the more open routing system and may be due to the influence of the Deck Handler’s queuing policy. The VBSC Handler prioritizes aircraft closest to their catapults specifically to avoid the conditions described for GC: aircraft at the “front” of operations are given instructions first in order to prevent them from obstructing traffic. The flow of traffic is kept moving, and the same effects observed for GC should not occur for VBSC.

In general, these results for the THIA aircraft safety measure suggest that the topology of the flight deck, as influenced by the use of the crew's "zone coverage" routing methods, is the most important driver of performance in this environment. Architectures using that routing method and not experiencing significant numbers of failures in operations (MC, LT, RT) provided the smallest values for the metric. The two architectures not using the crew routing (VBSC and SBSC), provided some of the worst values. While GC also used the crew routing, its high failure rates interacted with the crew routing to generate high THIA values. However, for these latter three architectures exhibiting very poor performance, results suggest that the use of the safety protocols is beneficial in lowering the rate of aircraft interactions on the flight deck. However, overconstraining operations in the T+A protocol provides only a slight improvement over the Temporal or Area Seapration cases at a significant penalty to LD values.

Similar trends appear in data taken at the 34 aircraft Density level (Figure 5-21). Kruskal-Wallis tests for SP main effects ($\chi^2(3) = 78.847, p < 0.0001$) and CA main effects ($\chi^2(5) = 885.2522, p < 0.0001$) again reveal significant results (full results appear in Appendix N). For Safety Protocols, a Steel-Dwass multiple comparisons test reveals that T+A (mean rank = 547.07) is significantly different from Area (mean rank = 738.10, $p < 0.0001$), Temporal (mean rank = 630.15, $p = 0.0266$), and Dynamic Separation (mean rank = 788.11, $p < 0.0001$). Temporal is also significantly different than Dynamic Separation ($p < 0.0001$) and Area ($p = 0.0028$), while Area and Dynamic Separation are not significantly different from one another ($p = 0.209$).

A Steel-Dwass test on the Control Architecture results shows significant pairwise differences exists between *all* cases except between MC and LT ($p = 0.5905$; MC and RT, $p = 0.0003$;VBSC and GC, $p = 0.0034$; all others, $p < 0.0001$). Recall that at the 22 aircraft setting, differences occurred primarily between the two groups of "manned" (MC, LT, RT) and "unmanned" (GC, VBSC, SBSC) architectures, where comparisons within groups were not significant. These data suggest that increasing the number of aircraft on the flight deck, which introduces new constraints in how aircraft can be routed on the flight deck and changes how the Deck Handler allocates aircraft through the mission, encourages further variations in the GC, VBSC, and SBSC architectures.

A review of results for the Duration of Tertiary Halo Incursions due to Aircraft (DTHIA) metric was also conducted, which appears in Appendix M. However, these results demonstrated the same general trend of results at the THIA metric. The trend observed in these metrics indicates that the safety of aircraft interactions is influenced largely by the selection of the topology of the flight deck and the role of crew in that topology. These results only partly support the hypotheses discussed in Section 5.4. While, overall, the trends in performance for control architectures and safety protocols were as expected, a variety of unexpected interactions existed within the system, due largely to the differences in the topology of the flight deck between the "zone coverage" and supervisory control

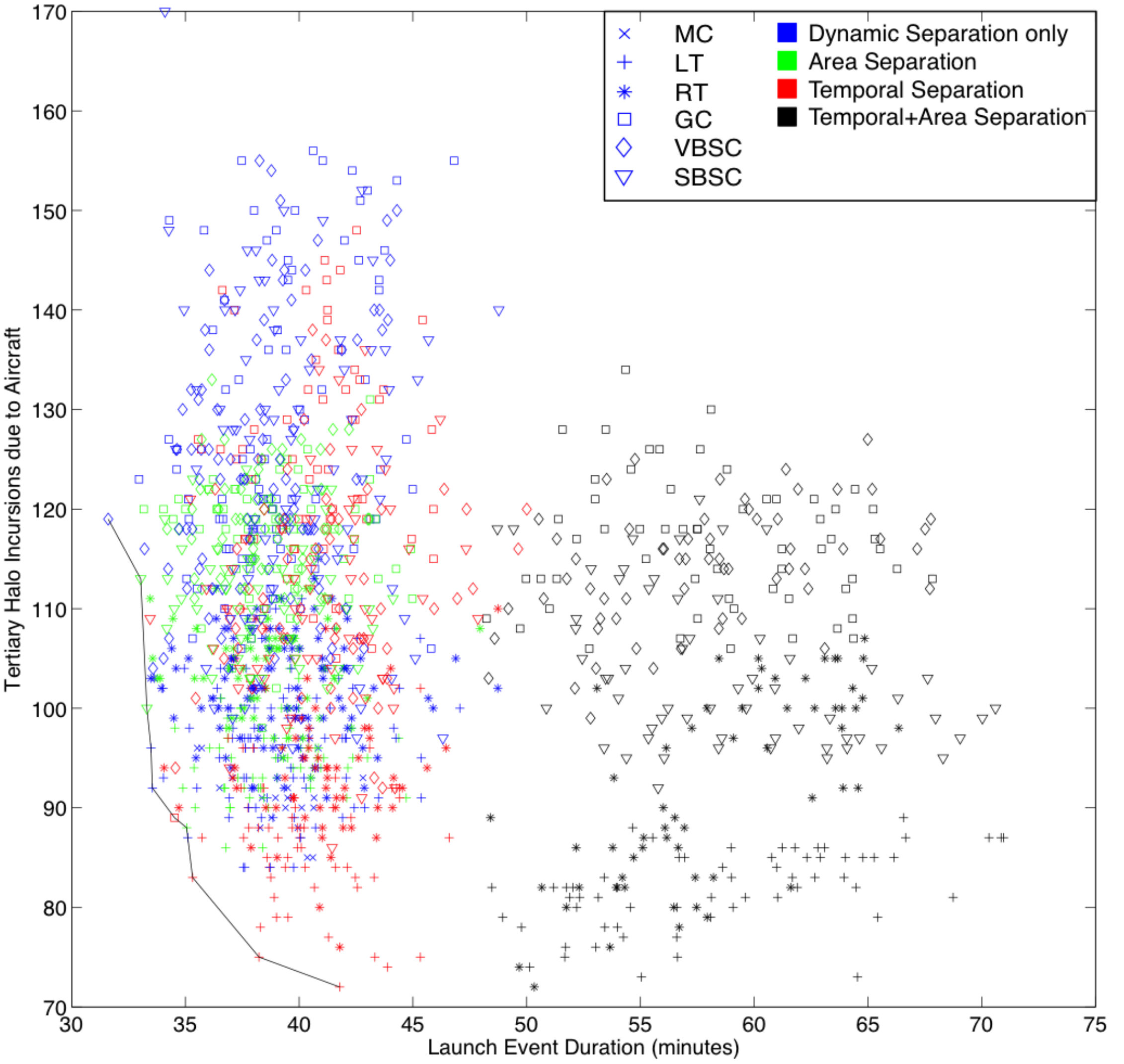


Figure 5-21: Pareto frontier plot of Tertiary Halo Incursions due to Aircraft (THIA) versus LD for 34 aircraft missions.

routing methods. These were significant enough that the SBSC control architecture exhibited both some of the best and some of the worst performance.

While these interactions highlight important considerations about the effects of crew interactions on aircraft safety, aspects of crew safety are a second important consideration in operations. Recall that the Pareto chart of the aggregated THI metrics showed trends quite different from the THIA metric. As noted previously, these differences are likely due to the crew component. However, the models of crew behavior implemented within MASCS were quite simple and do not reflect the true complexity of crew behavior when working in close proximity to aircraft. The PHI and Duration of Primary Halo Incursions (DPHI) metrics do provide some insight as to how vehicle routing affects the rate at which vehicles encounter crew on the deck, but they cannot truly offer any information on crew-vehicle interactions. To better characterize those interactions, future work should address the creation of more complex models of crew and vehicle motions at a much finer resolution than currently done in MASCS. An analysis of the Primary Halo Incursion (PHI) and Duration of Primary Halo Incursions (DPHI) measures for this current work can be found in Appendix M, which indicates that once again, the primary effect on operations was the choice of crew routing topology.

Several other safety metrics were defined in Section 5.3 that have not been discussed here. While these metrics were output from MASCS and were analyzed, the trends in performance are the same as those presented within this section. In particular, the predictive Collision measures were virtually identical to the Halo Incursion measures, which is likely an artifact of the slow speeds of transit on the flight deck (it takes several seconds for an aircraft to travel its body length). Pareto charts of these other metrics are provided in Appendix N. The next section reviews the final category of dependent variables included in MASCS: measures of efficiency.

5.5.3 Effects on Mission Efficiency

The design of the safety protocols and their interactions with the Composition settings were expected to drive significant variations in the efficiency measures, as well as for productivity and safety reviewed in earlier sections. While these measures may not be correlated directly to Launch event Duration, their performance is still of interest: longer taxi distances imply more wear on aircraft tires and on the deck itself, as well as increased fuel consumption. While these measures of efficiency may not be the primary concerns of stakeholders at this time, they may be sources of secondary benefits to operations.

The first of these measures, Total Taxi Distance (TTD), sums the total travel distance of all aircraft in the simulation, providing a measure of the efficiency with which aircraft were allocated to catapults by the Deck Handler planning heuristics, as well as the efficiency of routing in the zone coverage routing scheme. Figure 5-22 shows data for this metric at the 22 aircraft Density level. Similar to previous measures of safety, it can be seen that the 50% Area cases provide very low values of TTD, whilst the Temporal and Temporal+Area provides provide the worst overall. The performance of the Temporal cases is perhaps not surprising, given the previous discussions about the rate of transition from fore to aft and vice versa. The increased values at the T+A setting, however, come primarily from the unmanned vehicles that operate at the end of the mission schedule: some subset of these vehicles are assigned to the forward catapults, incurring much larger taxi distances than would be seen in the Area and Dynamic Separation cases.

A Kruskal-Wallis test using SP settings as the grouping variable reveals significant differences within the data ($\chi^2(3) = 972.0598, p < 0.0001$), and a Steel-Dwass test shows that the Temporal and T+A safety protocols generate increased taxi distances on the flight deck (full results appear in Appendix N). A slight trend does appear to exist for CAs within the larger pattern of SP, which demonstrates that the architectures utilizing the X-47 aircraft model have shorter total taxi distances on the flight deck. This is likely an artifact of the differences in shape of the vehicles: the X-47 aircraft used for GC, SBSC, and VBSC is shorter overall, and the gains in performance are likely the result of minor differences in starting locations and ending locations (at catapults), and also possibly for routing actions, because of this change in size. The plot of results for the 34 aircraft case appears in Appendix N but shows generally the performance as the results shown here for 22 aircraft.

Two additional productivity measures were also tracked: Total Aircraft taXi Time (TAXT) values (Figure 5-23) summed the total time in which vehicles are actively moving or otherwise available to move, whilst Total Aircraft Active Time (TAAT) measures time spent in motion as well as time spent at catapult performance the launch tasks. An error in coding lead to slightly

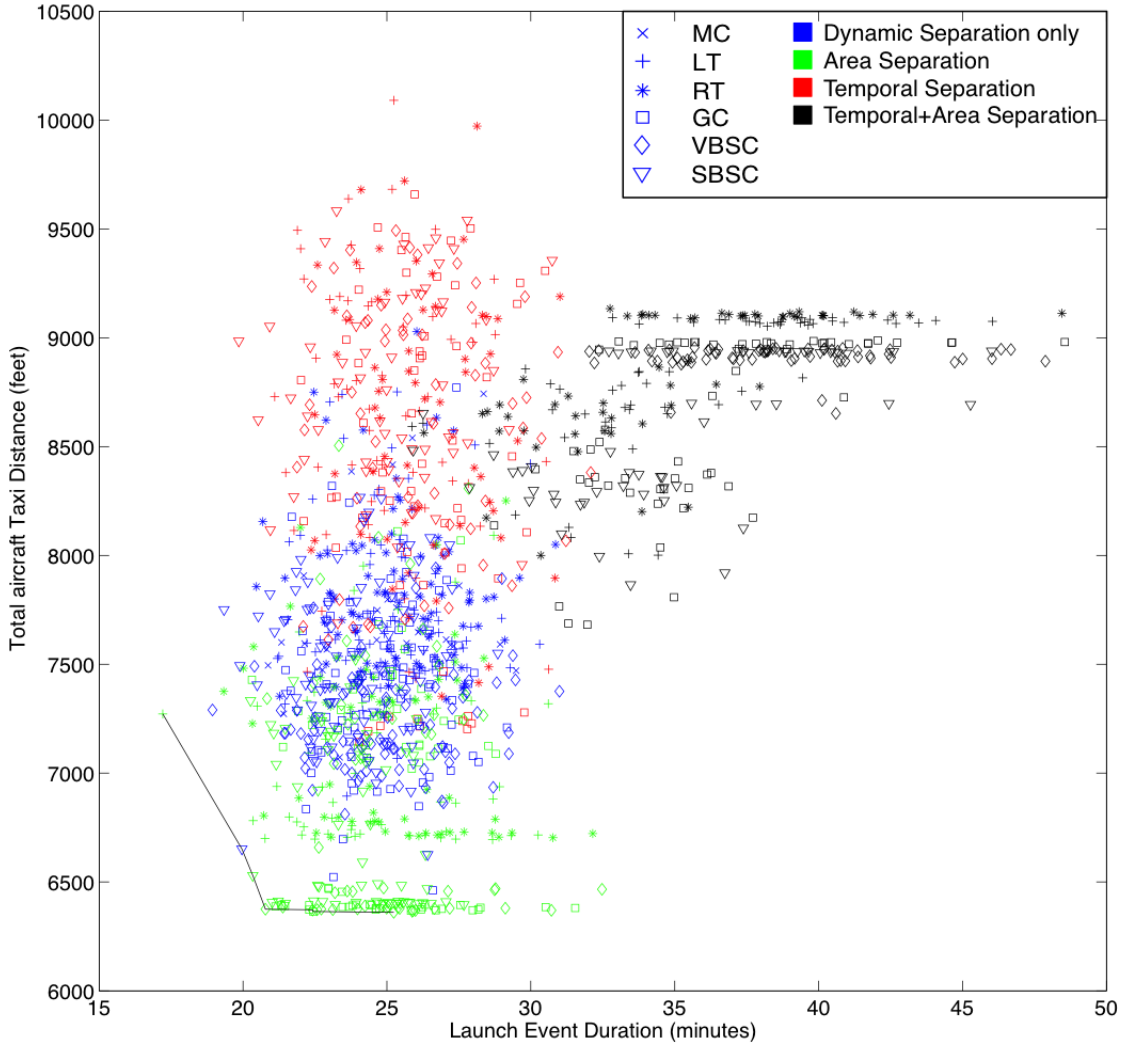


Figure 5-22: Pareto frontier plot of Total Taxi Distance (TTD) versus LD for 22 aircraft missions.

different calculations of these metrics for the HSC vehicles (SBSC and VBSC) and the remaining cases, based on the use of Aircraft Directors in zone coverage. For cases using zone coverage routing, TAXT included all time during which vehicles had an assigned Director and were not stopped due to the collision prevention or traffic management routines; if the vehicle did not have a Director assigned, or was otherwise stopped from moving, the time was not logged. For the remaining SBSC and VBSC vehicles, because Directors were not utilized, time spent waiting on taxi paths to open for motion was included in the time logging. This results in the significant variances amongst the CAs in Figure 5-23, which shows the results of TAXT for the 22 aircraft case. The data also appear highly correlated, which is expected given the nature of flight deck operations. Otherwise, aside from the significant disruptive effects of the T+A case, no major trends are apparent. Values of TAAT are almost identical to these measures and are not shown here (graphs appear in Appendix N).

5.6 Discussion

The Multi-Agent Safety and Control Simulation (MASCS) experimental matrix (Fig. 5-1) sought to examine the interactions and effects of four different variables on the safety and productivity of flight deck operations: Safety Protocols (SPs), Unmanned Aerial Vehicle (UAV) Control Architectures (CAs), mission Density (number of aircraft), and mission Composition (number of unmanned vehicles used in the mission). The results of the testing described in this chapter have shown that while variations in safety do occur in terms of the defined safety metrics, variations in mission productivity are minimal, especially for the key metric of Launch event Duration (LD). As a result, tradeoffs between safety and productivity only truly occur when operations are heavily constrained in the combined Temporal+Area Separation protocol. As compared to the baseline Manual Control (MC) case at 22 aircraft, the increased constraints of the T+A case increases the safety of aircraft interactions by only 8% (MC average = 59.53, average of all T+A cases = 54.54) while decreasing mission productivity by 44.3% (MC average = 24.97 minutes, T+A average = 36.17).

The causes of both the variations in safety measures and the lack of variation in productivity measures can be attributed largely to the human-generated heuristics of flight deck operations that dictate how aircraft are routed through the flight deck and queued at catapults. For trends in safety measures, a key variable in aircraft interactions under the Tertiary Halo Incursions due to Aircraft (THIA) metric was whether or not crew were being used in the current "zone coverage" routing topology. For control architectures operating under this topology and that did not experience high levels of failures or latencies in operations, the spacing afforded by the prior crew routing topology improved safety. The choice of Safety Protocol had little effect in these cases, as the routing topology already maintained sufficient spacing between vehicles. For control architectures not using this prior crew routing strategy, or that did while experiencing significant delays and latencies (the Gesture Control, GC, architecture), the safety of aircraft interactions was influenced

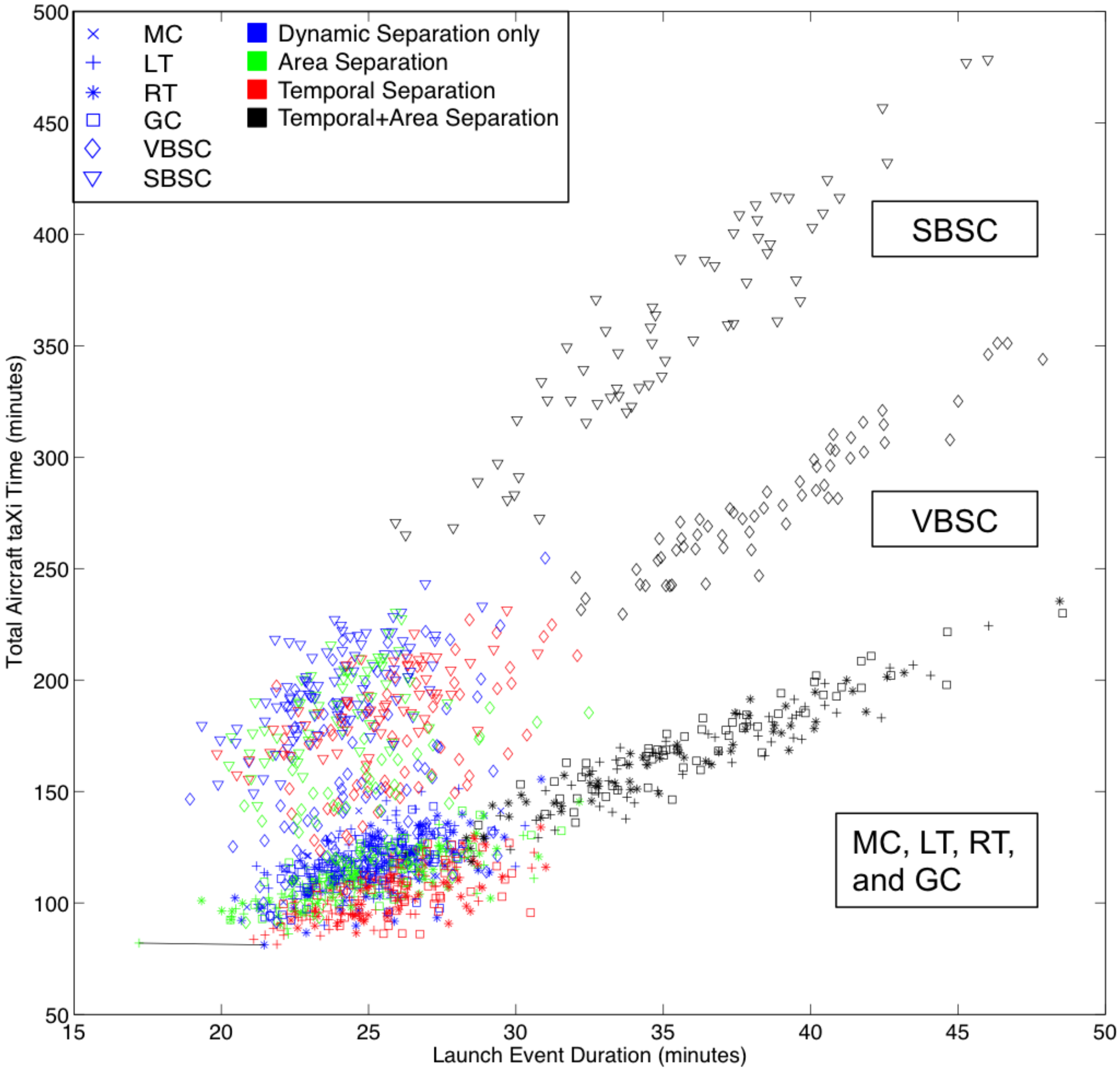


Figure 5-23: Pareto frontier plot of Total Aircraft taXi Time (TAXT) versus LD for 22 aircraft missions.

by the number of high-level constraints applied to operations. The Dynamic Separation only case, with no high-level constraints, provided the worst performance for these architectures. Applying one high-level constraint in either Area or Temporal Separation provided improved performance, with the combined Temporal+Area case providing the best.

Variations in productivity were also contingent on the role of crew, albeit in a different manner: their role in supporting the launch preparation tasks on the flight deck, in combination with the catapult queueing and Deck Handler mission planning. Here, the long average time to complete the launch preparation task enables queues of aircraft to be maintained at catapults. These catapult queues become the bottleneck in operations, as taxi operations appear to be efficient enough to supply aircraft to catapults before queues can be emptied regardless of the safety protocols or control architectures in use. However, further improvements to the queuing process and the launch preparation task should allow variations in productivity to arise between control architectures.

When considering the goals of this dissertation — to understand the effects of integrating unmanned vehicle control architectures on the safety and productivity of flight deck operations — the results of this chapter indicate that very little can be said about unmanned vehicle systems on their own. Rather, the results indicate that the primary influences in safety and productivity in flight deck operations comes from the interactions of the architectures with the crew on both a low level (interactions with crew as governed by heuristics for vehicle routing, the deck topology, the launch preparation process, and the catapult queueing process) and at a high level (the locations of crew on the deck and their role in routing vehicles). The selection of different control architectures or safety protocols only have effects in the context of how it changes vehicle interactions with crew. Accordingly, changes to the models of crew behavior may generate very different trends in results. Any future work addressing the results discussed in this chapter could place greater emphasis on investigating the role of the crew in all aspects of operations. This might include examining formal optimizations of crew behavior on the flight deck and simply altering the heuristics that guide their behavior during operations. Chapter 6 will address some of these issues, but will focus primarily on the launch preparation process and its effects on flight deck performance.

The data utilized in arriving at these conclusions was substantial — 92 runs, at thirty replications each, and a minimum of 20 minutes per run. Collecting this data in real-world trials would have required considerable resources and time in the field, disregarding the fact that many of the control architectures tested here do not yet exist. This demonstrates the utility of the use of simulations, and in particular, agent-based simulations for testing these systems. Modeling crew, aircraft, planning, and task execution as independent elements of the system has afforded an ability to examine and attribute results to the individual elements. These results have raised several additional questions, however, including what role the launch preparation time plays in regulating deck performance and whether any variations in control architecture productivity can be generated. The power of simulations such as MASCS is that these changes can be made easily within the system and enable an exploration of a variety of conditions quickly and efficiently. Chapter 6 explores how MASCS can be used in further exploring the key drivers of performance for UAV control architectures and for flight deck operations as a whole.

5.7 Chapter Summary

This chapter has described the process by which models of UAV performance on the aircraft carrier flight deck were generated for the MASCS simulation environment, the verification of these models, and the test and analysis of these models in terms of both the relative safety and productivity of operations. The results of these tests demonstrated a wide range of effects, as well as showing how varying definitions of safety metrics can provide wide-ranging results in terms of the performance of different systems. Overall, these results failed to confirm many of the hypotheses described in Section 5.4, most notably showing that the human heuristics that govern flight deck operations have significant effects. While the choice of unmanned vehicle control architecture does result in variations in safety metrics, these are primarily because of differences in when and how those heuristics are applied. Variations in productivity (in terms of Launch event Duration) were largely not observed, due to the bottleneck in catapult queueing that was created by the heuristics of the routing topology, catapult queueing process, and the role of the crew in launch preparation. While these effects means that very little can be said about how the choice of unmanned vehicle control architecture affects operations, these results highlight key concerns about the role of crew in flight deck operations that should be addressed in future work. Some of these concerns will be addressed in the next chapter, examining how the MASCS simulation can be used to explore improvements to the launch preparation time model that would remove the bottleneck in operations and how this affects productivity and safety on the flight deck.

Chapter 6

Exploring Design Options

> "Artificial intelligence is often described as the art and science of 'getting machines to do things that would be considered intelligent if done by people.' "
>
> *Alone Together: Why We Expect More from Technology and Less from Each Other (2012)*
>
> Sherry Turkle

Chapter 1 described the motivation for the development of the Multi-Agent Safety and Control Simulation (MASCS) environment: to develop a simulation methodology for simulating the integration of futuristic unmanned vehicle systems into mixed human-vehicle environments. This methodology was described in Chapter 3, calibrated and partially validated in Chapter 4, and used to test a set of candidate unmanned vehicle control architectures and operational safety protocols in Chapter 5. An added benefit of simulations like the MASCS environment is that it enables further investigations into the specific vehicle parameters the driver variations in performance in flight deck operations.

The results of Chapter 5 indicated that the interactions between agents in the system is affected not only by the types and number of vehicles active in the environment but by the constraints placed on their behavior due to safety protocols and to the structure and organization of operations. Neither of these factors are related to the choice of unmanned vehicle control architecture, which demonstrated significant and varied interactions with these factors. Understanding these interactions is especially important for systems that employ large numbers of unmanned vehicles, manned vehicles, and human collaborators, as testing such environments in the real world is both costly and may endanger the individuals involved in the test program. However, the utility of systems like MASCS is that they can quickly and easily be altered to incorporate new vehicle models, crew behaviors, environments, or other features of operations.

This chapter examines how a model like MASCS can be used to explore issues that affect the performance of the flight deck system as a whole. Potential points of investigation include process-level changes to operations, parameter-level changes to Unmanned Aerial Vehicle (UAV) models, and architectural changes to the environment itself. For example, MASCS can be used to examine how reductions in the number of crew used in "zone coverage" routing operations affect performance on the flight deck (an architectural change). MASCS can also be used to examine alternate versions of the unmanned vehicle control architectures (behavioral changes) defined in Chapter 5, altering the individual failure rates of vehicles, changing the parameters the define the latencies for vehicles in the system, or examining how the speed of aircraft travel affect mission performance. Additionally, reconfigurations of the flight deck itself, in terms of both the placement of aircraft (an operational change) as well as the location of the catapults (an architectural change) might also be tested.

Many other potential changes could be tested, but this chapter will address only a few. Chapter 5 suggested that the launch preparation task may have a significant role in regulating the pace of flight deck operations. The effects of modifying this task are examined first, reviewing how changes to this task might improve the productivity of flight deck operations for current Manual Control operations before continuing on to examine its effects on unmanned vehicle operations. This then extends to include launch preparation effects on the safety of operations, with the last section exploring individual parameter settings for the Gestural Control (GC) architecture to explore how its performance could be improved.

6.1 Exploration of Operational Changes

The results of Chapter 5 previously suggested that the combination of Deck Handler allocation strategies and large launch preparation times lead to the creation and continued existence of queues at catapults for all control architectures. With these queues in existence, the performance of the flight deck is dependent only on the launch preparation time, eliminating much of the possible variance in launch event duration. Improving the launch preparation process, whose characteristics are independent of any specific vehicle control architecture, should both reduce launch event duration values and eventually allow variations in the launch event duration values of the unmanned vehicle architectures to arise. This section explores modifications to this task for the baseline Manual Control (MC) architecture first, as it was partially validated against empirical data in Chapter 4 as is considered representative of current operations.

As described in Chapter 3, Section 3.2, the launch preparation task requires several tasks be performed manually by crew members on the flight deck. Appendix C described its definition as a normal distribution with a mean $\mu = 109.65$ seconds and standard deviation $\sigma = 57.80$ seconds

($N(109.65, 57.80^2)$). Improvements to this task can be accomplished both by reducing the mean and by reducing the standard deviation (square root of the variance) of the task. Reducing the mean value of the launch preparation time should have a strong effect in reducing the average launch event duration, as it should directly reduce the average time to launch an aircraft. Reducing the variance of the launch preparation task decreases the occurrence of very large, unpredictable outlier values and should also result in decreases in the average launch preparation time. Additionally, removing these large outlier values might also make it more difficult to sustain queues of aircraft at the catapults. Removing these outliers also has a side effect of increasing the predictability of the task, a benefit for crew and supervisors on the flight deck.

If the launch preparation process continues to require that crew perform actions manually, decreasing the mean might be accomplished by providing the crew with improved checklists, electronic checklists, or better training. These changes might also drive down the standard deviation, but a more effective way of reducing both the mean and the standard deviation would be by automating aspects of the launch preparation process. Checking the aircraft's weight and using it to set the catapult strength could be performed by wireless data transmission, while the physical connection of the aircraft to the catapult's "pendant" that propels the aircraft forward could perhaps be done through passive interlocking mechanisms that require no human intervention.

However, it is not clear how much the launch preparation time needs to be reduced to eliminate the queueing process and whether or not taxi operations can keep pace with these reductions. In this first section, the effects of varying the mean and standard deviation of the launch preparation task are explored in order to determine their effects on operations and whether or not taxi operations becomes a limiting factor in operations. The first series of tests systematically reduces the standard deviation σ of the launch preparation time model in increments of ten percent while holding the mean μ constant at 109.65 seconds. A second series holds standard deviation constant at 57.80 seconds while decreasing the mean. A third examines the combined effects, varying the mean while holding the standard deviation constant at its smallest value from the first set of tests (28.90 seconds).

The list of cases tested appears in Table 6.1 and were applied to the baseline MC control architecture that was previously tested in Chapter 4. Tests occurred only at the 22 aircraft Density setting, with thirty replications occurring for each launch preparation time model. Results for these tests appear in Figure 6-1 in line chart form. At each plotted point, whiskers indicate ±1 standard error in the data for the specified launch preparation model. The figure also includes linear fits for all three data series, with both the fit and the resulting R^2 value reported for each case.

As expected, reducing either the mean or the standard deviation has beneficial effects on the Launch event Duration (LD) values, with reductions in the mean providing stronger effects. Decreasing the standard deviation by 50% provides only a 10.48% reduction in average LD from the original launch preparation time model (Table 6.2). This same performance is provided by only a

Table 6.1: Settings for launch preparation time testing.

	Reducing Standard Deviation	Reducing Mean	Reducing Mean, 50% Standard Deviation
Baseline	$N(109.65, 57.80^2)$	$N(109.65, 57.80^2)$	$N(109.65, 28.90^2)$
-10%	$N(109.65, 52.02^2)$	$N(98.6832, 57.80^2)$	$N(98.6832, 28.90^2)$
-20%	$N(109.65, 46.24^2)$	$N(87.7184, 57.80^2)$	$N(87.7184, 28.90^2)$
-30%	$N(109.65, 40.46^2)$	$N(76.7536, 57.80^2)$	$N(76.7536, 28.90^2)$
-40%	$N(109.65, 34.68^2)$	$N(65.7888, 57.80^2)$	$N(65.7888, 28.90^2)$
-50%	$N(109.65, 28.90^2)$	$N(54.824, 57.80^2)$	$N(54.824, 28.90^2)$

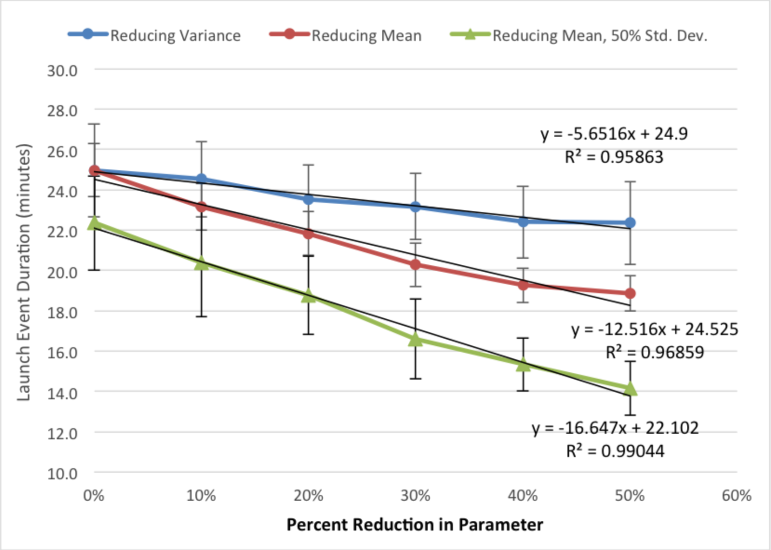


Figure 6-1: Effects of altering launch preparation mean and standard deviation on mean launch event duration for 22-aircraft Manual Control case.

20% reduction in the mean. Reducing the mean by 50% (and holding standard deviation constant) leads to a 25% reduction in average LD from the original LP model ($\mu = 24.97$ to $\mu = 18.86$). Reducing both the mean and standard deviation amplifies the beneficial effects of the two: at the 50% mean, 50% standard deviation condition, average LD is reduced by 43%, from 24.97 to 14.17 minutes.

The four end conditions (the original Launch Preparation (LP) model, standard deviation-50%, mean-50%, and mean-50% + standard deviation-50%) were repeated for the 34 aircraft case. The results of these tests appear in (Table 6.2) and show similar effects as that of the 22 aircraft tests. The standard deviation-50% case provides a 9.75% decreases in average LD from the original launch preparation time model, while the mean-50% case provides a 25.17% decrease in mean LD. Combining the two provides a 41.54% decrease in launch event duration. At this latter setting, the average time to launch a mission of 34 aircraft (22.78 minutes) requires less time than a team of 22 aircraft under the original launch preparation time model (24.97 minutes).

Table 6.2: Mean launch durations and percent reductions for the 34 aircraft Manual Control case for variations in launch preparation (LP) model.

	22 aircraft		34 aircraft	
	Mean Launch event Duration	% Reduction in Mean	Mean Launch event Duration	% Reduction in Mean
Baseline	24.97	-	38.97	-
SD -50%	22.35	-42.64%	35.17	-9.75%
Mean -50%	18.86	-51.61%	29.16	-25.17%
Mean -50%, SD -50%	14.17	-63.65%	22.78	-41.54%

These effects are of immediate practical significance to current Navy research and its development of a next generation of electromagnetic (as opposed to current steam-powered) launch catapults for the new *Ford* class of aircraft carriers. To generate the most benefit to operations, reducing both the mean and the standard deviation is suggested. However, it is not clear that these improvements in the launch preparation process are part of the objectives of the new catapult system being developed. The EMALS project seeks to replace the steam catapult system with an electromagnetic one, but of the six benefits listed by developer General Atomics on their website[1], only one relates to manpower or launch rates ("reduced manning workload"). Even so, "reduced manning workload" is likely far from complete automation of the process, which should further reduce the mean and the standard deviation of task process times. The decreases to mean and standard deviation tested in this section are modest compared to what might be possible with advanced automation of the launch preparation task.

The results of this section have demonstrated that reducing both the mean and the standard deviation of the launch preparation time improves LD values for flight deck operations. The most substantial reductions in the launch preparation time model (reducing both the mean and standard deviation by 50%) resulted in decreases in launch event duration of over 40% from the original LP model. These reductions in launch preparation time may be strong enough to break the queuing process for the different control architectures and allow variations in their LD values to arise. This is the focus of the testing described in the next section.

6.1.1 Effects of Launch Preparation Times on UAV Performance

In the previous tests in Chapter 5, very little variation in average launch event duration values was observed across the different unmanned vehicle control architectures. The only significant interactions that were observed occurred between the subset of the very best and very worst test cases. The cause of this was hypothesized to be continued existence of queues at the catapults, produced by the long launch preparation times combined with Handler allocation strategies. Decreasing the launch preparation time should at some point allow queues to clear rapidly enough that taxi operations

[1] http://www.ga.com/emals

cannot keep pace, allowing variations in launch event durations for the different control architectures to be observed. This is the focus of this section, beginning with an examination of the effects of changing Launch Preparation (LP) parameters on control architecture performance.

Effects of Reducing Launch Preparation Standard Deviation

A first series of tests examined the effects of reducing the standard deviation in the launch preparation time model on the launch event duration performance of the different UAV Control Architectures. These tests address whether the reduction in standard deviation, which reduces the largest launch preparation times that occur, reduces the ability of the control architectures to generate queues at the catapults. The Gestural Control (GC) and System-Based Human Supervisory Control (SBSC) control architectures were tested first based on their status as the best- and worst-performing architectures, respectively. Each was tested utilizing the 22 aircraft Density level and the 100% Composition level in order to maximize the effects of each control architecture. If reducing the standard deviation of the launch preparation task has an effect on queuing, it should result in significant differences between these two control architectures.

For each architecture, standard deviation values were decreased from the baseline value ($\sigma =$ 57.80 seconds) in increments of 10%. Thirty replications were performed for each control architecture at each standard deviation value. From the results of the previous section, the response of the GC and SBSC architectures is expected to be linear with respect to decreases in launch preparation standard deviation, although differences in performance may arise between the two. The results of these tests are shown in Figure 6-2, which includes the results for Manual Control (MC) from the previous section as a comparison point.

A fairly linear decrease is observed for each architecture, although there does appear to be significant noise within the data. The SBSC and MC data also appear to converge at the smaller standard deviation values (beginning at the -30% condition), suggesting that improving the launch preparation process also removes the advantages from the more open routing methods of SBSC. Conversely, it might also signify that the inefficiencies of the crew zone coverage routing are weaker at the improved launch preparation tempo. The GC architecture, however, maintains its relatively poor performance at each level.

At the smallest standard deviation value (28.90 seconds, original value -50%), the average time to complete a mission across the three control architectures decreases from 24.67 minutes to 22.73 minutes (a 7.87% drop in mean LD). The GC results (mean = 23.09, standard error = 0.24 minutes) appear to be significantly different from the SBSC results (mean = 22.41, s.e. = 0.22 minutes). Additional tests were then conducted for the remaining control architectures (Local Teleoperation (LT), Remote Teleoperation (RT), and Vehicle-Based Human Supervisory Control (VBSC)) using the standard deviation -50% setting at the 22 aircraft Density levels, and tests for all architectures were then conducted at the 34 aircraft level. These results appear in Figure 6-3 alongside results for

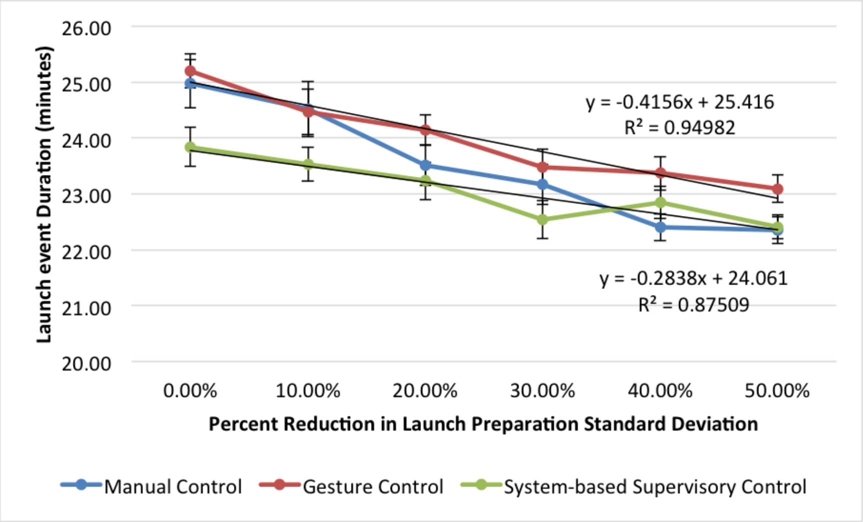


Figure 6-2: Line chart of mean Launch event Duration (LD) for the Manual Control (MC), Gestural Control (GC), and System-Based Human Supervisory Control (SBSC) control architectures for decreases in launch preparation time standard deviation. Whiskers indicate ±1 standard error in the data.

missions using the original launch preparation time model.

An ANOVA test of the results at the 22 aircraft level reveals significant differences across cases ($F(5,174) = 5.1771, p = 0.0002$). A Tukey HSD test reveals that significant pairwise differences exist between VBSC ($\mu = 22.14$ minutes) and RT ($\mu = 23.55$, $p = 0.0006$), VBSC and LT ($\mu = 23.18$, $p = 0.0067$), MC ($\mu = 22.35$) and RT ($p = 0.0296$), and SBSC ($\mu = 22.41$) and RT. Additionally, the GC ($\mu = 23.09$) and VBSC cases are marginally different ($p = 0.0738$). At the 34 aircraft density level, an ANOVA also reveals significant differences ($F(5,174) = 5.5665, p < 0.0001$). A Tukey HSD test shows significant pairwise differences only between the MC ($\mu = 35.17$) and RT ($\mu = 36.79$),

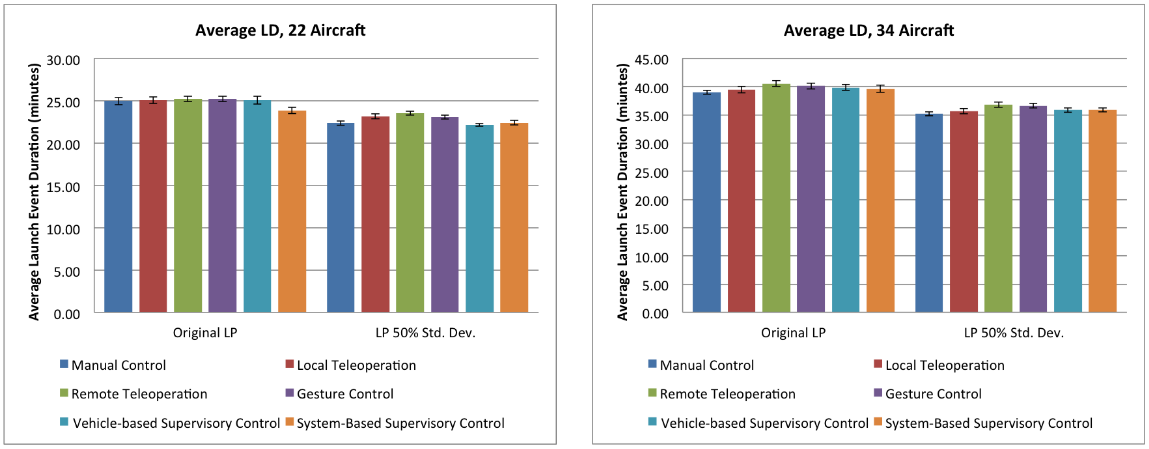


Figure 6-3: Column graphs depicting Launch event Duration (LD) values all control architectures under the 50% standard deviation, original mean launch preparation time model for 22 (left) and 34 (right) aircraft.

MC and GC ($\mu = 36.61$), and LT ($\mu = 35.66$) and RT cases.

In terms of practical differences, the range of average launch event duration values does not greatly change under the new reduced standard deviation launch preparation time model. Under the original LP model, the difference between best and worst was only 1.38 minutes for the 22 aircraft missions and 1.56 minutes for the 34 aircraft missions. Under the new LP model with standard deviation reduced by 50%, these differences increase slightly to 1.42 minutes for 22 aircraft and 1.61 minutes for 34 aircraft. However, the slight decrease in average launch event duration under the new launch preparation time model makes these differences significant in statistical tests. Even so, these differences are limited to comparisons of the very best (MC, LT, and SBSC) and the worst (GC and RT). Interestingly, at these settings, the VBSC architecture provides some of the best results, in opposition to some of the results presented in Chapter 5.

One explanation for this is that the periodic high-value launch preparation times that occur under the original, high standard deviation LP model cause an uneven rate of traffic flow of the deck. Extremely long launch times will be followed by comparatively shorter launch times, requiring that the Handler start operations for two vehicles in short succession. It is also likely that at least one launch has occurred at the other catapult during this time, further increasing the number of vehicles the Handler must being taxiing. When this occurs, the iterative interactions required from the Handler help to constrain operations. Under an LP model with a reduced standard deviation, the variability of the launch preparation task is reduced and should provide a steadier pace of operations. With launches occurring at a more consistent rate, the number of vehicles in the Handler queue at any one time should be smaller and remain fairly constant, easing the Handler's job of coordinating vehicle tasks. This is one example in which decreasing the random variation in the launch preparation process might benefit operations. However, given substantial reductions in the mean launch preparation time, the Handler may not be able to keep pace with operations and his performance may worsen as compared to the other architectures. These effects are examined in the next section.

Effects of Reducing Launch Preparation Mean

A second series of tests was conducted in which the standard deviation was held constant at its original value ($\sigma = 57.80$) and the mean was decreased in increments of 10%. These tests were then repeated, varying the mean while holding the standard deviation constant at the -50% settings ($\sigma = 29.80$). Tests again began with the GC and SBSC architectures using 22 aircraft and running only homogeneous (100% Composition) sets of aircraft under the Dynamic Separation (DS) protocol. The MC architecture results are again used as a baseline comparison case. The results using both the original and -50% standard deviation cases are shown in Figure 6-4.

As with the MC results shown previously, a linear response was expected for all control architectures at both standard deviation settings and is observed. The increasing benefits of reducing both

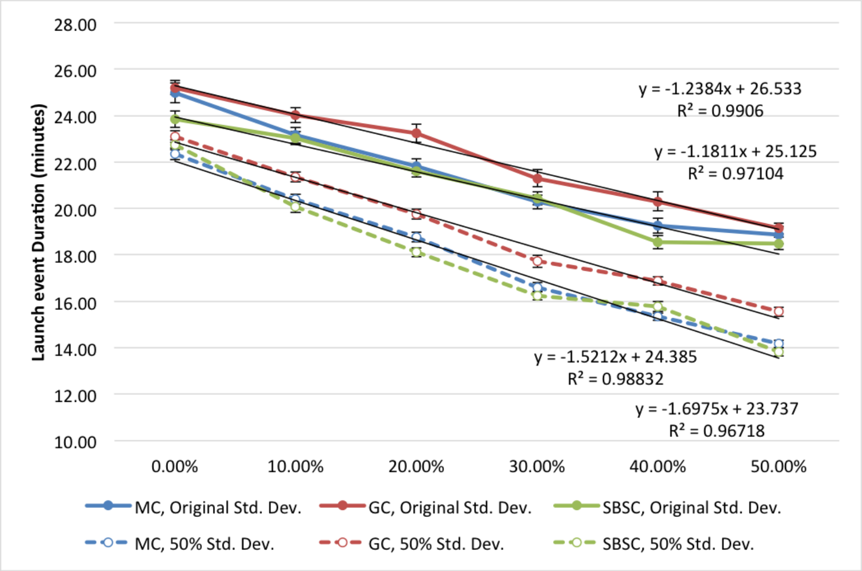


Figure 6-4: Line chart of Launch event Duration (LD) values for Manual Control (MC), Gestural Control (GC), and System-Based Human Supervisory Control (SBSC) missions using 22 aircraft varying the launch preparation mean. Whiskers indicate ±1 standard error.

mean and standard deviation also appear for the GC and SBSC control architectures, and it again appears that these effects are more beneficial for the SBSC architecture. At the 50% mean, 50% standard deviation condition, the GC (mean = 15.55, s.e. = 0.20 minutes) case again appears to be significantly different from the SBSC case (mean = 13.82, s.e. = 0.18 minutes), which also appears to be consistently similar to the MC results.

All six control architectures were then tested at the 50% mean, 50% standard deviation settings ($N(54.82, 29.8^2)$) using both 22 and 34 aircraft Density settings to examine what differences exist between architectures. The results of these tests appear in Figure 6-5. For 22 aircraft cases, an ANOVA again shows significant differences across cases ($F(5, 174) = 14.6874, p < 0.0001$) with a Tukey HSD test now showing ten significant or marginally significant pairwise differences ($p < 0.0328$ for all; full results appear in Appendix O) between cases. Of these, nine include comparisons between the "delayed" (RT, GC, VBSC) and non-delayed (MC, LT, SBSC) architectures. Two of these comparisons (LT vs. RT and LT vs. VBSC) are only marginally significant ($p = 0.0691$ and $p = 0.0723$, respectively). The tenth comparison that shows significant differences involves the LT and SBSC architectures ($p = 0.0328$). At the 34 aircraft level, a Kruskal-Wallis test shows significant results ($\chi^2(5) = 88.54, p < 0.0001$), with a Steel-Dwass non-parametric simultaneous comparisons test showing nine significant pairwise differences, once again occurring between the sets of "delayed" and "non-delayed" groups. All nine comparisons return highly significant differences ($p = 0.0002$ or less).

Overall, the results reported in this and the previous section suggest that, as hypothesized in

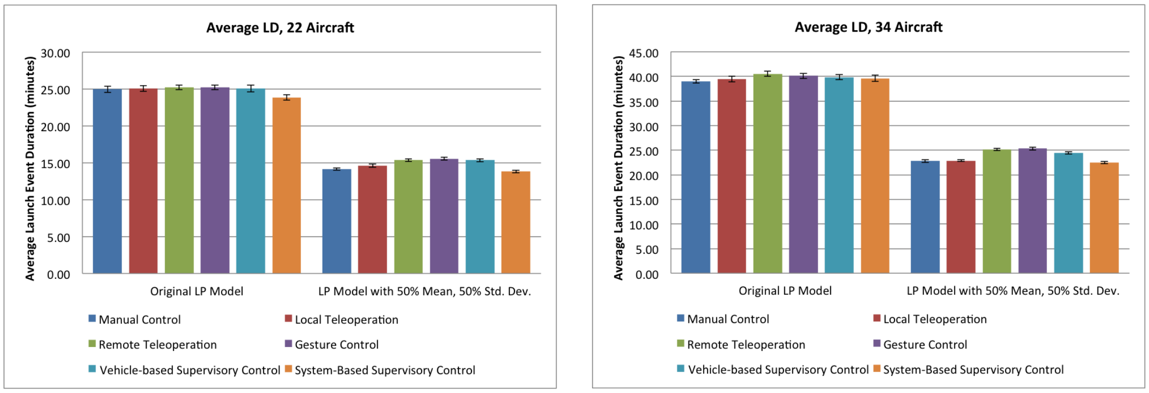


Figure 6-5: Column graphs depicting Launch event Duration (LD) values all control architectures under the 50% mean, 50% standard deviation launch preparation time model for 22 (left) and 34 (right) aircraft.

Chapter 5, the definition of the launch preparation time forms a choke point on operations. Decreasing the mean and standard deviation of the time required to complete this task provides significant benefits to operations. These changes also allow significant differences in control architectures to arise, providing similar trends in performance as observed for the single aircraft tests in Chapter 5: the “delayed” UAV control architectures of RT, GC, and VBSC perform worse than the remaining “non-delayed” architectures. However, for these data, the differences in best and worst performance of the control architectures at the mean-50%, standard deviation-50% cases (less than 2 minutes at 22 aircraft, less than 3 minutes at 34) are far smaller in magnitude than the benefits of changing the launch preparation model. For stakeholders working in this and other similar domains, if the implementation of autonomy is being done to increase the productivity of the system, those stakeholders should also question what other aspects of operations could benefit from increased automation. For the aircraft carrier flight deck, there are greater benefits to operational productivity in improving the launch preparation process than in automating individual vehicle behavior.

However, the launch preparation models used in this section might be improved further with even more substantial automation upgrades. As described previously in this chapter and earlier in Chapter 3 (Section 3.2), the tasks that are part of the launch preparation process are independent of the type of vehicle being used and rely heavily on the crew. Crew must check the weight of the vehicle, the settings of the catapult, and ensure that the nosewheel or the aircraft is properly aligned to the catapult’s “pendant,” the portion of the catapult mechanism that connect to the nosewheel and pulls the aircraft down the catapult. It is not unreasonable to imagine that these tasks could all be automated: wireless data processing could accomplish the first two tasks, while improved sensor systems and passive locking mechanisms could provide a secure, near-instant connection of aircraft to catapult. Such automation, similar to that done in the manufacturing domain, has the potential to even more significantly increase the performance of operations on the flight deck. The introduction of automation into this process, rather than into aircraft via the different control architectures, is explored in the next section.

6.1.2 Effects of Automating Launch Preparation Tasks

Extreme improvements to the launch preparation process could be achieved by completely automating all subtasks involved: automating the physical connection between the aircraft and the catapult mechanism as well as wirelessly communicating all information between the aircraft and the catapult control system. Doing so should enable large reductions in both the mean and the standard deviation of the launch task; at such levels, differences in the performance between the various control architectures would be due solely to inefficiencies in the taxi operations process. An extreme upper bound of performance might have an average of 5 seconds with a standard deviation of 1 second over the bounds [3,7] seconds, decreasing the time required to complete the launch preparation far below that of the values tested in the previous sections. This model ($N(5,1^2)$ seconds) was input into the MASCS model and a series of simulation trials were conducted using all control architectures at both the 22 and 34 aircraft Density settings at the 100% Composition DS settings. Mean LD values for these tests appear in Figure 6-6.

To begin, using an aggressive automated launch preparation time model results in substantial decreases in time to complete the mission as compared to the original launch preparation time model: an average of 8.57 minutes across all control architectures at the 22 aircraft level and 15.79 minutes at 34 aircraft. These are reductions of 65.7% and 59.5% from the average duration of the MC architecture under the original launch preparation time model (mean values of 24.97 and 38.97, respectively). Under this highly automated task, the time to complete the mission is solely dependent on the time required to taxi aircraft to catapults: aircraft launch so quickly that queues are rarely able to form, and even then only exist for a few seconds. Table 6.3 provides the means and

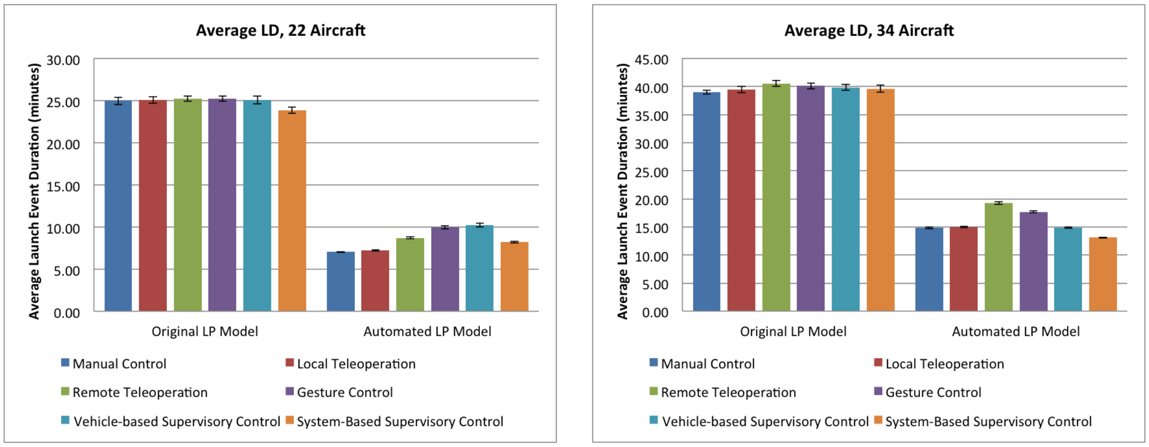


Figure 6-6: Column graphs depicting Launch event Duration (LD) values all control architectures under the "automated" launch preparation time model for 22 (left) and 34 (right) aircraft.

standard deviations of total waits time in queue for all aircraft in the 22 and 34 aircraft missions under both the original and automated launch preparation time models. While these values are cumulatively over 60 and 90 minutes under the original launch preparation time models at the two Density settings, respectively, average queue times are less than one minute under the automated launch preparation model.

Note, also, in the lefthand side of Figure 6-6 that for the 22 aircraft results the MC, LT, and SBSC architectures are again the best performers while RT, GC, and VBSC perform more poorly. However, unlike other tests, the gap between the two groups is fairly small: the SBSC and RT results are fairly close together. A Kruskal-Wallis test for data at the 22 aircraft Density level reveals significant differences ($\chi^2(5) = 144.48, p < 0.0001$), with a Steel-Dwass non-parametric simultaneous comparisons test showing significant pairwise differences for thirteen of fifteen possible comparisons. The only pairings that are not significantly different are GC (mean rank = 143.67) compared to VBSC (mean rank = 147.5), the two worst performers, and MC (mean rank = 28.43) paired with LT (mean rank = 37.2), the two best. All other comparisons are strongly statistically significant ($p = 0.0077$ or less). This also means that the two groups of "delayed" (RT, GC, and VBSC) and "non-delayed" architectures no longer exist as they have in previous results: MC and LT are both significantly different from SBSC (mean rank = 81.0, both at the $p < 0.0001$ level), while GC and VBSC are both different from RT (mean rank = 105.2, both at the $p < 0.0001$ level).

The "delayed" versus "non-delayed" trend of differences also disappears at the 34 aircraft level, albeit in a different fashion. From the results in the figure, it appears that the SBSC architecture is the best overall performer, while VBSC improves significantly and RT degrades substantially. A Kruskal-Wallis test again shows significance ($\chi^2(5) = 147.01, p < 0.0001$), with a Steel-Dwass test also showing multiple significant pairwise differences. In these results, all but three pairwise differences are strongly significant ($p < 0.0001$): MC (mean rank = 72.2), VBSC (mean rank = 74.7), and LT (mean rank = 78.17) are not significantly different from one another. However, SBSC (mean rank = 17.83) has significantly smaller values than each of these cases and is the lone best performer. Additionally, RT (mean rank = 159.93) and GC (mean rank = 140.17) are worse than the remaining cases and significantly different from one another. At the 34 aircraft Density level, one of the "delayed" architectures, VBSC, is now equivalent to two of the non-delayed architectures

Table 6.3: Means and standard deviations of wait times in catapult queues at both the 22 and 34 aircraft Density levels under the original and automated launch preparation models.

		Mean Wait Time in Queue (minutes)	Std Dev
22 aircraft	Original	69.9773	12.3197
	Automated	0.6236	0.4519
34 aircraft	Original	92.7235	13.4196
	Automated	0.8987	0.6173

(MC and LT), having flipped from being one of the worst performers at the 22 aircraft level to one of the best at the 34 aircraft level.

Looking across results at the two Density settings (22 and 34 aircraft), MC and LT had equivalent performance in each, which is generally to be expected given the minimal differences between the two architectures. The generally poor performance by both RT and GC is also to be expected; that GC is relatively better than RT at 34 aircraft and worse at 22 is an interesting result, however. Additionally, the large relative improvement in performance for the VBSC and SBSC cases at 34 aircraft is striking, especially for VBSC. These latter differences in performance are perhaps explained by the lack of the "zone coverage" routing structure, the common feature differentiating VBSC and SBSC from the other architectures. One explanation is that the "open" routing used by these two architectures is not optimized for the 22 aircraft setting, and the lack of constraints on taxi operations allows more aircraft to interfere with each other and trigger their respective collision prevention routines. For the other architectures, the use of the zone coverage routing leaves sufficient constraints in place that aircraft remain well-spaced and do not often conflict with one another. At the 34 aircraft level, the limitations on taxi paths due to the larger number of aircraft may be enough that the VBSC and SBSC do not encounter such issues and retain their beneficial performance. A similar explanation might explain why RT and GC differ in performance between the two density levels — the effects of GC delays and latencies are more strongly felt at the unconstrained 22 aircraft Density, with the constraints at the 34 aircraft level limiting the detrimental effects of that architecture.

In general, however, the role of the launch preparation time model has been clearly demonstrated by tests in this section both in terms of its effects on launch event duration values for flight deck operations as well as its effects in regulating UAV control architecture performance. Once the launch preparation time model was sufficiently reduced to eliminate the presence of queues on the flight deck, the differences in control architecture performance finally became visible. However, one caveat to these results are that the heuristics used by the Aircraft Director and Deck Handler agents were not designed for such fast operational tempos. The heuristics are quite likely not optimal for these operating conditions, and even better performance might be achievable if these heuristics are altered.

Additionally, the tests performed in this section highlight the importance of exploring all facets of the broader system, including processes defined independently of any of the unmanned vehicle control architectures. For a stakeholder selecting a control architecture based on mission performance, a failure to recognize the effects of tasks like the launch preparation process on operations could lead to errors in decision-making. The results observed in Chapter 5 under the original launch preparation time model might lead a decision-maker to conclude that all architectures are equivalent, or that they provide no overall benefit compared to manual control operations. However, if the broader system is slated to transition to a faster operating tempo, selecting GC or RT systems would clearly be incorrect for future operations. In the opposite direction as well, using the results from tests utilizing the “automated” launch preparation time model may lead stakeholders to select the LT or SBSC systems. If the system spends the majority of operations working within a slower tempo, then the benefits expected from these systems will never be realized and the selection of a different architecture might have been warranted.

To this point, this chapter has focused solely on launch event duration performance for the different UAV control architectures under the 100% Composition cases, to which the safety protocols defined in Chapter 5 cannot be applied. However, just like for the control architectures, the results previously discussed in Chapter 5 showed little variation in performance amongst the Area, Temporal, and Dynamic Separation safety protocols. It is likely that the lack of variation in safety protocol performance was also due to the effects of the sustained queueing of aircraft at catapults, and eliminating queuing through the use of the “automated” launch preparation time model should make the effects of the different safety protocols on launch event duration more visible. This chapter has also not yet explored the effects of the launch preparation time model on the results of the safety metrics, which were heavily influenced by the selection of safety protocols. The next section explores both of these topics, beginning with the effects of safety protocols on launch event durations under the new, automated launch preparation time model.

6.1.3 Interactions with Safety Protocols

Just as the original launch preparation time, with its large mean and standard deviation, obscured the effects of different control architectures, so too might it have obscured the effects of Safety Protocols on operations. Chapter 5 demonstrated that for LD performance, other than the severely-constrained Temporal+Area protocol, the only significant interactions between control architectures and safety protocols occurred between the Temporal Separation protocol and the three “delayed” UAV types. This was believed to have been caused by the increased use of forward-to-aft aircraft routing on the flight deck, the effects of which should be magnified at higher operational tempos and should further exacerbate errors and latencies in the three “delayed” control architectures. An additional series of tests were executed for the GC and SBSC Control Architectures under all possible settings of Safety Protocols (SPs), mission Density, and mission Composition settings using the new “automated” launch preparation time model. This section reviews the results of these tests on the Launch event Duration (LD) productivity measure and selected safety measures.

Effects on Productivity Measures

Figure 6-7 shows the average Launch event Duration (LD) values for the GC and SBSC architectures across all safety protocols at both 22 (left) and 34 (right) Density levels using the new “automated” launch preparation time. These figures also include the average LD of these architectures under the original launch preparation time model. The trends in the performance of the safety protocols under the automated launch preparation time model are generally the same for both mission Density levels: Dynamic and Area Separation provide the smallest LD values, while Temporal Separation and Temporal+Area provide the largest. Interestingly, SBSC appears to provide smaller LD values under Area and DS protocols, while GC performs slightly better at the Temporal and T+A settings.

Statistical tests verify that significant differences occur across Safety Protocols at all levels. For GC results at 22 aircraft, a Kruskal-Wallis test returns significance ($\chi^2(3) = 194.78, p < 0.0001$), with a Steel-Dwass nonparametric multiple comparisons test showing that all possible pairwise comparisons are significant at $p = 0.0048$ or less, with Area (mean rank = 58.97) providing the best performance, followed by DS (89.84), Temporal (195.85), then T+A (220.17). Similar results are seen for the SBSC data at 22 aircraft, with a Kruskal-Wallis test returning significance ($\chi^2(3) = 216.24, p < 0.0001$), with a Steel-Dwass nonparametric multiple comparisons test showing that all pairwise comparisons are significantly different with Area (mean rank = 59.817) providing the best performance.

At the 34 aircraft level, a Kruskal-Wallis test for SP results for the GC control architecture also shows significance ($\chi^2(3) = 209.13$), where a Steel-Dwass test demonstrates that all but one pairwise comparison is significantly different: the Dynamic Separation only (mean rank = 85.96) and Area Separation (mean rank = 59.82) protocols are not significantly different from one another ($p = 0.0527$). A Kruskal-Wallis test for SP results for the SBSC control architecture also shows significance ($\chi^2(3) = 220.73, p < 0.0001$) and a Steel-Dwass test also indicates all but one pairwise comparison is strongly significantly different. In this case, however, the Temporal (mean rank = 202.27) and Temporal+Area (mean rank = 218.73) are shown to only be marginally significantly different from one another ($p = 0.0473$).

Taken together, these results support the earlier results from Chapter 5 that indicated that the

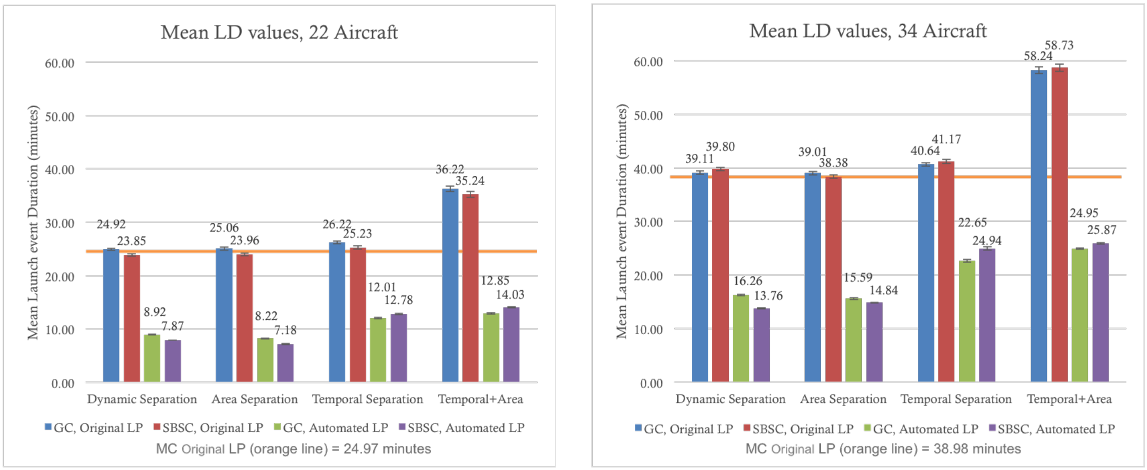


Figure 6-7: Column graphs depicting Launch event Duration (LD) values for the Gestural Control (GC) and SBSC Control Architectures run at the 22 (left) and 34 (right) aircraft Density levels under the "automated" launch preparation time for all Safety Protocols (SPs).

Temporal protocol had adverse effects on operations. These previous results from Chapter 5 showed that these effects only occurred for the "delayed" UAV control architectures, however; the new results from this Chapter suggest that the Temporal is detrimental to operations in and of itself. That this was not observed in Chapter 5 is likely due to the effects of the launch preparation time on operations, which may have served to limit the effects of this safety protocol just as it limited the differences in control architectures. Interestingly, the effects of the Temporal and T+A protocols are felt more strongly by the SBSC architecture than by GC; while SBSC provides better performance than GC under the Area Separation and Dynamic Separation protocols, it provides worse performance under Temporal and Temporal+Area. This is likely indicative of interactions with the open routing structure of the SBSC architecture and it not being designed for this operational tempo. This suggests that, while the zone coverage routing system does provide additional constraints on operations, the lack of constraints under the more open routing method may allow aircraft to more greatly interfere with one another's actions, limiting some aspects of operations. The zone coverage system may be beneficial in structuring operations and preventing these interferences and interactions from occurring.

The "automated" launch preparation time also has interesting interactions with mission Density. Increasing mission Density directly increases the time to complete a launch event, but the relative effects of mission Density differ between the original and automated launch preparation time models. Figure 6-8 calculates the percent increase in mean LD values when moving from the 22 to 34 aircraft Density level for the GC and SBSC control architectures under the two Launch Preparation (LP) time models. Under the original LP model, increasing mission Density from 22 to 34 aircraft results in a roughly 60% increase in LD for both the GC and SBSC architectures; this is basically equivalent to the percent increase in number of aircraft (12/22 = 54.54%). Under the automated launch preparation time model, the percent increase is closer to 90% for both architectures.

This difference in Density effects can be attributed to the effects of eliminating queues at the catapults. Under the original LP model, launch event durations are solely a function of the number of launches that are required due to the sustained existence of queues at the catapults. Increasing the number of aircraft in the mission simply adds to the number of consecutive launches that take place at the catapults. Under the automated LP model, these queues are eliminated and the efficiency of taxi operations becomes the critical function in operations. The stronger effects of increased Density under the automated launch preparation time model (increases of 90% from the 22 aircraft Density level) likely indicate that the taxi operations currently modeled in MASCS are inadequate for these higher-tempo operations. If taxi operations are appropriately redesigned for each control architecture, further decreases in launch event duration might be obtainable at both mission Density values. However, flight deck productivity is only one measure of performance for flight deck operations, and the changes to the launch preparation time models may also affect

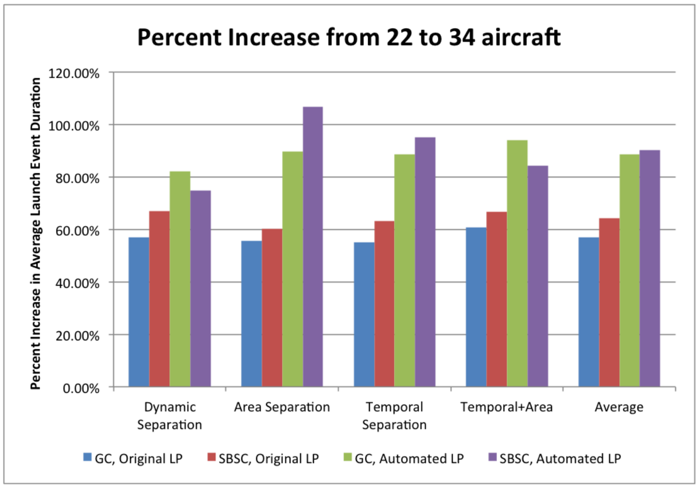


Figure 6-8: Percent increase in Launch event Duration when moving from 22 to 34 aircraft.

the results for the safety metrics. The next section reviews the effects of automating the launch preparation process on the safety measures for flight deck operations defined previously in Chapter 5.

Effects on Safety Measures

In Chapter 5, the analyses of safety measures addressing aircraft-aircraft interaction focused on the Tertiary halo radius (two wingspans away). Measures for crew interactions with vehicles were also developed; while these Primary Halo Incursion (PHI) and Duration of Primary Halo Incursions (DPHI) metrics are imperfect given the crew behavior models included in MASCS, they were analyzed in Appendix M. This section explores the effects of reducing the launch preparation mean and standard deviation on the Tertiary Halo Incursions due to Aircraft (THIA), PHI, and DPHI metrics. Figure 6-9 shows comparisons of THIA values between the original and automated launch preparation time models for each of the four Safety Protocols for the SBSC architecture. The same data is shown for GC in Figure 6-10.

Statistical tests verify that there are significant differences in performance between the original and automated launch preparation time models. The results of Wilcoxon Rank Sum tests comparing THIA values at the automated and original launch preparation time models at each Density level appear in Table 6.4. As can be seen, all four comparisons are strongly statistically significant. Practically, these results may be somewhat counterintuitive: that increasing the tempo of operations actually *increases* the safety of operations; this is not something that would typically be expected for most systems. It suggests, again, an interaction with the routing of aircraft on the deck. The increased tempo of operations speeds the rate at which catapults launch; while the rate of assignments by the Handler also increases, the speed at which aircraft taxi is left constant, leaving fewer aircraft near the catapults and one deck in general. This may also explain why THIA values for the automated launch preparation task at the 22 aircraft Density level appear to be invariant to Safety

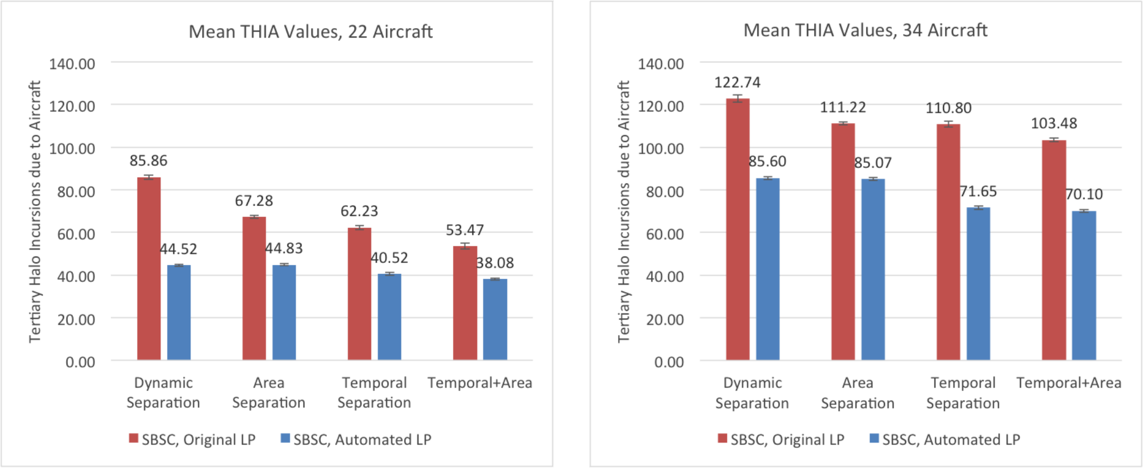


Figure 6-9: Column charts comparing Tertiary Halo Incursions due to Aircraft (THIA) values between the "automated" launch preparation time and the original launch preparation time for all treatments in the SBSC control architecture. Left: 22 aircraft cases. Right: 34 aircraft cases.

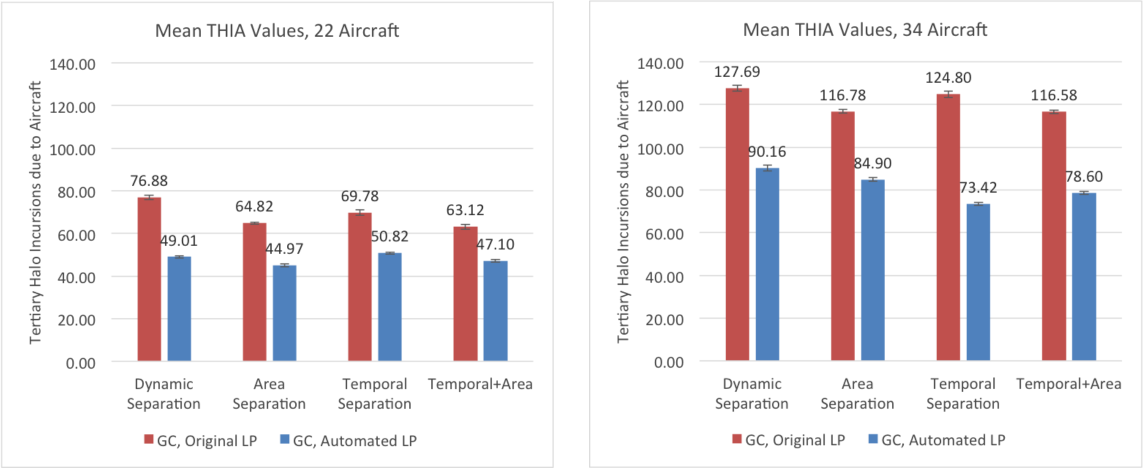


Figure 6-10: Column charts comparing Tertiary Halo Incursions due to Aircraft (THIA) values between the "automated" launch preparation time and the original launch preparation time for all treatments in the GC control architecture. Left: 22 aircraft cases. Right: 34 aircraft cases.

Protocols for both the GC and SBSC architectures: aircraft are not on the deck long enough that taxi operations are basically unconstrained and the safety protocols have little real effect.

A review of values for PHI (Fig. 6-11 and 6-12) indicate only slight differences in performance between the two launch preparation time models for the SBSC or GC architectures at either Density level. The PHI results in Appendix M showed that the major influence was in the selection of vehicle routing topology. As the launch preparation time does not affect this topology, the number of PHI incursions would not be expected to change.

However, the routing does occur at a faster pace, and DPHI values are shown to substantially decrease between cases (column charts also appear in Appendix O). This leads to a particularly

Table 6.4: Results of Wilcoxon Rank Sum Tests comparing THIA values between the original launch preparation time and the automated launch preparation time for the GC and SBSC Control Architectures. A value of $p < 0.013$ indicates significant differences between cases.

CA	Density	$\chi^2(1)$	DF	p-value
GC	22	376.92	1	$< 0.0001*$
GC	34	377.70	1	$< 0.0001*$
SBSC	22	328.34	1	$< 0.0001*$
SBSC	34	396.43	1	$< 0.0001*$

interesting set of implications. Increasing the rate at which launches occur by decreasing the mean and standard deviation of the launch preparation time reduces the total time of the mission, thereby reducing the total possible time during which crew interactions can occur. It does not change how aircraft are allocated, however, meaning that the total number of times crew interact with vehicles should (and does) remain unchanged. This means that the overall *rate* of crew incursions per minute of crew activity has increased, which in some environments might be considered the major driver of crew risk. This exemplifies not only a tradeoff between safety and productivity, but a tradeoff between different types of safety, one influenced by the speed of operations and one by the structure in which those operations occur. Which of these measures should be considered most important would require a detailed collection of data for current flight deck operations and accident rates that is outside the scope of this thesis and a potential area of future work. However, these results do provide evidence to support the idea that improving safety in one area may not improve safety in all areas.

Thus far, MASCS has focused on the assessment of previously-defined unmanned vehicle control architectures that were first described in Chapter 5 as models of mature futuristic systems. Tests

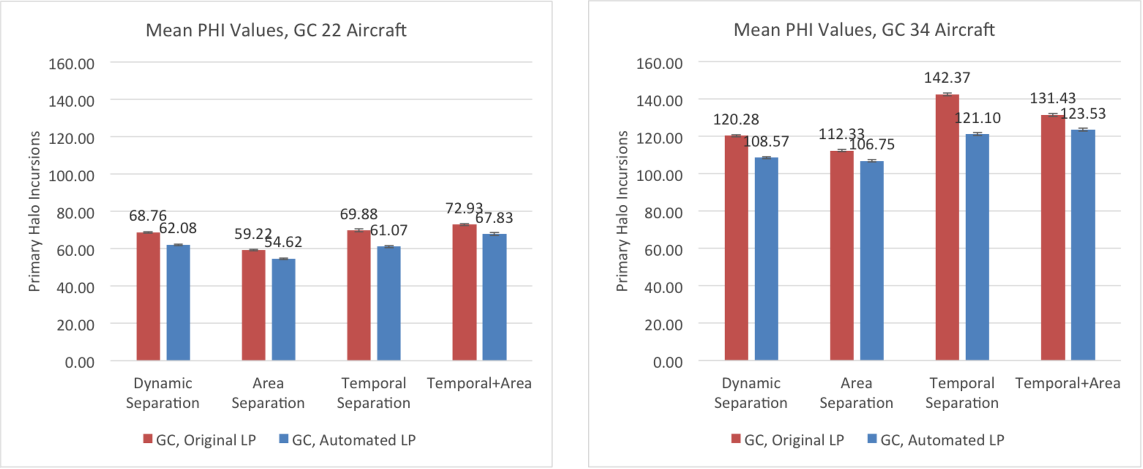


Figure 6-11: Column charts comparing Primary Halo Incursion (PHI) values between the "automated" launch preparation time and the original launch preparation time for all treatments in the GC control architecture. Left: 22 aircraft cases. Right: 34 aircraft cases.

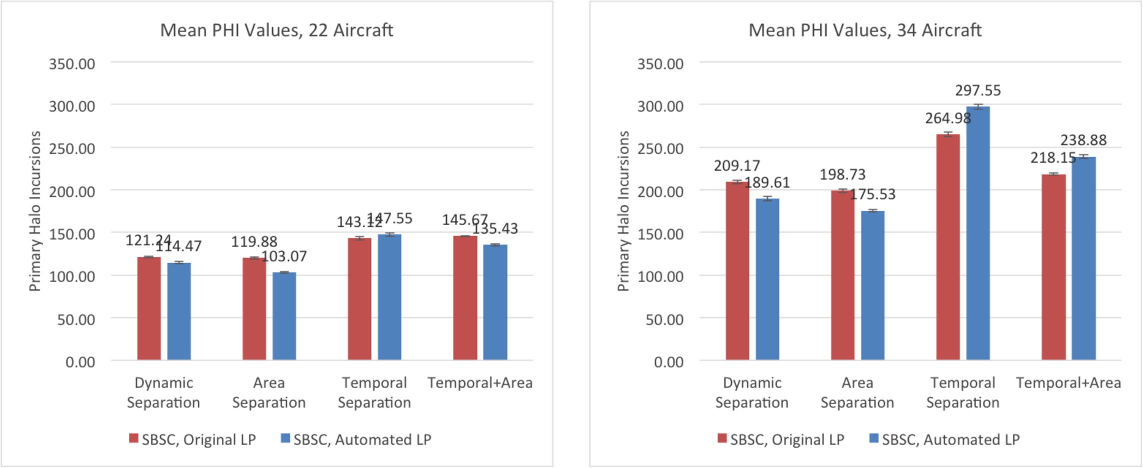


Figure 6-12: Column charts comparing Primary Halo Incursion (PHI) values between the "automated" launch preparation time and the original launch preparation time for all treatments in the SBSC control architecture. Left: 22 aircraft cases. Right: 34 aircraft cases.

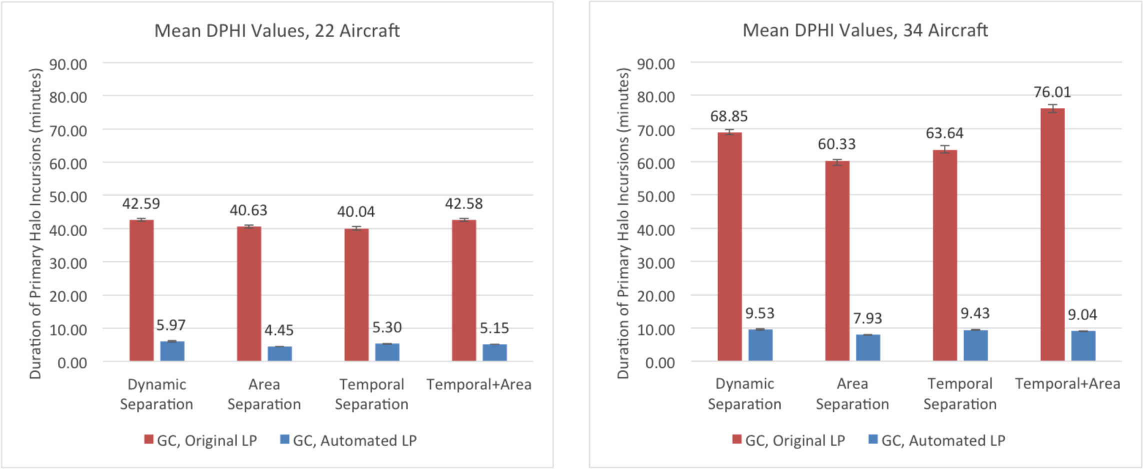


Figure 6-13: Column charts comparing Duration of Primary Halo Incursions (DPHI) values between the "automated" launch preparation time and the original launch preparation time for all treatments in the GC control architecture. Left: 22 aircraft cases. Right: 34 aircraft cases.

have focused on how these mature versions of these futuristic systems interact with process-level tasks in the environment, but the power of agent-based models like MASCS includes the ability to examine the effects of individual vehicle parameters on performance. This may be of use to stakeholders seeking to understand the relative cost-benefit ratio of investing in improvement to a current UAV control architecture or that seeks to understand exactly what type of performance is required for a current system to meet some performance threshold. The next section examines how simulations like MASCS could be used in exploring the design of individual control architectures.

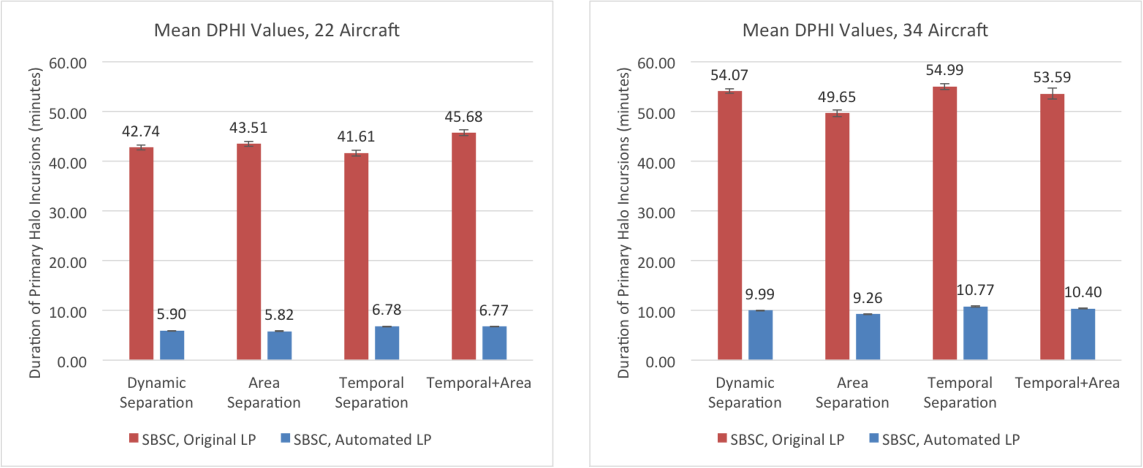


Figure 6-14: Column charts comparing Duration of Primary Halo Incursions (DPHI) values between the "automated" launch preparation time and the original launch preparation time for all treatments in the SBSC control architecture. Left: 22 aircraft cases. Right: 34 aircraft cases.

6.2 Exploring Control Architecture Parameters

MASCS might also be useful in determining the performance requirements of unmanned systems being developed for flight deck operations. In Chapter 5, a description and justification of the parameters used for the baseline control architectures was provided. However, any or all of these parameters may be modified and improved upon in future systems. It is not known which parameters, and at what values, significant differences in performance will occur. However, models such as MASCS can also be used in an exploratory fashion: given a description of the desired performance, one or more parameter settings for a given UAV control architecture could be varied until either the desired performance is achieved or the parameter value becomes unrealistic.

This can also serve as a way of providing direction for current autonomous systems research. Consider the case of the Gesture Control system, which had the worst overall performance for an individual aircraft and, at an increased operational tempo, the worst overall performance for a control architecture in homogeneous missions (100% UAV Composition). The GC model differed in several ways from the Manual Control baseline, and changes to one or more parameters may improve performance sufficiently to make the performance of the two architectures equivalent. Testing can also demonstrate at what point further improvement have no effect: a 95% success rate in gesture recognition may be sufficient to provide acceptable performance, and further investment to reach a 99% success rate would provide no additional benefits. Understanding each of these aspects of performance — which parameters are most important, what parameter settings generate equivalent performance, and at what point are further improvements no longer beneficial — could be used to help better characterize vehicle performance requirements in future systems. This section explores changes to these settings in the Gesture Control model, typically the worst-performing control architecture, to uncover at what parameter settings GC performance is becomes equivalent to the baseline Manual Control architecture and how those settings might vary across mission conditions.

A first series of tests examines the effects of varying GC parameters under the original launch preparation time model to determine whether or not significant effects are observed. Given the previous results, it is expected that the original launch preparation time model will again prevent any real differences in performance from appearing. For these tests, two different adjustments are made to the original Gesture Control system model. First, a “Low Failure” case is created, in which the failure rate is reduced from 11.18% to 1%. This assumes future systems that can reliably recognizing gestures and cope with variability both between individual Aircraft Directors and within a single Director (the precision of whose actions may degrade time). Failure rates are left as 1% as GC systems are not likely to ever achieve perfect recognition rates for the complex flight deck environment.

A “Low Latency” case can also be created, in which the processing of the system is reduced to a maximum of 0.5 seconds, reflecting the behavior of the fastest-processing systems reported in the literature. A third setting combines the two of these cases (Low Latency+Failure). Reducing the failure rate of the system would be expected to have a stronger effect on operations than reductions in processing latency, given the size of the time penalties for failures. However, it is doubtful that changes to these parameters would generate significant differences under the original launch preparation time, given its regulating influence as shown in previous tests.

The three modified Gesture Control models (Low Latency, Low Failure, and Low Latency+Low Failure) were implemented in MASCS and tested at the 22 and 34 aircraft Density settings under both the original and the “automated” launch preparation time models. Thirty replications were performed for each combination of Gesture control model, mission Density, and launch preparation time model. Average Launch event Duration (LD) values these tests appear in Figure 6-15, along with results for the baseline GC and MC models defined previously in Chapter 5 as a comparison. Improvements to the GC system latency and failure rate should improve its performance from that of the baseline GC model and move it closer to that of the baseline MC model.

As with previous results under the original Launch Preparation (LP) model, changes to the GC system parameters have no real effect on operations due to the constraints of the catapult queueing process. Once this queuing is eliminated under the aggressive settings of the automated LP model, LD values improve from left to right. The Low Failure and Low Latency cases are both slightly better than the baseline GC model, with further improvements occurring at the Low Latency+Low Failure case. At both the 22 and 34 aircraft Density levels, the performance of the Low Latency+Low Failure case is close to that of the baseline MC architecture.

Statistical tests show significant differences for the results at both the 22 (Kruskal-Wallis Test, $\chi^2(4) = 75.52, p < 0.0001$) and 34 aircraft (Kruskal-Wallis test, $\chi^2(4) = 111.75, p < 0.0001$) Density

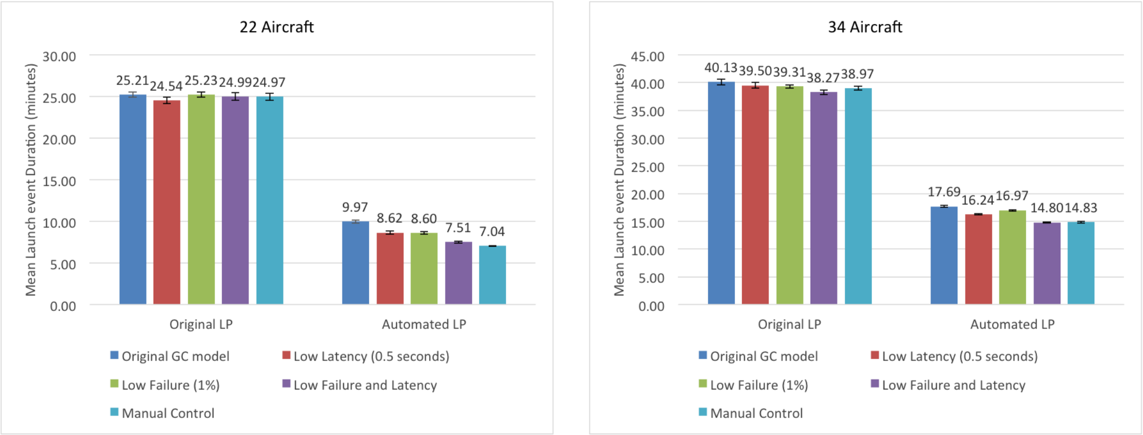


Figure 6-15: Column charts of mean launch event duration for variations of the Gestural Control (GC) model at both 22 (left) and 34 (right) aircraft Density level using the original launch preparation time model.

levels. A Steel-Dwass non-parametric simultaneous comparisons test at the 22 aircraft level reveals that all but one pairwise comparison are statistically significantly different: the Low Failure and Low Latency conditions are not significant different from one another (p = 1.0; all others, $p <$ 0.05). The Manual Control results are significant different from every GC model, including the Low Latency+Low Failure case (p = 0.308). This means that, for 22 aircraft, even significant improvements to the accuracy and processing speeds to GC vehicles are not sufficient to provide performance equivalent to Manual Control operations. However, in practical terms, the average mission times differ by only 0.47 minutes.

Interestingly, at the 34 aircraft level, increased accuracy and processing speeds *are* sufficient to provide equivalent performance: a Steel-Dwass test shows only one pairwise comparison that is not significant: Manual Control and the Low Latency+Low Failure (p = 0.9915). All others are significant at the $p = 0.05$ level (Low Latency and Low Failure, $p = 0.0056$; Low Latency and GC Baseline, $p = 0.0407$; all others, $p < 0.0001$). This again may indicate that the relative open 22 aircraft missions, as opposed to the somewhat constrained 34 aircraft missions, are more sensitive to deficiencies in vehicle taxi behavior. Even though the failures and latencies have been reduced for the GC models, the few failures that do occur, as well as their failures in catapult alignments, may still have substantial effects on performance. The limitations on taxi paths that occur at the 34 aircraft case might serve to dampen some of these effects or to dampen the performance of the MC model.

A final series of tests were taken in which the last major variable in GC, the time penalties due to gesture recognition failures, was reduced from 5-10 seconds to 1-2 seconds. Adding this case into the previous Kruskal-Wallis test, results are again significant ($\chi^2(5) = 109.60, p < 0.0001$) but a Steel-Dwass test now shows the MC case (mean rank = 39.83) is not significantly different from this new "Low Penalty" case (mean rank = 52.17, $p = 0.9930$). This Low Penalty case is also not significantly different from the Low Latency+Low Failure case (mean rank = 66.08, $p = 0.7722$), but the Low Latency+Failure case remains marginally significantly different from Manual Control ($p = 0.0454$).

It is interesting that the 22 aircraft case required additional improvements to GC time penalties to provide equal performance to Manual Control operations, but this is consistent with the results of previous sections that indicated different trends in results between the two Density settings. As noted in the previous discussions, this is likely an indication that the heuristics used by crew and the Deck Handler are not optimal for the aggressive automated launch preparation time model. However, even though the improved GC model is still not equivalent to Manual Control, the benefits to LD times at this automated launch preparation process far outweigh the less than 1 minute difference that remains between GC and MC. Although additional improvement to operations might occur if the crew routing methods were redesigned, investment in the automation of the launch process should become a priority to the Navy.

A final interesting question raised by these results is, for 22 aircraft missions using the lower time penalties and processing times, what accuracy rate is required for the GC architecture in order to generate performance equivalent to Manual Control? To answer this, additional trials were run for the GC model using the Low Latency and Low Penalty settings decreasing the failure rate from the baseline 11.18% to 10%, then in increments of 1%. The results of a subset of these tests appear in Figure 6-16, which also includes data for the baseline Gestural Control (GC) model used in Chapter 5 (fail rate = 11.18%), the GC Low Latency case (fail rate = 11.18%, max process latency = 0.5 seconds), and the baseline Manual Control (MC) model from Chapter 5.

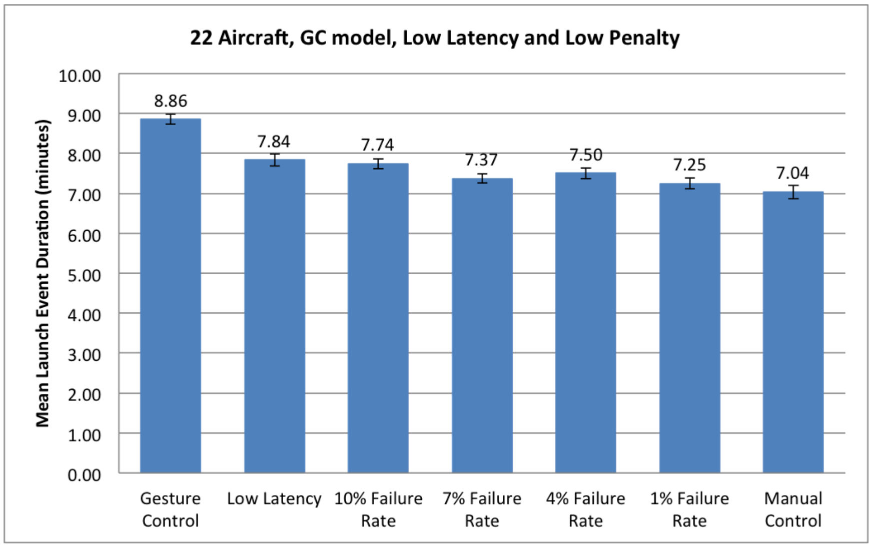


Figure 6-16: Plot of Launch Event Duration Means against Gestural Control (GC) failure rates under the original launch preparation time model. Whiskers indicate ±1 standard error in the data.

First, it can be seen that decreasing the latency provides an immediate benefit to operations; however, decreasing the failure rate has a relatively small effect once this decrease in process latency is applied. Statistical tests do indicate that there are significant differences between cases, however (Kruskal-Wallis test: $\chi^2(6) = 91.6848, p < 0.0001$). The results from a Steel-Dwass post-hoc nonparametric simultaneous comparisons test appear in Table 6.5. These results show that the Manual Control baseline is significantly different from all cases except the 1% and 7% cases. This is interesting, given that it would be expected that the 4% case should perform better than the 7% case. These variations are likely due to small inefficiencies in the Handler heuristics and taxi routing methods that have been described in other results. Although the 4% case is statistically significantly different form Manual Control, the difference is only 0.46 minutes.

This and the preceding sections have addressed how changes to parameters defining the process-level task of launch preparation and parameters related to the unmanned vehicle control architectures affect the performance of the flight deck system as a whole. Throughout these results, it has been pointed out that many remaining causes of inconsistencies in results and variations within the data are likely a function of the structure of operations. This includes both the behavior of the crew and the planning of operations, as well as extending into how aircraft are arranged on the flight deck prior to mission operations and the physical locations of the parking places and catapults. Simulations like MASCS could also be used to examine changes to these elements, although the redefinition of the decision-making rule bases required to do so would require more time and effort than is allowed within the scope of this thesis. Possible options for exploring these architectural changes are discussed in the next section.

6.3 Possible Future Architectural Changes

Much of Chapter 3 discussed current aircraft carrier flight deck operations, describing the organization of the flight deck, the structure of crew tasks and interactions, and the way in which supervisors (the Deck Handler) plan assignments for aircraft. Each of these is a significant feature of operations that required substantial research to model and implement within MASCS. They are also features of operations that are a core part of a crew's on-the-job education; both have grown organically over

Table 6.5: Results of a Steel-Dwass nonparametric simultaneous comparisons test of Gestural Control (GC) failure rate variations.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
MC	GC 1% Failure Rate	-2.5667	4.50925	-0.5692	0.9976
MC	GC 7% Failure Rate	-8.4333	4.50925	-1.87023	0.5003
MC	GC 4% Failure Rate	-15.7667	4.50925	-3.49652	0.0086*
MC	GC 10% Failure Rate	-18.2333	4.50925	-4.04354	0.001*
MC	GC Low Latency	-18.2333	4.50925	-4.04354	0.001*
MC	GC	-29.9667	4.509187	-6.64569	<.0001*

time and are not a formal part of a crew's education about the flight deck. To test fundamental changes to these schema would require significant retraining on the part of crew and may introduce additional risks to operations, as crew are being asked to work within an unfamiliar system in an unfamiliar fashion. However, because models of the current system were effectively constructed in MASCS, changes to these systems can also be modeled and investigated.

Given the budgetary constraints of 21st century operations, reducing the number of crew required for operations is a concern of the Navy. The new generation of *Ford*-class aircraft carriers already aims to reduce the required crew by about 700 people (USNFord, 2014). While the Aircraft Directors on the flight deck form only a small part of this complement of crew, MASCS could be used to explore the effects of reducing the number of Directors utilized in operations. Removing crew from the current "zone coverage" scheme would require a reconfiguration of Directors' areas of responsibility and how they connect to one another. MASCS currently hard-codes these connections into the system, but these could either be converted into a configuration file loaded on start or replaced by an algorithm that could determine connections automatically given the number and location of crew on the deck. These could then be used to examine the effects of the number and placement of crew on the flight deck and its effects on total launch event Duration and crew workload (which has not been a focus of this work). Similar investigations could also be made with the Escort crew, exploring how changes to the number of Escorts utilized, the number of starting locations, and the location of those starting locations affect the efficiency of operations. The configuration of parking locations is also based on past deck crew experience and might also be examined further. However, the parking locations currently utilized are closely tied to the planning heuristics of the Deck Handler, and changing one likely requires changing the architecture of the other. Parking spaces can be easily adjusted in system configuration files, but substantial changes to their organization might require significant changes to the Handler planning process.

The physical layout of the flight deck itself might also be explored: the locations and configuration of the launch catapults and the landing strip could be altered, two changes that would be also impossible to achieve in a real world test setting. The placement of catapults is of particular interest: the limitations in alternating launches within each catapult pair provides a significant constraint on operations. Deconflicting the catapults should be able provide significant benefits to operations. If catapult operations could be effectively parallelized in the current deck layout (aircraft could launch simultaneously without risking collision), parallel operations at the 22 aircraft Density settings return an average launch duration of 16.28 minutes (reduction of 34.81% from baseline) and 28.47 minutes at the 34 aircraft level (a 26.95% reduction from the baseline). These are in line with the results of the launch preparation mean -50% tests (18.86 and 29.16 minutes, respectively) from Section 6.1.1. Combining parallel operations with a reduced launch preparation time mean should combine to generate further advantages and should finally push the system into a state where the

Aircraft Directors and the Handler cannot keep up with the pace of launches.

Other changes involving the location of catapults, as well as the total number on the flight deck, could also be done easily within MASCS. How repositioning the catapults interacts with the current Safety Protocols might also be of interest, and new protocols could be designed to exploit the features of this new deck layout. For instance, the primary issues in performance regarding the Temporal and T+A protocols was in requiring an increased number of forward-aft transitions of aircraft on the flight deck. A rearrangement of parking locations and/or catapults might result in an orientation that does not require these disruptive actions. Extremely radical designs, such as a circular flight deck with catapults arranged around the edges of the deck, could be easily implemented and tested within MASCS. Although, such changes may also require changes to the rest of the ship's physical structure that may alter the shape of the hull and affect the ship's performance in the water.

MASCS might also be employed in exploring the use of alternative sets of Deck Handler heuristics, as it is not clear whether or not these assignment rules are optimal and whether or not they could be improved. Intelligent planning algorithms such as integer linear programs, Markov-based processes, auction-based algorithms, or other methods might provide alternative planning strategies that may be beneficial to operations, or might be able to offload some of the cognitive load of the Handler to the algorithm. The full development, testing, and verification of these algorithms would require a lengthy iterative process, for which the fully automated nature of MASCS would be beneficial. These planning algorithms might also be expected to have significant interactions with the layout of parking spaces on the flight deck, which could be jointly tested using the same optimization architectures as described above. The testing of novel route planning methods might also be conducted, replacing the basic taxi rules of the "zone coverage" system with algorithms such as A* and examining their effects on taxi time, taxi distances, and fuel consumption. Because of the modular nature of the MASCS system, integrating these new path planning and scheduling algorithms would be relatively simple: each could be written as their own independent piece of the codebase and adapted to take in the correct inputs from the MASCS system regarding aircraft and crew locations and provide the same outputs in terms of "tasks" as done in the current code. Within the other agent models of MASCS, simply rerouting the decision-making scripts to employ these new algorithms is a trivial adjustment.

Lastly, in line with the testing of novel planning algorithms, the concept of replanning schedules to recover from failures is also an area at which MASCS could be employed. If, during operations, a substantial mechanical failure or accident occurs on the flight deck, substantial modifications to the original schedule may be required. Such crisis events are also notoriously difficult to test in the real world in a way that replicates the same level of stress generated by real accidents. A model such as MASCS provides an alternative method of investigation.

6.4 Limitations of the MASCS Model

As with any simulation environment, a variety of limitations exist within MASCS. The core issue with any computerized simulation is that it is a system entirely based on mathematics and rules — any element of the real world that is required within the model must be formulated in these terms, as dictated by the specific type of computational model being generated. For an agent-based model specifically, this requires translating agent motion and decision-making into accurate movement models and accurate logical rule systems, with all requisite behavioral (and other) states included. With this comes a desire for simplicity and parsimony — reducing the complexity of the real world as much as possible in order to make modeling feasible. During the development of MASCS, many of the purely random behaviors that can occur in the world — weather, mechanical failures, human behavior outside the expected norm — were ignored in order to focus the analysis on the differences in performance of the different unmanned vehicle models. As such, in its current instantiation, the effects of interactions of these elements with other aspects of MASCS is not known. Additionally, even for the features that are modeled, the boundaries placed on the randomized mathematical distributions encoded in the system and the limited options in decision making processes limit the ability of MASCS to produce truly unique events.

Additionally, the parameterized distributions used within MASCS were defined based on a limited set of data that may not adequately represent the true environment. As best as possible, these parameters were based on empirical data from operations — videos of operations, live observations, and interviews with Subject Matter Experts (SMEs) — but these data were not collected as part of any internal Navy initiatives to study flight deck operations. Validation data, as well, was difficult to obtain and only covered a limited number of aspects of operations. To the author's knowledge, detailed databases containing this information do not exist in any formal capacity. While MASCS has been calibrated with respect to mission-level outputs of launch event duration and launch interdeparture rates, models of crew behavior and Handler planning heuristics have not been fully validated as data is not available regarding these aspects. Until such time as fully detailed data regarding individual crew behaviors and motion, Deck Handler planning, and possibly extending to rare and critical events is obtained, any model of flight deck operations will be difficult to fully validate.

The Control Architecture models integrated into MASCS also relied on a set of assumptions to define models of mature, futuristic control architectures. Because these models address vehicle systems that are not currently in existence, they cannot be formally validated against empirical data. The models may also not be reflective of the actual control architectures built for future operations, in terms of both low-level details of the architectures and in the settings of related parameters. Similarly, the behavior of immature versions of these control architectures when initially fielded could be substantially different from the architectures modeled here. As such, the results are limited to the definitions of the unmanned vehicle control architectures described in this work and the key parameters that define their behavior.

Even more importantly, the role of the crew and their presence in operations had substantial effects on safety and productivity performance, as previously discussed in Chapter 5, Section 5.6. The results and insights gained from the testing that was performed in this dissertation are limited by the definitions of crew behavior and their heuristics, including the topology of the crew zone coverage network. Alternative versions of any of these aspects may produce markedly different results for flight deck operations. Furthermore, the generalizability of these results to other domains is contingent on how similar human collaborator behavior in those environments is to that on the aircraft carrier flight deck.

This also includes the Deck Handler model and the assignment heuristics used to allocate aircraft on the flight deck. These heuristics were described previously in Chapter 3, Section 3.2.3 and were based on earlier work (Ryan, 2011; Ryan et al., 2014). The heuristics encoded into MASCS are meant to be representative of the general behavior of Handlers in the world. While the same set of heuristics was applied to all five UAV control architectures and the MC model in order to provide a consistent comparison, different sets of heuristics may provide different results. In addition, future operations may entail the use of different heuristics for each control architecture or safety protocol that could be formally optimized for each. However, previous research demonstrated that designing effective algorithms for this environment is difficult and may provide results no better than the human heuristics (Ryan et al., 2014).

Other structural aspects of the flight deck, in terms of the geometry of the space, initial vehicle locations, and vehicle destinations may also differ for other environments. On the flight deck, vehicles move only in the two-dimensional plane of the flight deck in what is technically an open space in which vehicles can transit in any unobstructed direction. Crew also generally attempt to move aircraft in the same direction in order to maintain an orderly flow of traffic, but this is not a hard constraint like it is for highways and other roadways. These aspects all influenced the definition of the safety protocols, and for other domains that do involve vertical motion (such as military aviation) or that have more or different restrictions on transit paths (highways, roadways in mines), the safety protocols defined here may not be entirely applicable and the results observed in MASCS testing may not apply. However, the models of unmanned vehicles used in the MASCS model should be easily transferable to other domains, and the general insight as to the role of crew in operations should also be true for other environments.

The MASCS model also includes only a limited number of tasks in the system model, all of which concern the movement of aircraft from parking locations to the launch catapults. Other environments may contain a richer set of actions that would be required in modeling, providing additional opportunities for control architecture failures, latencies, or other interactions between elements. However, it would be expected that the general features of the results described in Chapters 5 and 6, such as the effects of process level constraints and the ability of safety protocols to disrupt the flow of traffic in operations, would likely hold in some form. At minimum, the development of the current MASCS model of flight deck operations does provide a roadmap for the development and testing of simulations for other environments.

An additional operational limitation in MASCS is the time to complete individual runs. Even at ten times normal speed, a set of thirty replications takes between 60 and 180 minutes to complete, conditional on the specific settings of the mission profile. Because of the time to complete runs and the significant standard deviation in results, using MASCS within an optimization architecture would likely require many more runs to complete its exploration of a design space. Accelerating the speed of operations of MASCS would be a viable option, but the current architecture of the system means that the resolution of vehicle motion would decrease and a significant amount of information would be lost. Some other mechanism of updating the motions and actions of vehicles would be required in order to provide the performance necessary to facilitate deeper investigations into system parameters and optimality.

6.5 Chapter Summary

This chapter has explored some of the implications from the results displayed in Chapter 5, utilizing the MASCS simulation environment to explore the effects of changing both vehicle-independent, process-level tasks and UAV control architecture parameters on safety and productivity. Chapter 4 previously demonstrated that the MASCS model provides results in agreement with prior published data on and SME estimates of operations. This chapter has demonstrated that modifying the parameters of the launch preparation process eliminates the bottleneck it places on operations and greatly improves flight deck performance. In making such changes, MASCS has demonstrated that 22 aircraft mission durations can be reduced from 20 minutes to less than 10 minutes with an aggressive, automated launch preparation task. Furthermore, these results also indicate that at this faster operating tempo, the safety of aircraft interactions becomes robust to the selection of safety protocols and the total duration of crew interactions drops substantially.

Introducing this aggressive automated process also resulted in significant differences in Launch event Durations for the different control architectures, although these again ended to cluster into groups of “delayed” and non-delayed architectures. However, these results also suggest that at these extreme settings, the density of operations and its interactions with the scheduling of tasks and the taxi routing methods have significant effects on Launch event Duration values. The relative effects of increasing mission Density from 22 to 34 aircraft was much larger under the automated LP model than under the original, indicating that the heuristics used in these processes, which grew organically over time under much different operating conditions, are not well-constructed for an accelerated launch rate.

A review of safety metrics also demonstrated that increasing the speed of operations may not necessarily imply a reduction in safety. The automated launch preparation process does moderately decrease the number of aircraft safety incursions and greatly decrease the total duration of crew safety incursions but has no effect on the total number of crew incursions. Thus, in terms of two metrics, accelerating the pace of operations is a net benefit to safety, although this also implies the *rate* of crew safety incursions increases drastically. Additionally, variations in measures across safety protocols also appears to be lessened, implying that the safety protocols provide little benefit at this increase operational tempo.

The ramifications of the results in this and the previous chapter demonstrate the importance of systems like MASCS. Although the SBSC and GC systems are still years away from practical development, the results of this and the previous chapter have provided some understanding of the important features of their performance, how they might be integrated into operations, and what features of operations are especially interactive with these systems. MASCS has been able to explore what might be possible with candidate versions of these systems. Most generally, the results of this and the previous chapter show that the best performance in flight deck operations comes from either fully manual or fully autonomous operations: under either the MC or the SBSC architectures. The control architectures that lie between these models in the automation spectrum provide some form of detrimental performance due both to the parameters defining their behavior and the architecture of their interactions with the system.

To collect the data used to reach these conclusions in real-world field testing would consume an incredible amount of time and resources, and the instrumentation of data collection for the metrics defined here might also be problematic and difficult to achieve. By creating independent agent-based models that replicate the expected behavior of the different unmanned vehicle models, the MASCS model has afforded a better understanding of flight deck operations than might be achieved in years' worth of field trials. Overall, MASCS has generated not only a better understanding of how the flight deck works, but provided guidance on potential future research as to the choice of UAV systems to incorporate into operations and how they interact with candidate safety protocols for the environment.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

Conclusions

> "Robots are emotionless, so they don't get upset if their buddy is killed, they don't commit crimes of rage and revenge. But they see an 80-year-old grandmother in a wheelchair the same way they see a T80 tank; they're both just a series of zeros and ones."
>
> *"Military robots and the future of war,"*
> *TED Talk, Feb. 2009*
> Peter Singer

As unmanned and autonomous vehicle systems move from environments in which they operate primarily in isolation into Heterogeneous Manned-Unmanned Environments (HMUEs) that require close interaction with human collaborators, properly understanding which characteristics of the domain encourage safe and effective performance in these new domains is paramount. For a vehicle operating in isolation, it important to know the location of the vehicle, its current destination, and what transit path will move them to that point without overstressing the vehicle. For a vehicle operating in an HMUE, the same issues are also at play; however, the transit path, speeds of motions, and even the selection of the destination are influenced heavily by what types of actors are nearby. The decision-making rules and behaviors that provide desirable performance for a manned or unmanned vehicle operating on its own are not the same as the characteristics desirable for vehicles operating in HMUEs.

Understanding which characteristics are desirable is further complicated by the large scale of systems like the United States highway system. In most current practice, determining these characteristics occurs through a trial-and-error, one-factor-at-a-time style of testing: small adjustments are made to a system in the field that is then tested over a period of time. This process, and its reliance on physical systems operating in the world, brings with it several deficiencies and limitations: cost, safety concerns, limits on what can feasibly be tested with the current available hardware and software systems, and limitations on what can accurately be measured in the field. To this end, the Multi-Agent Safety and Control Simulation (MASCS) agent-based simulation environment was developed in order to facilitate the exploration of manned and unmanned vehicle behavior in one candidate HMUE, the aircraft carrier flight deck. This chapter reviews the motivation behind and the development and testing of the MASCS simulation model, as well as the contributions of this research and the questions that remain for future investigation.

7.1 Modeling HMUEs

7.1.1 The MASCS Model

The Multi-Agent Safety and Control Simulation (MASCS) model was constructed with the goal of modeling manned and unmanned vehicles acting within a candidate Heterogeneous Manned-Unmanned Environment (HMUE), such that the performance of these vehicles could be evaluated under realistic circumstances. This includes the effects not only of different vehicle control methods, but the effects of changes in how they are integrated into operations and the mission conditions under which they operate. These first two aspects, referred to as Control Architectures (CAs) and Safety Protocols (SPs), respectively, served as the primary focus of MASCS development and testing. A review of the existing research showed a spectrum of five different unmanned vehicle control architectures that were candidates for inclusion within MASCS, alongside a model of Manual Control that serves as a baseline comparison point. Unmanned vehicle control architectures ranged from Local Teleoperation (LT) and Remote Teleoperation (RT), which still require a remote human pilot, to Gestural Control (GC) (and other forms of advanced teleoperation) that require no pilot but do require instructions from the crew, to Vehicle-Based Human Supervisory Control (VBSC) and System-Based Human Supervisory Control (SBSC) that receive their instructions through a centralized control system staffed by a senior supervisor (the Deck Handler). Slight variations in the location of the human operator, the method of commanding the unmanned vehicle, and the types of commands provided differentiate these Unmanned Aerial Vehicle (UAV) CA from one another. These seemingly slight differences in architectures, however, are sufficient to generate significant differences in performance for individual vehicles.

An initial framework that mapped the general architecture of these systems in terms of the communication of information between human supervisors, manned and unmanned vehicles, and other human collaborators served as the foundation for the construction of the MASCS environment. This framework was then applied to aircraft carrier flight deck operations, chosen as an example HMUE based on its existence as a heterogeneous *manned* environment and the current research concerning the development and integration of unmanned vehicles into flight deck operations. A model of current manual flight operations was developed and partially validated based on data obtained from previous research, observations of operations (both in-person and recorded), and interviews with Subject Matter Experts (SMEs). This model then served as the template for the implementation of unmanned vehicles into flight deck operations.

However, the introduction of unmanned vehicles might also entail changes to the structure of operations in order to limit interactions between vehicles and other human collaborators. Past experience has demonstrated that robotic, unmanned, and autonomous systems are not often given free rein to operate in a system and are often isolated based on geographic area or in terms of temporal scheduling. These forms of Safety Protocols suggest a few simple ways in which the interactions of manned and unmanned vehicles can be limited to increase the safety of flight deck operations. However, the true effectiveness of these protocols and their interactions with system productivity are not readily known. Despite the ongoing development of unmanned vehicle systems for flight deck operations, significant questions remain as to how UAVs will be inserted into the mission planning and mission execution processes.

To incorporate both the variations between CAs and the different types of SPs, MASCS was constructed as an agent-based model. Such models rely on the definition of independent "agents" that replicate the decision-making behavior, task execution, and movement of the individual actors in the real-world system. Differences in CA can be mapped to differences in decision-making rules and parameters describing individual vehicle and crew behaviors, while changes in the SP are characterized by changes in the supervisory Deck Handler's planning activities. The modular design of the agents enabled these difference to easily be integrated into the simulation model. These features also are the primary inputs to the MASCS model and are discussed in the next section, along with the outputs of the simulation.

Model Inputs and Outputs

Chapter 3 detailed the creation of the original MASCS model of flight deck operations and the sources of data used in its construction. Constructing an agent-based model of flight deck operations first required the definition of (1) *agents* active within the environment, followed by the definition of (2) *states* that described the current conditions of the agent, (3) *decision-making rules* that define what actions should be taken to achieve its current goals, and (4) *parameters* that defined how those actions are performed. Each of these could be tailored by future researchers to explore new control architectures or new parameters for the currently defined architectures. Similar structures were also provided for independent task models, as well as for the model of the physical environment (the carrier flight deck).

- *Agents*: Creating MASCS requiring defining individual agents for all aspects of operations.

This included pilot/aircraft agents, crew on the flight deck, supervisory planning agents, equipment such as catapults, and the flight deck itself. Additional models were constructed for tasks executed by these agents, constructed in a modular fashion to make them easily adaptable to a variety of situations. For each of these agents and models, sets of states, parameters, and decision rules were defined individually based on prior observations of operations, a review of training manuals, and interviews with SMEs.

- Agent *states* describe the current characteristics of an agent in the world, including its current position, heading, assignments, and health. States describe the important characteristics of agents relevant both for their own activities and for the activities of other agents in the system.
- Agent *decision rules* describe how, given information on the current state of the agent and the observable states of other agents and the environment, how it should achieve its goals. These rules determine what the goal should be, then determine what tasks are required to reach that goal, then whether or not those tasks can be accomplished. Important decision rule states included modeling how Aircraft Directors move to place themselves within an aircraft's field of view, how Directors pass aircraft between one another in the "zone coverage" routing system, and how Handler's plan catapult assignments for aircraft in the system. Each of these decision rules were instrumental both in the design of the MASCS system and shown to have significant effects on system behavior. These are each system characteristics that might be altered under future investigations.
- Agent *parameters* define the characteristics of task performance and are often modeled as parameterized random distributions. The definitions of these parameters, along with changes to decision rules, define the differences between the various UAV control architectures modeled in MASCS. These include the speed at which aircraft can travel on the flight deck, time distributions characterizing the time required to complete certain tasks, and the failure rates of task execution. Failure rates and system processing latencies (caused by a variety of different factors) are also included here. Parameters for Manual Control operations were defined based on empirical evidence and SME advice, while UAV parameters were based on a combination of prior research and assumptions of how mature system technologies would behave. Less-mature systems, or other types of control architecture not considered here, may provide very different results than what was observed in this testing program and are a point of future investigation.

- *Task models*: Models of agent task execution were also required, consisting of decision-making rules, states, and parameters that describe under what conditions a task can be performed as well as how that task is executed when conditions permit. Of primary importance were models of agent motion in the world, which rely on interactions with agent states, the Handler assignment decision-making rule, and the Aircraft Director "zone coverage" routing system. Two other important tasks, the launch preparation time and acceleration time, are vehicle-independent are based on stochastic time distributions based on real-world observations. This models are modular in nature, defined independently of any agents in the system. As such, new models task models, or variations to current models, could also be introduced. Current models could also be adapted to new environments outside that of carrier flight deck operations.

- *Mission and Deck Configurations*: The characteristics of the mission to be executed and the arrangement of the flight deck also serve as important factors in the system. The number of aircraft included in the mission, mission Density, has fundamental effects on the taxi routes and other actions available to agents in the system. The percentage of unmanned vehicles used in the mission, mission Composition, affects how the Safety Protocols are executed. Additionally, the locations of aircraft on the flight deck and the physical layout of the deck itself form additional variables that were not explored in this work.
- *Metrics*: Analyzing operations required the definition of metrics in several categories addressing the productivity (expediency and efficiency) and safety of operations. The metrics provide a breadth of perspectives that enabled a detailed analysis of differences in behavior of Safety Protocols, Control Architectures, and their interactions with different mission configurations (mission Density and Composition).

Each of the elements described above could be altered by future researchers exploring different facets of operations. Additionally, the models themselves could be fundamentally altered: as the current focus of MASCS is on futuristic vehicle technologies using experienced operators, interactions between human and aircraft were treated as settings for a *combined* human-vehicle system. These models could be further differentiated, creating separate models for humans pilots/operators and vehicles in order to explore interactions between the two at a much finer level of detail. If these models, which would more explicitly explore human cognition and decision-making, were implemented, many of the task execution and failure rate models defined in this work would become the outputs of these new, more detailed models. Implementing more detailed human cognitive models might also require substantial changes to the decision-making rules in order to adapt to this new architecture. This modularity and the ability to easily introduce new agent models is one of the benefits of MASCS, which are discussed further in the next section.

Model Benefits

MASCS is, as far as it is known, the first simulation environment that models a variety of candidate unmanned vehicle control architectures in a partially validated model of a real-world HMUE. This has afforded an ability to examine features of unmanned vehicle performance and interaction with operational settings that has not been observed in other arenas. Due to the modular definition of agents and tasks within MASCS, the characteristics of individual agents, how they execute tasks, how they interact with one another, the physical organization of the flight deck, initial positions of aircraft and equipment, and the planning methods of the Deck Handler and other staff can all be investigated individually, without requiring modifications to other agents and decision-making routines. This framework also allows for the easy insertion of models of the different UAV CAs into the system, replacing manned aircraft with models of unmanned vehicle behavior and enabling comparison of performance across cases with requiring the construction of an entirely new model of the flight deck. Future models of unmanned aircraft could easily be generated and integrated with the current MASCS model, along with new operational settings and conditions.

The ability to make these changes individually enabled the creation of a test program that systematically varied the control architectures, safety protocols, and mission settings (both number of aircraft and percentage of UAVs included in the mission) in order to characterize differences in performance across cases as well as interactions between cases. This testing program and its results were examined in detail in Chapter 5, providing significant insight into potential outcomes of unmanned vehicle integration into flight deck operations. The most important lessons learned concern the role of the crew in affecting operations. The use of human crew in routing aircraft through the flight deck (the "zone coverage" system and its topology) was shown to have a significant effect on the results of safety measures in those tests. Additionally, the role of crew in the vehicle-independent launch preparation process was shown to serve a regulatory function on operations, limiting variations in the productivity of control architectures in terms of launch event duration.

Once these limitations in the launch process were removed after modeling an automated launch preparation process, some key aspects of productivity on the flight deck were observed. The unconstrained and highly autonomous SBSC architecture provided some advantages in operations over UAV control architectures experiencing latencies in operations (RT, GC, and VBSC). The architectures experiencing latencies in operations also performed most poorly when combined with the Temporal separation protocol, which itself provided poor results in general. The poor performance of the Temporal protocol suggests that for Deck Handlers scheduling flight deck operations, maintaining flexibility in aircraft assignments (Temporal Separation) is less important than in maintaining a consistent flow of traffic from aft to forward (as occurs in Area Separation). Ultimately, for UAV control architectures to have any benefits past that of the current Manual Control models, both the flow of traffic must be preserved and the launch preparation time must be improved: otherwise, the effects of these two elements greatly overshadow any differences in unmanned vehicle operations.

In terms of the safety of operations, significant interactions between control architectures, safety protocols, and mission settings were also observed. Measures of safety that examined the distances between aircraft (the Tertiary Halo Incursions due to Aircraft (THIA) measure) showed the best performance under the combined Temporal+Area protocol. However, improved performance on this measure came at the cost of mission expediency, as the heavy constraints of this protocol limited the ability to launch aircraft from the flight deck. While these results apply specifically to the design of modern flight deck operations, it does suggest a need to carefully question the organization and conduct of operations in other domains as they relate to and interact with their own unmanned vehicle systems.

The selection of control architecture also had interesting interactions with these safety measures, primarily due to the nature in which vehicles are routed on the flight deck. Under the THIA metric measuring the safety of aircraft interactions with one another, the SBSC systems provided both some of the very best and some of the very worst performance, depending on level of constraints applied by the safety protocols. This also has repercussions for real-world operations: the MASCS model did not assume that failures could occur in the collision prevention systems for any of the UAV control architectures. These results suggests that, for SBSC system reliant solely on their own autonomy to avoid accidents, imperfect compliance with collision prevention may make these some of the least safe systems to operate.

In general, MASCS has provided both an increased understanding of the dynamics of flight deck operations, the key features that govern mission productivity and efficiency, and pointed the way towards a better understanding of which characteristics of unmanned vehicles and of safety protocols generate desirable safety and productivity in this domain. The nature of vehicle interactions on the flight deck are similar to those in highway driving and mining environments, and the results described for SBSC here could extend to these domains in a limited fashion. Mining environments are similar with the exception that vehicles travel cyclically between different sites within the mine. Roadway driving is also similar, although the initial starting locations and destinations of aircraft are all dissimilar; the models of unmanned vehicle control architectures, however, should be similar to flight deck operations. The modularity of MASCS and its agent and task models means that it could easily be extended into models of different physical environments in order to verify exactly how well these results transfer into these new domains.

7.1.2 Model Confidence

Chapter 4 presented information regarding the calibration and partial validation of the MASCS model against data on current manual control flight deck operations, examining the performance at both the individual vehicle and mission levels, as well as sensitivity analyses of performance at both levels. These sensitivity analyses address the most important aspects of flight deck operations, namely vehicle speed, collision avoidance parameters, and launch preparation time. These tests examined the accuracy of the model with respect to replicating current operations as well as the stability and robustness of the model given changes in the task execution parameters.

Model Accuracy

Validation testing for the MASCS model of current aircraft carrier operations relied on a combination of reported performance data from Naval research institutions, video observations of operations, and subjective evaluations by SMEs. The model of individual manual control aircraft behavior was calibrated with recorded footage of an individual aircraft followed from initial parking location through the completion of the launch process. This video was catalogued, breaking the entire launch event into subtasks, each of which formed a comparison point for the simulation data. A replica scenario was developed and executed within MASCS, demonstrating that the simulation was capable of reproducing the video data within ±1 standard deviation on each metric.

More detailed data on mission performance was obtained from prior Navy studies. This data reported the Cumulative Density Function (CDF) of launch interdeparture times on the flight deck and expected values for the time to complete a mission of given size. A series of representative missions were replicated within MASCS, demonstrating that the simulation was capable of providing similar distributions of interdeparture times and total mission times in line with prior expectations and reports. The results of individual vehicle testing suggested that the models of individual aircraft motion and task execution were reasonable, while tests at the mission level demonstrated that the architecture of flight deck operations, including the assignment of aircraft to catapults and the routing of aircraft to those assignments, were also reasonable representations of the real system. However, there are a variety of aspects of operations that have not been formally validated, with individual crew behavior being the most prominent. While the model has been calibrated using data on current mission operations, new data may reveal significant flaws in the model.

Models of UAV behavior were based on prior research evidence and supplemented with assumptions about realistic performance for mature systems utilizing experienced operators. While the performance of these vehicles cannot be validated in any formal sense as they do not yet exist, tests of individual vehicles (replicating the individual vehicle mission used in validation) provides results in line with expectations: an inverted-U shape. Moving from LT to SBSC across the spectrum of automation complexity, the GC architecture provides the worst performance due to its delays in processing and failures in recognizing Directors and task instructions before improving. At the ends, the LT and SBSC systems provide the best performance and were not shown to be significantly different from Manual Control (MC). The remaining two cases of RT and VBSC were shown to be significantly worse than MC but still better than GC.

Model Robustness

Additional tests in MASCS examined simulation behavior given changes in internal and input parameters within the model. These tests specifically addressed the reaction of the simulation to changes in the number of aircraft and the number of catapults (input parameters) as well as changes in the taxi speed of aircraft and the related collision prevention parameters. Changes in the input parameters lead to substantial changes in how aircraft are prioritized and routed on the flight deck, while changes to internal parameter affect the execution of taxi operations upstream of the launch process. The tests conducted in Chapter 4 verified that the response of the flight deck system was as expected: linear with respect to the number of aircraft used in the mission, a slight difference (1-2 launches) between using 3 and 4 catapults, and that responses of the system to changes to taxi and motion behaviors, such as taxi speed and collision avoidance parameters, did not generate extreme variations in operations. Because these two parameters affect tasks upstream of the launch preparation task, their effects on mission performance are ultimately limited: differences in performance only occur if the disruptions to taxi operations break the maintenance of queues at catapults. The effects of varying the mean value of the launch preparation time model — a vehicle-independent task that occurs on the flight deck — was shown to have much stronger effect than either of these other parameters, as would be expected.

These results demonstrated that the MASCS model is capable of replicating the performance of the real world system, including its responses to changes in both internal and input parameters. This, in turn, signifies that models of Deck Handler planning heuristics and Aircraft Director traffic routing on the flight deck are both reasonable representations of the real system, including possessing the flexibility to handle different mission definitions. Establishing these decision-making rules was a complicated aspect of generating the agent models, and the work described in this dissertation provides a template for future researchers generating agent-based models of similar domains. However, as noted in Chapter 6, Section 6.4, one of the key limitations in MASCS development was the limited amount of data available on operations. If future data collection programs demonstrate that the data used in validating MASCS and in defining MASCS agent parameters was incorrect, MASCS should no longer be considered a valid model of operations. Input parameters for MASCS would have to be updated and the entire simulation revalidated to ensure its accuracy.

7.1.3 Model Applications

In general, MASCS is a complicated simulation designed for “what-if” scenario testing for aircraft carrier flight deck operations, designed to be flexible enough to test not only changes to the behavior and performance of individual agents within the system but also to test changes to the execution of tasks, in communication between agents, and in the methods of planning in the environment. Chapter 5 presented one way in which the MASCS environment can be used, testing different candidate UAV control architectures and safety protocols intended for use in these operations in order to examine whether or not differences exist in the performance of these systems and, if so, the magnitude of these differences and the sources of their variation. Chapter 6 explored another way in which MASCS can be utilized: as a design space exploration tool used to examine the effects of changing parameters for process-level tasks (the launch preparation time) and the control architectures on safety and productivity in the flight deck. A first series of tests explored altering the mean and the standard deviation of the launch preparation time, while a second series explored improvements to parameters for the GC architecture.

These results demonstrated that, for systems like the flight deck, where all vehicles head to one of many identical destination states and perform the same final terminal process, the characteristics of that process may ultimately affect judgments of the performance of the modeled systems. As shown in Chapter 6, at high mean, high standard deviation launch preparation times, no difference was observed between control architectures. If a stakeholder used these settings solely as their frame of reference, they would conclude that the use of unmanned vehicles not only had no impact on operations, but that it did not matter what control architecture they selected. Under a lower mean, lower standard deviation preparation time model, significant differences did arise between models, leading to very different conclusions about which UAV systems were most effective. However, in absolute terms, the differences in performance between cases was quite small.

Additional tests explored what improvements to control architecture parameters would be required if differences in performance did exist. Chapter 6 examined in depth the two most important parameters of the Gestural Control architecture, exploring how improvements in the processing latency and accuracy of these systems do or do not affect operations. This exploration lead to several interesting conclusions about the performance of these systems. Notably, under operations using the original launch preparation time model with large mean and standard deviation, significant improvements in these parameters had no further beneficial effect on operations: GC performance remained equivalent to MC performance due to the regulating effects of the launch preparation time. Only once this task was decreased were changes in GC parameter shown to have any effect. To reach performance equivalent to MC operations required substantial improvements to the GC parameters, but the practical differences between large and moderate improvements are fairly small.

Other tests examined the effects of changes to these processing on the performance of the different Safety Protocols. Here, other key aspects to the determination of "safety" in these environments was explored. Most intriguingly, results suggested that increasing the tempo of operations by reducing the launch preparation mean and standard deviation might actually be *beneficial* to operations overall, as it minimized the number of times aircraft come within a close distance of one another. This increased tempo also decreases the total amount of time in which crew are placed in close range to aircraft, but does not affect the total number of times this occurs. This implies an overall benefit to the safety of other aircraft on the flight deck, but an increased rate of risk generation for crew; as described above, balancing these considerations and understanding which are the most important are a key concern for stakeholders making decisions for these systems in the future.

Each of these explorations would have taken a substantial amount of time and money to conduct under real-world conditions. The ability of MASCS to quickly modify the conditions of the system, execute a series of tests (at typically ten time normal speed) then compile and review data provide an investigative tool not available to most current stakeholders. Additionally, the nature of MASCS provides the ability to test conditions that do not currently exist in the world: the effects of major improvement to system technology can be done without issue in such a simulation environment. As such, MASCS affords an ability to investigate system settings that do not currently exist and may not in the near term be achievable. This is one potential area of future work discussed in the next section.

7.2 Future Work

MASCS is, and was designed to be, a fairly simple agent-based model of the world. Each of every one of the logical decision-making rules utilized within it is based on if/then statements and other simple heuristic-based solutions. These models were based on a relatively limited amount of data on operations as compared to other environments. A significant source of future work in MASCS can address improving the complexity of the agent models, which in turn requires increased data collection and complexity of data collection flight deck operations, crew behavior, and unmanned vehicle performance.

As noted previously in Section 7.1.1, the pilot and aircraft models were treated as single entities designed to be representative of mature technologies with experienced operators. By definition, this negates any potential examination of the implementation of UAV models with novice operators without properly calibrated trust in the systems. This also negates investigating any effects of change in trust or improvement of skills over time, as well as eliminating the ability to examine how different human operator behaviors affect the performance of the systems. In order to create such models, the pilot/aircraft models must be further differentiated into two independent agents that interact with one another, potentially in the style of systems such as SOAR (see (Laird, 2008)) and ACT-R (see (Anderson & Schunn, 2000)). This requires a much greater level of detail in both human cognitive processes (memory, attention, decision-making and reaction times) as well as how these processes interact with the vehicles themselves. A reasonable first step in this would be a series of studies of naval aviators, understanding how they process information and execute tasks on the flight deck, followed by similar exercises with individuals using prototype or simulated UAV systems. Developing a model for any one of the control architectures alone may be considered a thesis-level research topic, and any single model could be incorporated as an extension of MASCS.

Similar exercises could occur for the Aircraft Directors and Deck Handler, each of whom had a single model of behavior developed for this work. While the models were shown to be flexible enough to handle a variety of operating conditions on the flight deck, other alternative models may also be possible. Errors on the part of these individuals were also largely eliminated from the model, and it may be of significant interest to characterize their effects. However, this again requires extensive additional data collection looking at individual behavior and the creation of more detailed models of cognitive processes, decision making, and errors. However, the generation of such a system would be of extraordinary assistance in terms of diagnostic evaluations of flight deck operations: the effects of an individual mistake could be demonstrated in simulation by simulating the mission both with and without the error, examining the differences in mission evolution and performance that results. This has significant potential as a training tool for new Directors and Deck Handlers, with the possibility of integrating the system as a real-time assessment tool onboard carriers. In this sense, the simulation could be converted into a decision-support system, allowing operators to gauge the effects of their decisions in real time and make appropriate adjustments.

Additionally, as noted in Chapter 5, the models of close-proximity crew interactions with vehicles were very simple. While the results of crew safety metrics were tracked and analyzed, the simplicity of the crew models precludes any real judgments about the safety of crew interactions with aircraft. To provide a true understanding of crew interactions with aircraft, more complex models that more effectively capture how crew interact with aircraft on the order of inches and feet would be required; the current iteration of MASCS does not address human or vehicle motion at that fidelity. However, the construction of such models of crew would provide benefits to both current manual and future unmanned operations, providing an understanding of the key behaviors of aircraft and pilots that place crew in danger (and vice versa). Such information could then be used to either improve crew and pilot training or improve the heuristics of vehicle routing on the flight deck.

Along a similar vein, a point of ongoing investigation is the development of planning algorithms for use in flight deck operations, as well as research into data mining and characterizing the key features of operations. It would be relatively easy to convert MASCS into a data generator for use in this research, replacing the Deck Handler with an automated scheduling algorithm or replacing the “zone coverage” crew routing with alternative path planning algorithms. Additional changes could introduce new metrics and measurement methods into the system for use in the data mining of operational performance. However, in order to transition the system into a true research tool usable by individuals outside the program designer, an automated scenario generation tool would aid in the creation of initial configuration files for the system and automated generation of output statistics across trials would also be useful. Making such changes would also facilitate the merging of the MASCS codebase with optimization toolkits such that would enable a broader exploration of the design space. Wrapping the MASCS system into a genetic algorithm or particle swarm optimization routine, which would have the authority to change the numbers and starting locations of aircraft, parameters within those aircraft, and possibly select different types of planning algorithms or heuristics would facilitate the search for a true “optimum” in the system. These searches could also be conducted for aspects of flight deck operations outside of the launch process, which are described in the next sections.

7.2.1 Additional Phases of Operations

Additionally, with budget concerns driving a desire to reduce staffing levels, simulations like MASCS could be used to examine how changes in the number of crew involved in flight deck operations drives efficiency. While the current instantiation of MASCS only utilized Aircraft Directors, numerous other personnel support operations prior to and after the launch phase. The development of MASCS focused on taxi operations prior to launch because this is the period of time on deck in which the behavior of UAVs would have the greatest effect. Prior to and after taxi operations, a variety of other processes that rely heavily on crew take place: fueling, loading weapons, landing the previous mission's aircraft, and the process of "respotting" the deck prior to the next series of launches. However, most of these tasks do not require much activity on the part of the aircraft and are instead rely mostly on the actions of crew. Aspects of these operations do, however, influence the launch mission and relate to an optimization of the placement of aircraft within it, which may be of interest to other related domains.

Recovery Operations

A first significant step would be the inclusion of the landing process into operations. The Center for Naval Analyses (CNA) report that provided guidance on the launch interdepature rates (Jewell, 1998) also provide guidance on the landing interarrival rates. Landings on the aircraft carrier flight deck proceed to the single landing strip, with landings occurring at a median of 57.4 seconds. After this, aircraft are immediately taxied out of the landing area and parked in areas forward of the landing strip, typically beginning with the parking spaces on catapults 1 and 2. Doing so ensures that the next aircraft coming in to land has no obstacles that would force it to abort its landing — a key issue when aircraft are potentially running low on fuel. The poor performance of RT, GC, and VBSC aircraft as demonstrated may pose a significant issue for these operations; extending MASCS to include these operations might be of significant use in examining the combat readiness of the different UAV types.

With the inclusion of recovery operations, an additional, more complicated type of operations referred to as "flex deck" operations could be examined. In this rather rare condition (typically occurring only during combat operations), the cyclic series of launch events is abandoned. Once an aircraft lands, it is immediately taxied out of the landing strip, refueled, weapons reloaded, and launched again as fast as possible. SMEs report that this puts tremendous stress on the crew to maintain order and adequately manage the flight deck. At such extremely high tempo operations, the "delayed" UAV systems may pose a serious problem in maintaining the required rate of launches and be a detriment to conducting this form of combat operations. To the knowledge of the author, very little research as to this type of operations has been conducted, perhaps due to the rarity of this type of operations and the stress placed on crew during its implementation. Extending MASCS to incorporate these types of mission affords a completely new ability to review performance at these unique operational conditions and may reveal important information about their utility in comparison to more standard cyclic operations.

Respotting Operations

As noted above, after aircraft land they immediately taxi out of the landing strip to the forward area of deck to clear space for the next aircraft arriving in the pattern. After all aircraft have landed, the deck is reconfigured by the Deck Handler and his staff to set up for the next set of launch operations, moving aircraft from their parking spots forward of the landing strip and reorienting them better position them for taxiing to catapults (the initial conditions under which the current iteration of MASCS operates). This layout is a form of spatial optimization — how to arrange a set of aircraft so as to provide open access paths and robustness schedule execution — that, while done quite well by Deck Handlers, is not well understood by other stakeholders. As MASCS already contains models of vehicle size and shape and the ability to execute a given launch schedule, additional modifications could be introduced to replicate both the process of vehicle relocation during respotting operations, as well as provide tools to analyze to efficiency of the new parking arrangement. Consider a model like MASCS existing within a tabletop configuration much like the "Ouija board" (Figure 7-1) currently used by Deck Handlers, in which the agents within MASCS could be dragged and dropped over the interface; a Handler could interact with the MASCS simulation much in the same fashion as he currently does with the Ouija board, but now have the ability to analyze the effectiveness of his current arrangement of aircraft, identify potential pitfalls in the schedule, and adjust for them accordingly.

7.2.2 Deck Configuration

Current use of MASCS in testing purposefully used static templates of aircraft initial parking locations in an attempt to maintain consistency across conditions. In reality, there are a multitude of possible parking arrangements that might be utilized by Deck Handlers, and testing the effects of these arrangements falls easily within the capabilities of the MASCS environment. This could be accomplished through the input of new configuration files, or through the use of the randomized assignment generator described in the beginning of this Section 7.2. Additionally, radical reconfigurations of the flight deck, including redesigns for future aircraft carriers, could also be explored within MASCS. However, such a major alteration to the system would require not only redesigning the physical geometry of the flight deck but also rebuilding the logic behind the Handler's decision-making rules for assigning aircraft to catapults as well as the zone coverage method of routing aircraft amongst crew. Neither of these latter actions may be trivial, although intermediate versions of a redesigned MASCS could be used as an aid to work with current operators to understand how

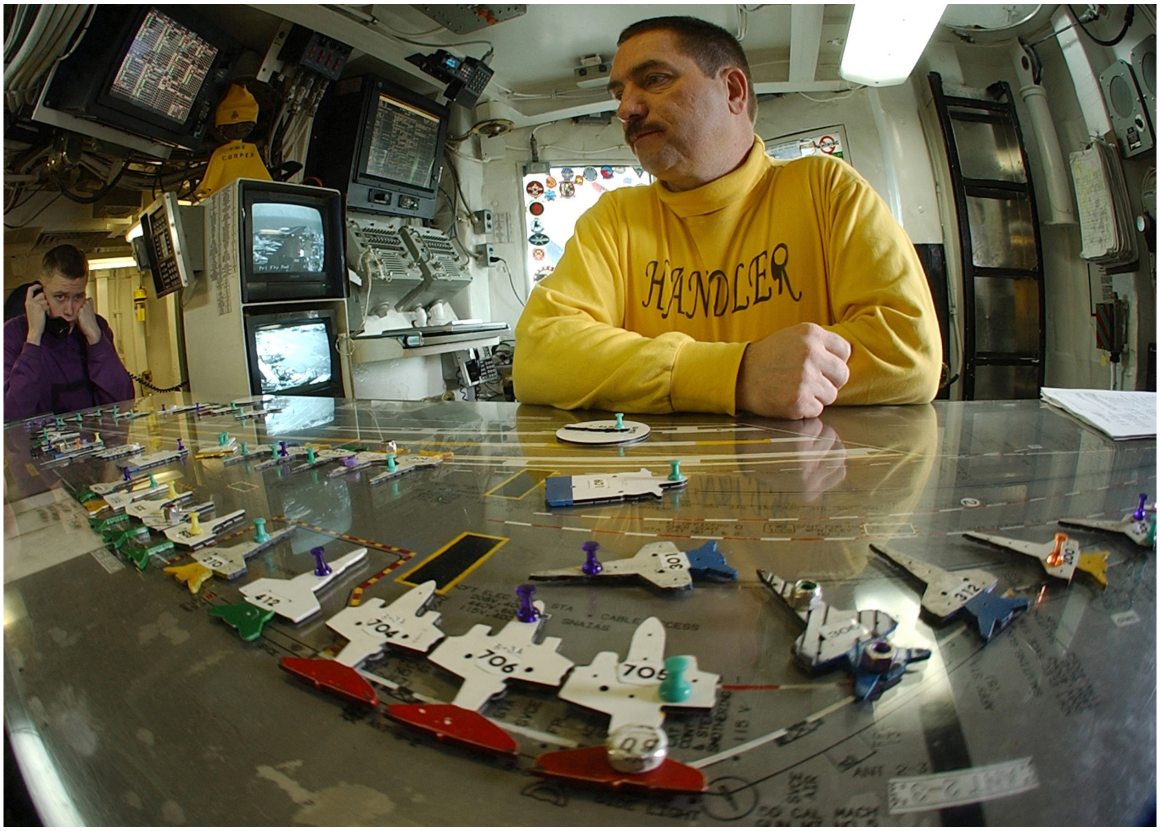


Figure 7-1: Picture of a Deck Handler, resting his elbows the "Ouija board," a tabletop-sized map of the flight deck used for planning operations. Scale, metal cutouts or aircraft are used to update aircraft positions and layouts, while colored thumbtacks are used to indicate fuel and weapons statuses.

they might adapt their heuristics to future operational environments.

7.2.3 Responding to Failures

Examining the responsiveness of the system to failures may be an additional extension of the MASCS model. In general, the MASCS missions assumed a fairly cooperative environment — no inclement weather, no mechanical failures on the part of aircraft or deck equipment, no random behaviors by crew, and no major issues that require a new plan be developed. Interviews with SMEs suggest that operations often require changes to plans in the midst of execution; consider a case where a launch catapult experiences a major issue and must shut down for the rest of the launch event. All aircraft currently assigned to the catapult must be reassigned and rerouted to the remaining catapults, which might also requires the relocation and movement of other aircraft on the flight deck to clear paths for the vehicles to travel. Given that issues with the "delayed" set of unmanned vehicles, such a drastic change in operations, requiring simultaneous action on the part of several agents, might place the system in a very poor position and limit its effectiveness.

Other aspects of interest include the current utilization of the crew. This might address levels of crew fatigue that occur during operations and the influences of both operational tempo as well as external environmental factors (rain, sun glare, etc.), or it might address how many crew are actually required to run the flight deck. Currently in MASCS, a team of sixteen Aircraft Directors are utilized in the zone coverage routing system. However, observations of the system recognize that the workload is not spread evenly amongst the crew, nor are many crew reaching peak utilization. While there are other benefits to having more crew than necessary on the flight deck (robustness of operations, safety concerns, etc.), systems such as MASCS could explore the effects of reducing this to smaller numbers of crew. Of particular interest would be the effects of crew reduction on the other aspects of flight deck operations, discussed earlier in Section 7.2.1, as well as the resulting effects on crew fatigue. Incorporating models the effects of fatigue and physiological state on crew errors may also prove useful in better estimating the rates of injuries, accidents, and other aspects of human performance on the flight deck.

7.2.4 Extensions to Other Domains

Lastly, and perhaps most importantly, alternative versions of this MASCS model could be developed for other domains, although requires a substantial rewriting of most likely every single agent and task defined within the system. However, the description of MASCS in Chapter 3 provides a template for the creation of alternative simulation environments. Of particular interest are models of the U.S. highway system or manufacturing environments, both of which are actively investigating the development of autonomous systems. Models of these environments may be created through similar methods as used in developing MASCS.

7.3 Contributions

The objective of this thesis was the development of an agent-based model of flight deck operations, a candidate Heterogeneous Manned-Unmanned Environment (HMUE), into which models of futuristic unmanned vehicle systems could be added and the effects of their integration into operations could be analyzed. In working towards this objective, several contributions have been made to the modeling and analysis of unmanned vehicle systems in complex sociotechnical systems, which are discussed below. This thesis also partially fills a gap in existing research, in which models of unmanned vehicle systems and their performance and interaction within human collaborators and other manned vehicles in the context of real-world, heterogeneous operations has gone largely unexamined.

Chapter 1 posed four research questions that guided the development of the MASCS model. The first two questions addressed the modeling of unmanned vehicle control architectures in HMUEs: "What are the specific agents and their related states, parameters, and decision-making rules required to construct an agent-based model of carrier flight deck operations" and "What are the key parameters that describe variations in unmanned vehicle control architectures and their interactions with the world [in that model]?" Answering these questions explored the creation of agent-based models of Heterogeneous Manned-Unmanned Environments and provided the following contributions:

- The establishment of a general framework of manned and unmanned vehicle operations in Heterogeneous Manned-Unmanned Environments that traces how information flows between human collaborators, manned vehicles, unmanned vehicles, and supervisors in the environment. This provided guidance as to the types of agents that should be defined within the system and served as a template for the construction of the MASCS model of operations.
- A description of the important states, parameters, and decision-making rules that define the key agents involved in flight deck operations, including models of motion and task execution in the environment. This provides guidance for future researchers seeking to build models of flight deck operations or of similar HMUEs.
- The identification of six key variables that define unmanned vehicle behavior in the context of flight deck operations. New models of unmanned vehicles were defined as a series of changes to a baseline manual control model, which was also created as part of this work. Verification tests for these models showed the same type of performance differences generally expected from the literature and demonstrates the utility of using an agent-based simulation for this research.

The final two sets of research questions addressed the effects of implementing unmanned vehicles in flight deck operations. The first examined changes in productivity: "In the context of the modeled control architectures, how do these key parameters drive performance on the aircraft carrier flight deck? How do these parameters interact with the structure of the environment, and how do the effects scale when using larger numbers of aircraft in the system?" The second set examined changes in safety: "If high-level safety protocols are applied to operations, what effects do they have on the safety of operations? Do the protocols leads to tradeoffs between safety and productivity in the environment?" This required the development and execution of a experimental program for the MASCS model, followed by additional tests that further explored the effects of different parameters within the system. This resulted in the following contributions:

- A series of performance metrics addressing the expediency, efficiency, and safety of flight deck operations were defined and implemented into the MASCS environment. These metrics described the overall functionality of the flight deck system while also reviewing individual vehicle behavior, the allocations of tasks across regions of the flight deck, and the proximity to which vehicles come to both one another and to crew.
- Variations in safety were driven largely by differences in the flight deck topology due to the heuristics of the crew zone coverage routing and the location of the crew in that topology.

For systems not using the crew zone coverage routing, improvements in aircraft safety were contingent on increasing high-level constraints in operations (separating aircraft based on time or schedule). Overconstraining operations provided some additional benefits to safety at a high cost to productivity.

- Variations in productivity, outside of the effects of overconstraining operations, were minimal due to the effects of the heuristics governing vehicle routing and catapult queueing and the utilization of the crew in the launch preparation process. Tests in Chapter 5 were unable to identify any real differences in productivity due to unmanned vehicle integration because of these factors. Key areas of future research concern reexamining the role of crew throughout these aspects of flight deck operations.
- When improvements to the catapult queueing process were made in Chapter 6, results showed that the occurrence of delays and latencies in operations is the key limitation in unmanned vehicle productivity. Further improvements to the planning heuristics and the topology of the flight deck (how vehicles are routed) may have further benefits to performance.
- At a much faster tempo of operations, variations in safety metrics across safety protocols for a given control architecture decreased, due to the reduction in time vehicles spent in queues and on the flight deck overall. This suggests that faster tempo operations (with vehicle transit speed left unchanged) may aid in minimizing the chances of accidents on the flight deck.

In all, this dissertation should serve as a guide to two potential populations. First, for researchers interested in developing agent-based models of human, manned vehicle, and unmanned vehicle behavior and interaction in the world, the creation of MASCS serves as a template for future models for similar environments. Second, for stakeholders interested in assessing the future effects of unmanned vehicle implementation into candidate environments, the results of the testing of MASCS provide guidance as to the key features of the environment that affect operations. In particular, these results have demonstrated that neither Safety Protocols nor Control Architectures can be examined on their own: significant interactions occur between them that are influenced by the context of conditions within the environment, the utilization of the crew, and the mission being conducted. The performance of the SBSC architecture, the most advanced system utilizing a high level of autonomy, suggests that increased autonomy does not necessarily imply increased safety across all types of measures.

As the transition of unmanned and autonomous vehicle systems from laboratory settings into the world at large accelerates, understanding exactly what the principle features of unmanned vehicle behavior are and how they interact with the environment is of significant importance. Given the significant cost of performing tests in the real world, the development of simulation models that adequately capture not only the motion of vehicles in the world, but their decision-making behavior and their interactions with other vehicles and people is of the utmost importance. MASCS provides one method of doing so and has in turn provided some significant findings as to the true worth of unmanned vehicles in one example domain and how they should be implemented. Extensions of models like MASCS that include more complex models of not only human performance but of vehicle hardware and sensors, even extended to hardware- or software-in-the-loop simulation, further enhance these capabilities and provide an even better understanding of these environments. For stakeholders such as the Federal Aviation Administration, the National Transportation Safety Board, and the National Highway Traffic Safety Administration, the utility of such models for reviewing the safe and effective of unmanned vehicles into their respective domains should be valuable not only in promoting the development of effective unmanned vehicles, but safe ones as well.

THIS PAGE INTENTIONALLY LEFT BLANK

References

Aircraft Carriers–CVN. (2014). Retrieved from http://ipv6.navy.mil/navydata/fact_display.asp?cid=4200&tid=200&ct=4. (Last accessed: April 29, 2014)

Alami, R., Albu-Schaffer, A., Bicchi, A., Bischoff, R., Chatila, R., De Luca, A., De Santis, A., Giralt, G., Guiochet, J., Hirzinger, G., Ingrand, F., Lippiello, V., Mattone, R., Powell, D., Sen, S., Siciliano, B., Tonietti, G., & Villani, L. (2006). Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges. In *Proceedings of Intelligent Robotics Systems (IROS'06).* Beijing, China.

Altenburg, K., Schlecht, J., & Nygard, K. E. (2002). An Agent-Based Simulation for Modeling Intelligent Munitions. In *Proceedings of the Second WSEAS International Conference on Simulation, Modeling and Optimization.* Skiathos, Greece.

Anderson, D., Anderson, E., Lesh, N., Marks, J., Perlin, K., Ratajczak, D., & Ryall, K. (1999). Human-Guided Simple Search: Combining Information Visualization and Heuristic Search. In *Proceedings of the 1999 Morkshop on New Paradigms in Information Visualization and Manipulation in conjunction with the Eighth ACM International Conference on Information and Knowledge Management.* Kansas City, MO.

Anderson, J. R., & Schunn, C. D. (2000). *Implications of the ACT-R Learning Theory: No Magic Bullets* (Technical Report). Department of Psychology, Carnegie Mellon University.

Archer, J. (2004). *Methods for Assessment and Prediction of Traffic Safety at Urban Interactions and their Application in Micro-simulation Modelling.* Academic thesis, Royal Institute of Technology, Stockholm, Sweden.

Archer, J., & Kosonen, I. (2000). The Potential of Micro-Simulation Modelling in Relation to Traffic Safety Assessment. In *Proceedings of the ESS Conference.* Hamburg, Germany.

Arthur, K. W., Booth, K. S., & Ware, C. (1993). Evaluating 3D Task Performance for Fish Tank Virtual Worlds. *ACM Transactions on Information Systems*, *11*(3), 239-265.

Atrash, A., Kaplow, R., Villemure, J., West, R., Yamani, H., & Pineau, J. (2009). Development and Validation of a Robust Speech Interface for Improved Human-Robot Interaction. *International Journal of Social Robotics*, *1*, 345-356.

Bandyophadyay, T., Sarcione, L., & Hover, F. S. (2010). A Simple Reactive Collision Avoidance Algorithm and its Application in Singapore Harbor. *Field and Service Robotics*, *62*, 455-465.

Batty, M., Desyllas, J., & Duxbury, E. (2003). Safety in Numbers? Modelling Crowds and Designing Control for the Notting Hill Carnival. *Urban Studies*, *40*(8), 1573-1590.

Bejczy, A. K., & Kim, W. S. (1990). Predictive Displays and Shared Compliance Control for Time-Delayed Telemanipulation. In *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems '90 (IROS '90) 'Towards a New Frontier of Applications'.* Ibaraki, Japan.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., & Basri, R. (2005). Action as Space-Time Shapes. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05).* Beijing, China.

Blom, H. A. P., Stroeve, S. H., & Jong, H. H. de. (2006). *Safety Risk Assessment by Monte Carlo Simulation of Complex Safety Critical Operations* (Technical Report No. NLR-TP-2006-684). National Aerospace Laboratory NLR.

Bonabeau, E. (2002). Agent-based Modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(Suppl. 3), 7280-7287.

Bonabeau, E., Hunt, C. W., & Gaudiano, P. (2003). Agent-Based Modeling for Testing and Designing Novel Decentralized Command and Control System Paradigms. Paper presented at *The International Command and Control Research and Technology Symposium.* Washington, D.C.

Bourakov, E., & Bordetsky, A. (2009). Voice-on-Target: A New Approach to Tactical Networking and Unmanned Systems Control via the Voice Interface to the SA Environment. In *Proceedings of the 14th International Command and Control Research and Technology Symposium.* Washington, D.C.

Bruni, S., & Cummings, M. (2005). Human Interaction with Mission Planning Search Algorithms. Paper presented at *Human Systems Integration Symposium (HSIS'05).* Arlington, VA.

Burdick, J. W., DuToit, N., Howard, A., Looman, C., Ma, J., Murray, R. M., & Wongpiromsarn, T. (2007). *Sensing, Navigation and Reasoning Technologies for the DARPA Urban Challenge* (Technical Report). California Institute of Technology/Jet Propulsion Laboratory.

Calhoun, G. L., Draper, M. H., & Ruff, H. A. (2009). Effect of Level of Automation on Unmanned Aerial Vehicle Routing Task. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 53, p. 197-201). San Antonio, TX.

Campbell, M., Garcia, E., Huttenlocher, D., Miller, I., Moran, P., Nathan, A., Schimpf, B., Zych, N., Catlin, J., Chelatrescu, F., Fujishima, H., Kline, F.-R., Lupashin, S., Reitmann, M., Shapiro, A., & Wong, J. (2007). *Team Cornell: Technical Review of the DARPA Urban Challenge Vehicle* (Technical Report). Cornell University.

Carley, K. (1996). *Validating Computational Models* (Technical Report). Pittsburgh, PA: Carnegie Mellon University.

Carley, K., Fridsma, D., Casman, E., Yahja, A., Altman, N., Chen, L.-C., Kaminsky, B., & Nave, D. (2003). BioWar: Scalable Agent-Based Model of Bioattacks. *IEEE Transactions on Systems, Man and Cybernetics–Part A: Systems and Humans*, *36*(2), 252-265.

Carr, D., Hasegawa, H., Lemmon, D., & Plaisant, C. (1993). The Effects of Time Delays on a Telepathology User Interface. In *Proceedings of the Annual Symposium on Computer Application in Medical Care.* Washington, D.C.

Ceballos, A., Gómez, J., Prieto, F., & Redarce, T. (2009). Robot Command Interface Using an Audio-Visual Speech Recognition System. In E. Bayro-Corrochano & J.-O. Eklundh (Eds.), *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (p. 869-876). Springer Berlin / Heidelberg.

Chen, X., Meaker, J. W., & Zhan, F. B. (2006). Agent-Based Modeling and Analysis of Hurricane Evacuation Procedures for the Florida Keys. *Natural Hazards*, *38*, 321-338.

Chen, Y. M., & Chang, S. H. (2008). An Agent-Based Simulation for Multi-UAVs Coordinative Sensing. *International Journal of Intelligent Computing and Cybernetics*, *1*(2), 269-284.

Cicirelli, F., Furfaro, A., & Nigro, L. (2009). An Agent Infrastructure over HLA for Distributed Simulation of Reconfigurable Systems and its Application to UAV Coordination. *Simulation*, *85*(1), 17-32.

Clare, A. S. (2013). *Modeling Real-time Human-automation Collaborative Scheduling of Unmanned Vehicles.* Doctoral dissertation, Massachusetts Institute of Technology.

Clothier, R. A., & Walker, R. A. (2006). Determination and Evaluation of UAV Safety Objectives. In *Proceedings of the 21st International Unmanned Air Vehicle Systems Conference* (p. 18.1-18.16). Bristol, United Kingdom.

Conover, W. J. (1980). Practical Nonparametric Statistics. *J. Wiley., New York.*

Conway, L., Volz, R. A., & Walker, M. W. (1990). Teleautonomous Systems: Projecting and Coordinating Intelligent Action at a Distance. *IEEE Transactions on Robotics and Automation*, *6*(2), 146-158.

Corner, J. J., & Lamont, G. B. (2004). Parallel Simulation of UAV Swarm Scenarios. In *Proceedings of the 2004 Winter Simulation Conference* (p. 355-363). Washington, D.C.

Crandall, J. W., Goodrich, M. A., Olsen, R. O., & Nielsen, C. W. (2005). Validating human-robot interaction schemes in multitasking environments. *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*, *35*(4), 438-449.

Cummings, M. L., Clare, A. S., & Hart, C. S. (2010). The Role of Human-Automation Consensus in Multiple Unmanned Vehicle Scheduling. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *52*(1).

Cummings, M. L., & Guerlain, S. (2007). Developing Operator Capacity Estimates for Supervisory Control of Autonomous Vehicles. *Human Factors*, *49*(1), 1-15.

Cummings, M. L., How, J., Whitten, A., & Toupet, O. (2012). The Impact of Human-Automation Collaboration in Decentralized Multiple Unmanned Vehicle Control. *Proceedings of the IEEE*, *100*(3), 660-671.

Cummings, M. L., & Mitchell, P. J. (2008). Predicting Controller Capacity in Remote Supervision of Multiple Unmanned Vehicles. *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*, *38*(2), 451-460.

Cummings, M. L., & Mitchell, P. M. (2005). Managing Multiple UAVs through a Timeline Display. Paper presented at *AIAA Infotech@Aerospace 2005.* Arlington, VA.

Cummings, M. L., Nehme, C. E., & Crandall, J. W. (2007). Predicting Operator Capacity for Supervisory Control of Multiple UAVs. In J. S. Chahl, L. C. Jain, A. Mizutani, & M. Sato-Ilic (Eds.), *Innovations in Intelligent Machines* (Vol. 70, p. 11-37). Springer Berlin Heidelberg.

Defense Science Board. (2012). *The Role of Autonomy in DoD Systems* (Technical Report No. July). Washington, D.C.: Department of Defense.

Dekker, S. W. A. (2005). *Ten Questions About Human Error: A New View of Human Factors and System Safety.* Mahwah, NJ: Lawrence Erlbaum Associates.

Dixon, S., Wickens, C. D., & Chang, D. (2005). Mission Control of Multiple Unmanned Aerial Vehicles: A Workload Analysis. *Human Factors*, *47*(3), 479-487.

Dixon, S. R., & Wickens, C. D. (2003). Control of Multiple-UAVs: A Workload Analysis. Paper presented at *The 12th International Symposium on Aviation Psychology.* Dayton, OH.

Dixon, S. R., & Wickens, C. D. (2004). *Reliability in Automated Aids for Unmanned Aerial Vehicle Flight Control: Evaluating a Model of Automation Dependence in High Workload* (Technical Report). Savoy, IL: Aviation Human Factors Division, Institute of Aviation, University of Illinois at Urbana-Champaign.

Doherty, P., Granlund, G., Kuchcinski, K., Nordberg, K., Skarman, E., & Wiklund, J. (2000). The WITAS Unmanned Aerial Vehicle Project. In *Proceedings of the 14th European Conference on Artificial Intelligence.* Berlin, Germany.

Edmonds, B. (1995). What is Complexity?–The philosophy of complexity per se with application to some examples in evolution. In F. Heylighen & D. Aerts (Eds.), *The Evolution of Complexity.* Dordrecht: Kluwer.

Edmonds, B. (2001). Towards a Descriptive Models of Agent Strategy Search. *Computational Economics*, *18*, 113-135.

Efimba, M. E. (2003). *An Exploratory Analysis of Littoral Combat Ships' Ability to Protect Expeditionary Strike Groups.* Master's thesis, Naval Postgraduate School, Monterey, CA.

Eichler, S., Ostermaier, B., Schroth, C., & Kosch, T. (2005). Simulation of Car-to-Car Messaging: Analyzing the Impact on Road Traffic. In *Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'05).* Atlanta, GA.

Elkins, L., Sellers, D., & Monach, W. R. (2010). The Autonomous Maritime Navigation (AMN) Project: Field Tests, Autonomous Behaviors, Data Fusion, Sensors, and Vehicles. *Journal of Field Robotics*, *27*(6), 790-818.

Epstein, J. M. (1999). Agent-Based Computational Models and Generative Social Science. *Complexity*, *4*(5), 41-60.

Epstein, J. M., & Axtell, R. M. (1996). *Growing Artificial Societies: Social Science from the Bottom Up.* Washington, D.C.: Brookings Institution Press.

Erlenbruch, T. (2002). *Agent-Based Simulation of German Peacekeeping Operations for Units up to Platoon Level.* Master's thesis, Monterey, CA.

Federal Aviation Administration. (2014a, January 6). *Fact Sheet–Unmanned Aircraft Systems (UAS).* Retrieved from http://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=14153. (Last accessed: April 29, 2014)

Federal Aviation Administration. (2014b). *Unmanned Aircraft (UAS) General FAQs* (No. July 27). Retrieved from http://www.faa.gov/about/initiatives/uas/uas_faq/#Qn1. Washington, D.C.: McGraw-Hill Companies, Inc. (Last accessed: July 27, 2014)

Fiscus, J. G., Ajot, J., & Garofolo, J. S. (2008). The Rich Transcription 2007 Meeting Recognition Evaluation. In R. Stiefelhagen, R. Bowers, & J. Fiscus (Eds.), *Multimodal Technologies for Perception of Humans.* Berlin: Springer-Verlag.

Fisher, S. (2008). *Replan Understanding for Heterogenous Unmanned Vehicle Teams.* Master's thesis, Cambridge, MA.

Forest, L. M., Kahn, A., Thomer, J., & Shapiro, M. (2007). The Design and Evaluation of Human-Guided Algorithms for Mission Planning. Paper presented at *Human Systems Integration Symposium (HSIS'07).* Annapolis, MD.

Fryman, J. (2014). Updating the Industrial Robot Safety Standard. In *Proceedings of ISR/Robotik 2014; 41st International Symposium on Robotics* (pp. 704–707). Munich, Germany.

Furui, S. (2010). HIstory and Development of Speech Recognition. In F. Chen & K. Jokinen (Eds.), *Speech Technology.* New York: Springer Science+Business Media.

Ganapathy, S. (2006). *Human-Centered Time-Pressured Decision Making in Dynamic Complex Systems.* Doctoral dissertation, Wright State University, Dayton, OH.

Gettman, D., & Head, L. (2003). *Surrogate Safety Measures From Traffic Simulation Models Final Report* (Technical Report). McLean, VA: U. S. Department of Transportation.

Gettman, D., Pu, L., Sayed, T., & Shelby, S. (2008). *Surrogate Safety Assessment Model and Validation: Final Report* (Technical Report). McLean, VA: U. S. Department of Transportation.

Gomez, J.-B., Ceballos, A., Prieto, F., & Redarce, T. (2009). Mouth Gesture and Voice Command Based Robot Command Interface. Paper presented at *The 2009 IEEE International Conference on Robotics and Automation (ICRA'09).* Kobe, Japan.

Gonzales, D., Moore, L., Pernin, C., Matonick, D., & Dreyer, P. (2001). *Assessing the Value of Information Superiority for Ground Forces–Proof of Concept* (Technical Report). Santa Monica, CA: RAND Corporation.

Hakola, M. B. (2004). *An Exploratory Analysis of Convoy Protection using Agent-Based Simulation.* Master's thesis, Naval Postgraduate School, Monterey, CA.

Hashimoto, T., Sheridan, T. B., & Noyes, M. V. (1986). Effects of Predictive Information in Teleoperation with Time Delay. *Japanese Journal of Ergonomics*, *22*(2).

Heath, B., Hill, R., & Ciarallo, F. (2009). A Survey of Agent-Based Modeling Practices (January 1998-July 2008). *Journal of Articifical Societies and Social Simulation*, *12*(4).

Heinzmann, J., & Zelinsky, A. (1999). A Safe-Control Paradigm for Human-Robot Interaction. *Journal of Intelligent and Robotic Systems*, *25*, 295-310.

Helbing, D., Farkas, I., & Vicsek, T. (2000). Simulating Dynamical Features of Escape Panic. *Nature*, *407*, 487-490.

Hennigan, W. J. (2013, July). *Navy drone X-47B lands on carrier deck in historic first.* Retrieved from http://articles.latimes.com/2013/jul/10/business/la-fi-mo-navy-drone-x47b-20130709. Los Angeles, CA. (Last accessed: April 29, 2014)

Hill, J. W. (1976). Comparison of Seven Performance Measures in a Time-Delayed Manipulation Task. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-6*(4), 286-295.

Ho, S.-T. T. (2006). *Investigating Ground Swarm Robotics Using Agent Based Simulation.* Master's thesis, Naval Postgraduate School, Monterey, CA.

Howe, A. E., Whitley, L. D., Barbulescu, L., & Watson, J. P. (2000). Mixed Initiative Scheduling for the Air Force Satellite Control Network. Paper presented at *The 2nd International NASA Workshop on Planning and Scheduling for Space.* San Francisco, CA.

Hu, J., Thompson, J. M., Ren, J., & Sheridan, T. B. (2000). Investigations into Performance of Minimally Invasive Telesurgery with Feedback Time Delays. *Presence*, *9*(4), 369-382.

Hu, X., & Sun, Y. (2007). Agent-Based Modeling and Simulation of Wildland Fire Suppression. In *Proceedings of the 2007 Winter Simulation Conference* (p. 1275-1283). Washington, D.C.

Huntsberger, T., Aghazarian, H., Howard, A., & Trotz, D. C. (2011). Stereo-vision Based Navigation for Autonomous Surface Vessels. *Journal of Field Robotics*, *28*(1), 3-18.

ISO. (unpublished). *ISO/PDTS 15066. Robots and robotic devices Industrial safety requirements Collaborative industrial robots* (Technical Report).

Jenkins, D., & Vasigh, B. (2013). *The Economic Impact of Unmanned Aircraft Systems Integration in the United States* (Technical Report). Arlington, VA: Association for Unmanned Vehicle Systems International.

Jewell, A. (1998). *Sortie Generation Capacity of Embarked Airwings* (Technical Report). Alexandria, VA: Center for Naval Analyses.

Jewell, A., Wigge, M. A., Gagnon, C. M. K., Lynn, L. A., Kirk, K. M., Berg, K. K., Roberts, T. A., Hale, A. J., Jones, W. L., Matheny, A. M., Hall, J. O., & Measell, B. H. (1998). *USS Nimitz and Carrier Airwing Nine Surge Demonstration* (Technical Report). Alexandria, VA: Center for Naval Analyses.

Jhuang, H., Serre, T., Wolf, L., & Poggio, T. (2007). A Biologically Inspired System for Action Recognition. Paper presented at *The IEEE 11th International Conference on Computer Vision.* Rio de Janeiro, Brazil.

Johnson, D. A., Naffin, D. J., Puhalia, J. S., Sanchez, J., & Wellington, C. K. (2009). Development and Implementation of a Team of Robotic Tractors for Autonomous Peat Moss Harvesting. *Journal of Field Robotics*, *26*(6-7), 549-571.

Jolly, K. G., Kumar, R. S., & Vijaykumar, R. (2009). A Bezier Curve Based Path Planning in a Multi-agent Robot Soccer System without Violating the Acceleration Limits. *Robotics and Autonomous Systems*, *57*, 23-33.

Kaber, D. B., Endsley, M., & Endsley, M. R. (2004). The Effects of Level of Automation and Adaptive Automation on Human Performance, Situation Awareness and Workload in a Dynamic Control Task. *Theoretical Issues in Ergonomics Science*, *5*(2), 113-153.

Kaber, D. B., Endsley, M. R., & Onal, E. (2000). Design of Automation for Telerobots and the Effect on Performance, Operator Situation Awareness and Subjective Workload. *Human Factors and Ergonomics in Manufacturing*, *10*(4), 409-430.

Kalmus, H., Fry, D. B., & Denes, P. (1960). Effects of Delayed Visual Control on Writing, Drawing, and Tracing. *Language and Speech*, *3*(2), 96-108.

Kanagarajah, A. K., Lindsay, P., Miller, A., & Parker, D. (2008). An Exploration into the Uses of Agent-Based Modeling to Improve Quality of Healthcare. In *Unifying Themes in Complex Systems* (p. 471-478). Springer.

Kennedy, W. G., Bugajska, M. D., Marge, M., Adams, W., Fransen, B. R., Perzanowski, D., Schultz, A. C., & Trafton, J. G. (2007). Spatial Representation and Reasoning for Human-Robot Collaboration. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence* (p. 1554-1559). Vancouver, Canada.

Kramer, L. A., & Smith, S. F. (2002). Optimizing for Change: Mixed-Initiative Resource Allocation with the AMC Barrel Allocator. Paper presented at *The 3rd International NASA Workshop on Planning and Scheduling for Space.* Houston, TX.

Kulic, D., & Croft, E. A. (2005). Safe Planning for Human-Robot Interaction. *Journal of Robotic Systems*, *22*(7), 383-396.

Kulic, D., & Croft, E. A. (2006). Real-time Safety for Human-Robot Interaction. *Robotics and Autonomous Systems*, *54*, 1-12.

Kumar, S., & Mitra, S. (2006). Self-Organizing Traffic at a Malfunctioning Intersection. *Journal of Artificial Societies and Social Simulation*, *9*(4).

Kurniawati, E., Celetto, L., Capovilla, N., & George, S. (2012). Personalized Voice Command Systems in Multi Modal User Interface. Paper presented at *The 2012 IEEE International Conference on Emerging Signal Processing Applications.* Las Vegas, NV.

Laird, J. E. (2008). *Extending the Soar Cognitive Architecture* (Technical Report). Division of Computer Science and Engineering, University of Michigan.

Lane, J. C., Carignan, C. R., Sullivan, B. R., Akin, D. L., Hunt, T., & Cohen, R. (2002). Effects of Time Delay on Telerobotic Contorl of Neutral Buoyancy Vehicles. In *Proceedings of the 2002 IEEE International Conference on Robotics & Automation* (p. 2874-2879). Washington, D.C.

LaPorte, T. R., & Consolini, P. M. (1991). Working in Practice but Not in Theory: Theoretical Challenges of "High-Reliability Organizations". *Journal of Public Administration Researcn and Theory*, *1*(1), 19-47.

Laptev, I., Marszalek, M., Schimd, C., & Rozenfeld, B. (2008). Learning Realistic Human Action from Movies. In *IEEE Conference on Computer Vision and Pattern Recognition* (p. 1-8). Anchorage, AK.

Larson, J., Bruch, M., Halterman, R., Rogers, J., & Webster, R. (2007). Advances in Autonomous Obstacle Avoidance for Unmanned Surface Vehicles. Paper presented at *AUVSI Unmanned Systems North America.* Washington, D.C.

Laskowski, M., McLeod, R. D., Friesen, M. R., Podaima, B. W., & Alfa, A. S. (2009). Models of Emergency Departments for Reducing Patient Waiting Times. *PLoS ONE*, *4*(7).

Laskowski, M., & Mukhi, S. (2008). Agent-Based Simulation of Emergency Departments with Patient Diversion. In D. Weerasinghe (Ed.), *ehealth 2008* (p. 25-37). Berlin: Springer.

Lauren, M. K. (2002). Firepower Concentration in Cellular Automaton Combat Models–An Alternative to Lanchester. *The Journal of the Operational Research Society*, *53*(6), 672-679.

Law, A. M. (2007). *Simulation Modeling and Analysis* (4th ed.). Boston: McGraw-Hill.

Lee, S. M., Pritchett, A. R., & Corker, K. (2007). Evaluating Transformations of the Air Transportation System through Agent-Based Modeling and Simulation. Paper presented at *FAA/Eurocontrol Seminar on Air Traffic Management.* Barcelona, Spain.

Leveson, N. G. (2004). *Model-Based Analysis of Socio-Technical Risk* (Technical Report). Cambridge, MA.

Leveson, N. G. (2011). *Engineering a Safer World: Systems Thinking Applied to Safety.* Cambridge, MA: Massachusetts Institute of Technology.

Liang, L. A. H. (2005). *The Use of Agent Based Simulation for Cooperative Sensing of the Battlefield.* Master's thesis, Naval Postgraduate School, Monterey, CA.

Lin, Z., Jiang, Z., & Davis, L. S. (2009). Recognizing Actions by Shape-Motion Prototypes. Paper presented at *The 12th International Conference on Computer Vision.* Kyoto, Japan.

Lockergnome. (2009, June). *USS Nimitz Navy Aircraft Carrier Operations.* Retrieved from http://www.youtube.com/watch?v=_2KYU97y2L8. (Last accessed: April 29, 2014)

Lum, M. J. H., Rosen, J., King, H., Friendman, D. C. W., Lendvay, T. S., Wright, A. S., Sinanan, M. N., & Hannaford, B. (2009). Teleoperation in Surgical Robotics–Network Latency Effects on Surgical Performance. Paper presented at *The 31st Annual International Conference of the IEEE EMBS.* Minneapolis, MN.

Lundell, M., Tang, J., Hogan, T., & Nygard, K. (2006). An Agent-Based Heterogeneous UAV Simulator Design. In *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering, and Data Bases* (p. 453-457). Madrid, Spain.

MacKenzie, I. S., & Ware, C. (1993). Lag as a Determinant of Human Performance in Interactive Systems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (p. 488-493). Amsterdam, Netherlands: ACM.

Macy, M. W., & Willer, R. (2002). From Factors to Actors: Computational Sociology and Agent-Based Modeling. *Annual Review of Sociology*, *28*, 143-166.

Maere, P. C., Clare, A. S., & Cummings, M. L. (2010). Assessing Operator Strategies for Adjusting Replan Alerts in Controlling Multiple Unmanned Vehicles. Paper presented at *The 11th IFAC/IFIP/IFORS/IEA Symposium on Analysis Design and Evaluation of Human Machine Systems.* Valenciennes, France.

Malasky, J., Forest, L. M., Khan, A. C., & Key, J. R. (2005). Experimental Evaluation of Human-Machine Collaborative Algorithms in Planning for Multiple UAVs. In *The 2005 IEEE International Conference on Systems, Man and Cybernetics* (Vol. 3, p. 2469-2475). Waikoloa, HI.

Mancini, A., Cesetti, A., Iuale, A., Frontoni, E., Zingaretti, P., & Longhi, S. (2008). A Framework for Simulation and Testing of UAVs in Cooperative Scenarios. *Journal of Intelligent and Robotic Systems*, *54*(1-3), 307-329.

Mar, L. E. (1985). *Human Control Performance in Operation of a Time-Delayed Master-Slave Manipulator.* Bachelor's thesis, Massachusetts Institute of Technology, Cambridge, MA.

Markoff, J. (2010). *Google Cars Drive Themselves, in Traffic.* Retrieved from http://www.nytimes.com/2010/10/10/science/10google.html?pagewanted=all&_r=0. Mountain View, CA: The New York Times. (Last accessed: April 29, 2014)

Marvel, J., & Bostelman, R. (2013). Towards Mobile Manipulator Safety Standards. In *Proceedings of the 2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE)* (pp. 31–36). Washington, D.C.: IEEE.

Matsusaka, Y., Fujii, H., Okano, T., & Hara, I. (2009). Health Exercise Demonstration Robot TAIZO and Effects of Using Voice Command in Robot-Human Collaborative Demonstration. Paper presented at *The 18th IEEE International Symposium on Robot and Human Interactive Communication.* Toyama, Japan.

McKinney, B. (2014). *Unmanned Combat Air System Carrier Demonstration (UCAS-D)* (Technical Report). Los Angeles, CA: Northrop Grumman Corporation. (Last accessed: April 29, 2014)

McLean, G. F., & Prescott, B. (1991). Teleoperator Task Performance Evaluation Using 3-D Video. Paper presented at *The IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing.* Victoria, BC, Canada.

McLean, G. F., Prescott, B., & Podhorodeski, R. (1994). Teleoperated System Performance Evaluation. *IEEE Transactions on Systems, Man, and Cybernetics*, *24*(5), 796-804.

McMindes, K. L. (2005). *Unmanned Aerial Vehicle Survivability: The Impacts of Speed, Detectability, Altitude, and Enemy Capabilities.* Master's thesis, Naval Postgraduate School, Monterey, CA.

Mekdeci, B., & Cummings, M. L. (2009). Modeling Multiple Human Operators in the Supervisory Control of Heterogeneous Unmanned Vehicles. Paper presented at *The 9th Conference on Performance Metrics for Intelligent Systems (PerMIS'09).* Gaithersburg, MD.

Mettler, L., Ibrahim, M., & Jonat, W. (1998). One Year of Experience Working with the Aid of a Robotic Assistant (the Voice-controlled Optic Holder AESOP) in Gynaecological Endoscopic Surgery. *Human Reproduction*, *13*(10), 2748-2750.

Milton, R. M. (2004). *Using Agent-Based Modeling to Examine the Logistical Chain of the Seabase.* Master's thesis, Naval Postgraduate School, Monterey, CA.

Mkrtchyan, A. A. (2011). *Modeling Operator Performance in Low Task Load Supervisory Domains.* Master's thesis, Massachusetts Institute of Technology, Cambridge, MA.

Munoz, M. F. (2011). *Agent-Based Simulation and Analysis of a Defensive UAV Swarm against an Enemy UAV Swarm.* Master's thesis, Naval Postgraduate School, Monterey, CA.

Naval Studies Board. (2005). *Autonomous Vehicles in Support of Naval Operations* (Technical Report). Washington, D.C.: Department of Defense.

Nehme, C. (2009). *Modeling Human Supervisory Control in Heterogeneous Unmanned Vehicle Systems.* Doctoral dissertation, Massachusetts Institute of Technology, Cambridge, MA.

Nikolaidis, S., & Shah, J. (2012). Human-Robot Interactive Planning using Cross-Training – A Human Team Training Approach. Paper presented at *AIAA Infotech@Aerospace 2012.* Garden Grove, CA.

Office of the Secretary of Defense. (2013). *Unmanned Systems Roadmap FY2013-2038* (Technical Report). Washington, D.C.: Office of the Secretary of Defense.

Olsen, D. R., & Wood, S. B. (2004). Fan-out: Measuring Human Control of Multiple Robots. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* Vienna, Austria.

Oving, A. B., & Erp, J. B. F. van. (2001). Driving with a Head-Slaved Camera System. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 45, p. 1372-1376). Minneapolis, MN.

Pan, X., Han, C. S., Dauber, K., & Law, K. H. (2007). A Multi-agent Based Framework for the Simulation of Human and Social Behaviors during Emergency Evacuations. *AI and Society*, *22*(2), 113-132.

Parasuraman, R., Galster, S., Squire, P., Furukawa, H., & Miller, C. (2005). A Flexible Delegation-Type Interface Enhances System Performance in Human Supervision of Multiple Robots: Empirical Studies with RoboFlag. *IEEE Transactions on Systems, Man, and Cybernetics*, *35*(4), 481-493.

Parasuraman, R., Galster, S. M., & Miller, C. A. (2003). Human Control of Multiple Robots in the RoboFlag Simulation Environment. Paper presented at *The IEEE International Conference on Systems, Man and Cybernetics.* Washington, D.C.

Park, K., Lee, S. J., Jung, H.-Y., & Lee, Y. (2009). Human-Robot Interface Using Robust Speech Recognition and User Localization Based on Noise Separation Device. Paper presented at *The 18th IEEE International Symposium on Robot and Human Interactive Communication.* Toyama, Japan.

Paruchuri, P., Pullalarevu, A. R., & Karlapalem, K. (2002). Multi Agent Simulation of Unorganized Traffic. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS'02)* (p. 176-183). Bologna, Italy.

Patvivatsiri, L. (2006). A Simulation Model for Bioterrorism Preparedness in an Emergency Room. In *Proceedings of the 2006 Winter Simulation Conference* (p. 501-508). Monterey, CA.

Peeta, S., Zhang, P., & Zhou, W. (2005). Behavior-based Analysis of Freeway Car-Truck Interactions and Related Mitigations Strategies. *Transportation Research Part B*, *39*, 417-451.

Perrow, C. (1984). *Normal accidents: Living with high-risk technologies.* Princeton, NJ: Princeton University Press.

Perzanowski, D., Schultz, A., Adams, W., Marsh, E., Gerhart, G. R., Gunderson, R. W., & Shoemaker, C. M. (2000). Using a Natural Language and Gesture Interface for Unmanned Vehicles. In *Proceedings of the Society of Photo-Optical Instrumentation Engineers* (Vol. 4024).

Pina, P. E., Cummings, M. L., Crandall, J. W., & Penna, M. D. (2008). Identifying Generalizable Metric Classes to Evaluate Human-Robot Teams. Paper presented at *Metrics for Human-Robot Interaction Workshop at the 3rd Annual Conference on Human-Robot Interaction.* Amsterdam, The Netherlands.

Pina, P. E., Donmez, B., & Cummings, M. L. (2008). *Selecting Metrics to Evaluate Human Supervisory Control Applications* (Technical Report). Cambridge, MA: MIT Humans and Automation Laboratory.

Pritchett, A. R., Lee, S., Huang, D., Goldsman, D., Joines, J. A., Barton, R. R., Kang, K., & Fishwick, P. A. (2000). Hybrid-System Simulation for National Airspace System Safety Analysis. In *Proceedings of the 2000 Winter Simulation Conference* (p. 1132-1142). Orlando, FL.

Rasmussen, S. J., & Chandler, P. R. (2002). MultiUAV: A Multiple UAV Simulation for Investigation of Cooperative Control. In *Proceedings of the 2002 Winter Simulation Conference* (p. 869-877). San Diego, CA.

Reason, J. (1990). *Human Error.* Cambridge, UK: Cambridge University Press.

Reason, J. T., Parker, D., & Lawton, R. (1998). Organisational Controls and Safety: The varieties of rule-related behaviour. *Journal of Occupational and Organisational Psychology*, *71*, 289-304.

Rio Tinto. (2014). *Mine of the Future.* Retrieved from http://www.riotinto.com/ironore/mine-of-the-future-9603.aspx.

Roberts, K. H. (1990). Some Characterstics of One Type of High Reliability Organization. *Organization Science*, *1*(2), 160-176.

Rochlin, G. I., La Porte, T. R., & Roberts, K. H. (1987). The Self-Designing High-Reliability Organization: Aircraft Carrier Flight Operations at Sea. *Naval War College Review*(42), 76-90.

Roth, E., Hanson, M. L., Hopkins, C., Mancuso, V., & Zacharias, G. L. (2004). Human in the Loop Evaluations of a Mixed-initiative System for Planning and Control of Multiple UAV Teams. In *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting* (p. 280-284). New Orleans, LA.

Rovira, E., McGarry, K., & Parasuraman, R. (2007). Effects of Imperfect Automation on Decision Making in a Simulated Command and Control Task. *Human Factors*, *49*(1), 76-87.

Roy, N., Baltus, G., Fox, D., Gemperle, F., Goetz, J., Hirsch, T., Margaritis, D., Montemerlo, M., Pineau, J., Schulte, J., & Thrun, S. (2000). Towards personal service robots for the elderly. Paper presented at *The Workshop on Interactive Robots and Entertainment (WIRE 2000)* (Vol. 25, p. 184).

Rucker, J. E. (2006). *Using Agent-Based Modeling to Search for Elusive Hiding Targets.* Master's thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH.

Ruff, H. A., Calhoun, G. L., Draper, M. H., Fontejon, J. V., & Guilfoos, B. J. (2004). Exploring Automation Issues in Supervisory Control of Multiple UAVs. In *Proceedings of the 2nd Human Performance, Situation Awareness, and Automation Conference (HPSAA II).* Daytona Beach, FL.

Ruff, H. A., Narayanan, S., & Draper, M. H. (2002). Human Interaction with Levels of Automation and Decision-Aid Fidelity in the Supervisory Control of Multiple Simulated Unmanned Air Vehicles. *Presence*, *11*(4), 335-351.

Ryan, J. C. (2011). *Assessing the Performance of Human-Automation Collaborative Planning Systems.* Master's thesis, Massachusetts Institute of Technology, Cambridge, MA.

Ryan, J. C., Banerjee, A. G., Cummings, M. L., & Roy, N. (2014). Comparing the Performance of Expert User Heuristics and an Integer Linear Program in Aircraft Carrier Deck Operations. *IEEE Transactions on Cybernetics*, *44*(6), 761–773.

Ryan, J. C., Cummings, M. L., Roy, N., Banerjee, A. G., & Schulte, A. S. (2011). Designing an Interactive Local and Global Decision Support System for Aircraft Carrier Deck Scheduling. Paper presented at *AIAA Infotech@Aerospace 2011.* St. Louis, MO, USA.

Sargent, R. G. (2009). Verification and Validation of Simulation Models. In *Proceedings of the 2009 Winter Simulation Conference* (p. 162-176). Orlando, FL.

Sato, A. (2003). *The RMAX Helicopter UAV* (Technical Report). Shizuoka, Japan: Yamaha Motor Co., LTD.

Scanlon, J. (2009). *How KIVA Robots Help Zappos and Walgreens* (No. April 10). Retrieved from http://www.businessweek.com/innovate/content/apr2009/id20090415_876420.htm. New York, NY: McGraw-Hill Companies, Inc. (Last accessed: April 29, 2014)

Schindler, K., & Gool, L. van. (2008). Action Snippets: How many frames does human action recognition require? Paper presented at *The IEEE Conference on Computer Vision and Pattern Recognition.* Anchorage, AK.

Schouwenaars, T., How, J., & Feron, E. (2004). Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees. Paper presented at *The AIAA Guidance Navigation and Control Conference and Exhibit.* Providence, RI.

Schuldt, C., Laptev, I., & Caputo, B. (2004). Recognizing Human Actions: A Local SVM Approach. In *Proceedings of the International Conference on Pattern Recognition.* Cambridge, UK.

Schultz, M., Schulz, C., & Fricke, H. (2008). Passenger Dynamics at Airport Terminal Environment. In W. W. F. Klingsch, C. Rogsch, A. Schadschneider, & M. Schreckenberg (Eds.), *Pedestrian and Evacuation Dynamics 2008.* Berlin: Springer-Verlag.

Schumann, B., Scanlan, J., & Takeda, K. (2011). Evaluating Design Decisions in Real-Time Using Operations Modelling. Paper presented at *The Air Transport and Operations Symposium.* Delft, The Netherlands.

Scott, S. D., Lesh, N., & Klau, G. W. (2002). Investigating Human-Computer Optimization. Paper presented at *CHI 2002.* Minneapolis, MN.

Scovanner, P., Ali, S., & Shah, M. (2007). A 3-Dimensional SIFT Descriptor and its Application to Action Recognition. Paper presented at *ACM Multimedia.* Augsburg, Bavaria, Germany.

Scribner, D. R., & Gombash, J. W. (1998). *The Effect of Stereoscopic and Wide Field of View Conditions on Teleoperator Performance* (Technical Report No. ARL-TR-1598). Aberdeen Proving Ground, MD.

Seetharaman, G., Lakhotia, A., & Blasch, E. P. (2006). Unmanned Vehicles Come of Age: The DARPA Grand Challenge. *Computer*, *39*(12), 26-29.

Sha, F., & Saul, L. K. (2007). Large Margin Hidden Markov Models for Automatic Speech Recognition. In B. Scholkopf, J. C. Platt, & T. Hofmann (Eds.), *Advances in Neural Information Processing Systems 19.* Cambridge, MA: MIT Press.

Sharma, S., Singh, H., & Prakash, A. (2008). Multi-Agent Modeling and Simulation of Human Behavior in Aircraft Evacuations. *IEEE Transactions on Aerospace and Electronic Systems*, *44*(4), 1477-1488.

Sheik-Nainar, M. A., Kaber, D. B., & Chow, M.-Y. (2005). Control Gain Adaptation in Virtual Reality Mediated Human-Telerobot Interaction. *Human Factors and Ergonomics in Manufacturing*, *15*(3), 259-274.

Sheridan, T. B. (1992). *Telerobotics, Automation and Human Supervisory Control.* Cambridge, MA: The MIT Press.

Sheridan, T. B., & Ferrell, W. R. (1963). Remote Manipulative Control with Transmission Delay. *IEEE Transactions on Human Factors in Electronics*, *4*(1), 25-29.

Sheridan, T. B., & Verplank, W. (1978). *Human and Computer Control of Undersea Teleoperators* (Technical Report). Cambridge, MA: MIT Man-Machine Systems Laboratory.

Simpson, R. C., & Levine, S. P. (2002). Voice Control of a Powered Wheelchair. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *10*(2), 122-125.

Sislak, D., Volf, P., Komenda, A., Samek, J., & Pechoucek, M. (2007). Agent-Based Multi-Layer Collision Avoidance to Unmanned Aerial Vehicles. In J. Lawton, J. Patel, & A. Tate (Eds.), *Knowledge Systems for Coalition Operation* (p. 1-6). Waltham, MA.

Smyth, C. C., Gombash, J. W., & Burcham, P. M. (2001). *Indirect Vision Driving with Fixed Flat Panel Displays for Near-unity, Wide, and Extended Fields of Camera View* (Technical Report). Aberdeen Proving Ground, MD.

Song, Y., Demirdjian, D., & Davis, R. (2012). Continuous Body and Hand Gesture Recognition for Natural Human-Computer Interaction. *ACM Transactions on Interactive Intelligent Systems*, *2*(1).

Spain, E. H., & Hughes, T. W. (1991). *Objective Assessments of Mobility with an Early Unmanned Ground Vehicle (UGV) Prototype Viewing System* (Technical Report). San Diego, CA: Naval Ocean Systems Center.

Starr, G. P. (1979). A Comparison of Control Modes for Time-Delayed Remote Manipulation. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-9*(4), 241-246.

Steele, M. J. (2004). *Agent-based Simulation of Unmanned Surface Vehicles.* Master's thesis, Naval Postgraduate School, Monterey, CA.

Sterman, J. D. (2000). *Business Dynamics: Systems Thinking and Modeling for a Complex World.* New York: Irwin/McGraw-Hill.

Stroeve, S. H., Bakker, G. J., & Blom, H. A. P. (2007). Safety Risk Analysis of Runway Incursion Alert Systems in the Tower and Cockpit by Multi-Agent Systemic Accident Modelling. Paper presented at *The 7th USA/Europe ATM R&D Seminar.* Barcelona, Spain.

Sugeno, M., Hirano, I., Nakamura, S., & Kotsu, S. (1995). Development of an Intelligent Unmanned Helicopter. In *Proceedings of 1995 IEEE International Conference on Fuzzy Systems, 1995 International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium* (Vol. 5, p. 33-34). Yokohama, Japan.

Sycara, K., & Lewis, M. (2008). *Agent-Based Approaches to Dynamic Team Simulation* (Technical Report No. NPRST-TN-08-9). Millington, TN: Navy Personnel Research, Studies, and Technology Division, Bureau of Naval Personnel.

Teller, S., Correa, A., Davis, R., Fletcher, L., Frazzolia, E., Glass, J., How, J. P., Jeon, J., Karaman, S., Luders, B., Roy, N., Sainath, T., & Walter, M. R. (2010). A Voice-Commanded Robotic Forklift Working Alongside Humans in Minimally-Prepared Outdoor Environments. Paper presented at *The IEEE International Conference on Robotics and Automation.* Anchorage, AK.

Teo, R., Jang, J. S., & Tomlin, C. J. (2004). Automated Multiple UAV Flight–The Stanford DragonFly UAV Program. Paper presented at *The 43rd IEEE Conference on Decision and Control.* Atlantis, Paradise Island, Bahamas.

Tesfatsion, L. (2003). Agent-Based Computational Economics: Modeling Economies as Complex Adaptive Systems. *Information Sciences*, *149*, 263-269.

The DARPA Urban Challenge: Autonomous Vehicles in City Traffic. (2009). In M. Buehler, K. Iagnemma, & S. Singh (Eds.), *Springer Tracts in Advanced Robotics.* Berlin Heidelberg: Sprinter-Verlag.

Thrun, S. (2010). *What we're driving at.* Retrieved from http://googleblog.blogspot.com/2010/10/what-were-driving-at.html. (Last accessed: April 29, 2014)

Tonietti, G., Schiavi, R., & Bicchi, A. (2005). Design and Control of a Variable Stiffness Actuator for Safe and Fast Physical Human/Robot Interaction. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation.* Barcelona, Spain.

Topf, A. (2011). *Rio Tinto boosts driverless truck fleet for use in Pilbara* (No. March 26). Retrieved from http://www.mining.com/2011/11/02/rio-tinto-boosts-driverless-truck-fleet-for-use-in-pilbara/. Vancouver, BC, Canada: Mining.com. (Last accessed: April 29, 2014)

Trouvain, B., Schlick, C., & Mevert, M. (2003). Comparison of a Map- vs. Camera-based User Interface in a Multi-robot Navigation Task. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics* (p. 3224-3231). Washington, D.C.

Vaughn, R. (2008). Massively Multi-Robot Simulation in Stage. *Journal of Swarm Intelligence*, *2*, 189-208.

Vries, S. C. de. (2005). *UAVs and Control Delays* (Technical Report). Soesterberg, The Netherlands: TNO Defence, Security, and Safety.

Wachter, M. D., Matton, M., Demuynck, K., Wambacq, P., Cools, R., & Compernolle, D. V. (2007). Template Based Continuous Speech Recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, *15*(4), 1377-1390.

Wahle, J., & Schreckenberg, M. (2001). A Multi-Agent System for On-Line Simulations Based on Real-World Traffic Data. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences.* Grand Wailea, Maui, HI.

Weibel, R. E., & Hansman Jr., R. J. (2004). Safety Considerations for Operation of Different Classes of UAVs in the NAS. Paper presented at *The AIAA's 4th Aviation Technology, Integration, and Operations Forum.* Chicago, IL.

Weick, K. E., & Roberts, K. H. (1993). Collective Mind in Organizations: Heedful Interrelating on Flight Decks. *Administrative Science Quarterly*, *38*(3), 357-381.

Wickens, C. D., & Hollands, J. G. (2000). *Engineering Psychology and Human Performance* (3rd ed.). Upper Saddle River, N.J.: Prentice Hall.

Wright, M. C. (2002). *The Effects of Automation on Team Performance and Team Coordination.* Doctoral dissertation, North Carolina State University, Raleigh.

Xie, Y., Shortle, J., & Donahue, G. (2004). Airport Terminal-Approach Safety and Capacity Analysis Using an Agent-Based Model. In *Proceedings of the 2004 Winter Simulation Conference* (p. 1349-1357). Washington, D.C.

Yoon, Y., Choe, T., Park, Y., & Kim, H. J. (2007). Safe Steering of UGVs in Polygonal Environments. Paper presented at *The International Conference on Control, Automation and Systems.* Seoul, South Korea.

Yuan, J., Liu, Z., & Wu, Y. (2009). Discriminative Subvolume Search for Efficient Action Detection. Paper presented at *The IEEE Conference on Computer Vision and Pattern Recognition.* Miami, FL.

Yuhara, N., & Tajima, J. (2006). Multi-Driver Agent-Based Traffic Simulation Systems for Evaluating the Effects of Advanced Driver Assistance Systems on Road Traffic Accidents. *Cognition, Technology, and Work*, *8*, 283-300.

Zarboutis, N., & Marmaras, N. (2004). Searching Efficient Plans for Emergency Rescue through Simulation: the case of a metro fire. *Cognition, Technology, and Work*, *6*, 117-126.

Appendices

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Safety Principles from Carrier and Mining Domains

This section contains tables of potential accidents in the aircraft carrier and mining domains, hazards that could cause those accidents, the safety principles that help to prevent those hazards, and the classes of safety principles they correspond to. Tables begin on the next page.

Table A.1: List of aircraft carrier of safety principles, part 1 of 3.

Accident	Hazard	Protocol	Class
Aircraft hit by UAV	Aircraft is too close to UAV in operation and unable to stop	Maintain proper spatial clearance	Collision Prevention
Aircraft hit by UAV	Aircraft is too close to UAV in operation and unable to stop	Automated vehicles segregated from manned	Restricted Areas
Catapult hits person	Person is inside safe boundary during catapult launch	Ensure safe area before beginning actions	Restricted Areas
Explosion and Fire	Accidental weapons firing on deck causes fires, explosions	Post landing, aircraft are immediately defueled and weapons are disarmed	Task Scheduling
Explosion and fire	Static charge on ordnance ignites fuel line, ignites weapons	Do not load fuel and weapons at same time (static discharge)	Task Scheduling
Person falls overboard	Person too close to deck edge	Crew should not cross the boundary line (deck edge)	Restricted Areas
Person hit by catapult	Person is inside safe boundary during catapult launch	Crew should not cross the boundary line (catapults)	Restricted Areas
Person hit by UAV	Person too close to UAV during operations, which was unable to stop	Maintain proper spatial clearance	Collision Prevention
Person hit by UAV	Person too close to vehicle during operations, which was unable to stop	Maintain proper spatial clearance	Collision Prevention
Person hit by UAV	UAV does not recognize presence of person	Automated vehicles segregated from manned	Restricted Areas
Person hit by UAV	Person too close to UAV during operations, which was unable to stop	Minimize the presence of extraneous personnel	Restricted Areas, Collision Prevention, Task Scheduling
Person hit by UAV	Person too close to UAV during operations, which was unable to stop	Wait for permission before entering area/using equipment	Restricted Areas, Collision Prevention, Task Scheduling
Person hit by UAV	Person too close to vehicle during operations, which was unable to stop	Wait for permission before entering area/using equipment	Restricted Areas, Collision Prevention, Task Scheduling
Person hit by UAV	UAV does not notice crew member in area	Wait for permission before entering area/using equipment	Restricted Areas

Table A.2: List of aircraft carrier safety principles, part 2 of 3.

Accident	Hazard	Protocol	Class
Person hit by vehicle	Vehicle does not notice crew member in area	Minimize the presence of extraneous personnel	Restricted Areas
Person hit by vehicle	Crew member enters area during unsafe time period	Wait for permission before entering area/using equipment	Restricted Areas, Task Scheduling
Person hit by vehicle	Vehicle does not notice crew member in area	Wait for permission before entering area/using equipment	Restricted Areas, Task Scheduling
Person hit by vehicle (landing aircraft)	Person is inside LZ during landing	Crew should not cross the boundary line (LZ)	Restricted Areas
Person hit by vehicle (launching aircraft)	Person is inside safe boundary during catapult launch	Crew should not cross the boundary line (catapults)	Restricted Areas
UAV hits person	UAV too close to person during operations and unable to stop	Maintain proper spatial clearance	Collision Prevention
UAV hits person	UAV too close to person during operations and unable to stop	Maintain awareness of surroundings	Maintain SA
UAV hits person	UAV too close to person during operations and unable to stop	Stay below safe driving speed	Collision Prevention
Vehicle crashes due to loss of fuel	Insufficient fuel level (due to leak or otherwise)	Maintain sufficient fuel levels	Task Scheduling
Vehicle hit by vehicle	Vehicle does not notice vehicle in area	Minimize the presence of extraneous personnel	Restricted Areas
Vehicle hits equipment	Aircraft too close to equipment during operations and unable to stop	Maintain proper spatial clearance	Collision Prevention
Vehicle hits equipment	Vehicle does not notice equipment in area/ vice versa	Maintain awareness of surroundings	Maintain SA
Vehicle hits equipment	Aircraft too close to equipment during operations and unable to stop	Stay below safe driving speed	Collision Prevention
Vehicle hits equipment	Vehicle moves without recognizing proximity of equipment	vehicles must remain stationary unless being directed by the AD	Collision Prevention

Table A.3: List of aircraft carrier safety principles, part 3 of 3.

Accident	Hazard	Protocol	Class
Vehicle hits person	Vehicle too close to person and could not stop in time	Maintain proper spatial clearance	Collision Prevention
Vehicle hits person	Vehicle does not notice crew member in area	Maintain awareness of surroundings (AD)	Maintain SA
Vehicle hits person	Vehicle does not notice crew member in area	Ensure safe area before beginning actions	Restricted Areas, Task Scheduling
Vehicle hits person	Vehicle moves unexpectedly	Tie down aircraft if not active	Task Scheduling
Vehicle hits person	Vehicle too close to person and could not stop in time	Apply minimum power needed for movement/orientation	Collision Prevention
Vehicle hits person	Vehicle too close to person and could not stop in time	Stay below safe driving speed	Collision Prevention
Vehicle hits person	Vehicle does not notice crew member in area	vehicles must remain stationary unless being directed by the AD	Collision Prevention
Vehicle hits vehicle	Vehicle does not notice vehicle in area	Maintain awareness of surroundings	Maintain SA
Vehicle hits vehicle	AD does not notice vehicle in area	Maintain awareness of surroundings (AD)	Maintain SA
Vehicle hits vehicle	Vehicle moves unexpectedly	Tie down aircraft if not active	Task Scheduling
Vehicle hits vehicle	Aircraft gets conflicting orders from multiple ADs	One AD in control of an aircraft at any time	Collision Prevention
Vehicle hits vehicle	Vehicle too close to vehicle and unable to stop in time	Apply minimum power needed for movement/orientation	Collision Prevention
Vehicle hits vehicle	Vehicle too close to vehicle and unable to stop in time	Stay below safe driving speed	Collision Prevention
Vehicle is hit by vehicle	Vehicle strays into path of other vehicle	Minimize the presence of extraneous personnel	Restricted Areas
Vehicle is hit by vehicle	Vehicle strays into path of other vehicle	wait for permission before entering area/using equipment	Restricted Areas, Task Scheduling
Vehicle is hit by vehicle (landing aircraft)	Person is inside LZ during landing	Vehicle should not cross the boundary line (LZ)	Restricted Areas
Vehicle is hit by vehicle (launching aircraft)	Verson is inside near catapult during launch	Vehicle should not cross the boundary line (catapults)	Restricted Areas

Table A.4: List of mining safety principles, part 1 of 3.

Accident	Hazard	Protocol	Class
Person (LV) hit by large vehicle	Vehicle moves without recognizing proximity of person	Maintain proper spatial clearance	Collision Prevention
Vehicle (auto) hits large vehicle	Vehicle (auto) unable to stop before hitting vehicle	Maintain proper spatial clearance	Collision Prevention
Vehicle hit by large vehicle	Vehicle unaware of location of vehicle	Maintain proper spatial clearance	Collision Prevention
Vehicle hits equipment	Vehicle unable to stop before hitting equipment	Maintain proper spatial clearance	Collision Prevention
Vehicle hits large vehicle	Vehicle unable to stop before hitting vehicle (Vehicle)	Maintain proper spatial clearance	Collision Prevention
Vehicle hits person (LV)	Vehicle unable to stop before hitting person (LV)	Maintain proper spatial clearance	Collision Prevention
Person hit by auto large vehicle	Vehicle does not recognize person	Automated vehicles segregated from manned	Restricted Areas
Vehicle hit by large vehicle (auto)	Vehicle does not recognize vehicle	Automated vehicles segregated from manned	Restricted Areas
Equipment hits person (LV)	Equipment moves without recognizing proximity of person	Ensure safe area before beginning actions	Restricted Areas, Collision Prevention, Task Scheduling
Equipment hits person (LV)	Equipment moves without recognizing proximity of person	Maintain awareness of surroundings	Restricted Areas, Collision Prevention, Task Scheduling
Equipment hits vehicle	Equipment moves without recognizing proximity of vehicle	Ensure safe area before beginning actions	Restricted Areas, Collision Prevention, Task Scheduling
Equipment hits vehicle	Equipment moves without recognizing proximity of vehicle	Maintain awareness of surroundings	Restricted Areas, Collision Prevention, Task Scheduling
Vehicle hits equipment	Vehicle too close to equipment during operations (could not stop)	Ensure safe area before beginning actions	Restricted Areas, Collision Prevention, Task Scheduling
Vehicle hits equipment	Vehicle unaware of location of equipment	Maintain awareness of surroundings	Restricted Areas, Collision Prevention, Task Scheduling

Table A.5: List of mining safety principles, part 2 of 3.

Accident	Hazard	Protocol	Class
Vehicle hits large vehicle	Vehicle does not notice equipment in area	Ensure safe area before beginning actions	Restricted Areas, Collision Prevention, Task Scheduling
Vehicle hits person (LV)	Vehicle unaware of location of person (LV)	Ensure safe area before beginning actions	Restricted Areas, Collision Prevention, Task Scheduling
Vehicle hits person (LV)	Vehicle unaware of location of person (LV)	Maintain awareness of surroundings	Restricted Areas, Collision Prevention, Task Scheduling
Vehicle hits vehicle	Vehicle unaware of location of vehicle	Maintain awareness of surroundings	Restricted Areas, Collision Prevention, Task Scheduling
Vehicle hit by large vehicle	Vehicle unable to see large vehicle at intersection	Drivers must ensure safe transit through intersections	Collision Prevention
Vehicle hits large vehicle	Vehicle unable to see large vehicle at intersection	Drivers must ensure safe transit through intersections	Collision Prevention
Vehicle hits person (LV)	Vehicle unable to see person (LV) at intersection	Drivers must ensure safe transit through intersections	Collision Prevention
Person (LV) hit by large vehicle	Vehicle unaware of location of parked person (LV)	Do not park larger vehicles next to larger vehicles	Task Scheduling
Person (LV) hit by large vehicle	Vehicle unable to see person (LV) at intersection	Give priority to large vehicles	Task Scheduling
Vehicle hit by large vehicle	Vehicle unaware of location of parked vehicle	Do not park larger vehicles next to larger vehicles	Task Scheduling
Person (LV) hit by auto large vehicle	Vehicle does not recognize person	Wait for permission before entering area	Restricted Areas, Task Scheduling
Person (LV) hit by equipment	Equipment moves without recognizing proximity of person	Wait for permission before using equipment	Restricted Areas, Task Scheduling
Person (LV) hit by large vehicle	Vehicle unaware of location of person (LV)	Minimize the presence of extraneous personnel	Restricted Areas
Person (LV) hit by large vehicle	Vehicle moves without recognizing proximity of person	Wait for permission before entering area	Restricted Areas, Task Scheduling

Table A.6: List of mining safety principles, part 3 of 3.

Accident	Hazard	Protocol	Class
Vehicle hit by equipment	Equipment moves without recognizing proximity of vehicle	Wait for permission before using equipment	Restricted Areas, Task Scheduling
Vehicle hit by large vehicle	Vehicle unaware of location of vehicle	Minimize the presence of extraneous personnel	Restricted Areas
Vehicle hit by large vehicle (auto)	Vehicle does not recognize vehicle	Wait for permission before entering area	Restricted Areas, Task Scheduling
Vehicle (auto) hits large vehicle	Vehicle (auto) unable to stop before hitting vehicle	Stay below safe driving speed	Collision Prevention
Vehicle hits equipment	Vehicle unable to stop before hitting equipment	Stay below safe driving speed	Collision Prevention
Vehicle hits large vehicle	Vehicle unable to stop before hitting vehicle	Stay below safe driving speed	Collision Prevention
Vehicle hits person (LV)	Vehicle unable to stop before hitting person (LV)	Stay below safe driving speed	Collision Prevention

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix B

MASCS Agent Model Descriptions

B.1 Task Models and Rules

Multi-Agent Safety and Control Simulation (MASCS) task models operate through the use of two main functions. The first, the *execute* function, exists within all tasks and updates all appropriate states and functions with the time of the elapsed update. An example of this appears as pseudocode in MoveForward, which provides pseudocode for the "moveForward" actions that replicates straight line motion on the flight deck. The task requires as input the current Agent that action is assigned to, that agent's initial position, desired final position, and current speed. From this, a series of variables are initialized in lines 2-7; the most important of these are that the total distance of travel *totalDistance* and total time of motion *totalTime.* The action will track the total time of task execution *currentTime* and at each update compare the value to *totalTime*; task execution stops when *currentTime*≥*totaltime.*

The script first tests whether or not the task should be updated (line 12): for motion tasks, if the Director is not visible to the pilot or if a collision is imminent, the task should not proceed. If these conditions are satisfied, the task proceeds: *currentTime* is updated with the value of *elapsedTime* and the vehicle positions are updated according to the elapsed time, speed, and angle or travel. This then repeats until the appropriate time has elapsed.

```
MoveForward($Aircraft, init.x, init.y, final.x, final.y, speed, elapsedTime$)
// Calculate geometry and distance of motion
$currentTime = 0$
$angle = \tan \frac{final.y - init.y}{final.x - init.x}$
$totalDistance = \sqrt{((final.x - init.x)^2 + (final.y - init.y)^2)}$
// Time equals distance divided by speed
$totalTime = \frac{totalDistance}{speed}$
// Total time of movement equals distance divided by speed
while $currentTime <= totalTime$
    // Continue to operate until we reach totalTime/total distance of movement
    // If Director is visible and aircraft not in danger of collision,
    // update currentTime and aircraft locations
    if $(Aircraft.Director.visible == \textsc{true})$ && $(Aircraft.collision \neq \textsc{true})$
        $currentTime+ = elapsedTime$
        $Aircraft.x = init.x + elapsedTime * speed * \cos angle$
        $Aircraft.y = init.y + elapsedTime * speed * \sin angle$
```

In the above example, checking the visibility of the Director and whether a collision is imminent are both logical rule sets. Pseudocode for collision prevention appears in CollisionPrevention, with pseudocode for Director visibility appearing DirectorVisibility. Aircraft models include a check for Director visibility while taxiing. The routine requires knowledge of the current aircraft, its current position and orientation, and its available field of view. It also requires a maximum distance threshold, describing the general area in which the Director should be (see Figure 3-5 in Chapter 3). The routine then checks both the current angle between the Aircraft and the Director and their distance between them; if either is not at an acceptable value, another routine not shown here calculates the shorter straight-line path to move the Aircraft Director within the Aircraft's field of view at the nearest aircraft wingtip.

```
DirectorVisibility($Aircraft, aircraft.angle, Director, distanceThreshold$)
// Calculate geometric relationships
$angle = \tan \frac{currentPos.y - directorPos.y}{currentPos.x - directorPos.x}$
$diffAngle = currentAngle - angle$
$distance = \sqrt{((finalPos.x - initPos.x)^2 + (finalPos.y - initPos.y)^2)}$
if $(distance > distanceThreshold) || (diffAngle > Aircraft.fieldOfView)$
    Run $Aircraft.findAlignmentPoint$
    // Determines closest point where distance $distance \leq distanceThreshold$
    // and $diffAngle \leq Aircraft.fieldOfView$
else Continue
```

While in motion, Aircraft are also continually applying a set of rules that act to prevent them from colliding with other aircraft. Pseudocode for this appears in CollisionPrevention. The routine first requires knowledge of the current aircraft in motion, its position and heading, as well as a list of all aircraft currently on the flight deck and a distance threshold *d*. The routine then checks the distance between the current aircraft in motion (Aircraft) and all other vehicles active on the deck. If another aircraft is both within the distance *d* and in front of Aircraft (angle between them is 90^{circ} or less), the current aircraft in motion must stop. “Aircraft” will remain stopped until the other aircraft moves out of the way. As mentioned above, this routine is always in operation while the aircraft is in motion.

```
CollisionPrevention($Aircraft, AircraftList, Director$)
for $i = 0$ to $AircraftList.size()$
    $angle = \tan \frac{Aircraft.y - Director.y}{Aircraft.x - Director.x}$
    $diffAngle = Aircraft.angle - angle$
    if $|diffAngle| \leq 90$
        if $distance < d$
            // Aircraft is in front of us and in danger of collision
            return TRUE
        else
            // Aircraft is in front of us, but far enough away
            return FALSE
    else
        // Aircraft is not in front of us
        return FALSE
```

Time-based actions address non-motion tasks executed by aircraft and typically involve interactions with other agents (vehicle, crew, or equipment agents). For example, the set of tasks performed during launch preparations require no physical movement on the part of the aircraft or crew. The Action that replicates these functions in MASCS simply counts up to a specific time value before completing execution. The acceleration of the aircraft down the catapult is also time-based, using the time required to complete the task to decompose the required acceleration of the vehicle to flight speed. Data on these two parameters were compiled from video recordings of operations compiled primarily from the website Youtube.com. These data were then each fit into a mathematical distribution, the results of which appear in Appendix C.

Executing these tasks is similar to executing motion based tasks as shown in MoveForward. For these tasks, however, the time is provided by a specified parametric distribution. Pseudocode for the launch preparation time appears in Launch Preparation-Initialize and Launch Preparation-Execute; pseudocode for the acceleration task is a combination of MoveForward and Launch Preparation-Initialize, first sampling the completion time from a candidate distribution then iteratively updated vehicle speed and position over the provided time.

```
Launch Preparation-Initialize($mean, stddev, mininum$)
// Initialize variables
$distribution = newGaussian(mean, stddev)$
$totalTime = distribution.sample()$
// Randomly sample distribution to get task time
while $(totalTime \leq (mean - 3.9 * stddev)$ —— $(totalTime \geq (mean + 3.9 * stddev))$
    $totalTime = distribution.sample()$
```

```
LAUNCH PREPARATION-EXECUTE($totalTime, elapsedTime$)
// Initialize variables
$currentTime = 0$
// Execution loop
while $currentTime <= totalTime$
    $currentTime+ = elapsedTime$
```

B.2 Director models and rules

The collision prevention rule base was described above in CollisionPrevention. The logic behind it is rather simple and is vehicle-centric in nature: it only describes what the behavior of an individual aircraft should be and makes no judgments about the efficiency of the system. Employing only this simple collision prevention rule base, aircraft can easily find themselves simultaneously blocking each other and "locking" the deck. This requires action on the part of the Aircraft Directors to control traffic on the flight deck, applying rules and priorities to aircraft prevent any motion conflicts from occurring. Pseudocode for this appears in TrafficDeconfliction. These rules first rely on a set of traffic management lists and assumes that an aircraft has already been provided a catapult assignment (this logic is described later in Handler-Assign). One management list controls taxi operations in the Fantail area for aircraft heading to aft catapults, or for aircraft parked aft heading to forward catapults. Another list manages the "Street" area, managing traffic for aircraft heading to forward catapults and aircraft parked forward that are assigned to aft catapults. A third list manages the "Box" area near catapults 1 and 2, managing operations of aircraft parked far forward on the flight deck.

Each list operates similarly: as aircraft receive assignments, they are added to the appropriate list(s). Once they have moved through the area, they are removed from the list. The first aircraft in the list has top priority in taxi operations, although the lists are checked to ensure that aircraft with higher overall launch priorities are ranked higher than others. While an aircraft is taxiing, these rules are continually checked to ensure that they have permission to move, much in the same way that the collision prevention routines is checked. Given a current aircraft in motion ($Aircraft$), it positions in the appropriate list is checked. If it is the first in the list, it is allowed to move. Otherwise, a series of additional checks are performed to determine if there actually is a traffic conflict. For instance, if the first aircraft in the list ($firstAircraft$) is assigned to a forward catapult, no conflict exists if $Aircraft$ is parked aft of $firstAircraft$ and assigned to an aft catapult. The same is true if $firstAircraft$ is assigned aft and $Aircraft$ is parked forward and assigned forward.

Operations may be allowed to occur if both are heading the same direction, however. If both are assigned forward and $Aircraft$ is aft of $firstAircraft$ (following behind it), no conflict should ever exist. If $Aircraft$ is parked forward of $firstAircraft$ (ahead of it), it must be several aircraft lengths ahead of it in order to prevent a conflict from occurring. So long as this is true, $Aircraft$ is allowed to move.

Other rules for Directors dictate how they transfer aircraft between one another on the flight deck. This requires knowing the current assignment of the aircraft, its positions and heading, and the network of Directors on the deck. This network is known to each Director and includes an understanding of their positions on the flight deck. Pseudocode for this routine appears in DirectorHandoff, which functions as a search loop over all available connections. It begins by defining a set of variables for search, the most important of which are the current distance between the aircraft and its assigned catapult (destDistance) and the current angle to that catapult (destAngle). The Director checks their connection network to see which options are available. The routine then iterates over this list, calculating the distance and angle between the connection and and the assigned catapult. The difference between the aircraft's angle to the catapult and the new Director's angle to the catapult are compared; this should be as small as possible. The distance of the new Director to the catapult is also compared, and should be both as small as possible and smaller than the aircraft's distance from the catapult. Best values for each are stored during the iteration, and the Director that is closest to the catapult and closest to the straight line path to the catapult is selected.

```
DirectorHandoff($Aircraft, DirectorConnections, Destination$)
// Initialize variables
$destAngle = \tan \frac{Aircraft.y - destination.y}{Aircraft.x - destination.x}$
$bestDist = 1000000$
$bestAngle = 180$
$bestIndex = -1$
// Iterate through list of connections to find best option
for $i = 0$ to $DirectorConnections.size()$
    $currentDir = DirectorConnections.get(i)$
    $deltaX = currentDir.x - destination.x$
    $deltaY = currentDir.y - destination.y$
    $currentAngle = |\tan \frac{deltaY}{deltaX} - destAngle|$
    // currentAngle compares angle between Director and destination to angle between
    // aircraft and destination; looking for Directors closest to that path.
    $currentDist = \sqrt{((deltaX)^2 + (deltaY)^2)}$
    if $currentDist < bestDist$ & $currentAngle < bestAngle$
        $bestIndex = i$
        $bestDist = currentDist$
        $bestAngle = currentAngle$
if $bestIndex \neq -1$
    return $DirectorConnections.get(bestIndex)$
```

The Deck Handler allocation routine requires the definition of two different heuristic sets: the first (Handler-Assign) assigns aircraft to catapults and used the second (Handler-RouteCheck) to determine whether the route to the catapult is feasible. The assignment heuristic checks if any aircraft are parked in the center parking spaces in the flight deck in between the “Street” and “Fantail” areas (identification numbers 8 and 9 in the MASCS simulation). If any aircraft are parked in these area, aircraft cannot be taxied forward to aft, or vice versa.

```
TrafficDeconfliction($Aircraft, AircraftList, threshold$)
for $i = 0$ to $AircraftList.size()$
    if $AircraftList.indexOf(this) \neq 0$
        $firstAircraft == AircraftList.getFirst()$
        $deltaX = firstAircraft.x - Aircraft.x$
        $deltaY = firstAircraft.y - Aircraft.y$
        $distance = \sqrt{((deltaX)^2 + (deltaY)^2)}$
        if $firstAircraft.catapult < 3$
            // $firstAircraft$ is assigned to head forward
            if $(Aircraft.x - firstAircraft.x) < 0$
                Continue
                // $Aircraft$ is aft of $firstAircraft$, which is heading forward;
                // no conflict exists
            elseif $(Aircraft.catapult < 3)$ &&
                $(Aircraft.x - firstAircraft.x) > 0$ && $distance > threshold$
                Continue
                // $Aircraft$ is forward of $firstAircraft$, heading forward, and
                // far enough ahead of $firstAircraft$ that no conflict exists
            else Pause
                // A routing conflict exists
        else
            // $firstAircraft$ is assigned to head aft
            if $(Aircraft.x - firstAircraft.x) > 0$
                Continue
                // $Aircraft$ is forward of $firstAircraft$, which is heading aft;
                // no conflict exists
            elseif $(Aircraft.catapult > 2)$ && $(Aircraft.x - firstAircraft.x) < 0$
                                  && $distance > threshold$
                Continue
                // $Aircraft$ is aft of $firstAircraft$, heading aft, and far enough ahead of
                // $firstAircraft$ that no conflict exists
            else Pause
                // A routing conflict exists
```

```
Handler-RouteCheck($Aircraft, Catapult, ParkingSpaces$)
if $Aircraft.parkingArea ==$ AFT && $Catapult.number < 3$
    // aircraft parked aft and heading to forward catapult
    if $ParkingSpace.8.occupied ==$ TRUE || $ParkingSpace.9.occupied ==$ TRUE
        return FALSE
    else return TRUE
elseif $Aircraft.parkingArea ==$ FORWARD && $Catapult.number > 2$
    // aircraft parked forward and heading to aft catapult
    if $ParkingSpace.8.occupied ==$ TRUE || $ParkingSpace.9.occupied ==$ TRUE
        return FALSE
    else return TRUE
else return TRUE
```

```
Handler-Assign($UnassignedAircraft, CatapultList$)
// Initialize variables
$bestDist = 1000000$
$bestIndex = -1$
// Iterate over list of aircraft without assignments; only look at five highest priority
for $i = 1$ to $5$
    $currentAircraft = AircraftList.get(i)$
    // iterate over all catapults
    for $j = 1$ to $4$
        if $CatapultList(j).queue.size() == 0$
            if $routeCheck ==$ TRUE
                $deltaX = currentAircraft.x - CatapultList(j).x$
                $deltaY = currentAircraft.y - CatapultList(j).y$
                distance = $\sqrt{((deltaX)^2 + (deltaY)^2)}$
                if $distance < bestDist$
                    $bestDist = distance$
                    $distance = 0$
                    $bestIndex = i$
    if $bestIndex \neq -1$
        $currentAircraft.catapult == i$
        $UnassignedAircraft.remove(currentAircraft)$
    else
        $bestDist = 1000000$
        if $CatapultList(j).queue.size() == 1$
            if $routeCheck ==$ TRUE
                $deltaX = currentAircraft.x - CatapultList(j).x$
                $deltaY = currentAircraft.y - CatapultList(j).y$
                distance = $\sqrt{((deltaX)^2 + (deltaY)^2)}$
                if $distance < bestDist$
                    $bestDist = distance$
                    $distance = 0$
                    $bestIndex = i$
    if $bestIndex \neq -1$
        $currentAircraft.catapult == i$
        $UnassignedAircraft.remove(currentAircraft)$
```

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix C

MASCS Input Data

Turn rate observations:

Table C.1: Observed turn rates for F-18 aircraft during manually-piloted flight deck operations.

time (sec)	Estimated Angle (degrees)	Estimated Rate (degrees/second)	Source
7.50	180.00	24.00	http://youtu.be/RGQy5ImbnNI
12.00	180.00	15.00	http://youtu.be/RGQy5ImbnNI
3.50	90.00	25.71	http://youtu.be/7EaPe7zGZAw
1.00	15.00	15.00	http://youtu.be/_2KYU97y2L8
10.00	90.00	9.00	http://youtu.be/7G3MujUry-I
Average		17.75	

The Launch Preparation Time and Takeoff Time distributions discussed in Chapter 3 were based on empirical observations from video recordings of operations. Table C.2 contains the observations related to the launch preparation time, while Table C.3 contains observations related to the Takeoff time.

Table C.2: Observations of launch preparation time.

61.00	74.00	141.00
73.00	75.00	153.00

Table C.3: Observations of takeoff (acceleration to flight speed) time.

2.30	2.50	2.80	3.00
2.30	2.50	2.80	3.00
2.40	2.60	2.80	3.10
2.50	2.60	2.90	3.10
2.50	2.70	2.90	3.20
2.50	2.70	2.90	3.20
2.50	2.80	3.00	3.20

Because of the low number of observations for launch preparation time, any distribution fit to the data is likely to be inaccurate. For this data, a normal distribution was fit that placed the maximum (153 seconds) and minimum (61 seconds) observations at certain intervals from the mean, influenced by an understanding of the maximum and minimum feasible launch times. The final selected distribution appears in Fig. C-1, with the six observed points overlaid. This distribution utilizes a mean of 109.648 seconds and a standard deviation of 57.8 seconds, which places the maximum observations (153 seconds) at the 75th percentile and the minimum (60 seconds) at the 20th. The minimum feasible launch time is 30 seconds; within Multi-Agent Safety and Control Simulation (MASCS), any random samples that occur below this threshold are ignored. The maximum allowed time to sample would be equal to the mean times 3.9 standard deviations, equal to 335 seconds (about 6.5 minutes). Beyond a distance of 3.9 standard deviations, values are considered to be extreme (99.99% of all values lie within ±3.9 standard deviations of the mean).

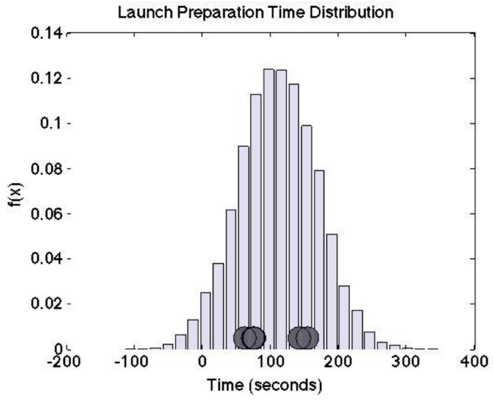


Figure C-1: Probability density function of the modeled launch preparation time, with empirical observations overlaid.

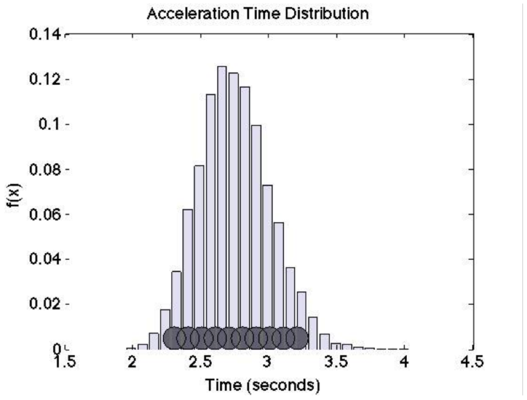


Figure C-2: Probability density function of the modeled launch acceleration time, with empirical observations overlaid.

Figure C-2 shows the resulting distribution of takeoff times, based on a Lognormal distribution fit to the data in Table C.3. The distribution fit was performed in the Easyfit program, which returned the following information from the Kolmogorov-Smirnov test (Table C.4):

Table C.4: Results of Easyfit fit of Lognormal distribution to takeoff acceleration time data.

	Lognormal
K-S statistic	0.1513
p	0.49598
Shape parameter σ	0.09891
Location parameter μ	1.0106

Appendix D

Example Scenario Configuration Text

D.1 Example 1: Manually piloted Aircraft

This first line of text initializes an F-18 type aircraft (FMAC) named "F18 40" at parking space 9, spot 1, fuel level at 100% (1.0). Line two assigns a wait task of 5000 ms to the vehicle (spacing its addition into the simulation). Line 3 initializes a Taxi task that will send the vehicle to a catapult; the assignment of -1 implies that the Deck Handler algorithm will make this assignment some time in the future. Line 4 assigns a takeoff task to the aircraft, which will be at the same aircraft as the Taxi task. Line 5 assigns the aircraft to fly away from the aircraft off to Mission. The Launch preparation tasks discussed in Chapter 3 are contained within the Takeoff task structure.

```
Aircraft FMAC F18-40 Parked 0 0 9 1 1.0 0
Task Wait 5000 0
Task Taxi -1 Catapult
Task Takeoff -1 0
Task GotoMission 2000 North
```

D.2 Example 2: Crew

The first line below creates a "Crew" agent, defined as "D16" and assigned as an Aircraft Directors. It is initially placed as deck coordinates (1355, 1050) and assigned to the group Deck 3.

```
Crew D16 director 1355 1050
```

D.3 Example 3: Deck equipment

The first line below creates a Catapult agent, replicating catapult 1, that it is operational (the second numeral 1), with starting point (1656, 1124) and ending point (2362,1072) in the deck coordinates. The trio of lines that follow define a pair of parking locations on the flight deck. The first line initializes a parking space structure (regions on the flight deck) named "Parking Space 1" and assigns a parking angle of −140° to aircraft. The second line defines parking spot 0 within that region, that it is availabled, and places it at coordinates (2340, 1100). The second line defines parking spot 1 within that space, at coordinates (2240, 1110).

```
Catapult 1 1 1656 1124 2362 1072

ParkingSpace 1 -140
ParkingSpot 0 1 2340 1100
ParkingSpot 1 1 2240 1110
```

Appendix E

Center for Naval Analyses (CNA) Interdeparture Data

Figure E-1 shows the data reported by Jewell (Jewell, 1998) for launch interdeparture rates on the aircraft carrier flight deck including a 3rd order polynomial fit of the data. Jewell notes that the data comes from one week of operations on a carrier used in a previous study. Jewell's original presentation of this data did not include the 3rd order polynomial shown in the figure. In order to extract the numerical values from this chart, the figure was imported into the Engauge Digitizer software package[1]. This software is capable of extracting numerical relationships from a chart given the input image and specifications of the x- and y-axes and their ranges. The Engauge software returned a set of 102 points comprising the information presented in Fig. E-1. These points were then fit in the JMP PRO 10 statistical program with a 3rd-order polynomial fit, with both the F test of model fit and coefficient tests returning significant results (Tables E.1 and E.2). The equation is given as:

$$p = 0.3947704 + 0.004392 * time - 3.99E-05 * (time - 86.5334)^2 + 1.67E-07 * (time - 86.5334)^3 \quad (E.1)$$

Table E.1: Results of F test of model fit for JMP PRO 10 fit of CNA CDF data with 3rd order polynomial.

Source	**DF**	**Sum of Squares**	**Mean Square**	**F ratio**	**Prob > F**
Model	3	8.7080979	2.9027	9492.7616	<.0001
Error	98	0.0299665	0.00031		
C. Total	101	8.7380644			

[1] http://digitizer.sourceforge.net/

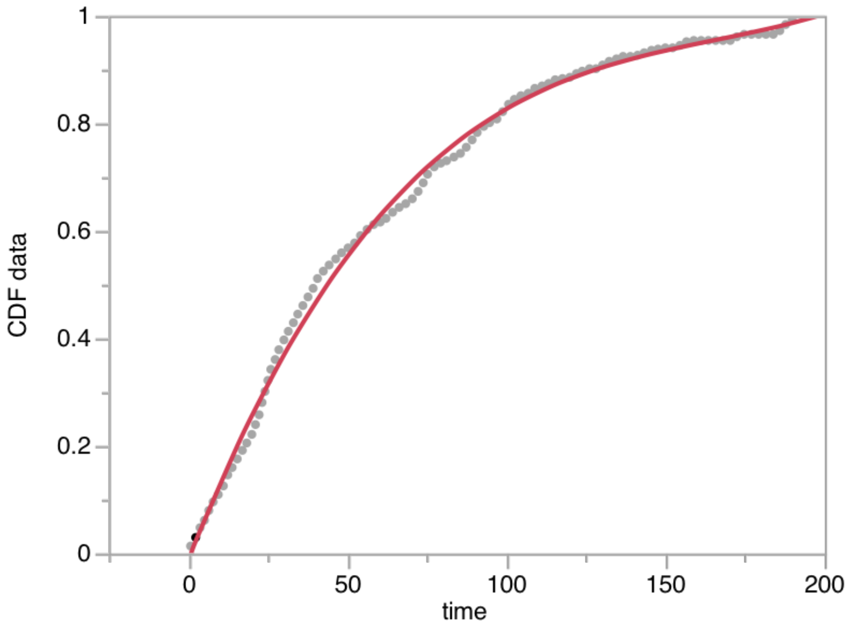


Figure E-1: Cumulative Density Function (CDF) of launch interdepartures from (Jewell, 1998).

Table E.2: Results of t tests for coefficients of JMP PRO 10 fit of CNA CDF data with 3rd order polynomial.

Term	**Estimate**	**Std Error**	**t Ratio**	**Prob$> \|t\|$**
Intercept	0.3947704	0.007481	52.77	<.0001
time	0.004392	0.000078	56.28	<.0001
$(\text{time-86.5334})^2$	-3.99E-05	7.00E-07	-56.96	<.0001
$(\text{time-86.5334})^3$	1.67E-07	1.32E-08	12.59	<.0001

This equation was then converted into a Probability Density Function (PDF) by taking the first differential with respect to X. Since the data represents true interdeparture rates for a system, it should be characterizable as a negative exponential function. The PDF developed from the differentiation of the CDF equation is plotted in blue in Figure E-2. These data points were then imported into JMP PRO 10 to formally fit a negative exponential to the data, done by transforming the Y variable (in this case, the probability value) into log(Y). The results of this fit also returned statistical significance (Table E.3 and E.4) for the model

$$Log(P) = -4.128056 - 0.0155665 * time \tag{E.2}$$

which is virtually identical to the more typical negative exponential form $\lambda * e^{-\lambda x}$,

$$P = 0.0155665 * e^{-0.0155665 * time} \tag{E.3}$$

which is treated as the appropriate negative exponential model. For this model, the value of λ corresponds to the average departure rate per second; the inverse of this corresponds to the average time between arrivals (for this model, 64.25 seconds).

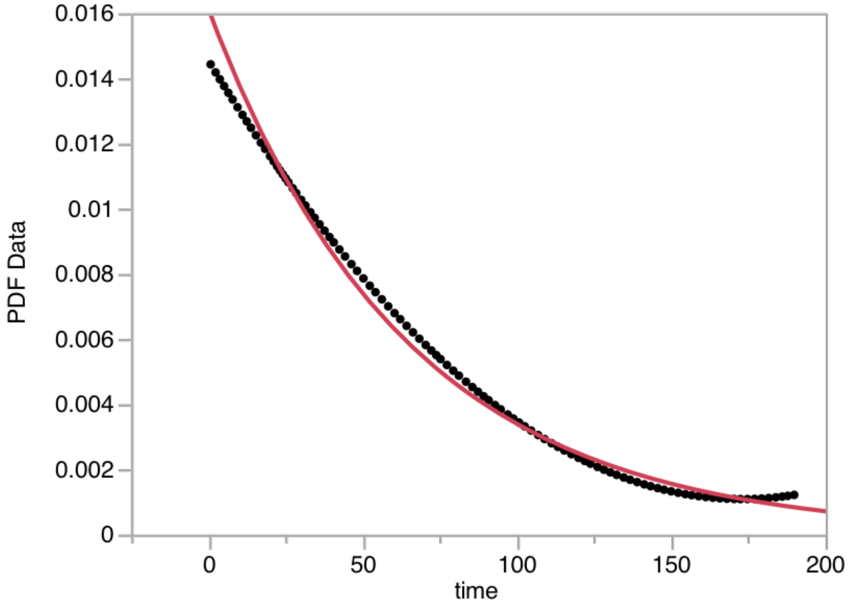


Figure E-2: PDF of launch interdepartures from (Jewell, 1998)

Table E.3: Results of F test of model fit for JMP PRO 10 fit of CNA PDF data with exponential function.

Source	**DF**	**Sum of Squares**	**Mean Square**	**F Ratio**	Prob > F
Model	1	78.076245	78.0762	8095.805	<.0001
Error	100	0.964404	0.0096		
C. Total	101	79.040648			

Table E.4: Results of t tests for coefficients of JMP PRO 10 fit of CNA PDF data with exponential function.

Term	**Estimate**	**Std Error**	**t Ratio**	**Prob>** $\lvert t \rvert$
Intercept	-4.128056	0.017851	-231.2	<.0001
time	-0.015566	0.000173	-89.98	<.0001

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix F

Multi-Agent Safety and Control Simulation (MASCS) Single Aircraft Test Results

Table F.1: Test results for distribution fits of the MASCS single aircraft calibration testing.

	Launch Duration	Taxi Duration	Takeoff Duration	Launch Preparation
K-S statistic	0.05773	0.06504	0.04409	0.05413
p	0.25992	0.15113	0.5886	0.33108
Mean μ	209.99	83.943	2.7513	120.16
Standard deviation σ	49.628	6.0198	0.25651	48.842

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix G

Description of Mission Queuing

This section provides a more extended description of the process of queuing at catapults on the flight deck and how this process influences the execution of the mission. Typically, operations allow one aircraft on the catapult and one more parked behind the jet blast deflector; this implies a queue size of two for an individual catapult and a size of four for a pair. Aircraft are allocated to a catapult as soon as a position in the queue opens. With four catapult open, each pair services an equal number of aircraft and the expectations are as listed above — allocations should be equally divided between forward and aft catapults. However, if only three catapults are used, the larger queue size of the pair of catapults interacts with the allocation pattern; as the pair is able to "store" more aircraft in their vicinity, the allocations may be slightly inefficient. At the end of the mission, if all three catapults have full queues, the pair will take four rounds of launches to clear its queue of four aircraft, with the single catapult requiring only two to clear its two aircraft. Figure G-1shows this allocation as time evolves. The leftmost image shows the initial allocation of launches to a team of four catapults. The queue at each catapult, over time, is noted in the numbered columns (catapult 1 on the left, to catapult 4 on the right). Colored blocks within each column denote (a) the presence of an aircraft in that catapult's queue and (b) the order in which they were allocated. Thus, block 1 was the first aircraft allocated, assigned to catapult 1. Block 6 was the sixth aircraft allocated to launch, assigned to catapult 2. The staggered, "checkered" pattern is the table is due to the alternating launches between pairs of catapults; while aircraft 2 was assigned second, it was assigned to catapult 2 and must wait for catapult 1 to launch aircraft 1. Similar patterning is shown for aircraft at catapults 3 and 4. In this case, aircraft 1 is shown as a red block; this is used to indicate the aircraft is in preparation to launch. In the middle image, aircraft 1 is now crossed out with a black line, indicating it has launched. Now that there is a spot free at the catapult, aircraft 9 is assigned to take its place (recall from Chapter 3, the Handler routine assigns aircraft as catapults become available). This pattern continues moving forward; in the rightmost image, after aircraft 3 and 4 have launched, aircraft 10 and 11 have been assigned, respectively, and aircraft 4 and 5 being preparing for launch. Continuing this build-out as shown, each pair of catapults launches one half of the allotted aircraft, and the time to complete the mission is approximate as $N/2 * \mu_launch$

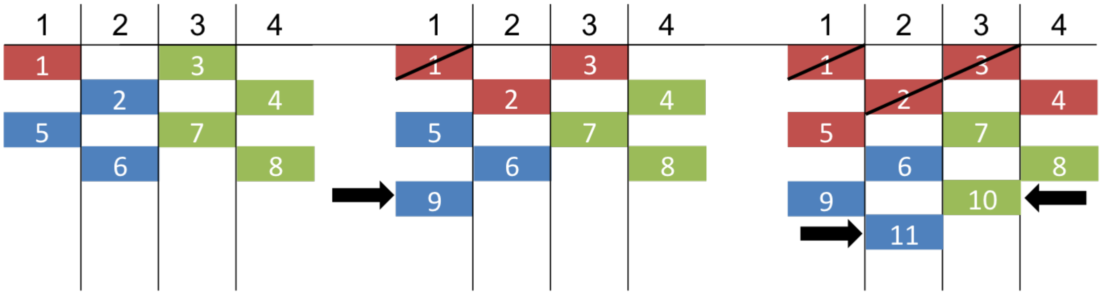


Figure G-1: Pattern of allocation to catapults. Blue/green coloring corresponds to pairing of catapults 1/2 and 3/4, respectively. Red indicates the launch is in process. A black diagonal line indicates that the launch has occurred and the aircraft is no longer in the system.

With only three catapults operating, the allocation pattern changes slightly (Figure G-2). One catapult (in this case, catapult 2) is no longer paired and free to operate on its own. The leftmost image in the figure shows the initial allocation of aircraft; four go to catapults 3 and 4, staggered in an alternating pattern. Only two can go to catapult 2, but launch consecutively. The remaining two images in the figure show how the allocation builds out from there; after aircraft 1 and 2 launch, aircraft 3 and 4 begin processing, and aircraft 7 and 8 are allocated. This continues for the rest of the allocation, with catapult 2 launching about as many aircraft as the paired catapults 3 and 4.

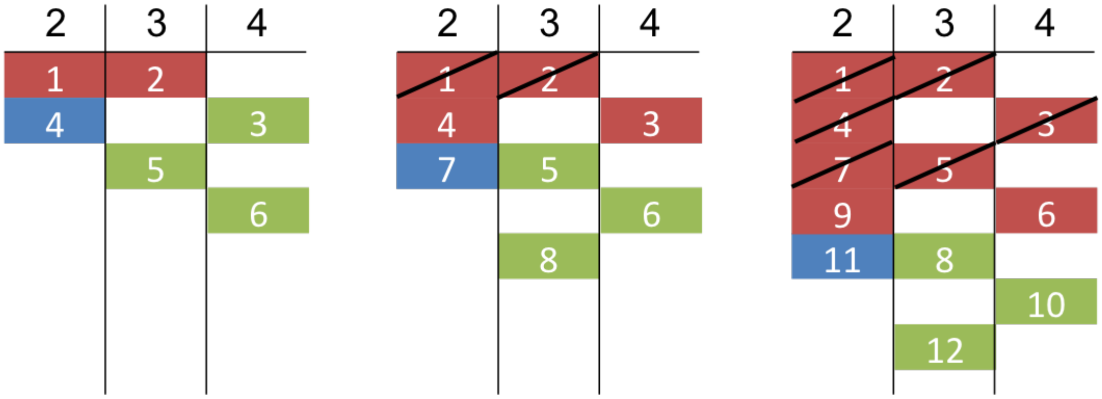


Figure G-2: Pattern of allocation to a set of three catapults. Blue/green coloring corresponds to pairing of catapults 1/2 and 3/4, respectively. Red indicates the launch is in process. A black diagonal line indicates that the launch has occurred and the aircraft is no longer in the system.

Figure G-3 shows how this allocation plays out through a 22 aircraft launch event. As described above, the four catapult case maintains an even allocation across the pairs and both complete at the same time (11 rounds of launches). For the three catapult case, because of the imbalance in allocation the results from the queuing arrangement, catapult 2 finishes after only 10 launches and the catapult 3/4 pair finishes after 12 rounds, one more than the four catapult case. This implies that the three catapult case should require one additional round of launches than the four catapult case under this idealized queue model of the flight deck. Given that there will be significant stochasticity in the execution of the model, and that assignments are dependent on the geometric arrangement of the flight deck at the time of assignment, it is not unreasonable to assume that this difference will not be precisely equal to one, but should be close.

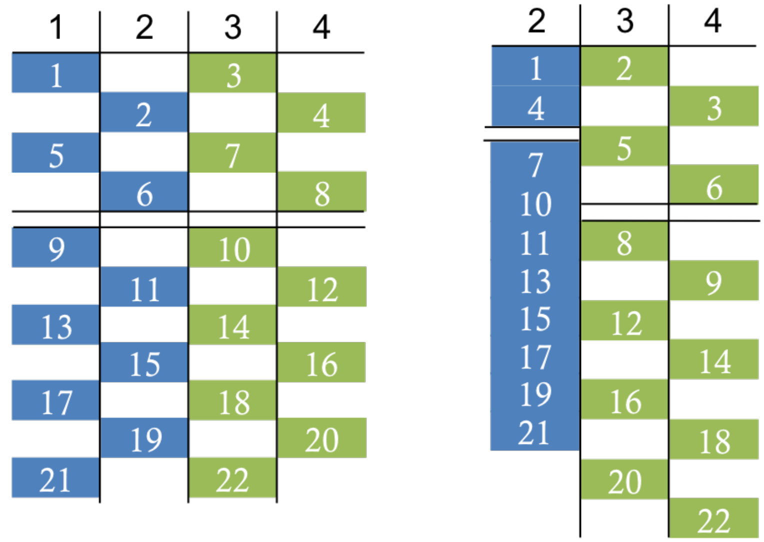


Figure G-3: Build-out of 22 aircraft launch events with four (left) and three (right) available catapults.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix H

Multi-Agent Safety and Control Simulation (MASCS) Mission Test Results

Table H.1: Results of distribution fits of interdeparture times for CNA and MASCS test data. Results include λ and p-values for fits from Easyfit along with λ values and 95% confidence intervals from JMP PRO 10.

Data source	Easyfit λ value	EasyFit K-S test p-value	JMP λ	Lower range	Upper range
CNA	0.0156	0.0717	0.0156	-	-
18-3	0.0144	0.2813	0.0144	0.0108	0.0187
18-4	0.0164	0.1554	0.0164	0.0123	0.0213
20-3	0.0153	0.1251	0.0153	0.0114	0.0198
20-4	0.0160	0.1358	0.0159	0.0120	0.0207
21-3	0.0141	0.2755	0.0141	0.0105	0.0184
21-4	0.0152	0.1596	0.0152	0.0113	0.0197
22-3	0.0145	0.4500	0.0145	0.0108	0.0190
22-4	0.0154	0.2446	0.0154	0.0115	0.0201
23-3	0.0144	0.1782	0.0144	0.0108	0.0188
23-4	0.0153	0.2254	0.0153	0.0115	0.0199
24-3	0.0146	0.0999	0.0146	0.0109	0.0191
24-4	0.0159	0.2565	0.0159	0.0119	0.0208
26-3	0.0150	0.1958	0.0150	0.0110	0.0198
26-4	0.0159	0.2126	0.0159	0.0117	0.0210
30-3	0.0147	0.1089	0.0147	0.0110	0.0191
30-4	0.0154	0.1900	0.0154	0.0115	0.0200
34-3	0.0149	0.1981	0.0149	0.0111	0.0195
34-4	0.0153	0.3791	0.0153	0.0114	0.0200

Table H.2: JMP PRO 10 output for F-test of model fit for LD values using three catapults.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Model	1	254.67542	254.675	1735.526	<.0001
Error	7	1.0272	0.147		
C. Total	8	255.70262			

Table H.3: JMP PRO 10 output for t-test of coefficients for linear fit of LD values using three catapults.

Term	Estimate	Std Error	t Ratio	Prob > $\|t\|$
Intercept	2.4130162	0.65966	3.66	0.0081
Num_air	1.1130865	0.026719	41.66	<.0001

Table H.4: JMP PRO 10 output for F-test of model fit for LD values using four catapults.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Model	1	267.30216	267.302	776.9774	<.0001
Error	7	2.4082	0.344		
C. Total	8	269.71036			

Table H.5: JMP PRO 10 output for t-test of coefficients for linear fit of LD values using four catapults.

Term	Estimate	Std Error	t Ratio	Prob> $\|t\|$
Intercept	0.0860649	1.010041	0.09	0.9345
Num_air	1.1403459	0.04091	27.87	<.0001

Table H.6: JMP Pro 10 output of Levene test of equal variances for launch preparation sensitivity tests using 22 aircraft and 3 catapults.

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	0.7676	2	87	0.4672
Brown-Forsythe	0.2117	2	87	0.8097
Levene	0.2099	2	87	0.8111
Bartlett	0.7683	2	.	0.4638

Table H.7: JMP Pro 10 output of an ANOVA test for launch preparation sensitivity tests using 22 aircraft and 3 catapults.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
22-3	2	259.75601	129.878	29.1344	<.0001
Error	87	387.83637	4.458		
C. Total	89	647.59238			

Table H.8: JMP Pro 10 output of Tukey test for launch preparation sensitivity tests using 22 aircraft and 3 catapults.

Level	- Level	Difference	Std Err Dif	Lower CL	Upper CL	p-Value
22-3 +10%	22-3 -10%	4.155106	0.5451538	2.855189	5.455023	<.0001
22-3 +10%	22-3 base	2.27535	0.5451538	0.975433	3.575267	0.0002
22-3 base	22-3 -10%	1.879756	0.5451538	0.579839	3.179673	0.0025

Table H.9: JMP Pro 10 output of Levene test of equal variances for taxi speed sensitivity tests using 22 aircraft and 3 catapults.

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	0.3054	2	87	0.7376
Brown-Forsythe	0.1777	2	87	0.8375
Levene	0.2357	2	87	0.7906
Bartlett	0.2962	2	.	0.7436

Table H.10: JMP Pro 10 output of an ANOVA test for taxi speed sensitivity tests using 22 aircraft and 3 catapults.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
22-3	2	12.48402	6.24201	1.0913	0.3403
Error	87	497.60072	5.71955		
C. Total	89	510.08474			

Table H.11: JMP Pro 10 output of Levene test of equal variances for collision prevention parameter sensitivity tests using 22 aircraft and 3 catapults.

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	1.1804	2	87	0.312
Brown-Forsythe	1.4554	2	87	0.2389
Levene	1.4402	2	87	0.2425
Bartlett	1.1048	2	.	0.3313

Table H.12: JMP Pro 10 output of an ANOVA test for collision prevention parameter sensitivity tests using 22 aircraft and 3 catapults.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
22-3	2	17.85385	8.92693	1.5253	0.2233
Error	87	509.18622	5.85272		
C. Total	89	527.04007			

Table H.13: JMP Pro 10 output of Levene test of equal variances for taxi speed sensitivity tests using 22 aircraft and 4 catapults.

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	0.184	2	87	0.8323
Brown-Forsythe	0.5919	2	87	0.5555
Levene	0.5318	2	87	0.5895
Bartlett	0.142	2	.	0.8676

Table H.14: JMP Pro 10 output of an ANOVA test for taxi speed sensitivity tests using 22 aircraft and 4 catapults.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
22-4	2	10.33376	5.16688	1.1123	0.3334
Error	87	404.14883	4.64539		
C. Total	89	414.48259			

Table H.15: JMP Pro 10 output of a Levene test of equal variances for collision prevention parameter sensitivity tests using 22 aircraft and 4 catapults.

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	2.2512	2	87	0.1114
Brown-Forsythe	1.4837	2	87	0.2325
Levene	1.5309	2	87	0.2221
Bartlett	2.6178	2	.	0.073

Table H.16: JMP Pro 10 output of an ANOVA test for collision prevention parameter sensitivity tests using 22 aircraft and 4 catapults.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
22-4	2	43.43476	21.7174	4.6278	0.0123
Error	87	408.27087	4.6928		
C. Total	89	451.70563			

Table H.17: JMP Pro 10 output of a Tukey test for collision prevention parameter sensitivity tests using 22 aircraft and 4 catapults.

Level	- Level	Difference	Std Err Dif	Lower CL	Upper CL	p-Value
22-4 CA+25%	22-4 CA-25%	1.643544	0.5593311	0.309822	2.977267	0.0116
22-4 base	22-4 CA-25%	1.203622	0.5593311	-0.130101	2.537345	0.0855
22-4 CA+25%	22-4 base	0.439922	0.5593311	-0.893801	1.773645	0.7123

Table H.18: JMP Pro 10 output of a Levene test of equal variance for taxi speed sensitivity tests using 34 aircraft and 3 catapults.

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	0.6872	2	87	0.5057
Brown-Forsythe	0.7932	2	87	0.4556
Levene	0.8184	2	87	0.4445
Bartlett	0.8447	2	.	0.4297

Table H.19: JMP Pro 10 output of an ANOVA test for taxi speed sensitivity tests using 34 aircraft and 3 catapults.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
34-3	2	83.3424	41.6712	4.9753	0.009
Error	87	728.6756	8.3756		
C. Total	89	812.018			

Table H.20: JMP Pro 10 output of a Tukey test for taxi speed sensitivity tests using 34 aircraft and 3 catapults.

Level	- Level	Difference	Std Err Dif	Lower CL	Upper CL	p-Value
34-3 taxi-25	34-3 taxi+25	2.328156	0.747243	0.546357	4.109954	0.007
34-3base	34-3 taxi+25	1.483278	0.747243	-0.29852	3.265076	0.122
34-3 taxi-25	34-3base	0.844878	0.747243	-0.93692	2.626676	0.4979

Table H.21: JMP Pro 10 output of a Levene test of equal variance for collision prevention parameter sensitivity tests using 34 aircraft and 3 catapults.

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	1.3729	2	87	0.2588
Brown-Forsythe	1.6028	2	87	0.2072
Levene	1.7374	2	87	0.182
Bartlett	1.4627	2	.	0.2316

Table H.22: JMP Pro 10 output of an ANOVA test for collision prevention parameter sensitivity tests using 34 aircraft and 3 catapults.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
34-3 C	2	3.90398	1.95199	0.2293	0.7956
Error	87	740.75	8.51437		
C. Total	89	744.65397			

Table H.23: JMP Pro 10 output of a Levene test of equal variance for collision prevention parameter sensitivity tests using 34 aircraft and 4 catapults.

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	0.6346	2	87	0.5326
Brown-Forsythe	0.4903	2	87	0.6141
Levene	0.4355	2	87	0.6484
Bartlett	0.5318	2	.	0.5876

Table H.24: JMP Pro 10 output of a an ANOVA for launch preparation time sensitivity tests using 34 aircraft and 4 catapults.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
34-4	2	517.4101	258.705	34.4374	<.0001
Error	87	653.5734	7.512		
C. Total	89	1170.9835			

Table H.25: JMP Pro 10 output of a Tukey test for launch preparation time sensitivity tests using 34 aircraft and 4 catapults.

Level	- Level	Difference	Std Err Dif	Lower CL	Upper CL	p-Value
34-4 +10%	34-4 -10%	5.873072	0.7076882	4.185593	7.560552	<.0001
34-4 +10%	34-4 base	2.964311	0.7076882	1.276831	4.651791	0.0002
34-4 base	34-4 -10%	2.908761	0.7076882	1.221281	4.596241	0.0003

Table H.26: JMP Pro 10 output of Levene test of equal variance for taxi speed sensitivity tests using 34 aircraft and 4 catapults.

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	2.7878	2	87	0.0671
Brown-Forsythe	2.3031	2	87	0.106
Levene	2.6456	2	87	0.0767
Bartlett	2.9422	2	.	0.0528

Table H.27: JMP Pro 10 output for an ANOVA test for taxi speed sensitivity tests using 34 aircraft and 4 catapults.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
34-4	2	55.10413	27.5521	3.8031	0.0261
Error	87	630.28446	7.2446		
C. Total	89	685.38859			

Table H.28: JMP Pro 10 output of a Tukey test for taxi speed sensitivity tests using 34 aircraft and 4 catapults.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
34-4	2	55.10413	27.5521	3.8031	0.0261
Error	87	630.28446	7.2446		
C. Total	89	685.38859			

Table H.29: JMP Pro 10 output of Levene test of equal variance for collision prevention parameter sensitivity tests using 34 aircraft and 4 catapults.

Test	F Ratio	DFNum	DFDen	Prob > F
O'Brien[.5]	1.2733	2	87	0.2851
Brown-Forsythe	1.0896	2	87	0.3409
Levene	1.078	2	87	0.3448
Bartlett	1.1711	2	.	0.31

Table H.30: JMP Pro 10 output for an ANOVA test for collision prevention parameter sensitivity tests using 34 aircraft and 4 catapults.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
34-4	2	76.49218	38.2461	5.4986	0.0056
Error	87	605.13776	6.9556		
C. Total	89	681.62995			

Table H.31: JMP Pro 10 output for a Tukey test for collision prevention parameter sensitivity tests using 34 aircraft and 4 catapults.

Level	- Level	Difference	Std Err Dif	Lower CL	Upper CL	p-Value
34-4 +25%	34-4 CA-25%	2.00675	0.6809604	0.383	3.630498	0.0114
34-4 base	34-4 CA-25%	1.900217	0.6809604	0.27647	3.523964	0.0176
34-4 +25%	34-4 base	0.106533	0.6809604	-1.51721	1.730281	0.9866

Appendix I

Unmanned Aerial Vehicle (UAV) Models

Chapter 5 discusses the Multi-Agent Safety and Control Simulation (MASCS) experimental test program and provided an overview of the Unmanned Aerial Vehicle (UAV) models encoded into MASCS. This appendix provides additional information regarding those models.

I.1 Local Teleoperation

Local Teleoperation (LT) relocates the pilot from the cockpit of the aircraft to a Ground Control Station (GCS) nearby that enables them to control their vehicles using the same control inputs they would normally supply if they were within the vehicle (”stick and rudder inputs”). “Local” teleoperation assumes that this control is done via Line-Of-Sight (LOS) communications and that there is no significant latency in operations. Additionally, the use of a GCS implies that the operator is now receiving visual information about the world through cameras on board the vehicle. These may have a limited field of view and thus impair the operator’s ability to observe the location of Directors and other vehicles in the world.

Research by several groups has addressed teleoperated driving ((McLean & Prescott, 1991; McLean et al., 1994; Spain & Hughes, 1991; Scribner & Gombash, 1998; Oving & Erp, 2001; Smyth et al., 2001)), a task similar to what will occur with teleoperated aircraft on the flight deck. The research has shown that the move from direct viewing within the vehicle to the use of a teleoperated display with computer/video screen has some detrimental effect. The increase in task completion time for teleoperated systems ranged from 0% to 126.18%, with a mean of 30.27% and a median of 22.12%. The tasks used in these experiments typically required drivers to navigate a marked course that required several turning maneuvers. In these tasks, drivers showed a significantly higher error rate (defined as the number of markers or cones struck during the test) as compared to non-teleoperated driving. Percent increases in errors ranged from 0% to 3650.0% with a median of 137.86% and a mean of 60%. However, these values are difficult to translate into direct error rates on the flight deck. Conceptually, they signify a difficulty in high-precision alignment tasks, a skill required while aligning and driving on to a catapult. A failure to align correctly to a catapult then requires other crew to physically push the vehicle off of the catapult to realign it. Interviews and observational data suggests this takes at least 15 to 45 seconds.

On the aircraft carrier flight deck, teleoperated vehicles would still be directed around the deck through the zone coverage routing structure. The research described above suggests that operators would drive more slowly while under guidance from Directors and would have more difficulty in finding Directors on the deck due to their constrained field of view. The research does not suggest, however, that operators would fail to notice their designated Directors once they lie within their field of view, nor that that they would fail to properly comprehend instructions. Additionally, though not explored in the research, the operator's understanding of context should prevent her from executing an incorrect action. However, because of the limitations of view field and lack of tactile and proprioceptive sensory inputs, it is likely that local teleoperation systems will have difficult in properly aligning to catapults.

I.1.1 Local Teleoperation Safety Implications

Just as with manual control, safety in local teleoperation is provided by a combination of the zone coverage routing structure, the planning of aircraft assignments, and reactive collision avoidance (Dynamic Separation) on the part of the UAV operator. However, this latter property will be affected by the limited view field of the cameras and the screen on the GCS and thus should be less effective than manual control operations.

Among the available safety protocols, it would be expected that Area+Temporal would be the worst selection for productivity, while Area Separation would be the worst for safety metrics. The problems lie in confining vehicles to small areas: in Area+Temporal, manned aircraft are confined to a small area and cannot exploit the efficiency of using all four catapults simultaneously. In Area Separation, UAVs are confined to operating in a single highly-congested area where the limited field of view should decrease their ability to avoid imminent collisions. Accordingly, it would be expected that Temporal Separation would be best for both safety and productivity — it allows the manned aircraft (able to safely taxi high-congestion areas) to depart first, freeing up the remaining space on the deck for the unmanned vehicles and increasing their efficiency. This both decreases any detrimental effects of safety on the part of the UAVs while also increasing their productivity by providing an open deck. Under this same logic, Area+Temporal should also do well in safety measures as it operates in largely the same manner as Temporal Separation. Its performance in productivity will be worse as it overly constrains manned aircraft access to catapults during their phase of operations.

I.2 Remote Teleoperation

Remote Teleoperation (RT) systems are almost identical to Local Teleoperation systems (ground control station with view constrained by cameras), with the lone change being that the operator is now much farther away (hundreds to thousands of miles) and cannot communicate with the vehicle using LOS methods. This typically requires the use of satellite communications, at the cost of significant latency in transmissions. The first effect is that there is a delay in beginning to execute actions. Once the Director issues a command to the UAV operator, the video feed must first travel to the operator, whose response must then travel back to the vehicle for execution. Secondly, lag effects how operators input tasks to the system. Research has shown that a naive teleoperator inputs tasks assuming there is no lag, requiring multiple corrective actions to finally reach the target and much greater amounts of time. Sheridan and Ferrell (Sheridan & Ferrell, 1963) observed that more experienced teleoperators used an “open-loop” control pattern, inputting tasks anticipating the lag, then waiting for the tasks to complete before issuing new commands. Using this strategy prevents the error in one task from overly affecting the next task, thus reducing the overall number of errors and time required to complete the tasks at hand. However, it still requires much more time than non-latency tasks. This has important repercussions in modeling how operators execute taxi commands on deck and is discussed in the next section.

A review of literature regarding these experiments, including the use of remotely teleoperated ground vehicles, shows that lag is multiplicative in its effects even in the best case: one second of latency translates into much more than one second of additional time. Fifteen different papers (Kalmus et al., 1960; Sheridan & Ferrell, 1963; Hill, 1976; Starr, 1979; Mar, 1985; Hashimoto et al., 1986; Bejczy & Kim, 1990; Conway et al., 1990; Arthur et al., 1993; Carr et al., 1993; MacKenzie & Ware, 1993; Hu et al., 2000; Lane et al., 2002; Sheik-Nainar et al., 2005; Lum et al., 2009), totaling 56 different experimental tests, show that the percentage change in task completion time *per second of lag* ranges from 16% to 330%, with a mean of 125% and a median of 97%. Thus, at the median, one second of lag results in a near doubling of the time to complete a task (a 100% increase); two seconds of lag equals a 200% increase. These tests addressed both cases where operators where directly observing the system they were controlling, or while receiving information via a video or computer display.

Remotely teleoperated vehicles are still expected to operate in a zone coverage routing scheme, with Aircraft Directors using hand gestures to provide instructions that captured by vehicle cameras and relayed to the remote teleoperators. The ability of UAV operators to detect Directors is affected by the limited field of view of the onboard cameras, just as in local teleoperation, and they are still expected to properly comprehend instructions when provided. Their ability to execute those actions is affected by the signal latency; taxi speeds are likely to be greatly reduced, and high-accuracy alignment tasks (aligning to catapult) are likely to suffer much higher error rates and take more time. Furthermore, the "open-loop" nature of control means that they will likely not be perfect in adhering to Director instructions as to when to stop taxiing or turning.

I.2.1 Remote Teleoperation Safety Implications

Just as with manual control and local teleoperation, safety in remote teleoperation is provided by a combination of the zone coverage routing structure, the planning of aircraft assignments, and reactive collision avoidance (Dynamic Separation) on the part of the UAV operator. However, this latter property will be affected by the limited view field of the cameras, the screen on the GCS, and also signal latency, making it even less effective than local teleoperation systems. The existence of signal latency also means that vehicles will have trouble stopping in Dynamic Separation events, and the likelihood of collisions in congested traffic areas will be greatly increased.

The performance of Remote Teleoperation safety protocols will be quite similar to that of Local Teleoperation — the same concerns about operations apply to all phases. However, because of the existence of latency in the signal, the performance of Remote Teleoperation along all measures (productivity and safety) is expected to be worse than Local Teleoperation and worst out of all available control architectures.

I.3 Gestural Control (advanced teleoperation)

The third control architecture, Gestural Control (GC), places the operator in the physical environment nearby the unmanned vehicle they are controlling. Here, control inputs are communicated via hand and arm movements, providing basic action commands such as "drive forward" to unmanned vehicles. On the aircraft carrier flight deck, this means the elimination of the pilot/operator all together: the Aircraft Directors on deck now assume direct control over the unmanned vehicles. Gestural control UAVs would be designed to accept the same hand signals as human pilots, however, meaning that this would be a relatively simple transition in operations. These vehicles would again be routed in zone coverages, and the evolution of operations on the flight deck will look very much like that of manual control operations. However, the ability of vehicles to detect Directors, comprehend their commands, and execute tasks will not be as reliable and human pilots.

Gestural-control systems rely on stereo-cameras to detect visual and depth information about the world. Onboard processing uses this information to develop a model of the world, denoting what objects should be monitored before tracking and interpreting their gestures. A review of recent research shows many systems with very high accuracy rates for data sets involving only a small number (less than 20) of simple actions in relatively simple, "clean" (very little clutter) environments. For larger action sets, involving more complex hand and arm gestures, accuracy is still fairly high, but often well below 100%. Over all of the literature reviewed, systems display a mean failure rate of 11.16% and median rate of 8.93% for mostly simple, clean data sets. However, Song et al. (Song et al., 2012) provide an example of a system built specifically for the aircraft carrier flight deck environment. Their most recent results achieve almost 90% accuracy for a limited set of gestures and requires 3 seconds of processing time to register commands. This system utilizes a Bumblebee stereo camera, which has optional fields of view of 43°, 65°, or 100°.

Due to the nature of the data sets used most broadly in the research, it is difficult to define exactly how well these systems would perform on the flight deck. A variety of issues — sun glare off of the ocean, multiple similarly-dressed individuals, a continually changing background, and the variability of the humans providing gestures – can substantially affect the systems' ability to detect operators and comprehend the instructions they provide. Song et al.'s current 10% failure rate in task comprehension would increase substantially when expanded to the full vocabulary of tasks (over 100 distinct commands) and placed in the world. Given this alternative, maintaining a failure rate equivalent to the mean error rate of 11.16% may be permissable, given certain safety features on the vehicles. Typically, these systems include a default "stop" command, such that if the tracked individual providing commands is lost, or a match for the gesture is not found, the vehicle comes to a complete halt. Modeling in MASCS will make the assumption that this is a permissable error rate for Director and vehicle detection (within the vehicle's field of view) and task comprehension. For each case, failure to detect/comprehend results in a penalty of 10 seconds, modeling the time required for a Director to recognize the failure and re-issue the command/be recognized by the system.

I.3.1 Gesture Control Safety Implications

Safety in Gestural Control, just as in local and remote teleoperation, is again predicated upon the orderly assignment of aircraft tasks, the structure of zone coverage routing and traffic management, and in this case the ability of vehicles themselves to monitor nearby activity and avoid collisions (Dynamic Separation). Just as in the teleoperation conditions, the ability of vehicles to do this is mitigated by the failure rate of recognition and the required processing time. An incoming vehicle may be fully within the field of view, but if the vehicle fails to recognize it as a vehicle or takes too long to process, it will fail to stop and a collision might still occur.

In terms of safety, it is unclear as to which safety protocols perform best for these systems. Gesture-controlled vehicles, because of the default stop command, should be better able to taxi in high traffic areas but with some cost to productivity (they will likely stop more often due to the sheer number of things in their fields of view). However, as with the teleoperation systems, reducing the density of vehicles on the deck by moving manned aircraft early in the mission (through Temporal Separation or Area+Temporal) should increase both productivity and safety in the system. Temporal

Separation should also provide the best performance in productivity, moving manned aircraft first and allowing UAVs more open space to operate on the flight deck. However, it is unclear which of the remaining protocols will provide the worst overall performance.

I.4 Human Supervisory Control Systems

The last two forms of control architectures are variations on more modern "human supervisory control" systems. In both cases, vehicles contain advanced autonomous capabilities and only require abstract task descriptions such as "go to this area and monitor for any activity" from the operator. The vehicles include sufficient guidance, navigation, and control systems onboard the vehicle to understand where they are in the world, where their final destination is, how to travel to that destination, and be capable of executing tasks upon arrival. This typically requires a suite of sensors (GPS, inertial measurement units, and various cameras and range detection systems) to acquire data for the logic systems onboard.

For these systems, a human "Supervisor" manages all vehicles through a centrally networked computer system, meaning that Directors on deck are no longer responsible for communicating instructions to vehicles. Information on the status of all vehicles is fed to the central system, where the operator decides what tasks vehicles should perform at what times. In some cases, the operator is assigning tasks individually to each vehicle; in others, the operator may replan all tasks at once (likely with the help of an intelligent planning algorithm). The latter may also include a path planning algorithm that will attempt to pre-plan all vehicle paths to minimize the likelihood of collisions or traffic congestion. The autonomy embedded in the vehicle is responsible for resolving the abstract tasks provided by the Supervisor into low-level control inputs for the throttle, steering, and other mechanisms. These vehicles would also contain reactive collision avoidance mechanisms (Dynamic Separation systems) in the case that a crew member or other vehicle crosses their path without warning. While cameras might be used for a variety of sensing, it is probably that all crew and vehicles would also be equipped with GPS beacons and other sensors that broadcast their location to the autonomous vehicle network, ensuring that they are seen and accounted for before a collision. Despite the advanced capabilities, due to the environmental conditions on deck (pitching, rolling, etc.) it is still likely that vehicles will have difficulty in high-precision tasks like aligning to the catapult and navigating through close traffic. While the advanced autonomy does not require crew on deck to provide tasks, this crew may be required as safety escorts to monitor the movement of the vehicle and help correct any errors in catapult alignment. This would require fewer crew than standard manual control operators, and in combination with the advanced networked planning, would allow unmanned vehicles to travel faster than in other control architectures. Speeds equivalent to manual control cases are not unreasonable.

As noted above, these systems take one of two different forms. In the first (Vehicle-based Human

Supervisory Control, VH), the operator provides tasks to vehicles individually, addressing them in series. Once a vehicle finishes its tasks, it must wait for the operator to provide new instructions. Servicing each vehicle takes some variable amount of time dependent on the operator's current workload, how well they are aware of the state of the system, and what tasks are being issued. Past research has shown that operators can control up to 8 ground vehicles of 12 aircraft (Cummings & Mitchell, 2008), suitable for a flight deck where no more than 8-12 vehicles are taxiing at any given time. Previous work by Nehme (Nehme, 2009) modeled human operators performing a variety of tasks, with values ranging from averages of 1-2 seconds to 25 seconds for various types of tasks ranging from defining waypoints to verifying sensor data. In MASCS, waypoint definition is a major responsibility of the Supervisor in Vehicle-Based Human Supervisory Control (VBSC), and the Supervisor is responsible not only for assigning waypoints but also for defining vehicle paths on the flight deck.

In system-based operations (SH), the operator uses a planning algorithm to replan tasks for all vehicles simultaneously. SH systems also require the existence of a path planning agent to handle the deconfliction of vehicle paths on the flight deck. The planning of tasks must happen once at the beginning of the mission, and because schedules degrade over time due to uncertainty in the world, may happen periodically throughout. Each time replanning occurs, it takes some time to accomplish, during which vehicles may be waiting for instructions. This replanning may also significantly change prior instructions, rerouting aircraft to new destinations. Such systems typically assign a queue of tasks to a single vehicle for execution over a period of time, allowing the operator to focus on monitoring the performance of the system as a whole, rather than individual vehicles. In past work with the Deck operations Course of Action Planner (DCAP) system built for carrier deck planning (Ryan et al., 2011; Ryan, 2011), the time required to replan ranged from 5 to 20 seconds, with replans only occurring in the event of failures (once per 20 minute mission). In a similar system termed OPS-USERS (Fisher, 2008; Cummings, Clare, & Hart, 2010; Clare, 2013), replanning times ranged from 2 to 15 seconds. Replans should not occur after the last vehicle is allocated, however, as the cost of making changes outweighs the benefits.

In both cases, vehicles are not engaging in visual detection as in the previous control architectures; they receive their tasks from the central network through direct communication and translation. This also means that task comprehension does not occur in similar fashions to other architectures and should be more reliable. For both of these elements, failure rates are not applicable for the system. However, vehicles may still have issue in executing tasks. The difficulties affecting Gestural Control automation are the same for V- and SH systems, and failure rates should be the same.

I.4.1 Human Supervisory Control (HSC) Safety Implications

For both of these systems, Dynamic Separation is accomplished through collision avoidance systems onboard the vehicles that utilize both local sensor information and data from the network to monitor the location of nearby vehicles. For avoiding other networked vehicles, these systems should be flawless in their execution, performing as well as human operators in navigating the deck.

In terms of safety protocol performance, both VBSC and System-Based Human Supervisory Control (SBSC) systems should perform better than all previous control architectures, given their higher expected reliability and improved autonomy. Because of this performance, it is unclear that any one safety protocol will generate better performance than another in terms of either productivity or safety. These systems are the closest, in terms of performance, to manually controlled vehicles and do not have the same limitations and errors of other systems. While some variation in performance would not be surprising, it is unclear as to what these variations will be.

Appendix J

UAV Verification

Chapter 5 described the development of the five Control Architecture models and included results showing the differences in performance of the full models of each. This appendix provides additional data showing the effects of introducing individual elements to each of the models. As noted previously in Figure 5-5, creating the UAV control architecture models required modifying parameters in or introducing new parameters to the baseline Manual Control model. For some control architectures, this required few changes, while others required more substantial changes either to the models or to the structure of operations. For each model, parameters were introducing individually then tested using the single aircraft calibration scenario with 300 replications being performed. This Appendix presents the data for each of these individual vehicle models to highlight how changes in parameters affected the performance of individual vehicle models.

For Local Teleoperation (LT), the two major changes were to slightly restrict the field of view from 85° to 65° and to increase the rate of catapult alignment failures from 0% to 5%. The results of this appear in Figure J-1, which plots three lines: the first to show the results of the baseline Manual Control model (gray), one to show the empirical observation used in calibration testing (orange), and a third for the UAV model (blue), which includes whiskers for standard deviation. The LT model has two data points: the first point, labeled "LT +FOV" denotes that the first variable changed was the field-of-view of the aircraft. The next point, "LT +CatFail" adds the 5% chance to fail at catapult alignment. As can be seen from the chart, neither of these changes makes much difference to results, which are quite similar to the original Manual Control (MC) baseline.

Figure J-2 shows similar data for the Remote Teleoperation (RT) model. The first change here introduced both FOV and catapult alignment failure changes, given the lack of effect of field of view on its own for the LT model. The second change introduced the 1.6 second latency at the start of each commanded task, with the third introducing modifications to the end of tasks due to operator behavior. For this model, the introduction of the initial latency has the greatest effect on results,

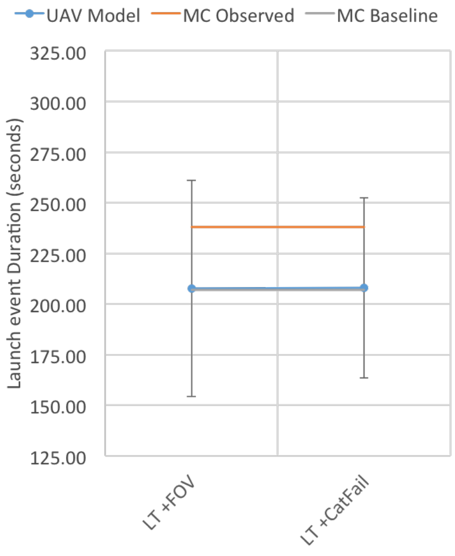


Figure J-1: Mean launch event duration time for Local Teleoperation (LT) control architecture with incremental introduction of features. Whiskers indicate ±1 standard deviation.

increasing mission duration by 13 seconds on average (216.21 to 228.96 seconds). However, data for each case is still within one standard deviation of the original MC baseline.

Data for the Gestural Control (GC) is shown in Figure J-3, with the first model again beginning with changes to the field of view and catapult alignment failure rate. The second data point include the effects of failing to recognize and Director and the associated time penalties. This appears to have no effect on results. The third model introduces the effects of failing to recognize commands, which could occur multiple times in series with time penalties accruing for each. This clear shows a significant change in operations, increasing the average launch event duration from 200.65 seconds to 221.66 seconds. The final model adds further latency into the system, modeling the processing time of the GC vehicles (uniform, 0-3 seconds). This has even further effects, increasing the average launch event duration to 248.62 seconds and moving nearly a full standard deviation away from the MC baseline.

Introducing the VH model required not only changes to vehicle parameters but also changes to their interactions with the crew. The first model in Figure J-4 removes the Director crew from operations; vehicles are no longer constrained to the zone coverage routing system. These results suggest that the removal of the crew may provide a slight benefit to operations, as the mean time to complete the launch event is smaller than the MC Baseline (196.38 seconds vs. 209.99 seconds). The second model catapult alignment failures to 2.5%, again with only minor effects. The third model introduces the Handler Queueing system, in which vehicles must repeatedly request instructions from the Handler. This results in a substantial increase in launch event duration, increasing the average time from 198.39 seconds to 229.31 seconds, and has an effect similar to that of the GC

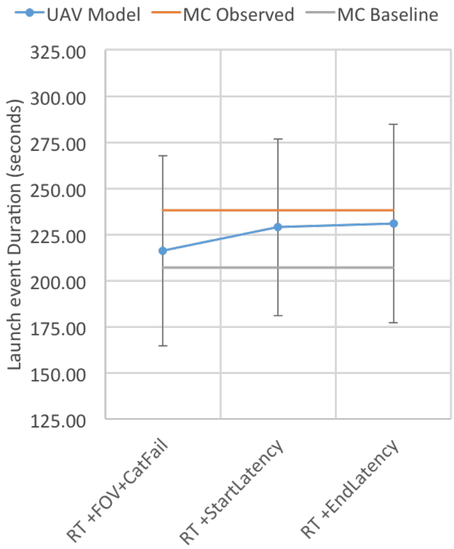


Figure J-2: Mean launch event duration time for Remote Teleoperation (RT) control architecture with incremental introduction of features. Whiskers indicate ±1 standard deviation.

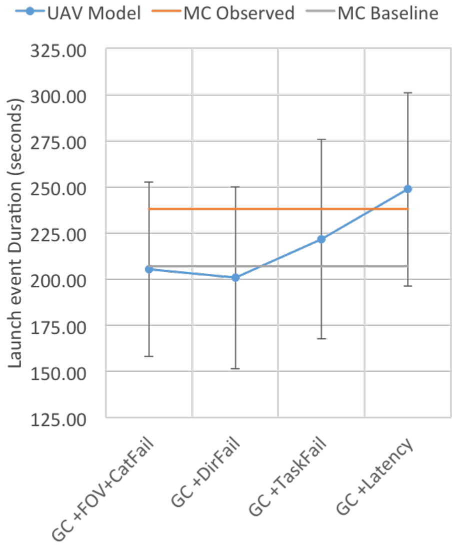


Figure J-3: Mean launch event duration time for Gestural Control (GC) control architecture with incremental introduction of features. Whiskers indicate ±1 standard deviation.

processing latency and the RT communications latency.

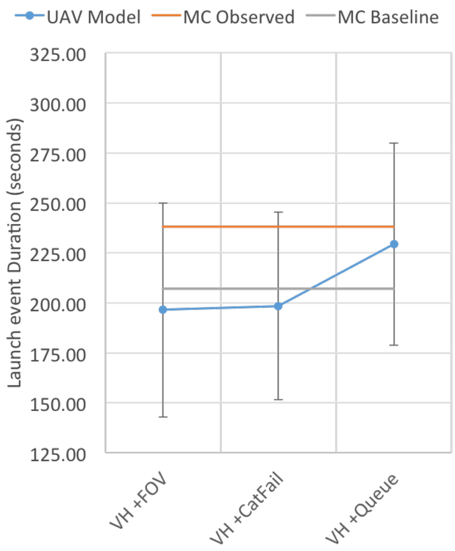


Figure J-4: Mean launch event duration time for Vehicle-Based Human Supervisory Control (VBSC) control architecture with incremental introduction of features. Whiskers indicate ±1 standard deviation.

The System-Based Human Supervisory Control (SBSC) model was the final to be introduced, which requires only two real changes. The first model (Figure J-5) removes the Director crew from operations and eliminates the use of the zone coverage routing scheme. This, like for the VH model, seems to provide a slight benefit to the speed of operations, lowering the mean launch event duration to 193.87 seconds. The second model increases the rate of catapult alignment failures to 2.5%, again with very little differences in operations. However, because not latencies are introduced, for this model, it retains slightly better performance than the MC case in its full implementation.

Figure J-6 plots the data from the previous figures on the same axis, providing a comparison of performance across all of the models. In general, a few conclusions can be drawn across these cases. First, that the modeled changes in field of view and catapult failures appear to no real effect on operations. Field of view results in only a minor change in how Aircraft Director crew align to unmanned aircraft during operations; catapult alignment failures occur at a fairly low rate, and the time penalties associated with these failures are on the order of the standard deviation of the launch preparation time process. Additionally, these results suggest that latency in operations, be it from communications, software processing, human interaction, or in the form of repeated failures with time penalties, is quite detrimental to operations. The reason for this can be seen in Figure J-7, which plots the taxi duration component of the launch event. Within this figure it can be seen that significant differences occur between taxi times within the various models. Because the launch preparation and launch acceleration time models do not vary for the unmanned vehicle control

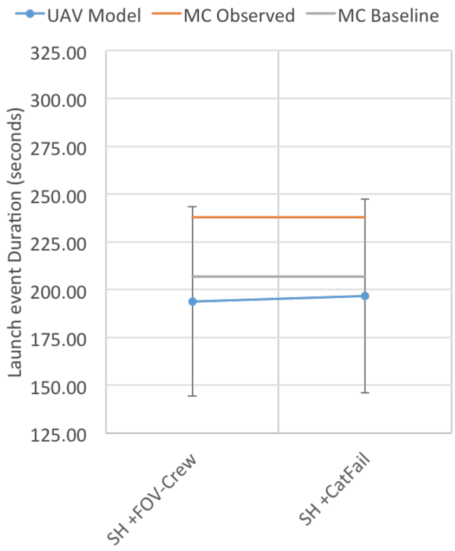


Figure J-5: Mean launch event duration time for System-Based Human Supervisory Control (SBSC) control architecture with incremental introduction of features. Whiskers indicate ±1 standard deviation.

architectures, changes in taxi time are the primary source of variation here. This has important ramifications for mission testing, where delays in taxi operations may cause significant disruptions to the flow of traffic on deck and, in the aggregate, severely hamper mission operations.

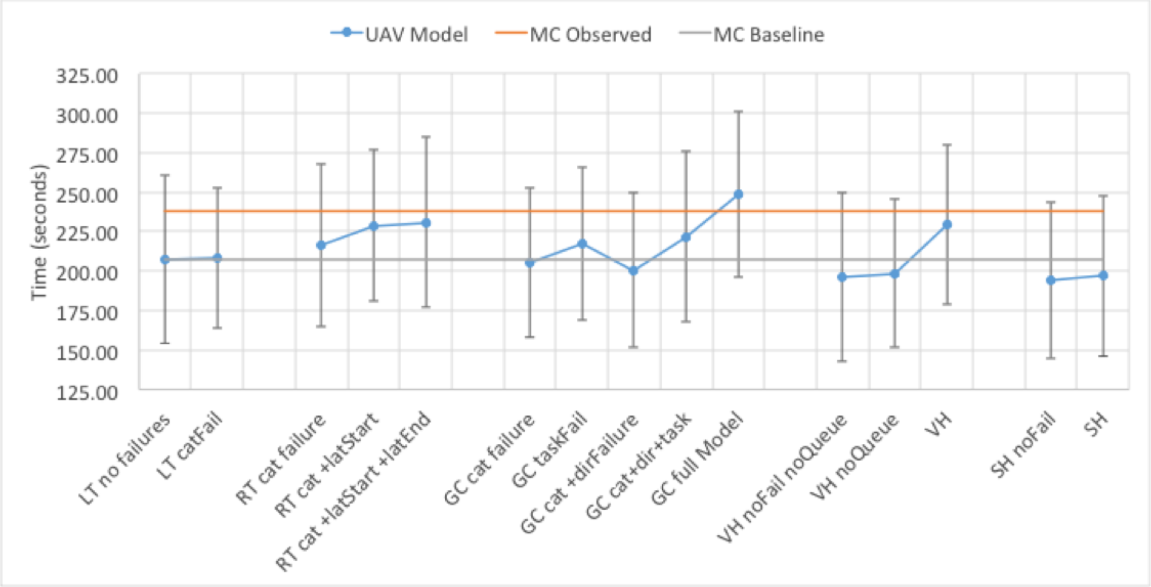


Figure J-6: Mean launch event duration time for all control architectures with incremental introduction of features. Whiskers indicate ±1 standard deviation.

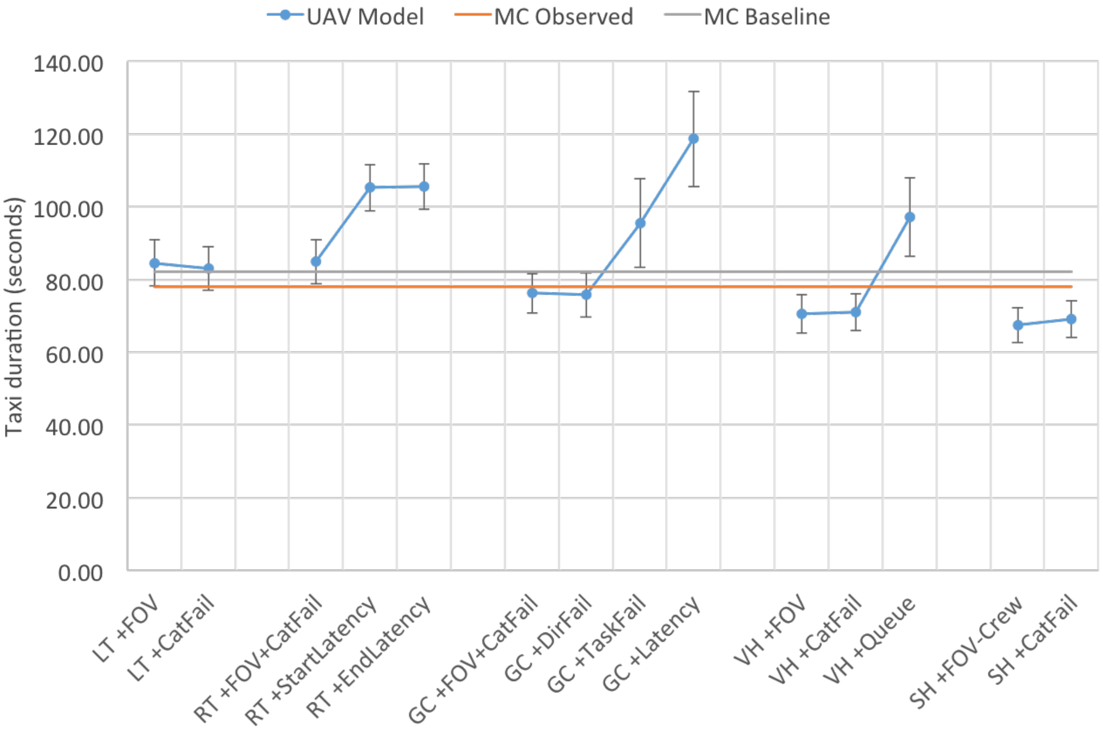


Figure J-7: Mean taxi duration time for Local Teleoperation (LT) control architecture with incremental introduction of features. Whiskers indicate ±1 standard deviation.

Appendix K

Launch Event Duration (LD) Results

The time required to complete a launch event served not only as an important validation metric for the baseline manual control Multi-Agent Safety and Control Simulation (MASCS) simulation but is also the most critical evaluation metric for Subject Matter Experts (SMEs) in carrier operations. Chapter 5 provided a summary of the results of testing, indicating the few meaningful differences in performance were found between Control Architectures, Safety Protocols, mission settings, and interactions of each. This section provides additional detail on the tests that were performed.

K.1 Effects due to Safety Protocols

Figure K-1 provides boxplots of Launch Duration values according to Safety Protocol (SP). Presented in this fashion, the expected detrimental effect of the Temporal+Area Separation (T+A) safety protocol is clearly seen. If these results are separated based on mission UAV Composition (25% or 50%), the percent increase in Launch event Duration (LD) for the T+A cases as compared to the DS only baseline averaged 38.89% for the 50% Composition level and 57.45% for the 25% level. It also appears that the Temporal safety protocol produces slightly higher LD values than the Area and DS only cases. Applying a Kruskal-Wallis nonparametric one-way analysis of variance to the Area, Temporal, and DS only safety protocols (ignoring the T+A case) reveals significant differences at both the 22 aircraft level ($\chi^2(2) = 34.2927, p < 0.0001$) and 34 aircraft levels ($\chi^2(2) = 100.6313, p < 0.0001$). Post-hoc Steel-Dwass multiple comparisons tests reveal that at the 22 aircraft level, the Temporal Separation protocol (mean rank = 629.62) is different from both Area (497.488) and Dynamic Separation (511.682) protocols, with the latter two not significantly different from one another. At the 34 aircraft level, significant differences exist for all pairwise comparisons between the three protocols. This fails to confirm the hypothesis regarding Temporal protocol performance: the Temporal protocol was expected to have equivalent performance to the DS only safety protocol, given that it preserved the same flexibility in planning operations. Area was expected to perform more poorly given the constraints it places on the assignment of vehicles. These results suggest quite the opposite, that the preservation of the flow of vehicles on the deck (which is unchanged under the Area protocol) is more beneficial than preserving flexibility in catapult assignments.

K.2 Effects due to Control Architectures

Figure K-2 contains a second set of boxplots, now comparing LD values across Control Architectures, excluding the T+A cases and their substantial effects. From this figure it can be seen that while some fluctuations in mean LD do occur across architectures, there do not seem to be any large deviations within the set. However, Kruskal-Wallis non-parametric one-way analysis of variance tests using Control Architecture (CA) as the grouping variable show that significant differences exist at the 22 aircraft level ($\chi^2(5) = 37.7272, p < 0.0001$) but not at the 34 aircraft level ($\chi^2(5) = 5.8157, p = 0.3246$). A post-hoc Steel-Dwass non-parametric simultaneous comparisons test for the 22 aircraft data reveals that significant pairwise differences exist between the System-Based Human Supervisory Control (SBSC) (mean rank = 438.6) case and each of the Remote Teleoperation (RT) (605.7), Gestural Control (GC) (582.2), and Vehicle-Based Human Supervisory Control (VBSC) (562.0) cases. The Local Teleoperation (LT) (mean rank = 515.2) and RT cases are also shown to be significantly different from one another, but no other comparisons result in significance. However, in practical terms, the differences are quite small: the three largest differences in mean LD occur between SBSC (μ = 24.27 minutes) and RT (μ = 25.50), GC (μ = 25.33 minutes), and VBSC (μ = 25.30), providing differences of 1.23, 1.06, and 1.03 minutes, respectively.

These results only partly support the hypotheses regarding CA performance: the SBSC case does trend towards better and the GC towards worse performance at the 22 aircraft level, but no differences appear at the 34 aircraft level. However, the hypotheses did not expect significantly poor performance for the RT and VBSC cases, as the delays generated in their operations were not

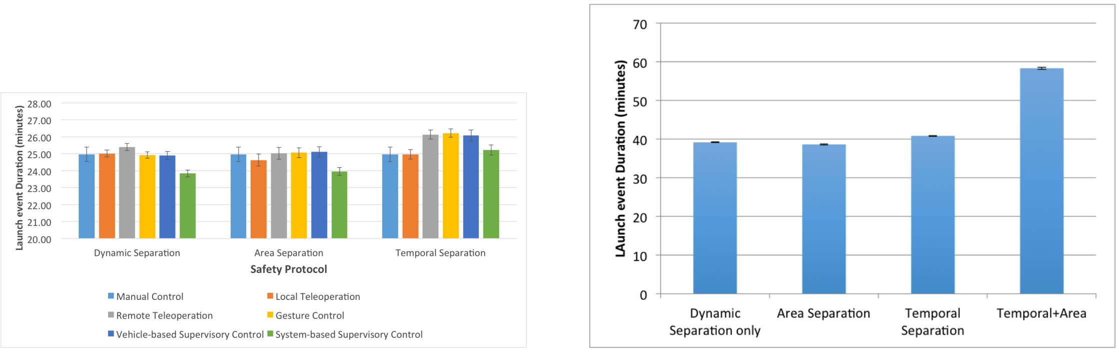


Figure K-1: Boxplots of Launch event Duration (LD) for each Safety Protocol at the 22 (left) and 34 (right) aircraft Density levels.

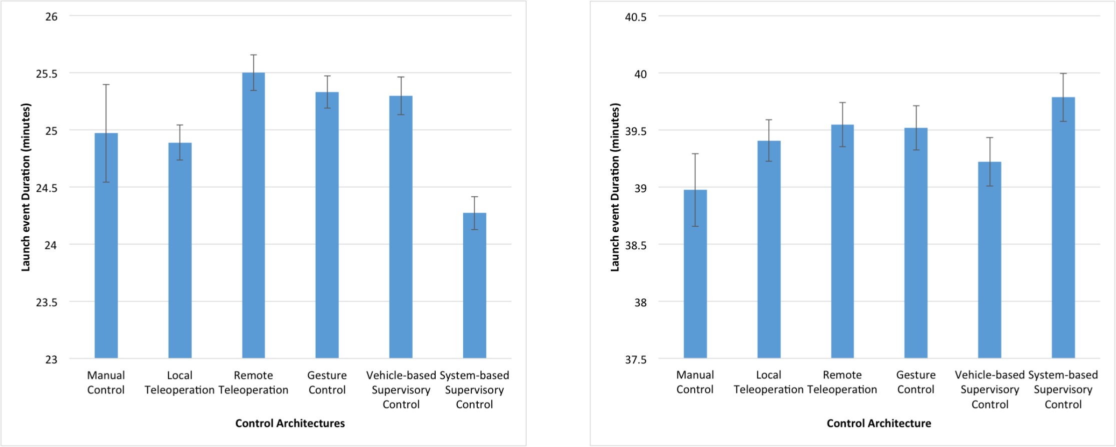


Figure K-2: Boxplots of Launch event Duration (LD) for each Control Architecture at the 22 (left) and 34 (right) aircraft Density levels, with Temporal+Area safety protocols excluded.

expected to be as severe. Additionally, the lack of variation at the 34 aircraft level suggests that the changes in deck traffic flow that occur at higher numbers of aircraft (the obstruction of taxi routes and the temporary loss of the fourth catapult) may have a substantial dampening effect on operations.

Interestingly, these conclusions are not supported when examining only the homogeneous cases (where Composition = 100% and all vehicles within the mission are identical). Boxplots of these results appear in Figure K-3, with little visible variation in results. This is supported by a statistical tests across each of these two sets of data. Statistical tests at the 22 aircraft (ANOVA, $F(5, 174) = 1.906, p = 0.096$) and 34 aircraft levels (Kruskal-Wallis, $\chi^2(5) = 4.399, p = 0.493$) reveal no significant differences amongst the data at the $\alpha = 0.05$ level. This is an interesting result given the differences in the omnibus test on CA in the preceding paragraph and implies that the significant differences in the earlier test on CA must be due to interactions of the Control Architectures with the other settings.

K.3 Effects due to Interactions

In order to better identify the root causes of these variations, another series of statistical tests were applied for each Density values (22 or 34 aircraft) using the full treatment specification (CA + Composition + SP) as the grouping variable. Figure K-4 provides a column chart of these results, with Table K.1 providing the means and standard deviations of each case. These data and the statistical tests exclude the T+A cases, however, as their effects are already shown to be significant. This results in one-way analyses of variance for 36 cases within each Density setting. Kruskal-Wallis tests showed significant differences in the data at both the 22 aircraft ($\chi^2(35) = 108.999, p < 0.0001$) and 34 aircraft levels ($\chi^2(35) = 176.264, p < 0.0001$). A post-hoc Steel-Dwass non-parametric multiple comparisons test at the 22 aircraft level reveals seventeen significant pairwise comparisons

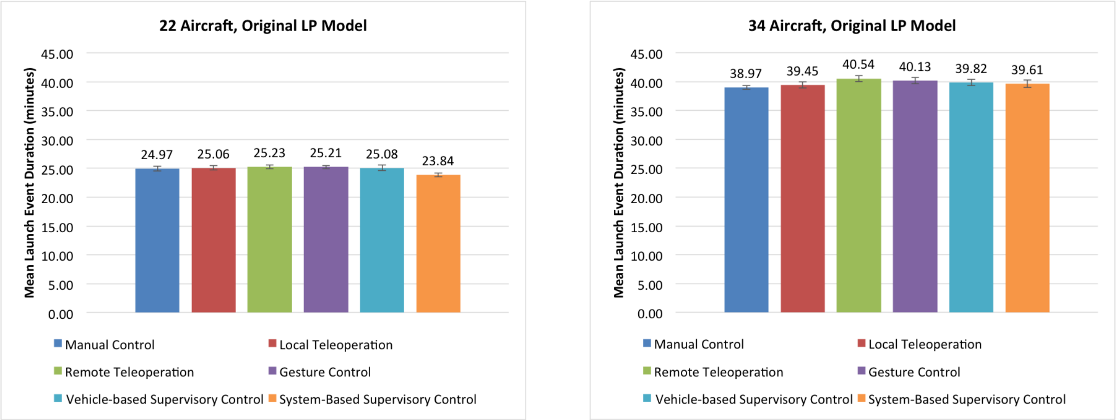


Figure K-3: Boxplots of Launch event Duration (LD) for each Control Architecture at the 100% Composition settings for 22 (left) and 34 (right) aircraft Density levels.

that all show tests using the SBSC aircraft to be significantly better than cases using the RT, GC, or VBSC aircraft. Most often, these comparisons involve the SBSC 25% DS only, 25% Area, and 100% DS only cases compared to the RT, GC, or VBSC 25% Temporal or 50% Temporal cases. A Steel-Dwass multiple comparisons test at the 34 aircraft Density levels returns fifty-six significantly different pairwise comparisons, all but ten of which involve the 50% Temporal protocol applied to a Control Architecture (a full list of the significant pairwise differences can be found in Appendix L). Thirty-five of the 46 50% Temporal protocol cases involve the use of the RT and VBSC systems, with the remainder spread across the other UAV control architectures. Of the ten that do not include the 50% Temporal protocol, seven involve the 25% Temporal protocol.

These results suggests that these the use of the 25% and 50% Temporal protocol cases are creating a significant detrimental effect in operations, with especially adverse interactions with the three "delayed" UAV cases of RT, GC, and VBSC. These performance variations may likely be the result of changes in the manner in which aircraft are routed on the flight deck. The Deck Handler heuristics, in the abstract, attempt to keep aircraft routed in an aft-to-forward direction, which in turn means limiting the assignment of forward-parked aircraft to aft catapults. The design of the Temporal Separation force the Handler to make just these assignments, as all manned aircraft are parked forward and must launch before UAVs begin operations. Under the Area Separation protocol, even though the flexibility of aircraft assignments is limited by the division of the deck, aircraft parked forward are still most often assigned to forward catapults, preserving this flow of traffic. Furthermore, in light of the significant main effects of control architectures observed earlier across all cases and the distinct lack of differences in the homogeneous cases, it is likely that Temporal Protocol performance is main driver behind the differences observed across control architectures. More specifically, that the interactions of the Temporal protocol with the "delayed" UAV types are substantial enough to create a main effect with that data.

Table K.1: Descriptive statistics for SP*Composition settings across all Control Architectures (CAs) at the 22 aircraft Density level.

Level	Mean	Std Dev	Std Err Mean	Lower 95%	Upper 95%
100% Dynamic Separation Only	24.90	2.11	0.16	24.59	25.21
25% Dynamic Separation only	24.82	2.12	0.17	24.47	25.16
50% Dynamic Separation	24.75	1.94	0.16	24.44	25.06
25% Area Separation	24.42	2.24	0.18	24.06	24.78
50% Area Separation	25.09	2.55	0.21	24.68	25.50
25% Temporal Separation	25.38	2.24	0.18	25.02	25.74
50% Temporal Separation	26.07	2.24	0.18	25.71	26.43
25% Temporal+Area	38.63	3.33	0.27	38.09	39.17
50% Temporal+Area	33.70	3.85	0.31	33.08	34.32

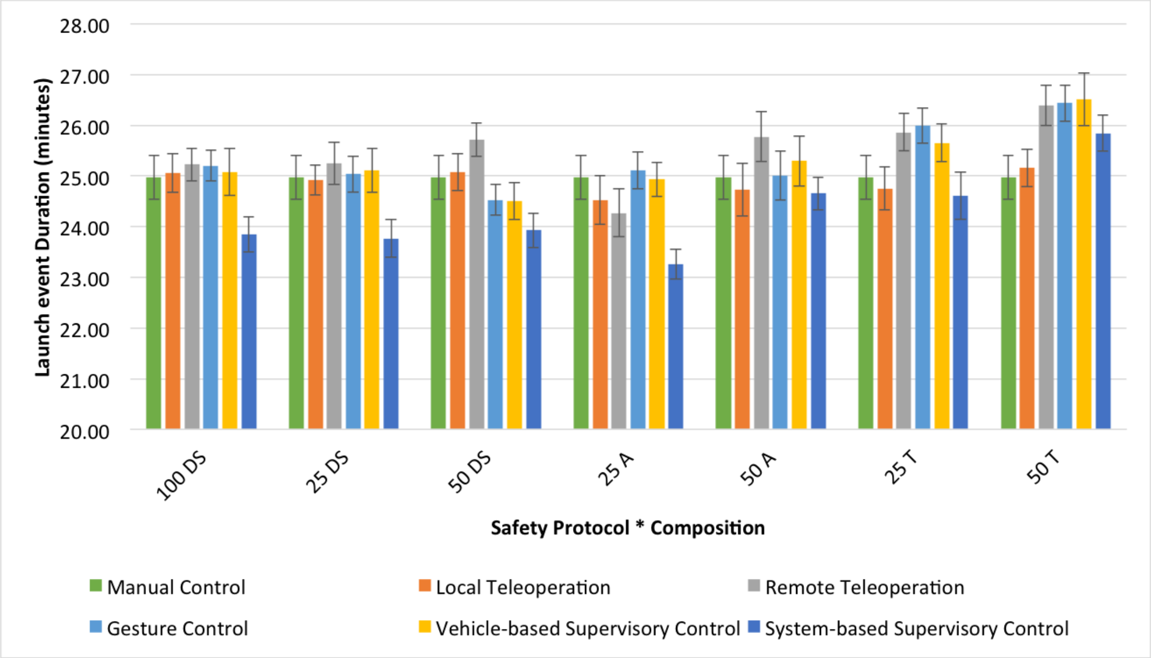


Figure K-4: Column chart comparing Launch event Duration (LD) values across all Control Architectures (CAs) sorted by Safety Protocol (SP)-Composition pairs. Whiskers indicate ±1 standard error.

Table K.2: Results of ANOVAs for LD results across control architectures for each SP*Composition setting.

SP*COMP	DF1	DF2	F	p value	Tukey results
100% DS	5	174	1.906	0.0956	n/a
25% DS	4	145	2.4906	0.0458*	RT v. SH, p=0.0503
50% DS	4	145	3.9162	0.0048*	RT v. SH, p=0.0027
25% Area	4	145	3.3537	0.0117*	GC v. SH, p=0.0106;VBSC v SH, p=0.0285
50% Area	4	145	0.9684	0.4268	n/a
25% Temporal	4	145	2.6199	0.0374*	None
50% Temporal	4	145	2.0286	0.0934	n/a

K.4 LD Effects Summary

This section has reviewed the effects of Control Architectures (CAs), Safety Protocols (SPs), mission Density, and mission Composition on Launch event Duration, the primary productivity metric in which Subject Matter Experts are concerned. This data has demonstrated that significant effects do exist between Control Architectures (CAs) and between Safety Protocols (SPs), but that these differences are largely due to interactions of CA with the other independent variables. It was earlier hypothesized that the choice of CA would provide a main effect in results, with the GC case being the worst and SBSC being the best. While data did demonstrate a significant difference between these two cases, further tests at finer resolutions indicated that these effects are more likely due to adverse interactions of the GC setting with the Temporal Separation protocol using 50% Unmanned Aerial Vehicle (UAV) Composition. Adverse interactions were also observed between the 50% Temporal Separation setting for the RT and VBSC control architectures, which was not expected. In fact, at the 34 aircraft Density setting, significant effects were observed for the 50% Temporal case across all control architectures and was the primary driver of variations at that level. No real main effects were observed for control architectures at the 34 aircraft level.

These results also fail to confirm the earlier hypothesis regarding the performance of the Safety Protocols in relation to the Launch event Duration measure, which presumed that the Temporal protocol, preserving flexibility in assignments, would outperform the T+A and Area protocol cases. In fact, the reverse was proven to be true in the case of the Area protocol: the preserving the flow of traffic on deck is of greater importance than flexibility in assignments. Additionally, results also suggest that the structure of operations forms a significant constraint on the performance of the UAV control architectures, as the effects at the 22 aircraft level often differed from the effects at the 34 aircraft level. Together with the effects of the Safety Protocol, these results suggest that preserving a general aft-to-forward flow of traffic on the flight deck is the most important feature in operations. Disruptions to this process, either due to delays on the part of UAV Control Architectures or to Safety Protocols, drives performance on these measures. Perhaps not unsurprisingly, the three CAs that continued to show poor performance — RT, GC, and VBSC — are the three cases that involve some sort of latency in operations. Even though the latencies were thought to have only a minimal effect on operations, they appear to be substantial enough to fully disrupt the flow of traffic on the flight deck. These disruptions also appear to be exacerbated by interactions with the Temporal Separation safety protocol. This may be due to the fact that the Temporal protocol forces all UAVs to be active at the same time; under the Area protocol, deck activity remains roughly split between manned and unmanned.

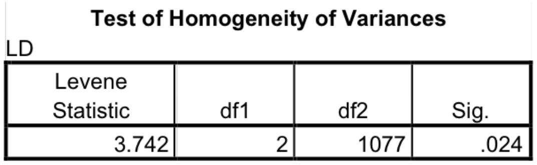
Even so, when missions used only one type of aircraft (the 100% Composition setting), no significant differences in performance were observed. Even though the use of more of the "delayed"

UAVs cases might be thought to further compound the delays on the flight deck, allowing the planning system to retain its original flexibility and flow seems to mitigate the detrimental effects of the "delayed" UAV systems. However, observations of simulation execution do note significant disruptions in taxi operations for these control architectures; in many cases, vehicles are arriving at catapults just as the previous aircraft is launching. Although there is an effect, it is not felt because the vehicle arrives in time (just barely) to prepare for its own launch. Reducing the launch preparation time mean may break this ability to queue at catapults and lead to significant effects at the 100% Composition levels (the effects of which will be explored in Chapter 6). In general, however, these results suggest that the different control architectures and safety protocols do have significant interactions with the structure of flight deck operations, at least in terms of the time to complete launch events. The next section examines the performance of the MASCS simulation in terms of the Safety metrics defined previously in Section 5.3, building Pareto frontiers of safety metrics vs. LD and examining how the effects of CA and SP discussed in this section affect the level of risk in flight deck operations.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix L

Statistical Tests on Launch Event Duration (LD)

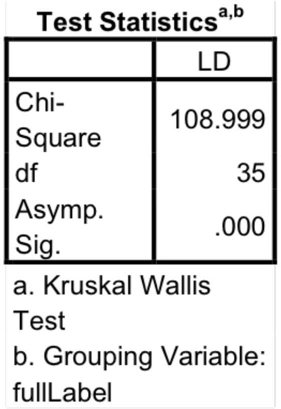


Test of Homogeneity of Variances

LD

Levene Statistic	df1	df2	Sig.
3.742	2	1077	.024

Figure L-1: SPSS output of Levene Test of Equal Variances for the 22 aircraft cases versus SP, with T+A safety protocols excluded.



Test Statistics[a,b]

	LD
Chi-Square	108.999
df	35
Asymp. Sig.	.000

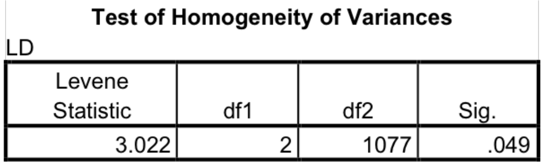
a. Kruskal Wallis Test

b. Grouping Variable: fullLabel

Figure L-2: SPSS output of non-parametric Kruskal-Wallis one-way test of variances for the 22 aircraft cases versus SP, with T+A safety protocols excluded.

Table L.1: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for 22 aircraft cases versus SP, with T+A safety protocols excluded.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
T	DS	87.37083	16.58243	5.268879	<0.0001
T	A	70.7	14.15391	4.995085	<0.0001
DS	A	12.44479	16.58243	0.75048	0.7333

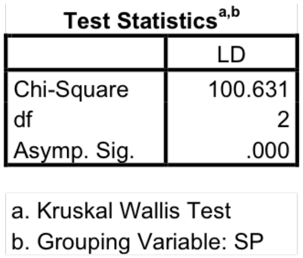


Test of Homogeneity of Variances

LD

Levene Statistic	df1	df2	Sig.
3.022	2	1077	.049

Figure L-3: SPSS output of Levene Test of Equal Variances for the 34 aircraft cases versus SP, with T+A safety protocols excluded.



Test Statistics[a,b]

	LD
Chi-Square	100.631
df	2
Asymp. Sig.	.000

a. Kruskal Wallis Test

b. Grouping Variable: SP

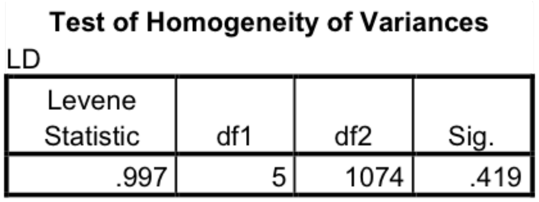
Figure L-4: SPSS output of non-parametric Kruskal-Wallis one-way test of variances for the 34 aircraft cases versus SP, with T+A safety protocols excluded.

Table L.2: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for 34 aircraft cases versus SP, with T+A safety protocols excluded.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
T	A	135.5767	14.15392	9.578739	<0.0001
T	DS	131.5627	16.58243	7.933861	<0.0001
DS	A	39.7692	16.58243	2.398271	0.0435

Table L.3: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for 22 aircraft cases versus CA, with T+A safety protocols excluded.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
SBSC	GC	-57.1429	11.84624	-4.82371	<0.0001
SBSC	RT	-64.6952	11.84624	-5.46125	<0.0001
VBSC	SBSC	46.7238	11.84624	3.94419	0.0011
RT	LT	35.5286	11.84624	2.99914	0.0324

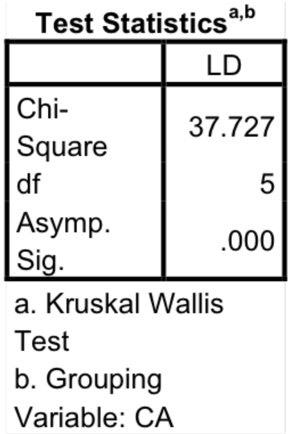


Test of Homogeneity of Variances

LD

Levene Statistic	df1	df2	Sig.
.997	5	1074	.419

Figure L-5: SPSS output of Levene Test of Equal Variances for the 22 aircraft cases versus CA, with T+A safety protocols excluded.



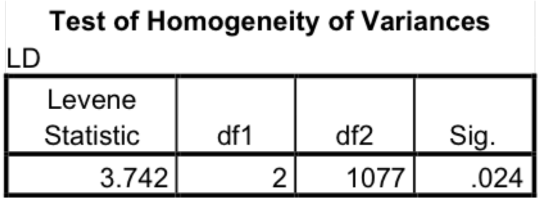
Test Statistics[a,b]

	LD
Chi-Square	37.727
df	5
Asymp. Sig.	.000

a. Kruskal Wallis Test

b. Grouping Variable: CA

Figure L-6: SPSS output of non-parametric Kruskal-Wallis one-way test of variances for the 22 aircraft cases versus CA, with T+A safety protocols excluded.

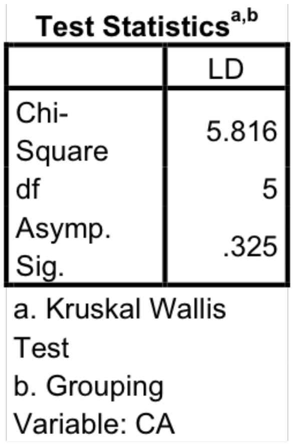


Test of Homogeneity of Variances

LD

Levene Statistic	df1	df2	Sig.
3.742	2	1077	.024

Figure L-7: SPSS output of Levene Test of Equal Variances for the 34 aircraft cases versus CA, with T+A safety protocols excluded.



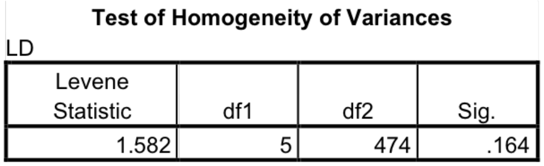
Test Statistics[a,b]

	LD
Chi-Square	5.816
df	5
Asymp. Sig.	.325

a. Kruskal Wallis Test

b. Grouping Variable: CA

Figure L-8: SPSS output of non-parametric Kruskal-Wallis one-way test of variances for the 34 aircraft cases versus CA, with T+A safety protocols excluded.

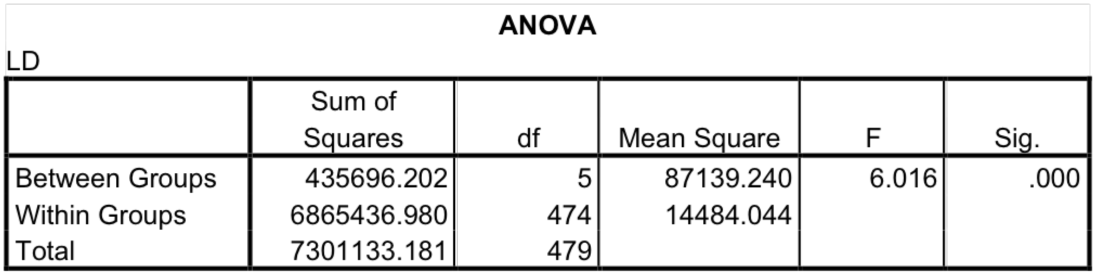


Test of Homogeneity of Variances

LD

Levene Statistic	df1	df2	Sig.
1.582	5	474	.164

Figure L-9: SPSS output of Levene Test of Equal Variances for the 22 aircraft cases versus CA for Dynamic Separation.

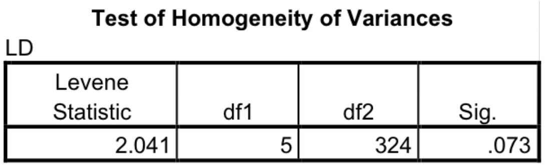


ANOVA

LD

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	435696.202	5	87139.240	6.016	.000
Within Groups	6865436.980	474	14484.044		
Total	7301133.181	479			

Figure L-10: SPSS output of an ANOVA test for the 22 aircraft cases versus CA for Dynamic Separation.



Test of Homogeneity of Variances

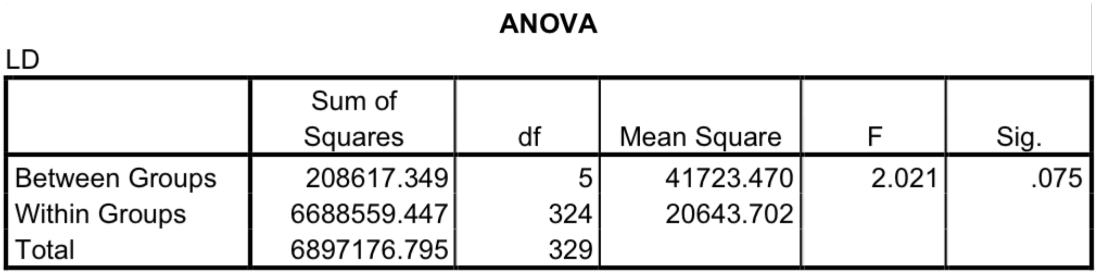
LD

Levene Statistic	df1	df2	Sig.
2.041	5	324	.073

Figure L-11: SPSS output of Levene Test of Equal Variances for the 22 aircraft cases versus CA for Area Separation.

Table L.4: JMP PRO 10 output for Tukey HSD Multiple Comparisons tests for the 22 aircraft cases versus CA for Dynamic Separation.

Level	- Level	Difference	Std Err Dif	Lower CL	Upper CL	p-Value
RT	SBSC	93.04478	17.94067	41.7089	144.3807	¡.0001
LT	SBSC	70.27722	17.94067	18.9413	121.6131	0.0014
MC	SBSC	67.49522	25.37194	-5.1047	140.0951	0.0854
GC	SBSC	64.62133	17.94067	13.2855	115.9572	0.0047
VBSC	SBSC	63.28444	17.94067	11.9486	114.6203	0.0061
RT	VBSC	29.76033	17.94067	-21.5755	81.0962	0.5598
RT	GC	28.42344	17.94067	-22.9124	79.7593	0.6094
RT	MC	25.54956	25.37194	-47.0503	98.1495	0.9155
RT	LT	22.76756	17.94067	-28.5683	74.1034	0.8018
LT	VBSC	6.99278	17.94067	-44.3431	58.3287	0.9988
LT	GC	5.65589	17.94067	-45.68	56.9918	0.9996
MC	VBSC	4.21078	25.37194	-68.3891	76.8107	1
MC	GC	2.87389	25.37194	-69.726	75.4738	1
LT	MC	2.782	25.37194	-69.8179	75.3819	1
GC	VBSC	1.33689	17.94067	-49.999	52.6728	1

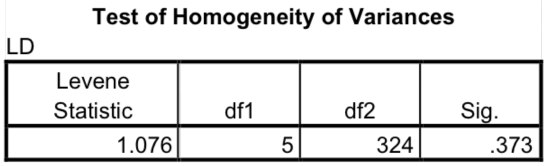


ANOVA

LD

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	208617.349	5	41723.470	2.021	.075
Within Groups	6688559.447	324	20643.702		
Total	6897176.795	329			

Figure L-12: SPSS output of an ANOVA test for the 22 aircraft cases versus CA for Area Separation.

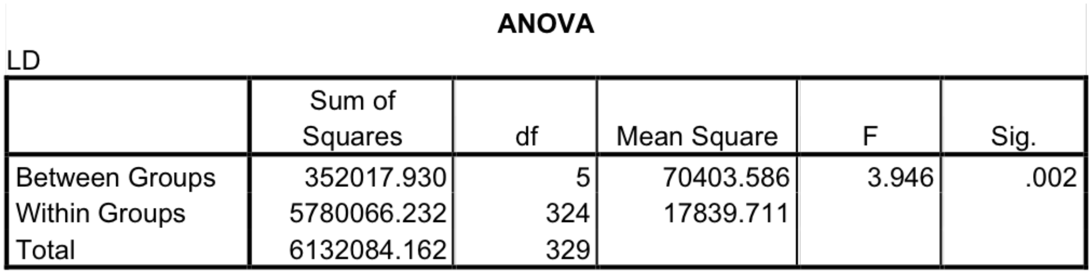


Test of Homogeneity of Variances

LD

Levene Statistic	df1	df2	Sig.
1.076	5	324	.373

Figure L-13: SPSS output of Levene Test of Equal Variances for the 22 aircraft cases versus CA for Temporal Separation.



ANOVA

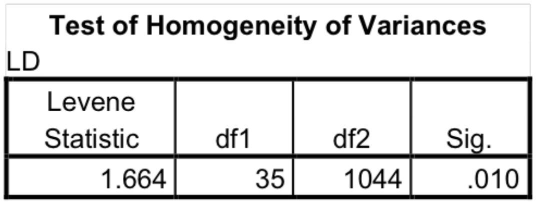
LD

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	352017.930	5	70403.586	3.946	.002
Within Groups	5780066.232	324	17839.711		
Total	6132084.162	329			

Figure L-14: SPSS output of an ANOVA test for the 22 aircraft cases versus CA for Temporal Separation.

Table L.5: JMP PRO 10 output for Tukey HSD Multiple Comparisons tests for the 22 aircraft cases versus CA for Temporal Separation, with T+A safety protocols excluded.

Level	- Level	Difference	Std Err Dif	Lower CL	Upper CL	p-Value
GC T	LT T	75.73367	24.38559	5.8235	145.6438	0.025
GC T	MC DS	74.867	29.86613	-10.7551	160.4891	0.1251
RT T	LT T	70.32617	24.38559	0.416	140.2363	0.0477
RT T	MC DS	69.4595	29.86613	-16.1626	155.0816	0.1868
VH T	LT T	67.619	24.38559	-2.2911	137.5291	0.0645
VH T	MC DS	66.75233	29.86613	-18.8697	152.3744	0.2246
GC T	SH T	59.43167	24.38559	-10.4785	129.3418	0.1467
RT T	SH T	54.02417	24.38559	-15.886	123.9343	0.2335
VH T	SH T	51.317	24.38559	-18.5931	121.2271	0.2876
SH T	LT T	16.302	24.38559	-53.6081	86.2121	0.9852
SH T	MC DS	15.43533	29.86613	-70.1867	101.0574	0.9955
GC T	VH T	8.11467	24.38559	-61.7955	78.0248	0.9995
GC T	RT T	5.4075	24.38559	-64.5026	75.3176	0.9999
RT T	VH T	2.70717	24.38559	-67.203	72.6173	1
MC DS	LT T	0.86667	29.86613	-84.7554	86.4887	1

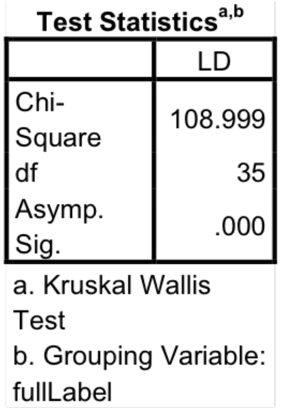


Test of Homogeneity of Variances

LD

Levene Statistic	df1	df2	Sig.
1.664	35	1044	.010

Figure L-15: SPSS output of Levene Test of Equal Variances for the 22 aircraft cases versus the full treatment definition, with T+A safety protocols excluded.



Test Statistics[a,b]

	LD
Chi-Square	108.999
df	35
Asymp. Sig.	.000

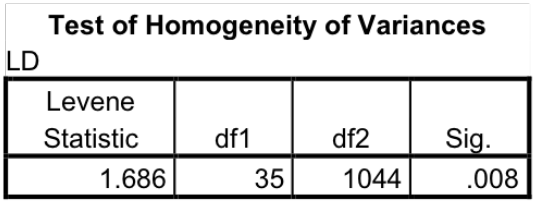
a. Kruskal Wallis Test

b. Grouping Variable: fullLabel

Figure L-16: SPSS output of non-parametric Kruskal-Wallis one-way test of variances for the 22 aircraft cases versus the full treatment definition, with T+A safety protocols excluded.

Table L.6: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests (significant results only) for 22 aircraft cases versus full treatment definitions, with T+A safety protocols excluded.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
22 SBSC 25 A	22 GC 50 T	-23.5667	4.50925	-5.22629	<0.0001
22 SBSC 25 A	22 RT 50 T	-22.0333	4.50925	-4.88625	0.0006
22 SBSC 25 A	22 GC 25 T	-21.9	4.50925	-4.85668	0.0007
22 SBSC 50 T	22 SBSC 25 A	21.0333	4.50925	4.66449	0.0017
22 SBSC 25 A	22 RT 25 T	-20.3667	4.50925	-4.51664	0.0033
22 SBSC 25 A	22 RT 50 DS	-20.2333	4.50925	-4.48707	0.0038
22 SBSC 25 DS	22 GC 50 T	-19.7	4.50925	-4.3688	0.0064
22 VBSC 50 T	22 SBSC 25 A	19.4333	4.50925	4.30966	0.0082
22 SBSC 50 DS	22 GC 50 T	-19.1	4.509187	-4.2358	0.0111
22 VBSC 25 T	22 SBSC 25 A	18.9667	4.50925	4.20617	0.0125
22 SBSC 100 DS	22 GC 50 T	-18.9	4.50925	-4.19138	0.0133
22 SBSC 25 DS	22 RT 50 T	-18.8333	4.50925	-4.1766	0.0141
22 SBSC 100 DS	22 RT 50 T	-18.3	4.50925	-4.05832	0.0223
22 SBSC 100 DS	22 GC 25 T	-17.7667	4.50925	-3.94005	0.0347
22 SBSC 25 DS	22 GC 25 T	-17.7667	4.50925	-3.94005	0.0347
22 SBSC 50 DS	22 RT 50 T	-17.7	4.509187	-3.92532	0.0366
22 SBSC 25 A	22 RT 100 DS	-17.5667	4.50925	-3.8957	0.0407

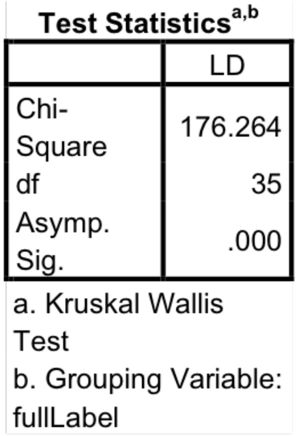


Test of Homogeneity of Variances

LD

Levene Statistic	df1	df2	Sig.
1.686	35	1044	.008

Figure L-17: SPSS output of Levene Test of Equal Variances for the 34 aircraft cases versus the full treatment definition, with T+A safety protocols excluded.



Test Statistics[a,b]

	LD
Chi-Square	176.264
df	35
Asymp. Sig.	.000

a. Kruskal Wallis Test

b. Grouping Variable: fullLabel

Figure L-18: SPSS output of non-parametric Kruskal-Wallis one-way test of variances for the 34 aircraft cases versus the full treatment definition, with T+A safety protocols excluded.

Table L.7: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests (significant results only) for 34 aircraft cases versus full treatment definitions, with T+A safety protocols excluded (part 1).

Level	-Level	Score Mean Difference	Std Err Dif	Z	p-Value
34 VBSC 50 T	34 VBSC 50 DS	27.6333	4.50925	6.12814	<0.0001
34 RT 50 T	34 RT 50 A	25.2333	4.50925	5.59591	<0.0001
34 VBSC 50 T	34 RT 50 A	25.2333	4.50925	5.59591	<0.0001
34 RT 50 T	34 LT 25 A	24.1	4.50925	5.34457	<0.0001
34 VBSC 50 T	34 VBSC 25 A	24.0333	4.50925	5.32979	<0.0001
34 VBSC 50 T	34 LT 25 A	23.6333	4.50925	5.24108	<0.0001
34 VBSC 50 T	34 RT 25 DS	23.1667	4.50925	5.13759	0.0002
34 RT 50 T	34 RT 25 DS	22.7667	4.50925	5.04888	0.0003
34 RT 50 T	34 MC 100 DS	22.3667	4.50925	4.96017	0.0004
34 VBSC 50 T	34 MC 100 DS	22.1	4.50925	4.90104	0.0005
34 VBSC 50 T	34 SBSC 25 A	21.8333	4.50925	4.8419	0.0007
34 RT 50 T	34 GC 50 DS	21.1667	4.50925	4.69406	0.0015
34 VBSC 50 T	34 GC 50 DS	21.1667	4.50925	4.69406	0.0015
34 RT 50 T	34 GC 25 DS	20.9	4.50925	4.63492	0.002
34 VBSC 50 T	34 GC 25 DS	20.9	4.50925	4.63492	0.002
34 VBSC 50 T	34 VBSC 50 A	20.9	4.50925	4.63492	0.002
34 VBSC 50 T	34 GC 50 A	20.7	4.50925	4.59056	0.0024
34 SBSC 25 T	34 RT 50 A	20.5	4.50925	4.54621	0.0029
34 RT 50 T	34 GC 50 A	20.3667	4.50925	4.51664	0.0033
34 RT 50 T	34 RT 50 DS	19.8333	4.50925	4.39837	0.0056
34 VBSC 50 T	34 RT 50 DS	19.6333	4.50925	4.35401	0.0068
34 VBSC 50 T	34 VBSC 25 DS	19.5	4.50925	4.32444	0.0077
34 VBSC 50 T	34 SBSC 50 A	19.1667	4.50925	4.25052	0.0104
34 SBSC 25 T	34 LT 25 A	18.8333	4.50925	4.1766	0.0141
34 VBSC 50 T	34 LT 50 A	18.7667	4.50925	4.16182	0.0149
34 LT 50 T	34 LT 25 A	18.4333	4.50925	4.08789	0.0199
34 RT 50 T	34 LT 50 A	18.4333	4.50925	4.08789	0.0199
34 VBSC 50 T	34 RT 25 A	18.3667	4.50925	4.07311	0.0211
34 RT 50 T	34 RT 25 A	18.3	4.50925	4.05832	0.0223

Table L.8: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests (significant results only) for 34 aircraft cases versus full treatment definitions, with T+A safety protocols excluded (part 2).

Level	-Level	Score Mean Difference	Std Err Dif	Z	p-Value
34 RT 50 T	34 LT 25 T	18.2333	4.50925	4.04354	0.0236
34 SBSC 25 T	34 SBSC 25 A	18.2333	4.50925	4.04354	0.0236
34 SBSC 50 T	34 RT 50 A	18.0333	4.50925	3.99919	0.0279
34 VBSC 50 T	34 LT 25 T	17.9	4.50925	3.96962	0.0312
34 RT 50 T	34 LT 50 DS	17.7	4.50925	3.92526	0.0366
34 RT 25 DS	34 GC 50 T	-17.5	4.50925	-3.88091	0.0429
34 SBSC 25 A	34 LT 50 T	-17.5667	4.50925	-3.8957	0.0407
34 VBSC 50 DS	34 GC 100 DS	-17.6333	4.50925	-3.91048	0.0386
34 VBSC 50 DS	34 SBSC 50 DS	-17.7667	4.50925	-3.94005	0.0347
34 VBSC 25 A	34 LT 50 T	-17.9	4.50925	-3.96962	0.0312
34 VBSC 50 DS	34 RT 25 T	-18.2333	4.50925	-4.04354	0.0236
34 VBSC 25 A	34 GC 50 T	-18.3667	4.50925	-4.07311	0.0211
34 RT 50 A	34 LT 50 T	-18.6333	4.50925	-4.13225	0.0168
34 VBSC 25 DS	34 RT 50 T	-18.9	4.50925	-4.19138	0.0133
34 SBSC 50 A	34 RT 50 T	-19.3667	4.50925	-4.29488	0.0087
34 VBSC 25 A	34 SBSC 25 T	-19.3667	4.50925	-4.29488	0.0087
34 VBSC 50 DS	34 RT 100 DS	-19.5	4.50925	-4.32444	0.0077
34 VBSC 50 DS	34 GC 25 T	-19.9	4.50925	-4.41315	0.0053
34 RT 50 A	34 GC 50 T	-19.9667	4.50925	-4.42794	0.0049
34 VBSC 50 A	34 RT 50 T	-20.5667	4.50925	-4.561	0.0027
34 VBSC 50 DS	34 SBSC 50 T	-20.8333	4.50925	-4.62013	0.0021
34 SBSC 25 A	34 RT 50 T	-21.5	4.50925	-4.76798	0.001
34 VBSC 50 DS	34 LT 50 T	-22.9	4.50925	-5.07845	0.0002
34 VBSC 50 DS	34 GC 50 T	-23.1	4.50925	-5.1228	0.0002
34 VBSC 50 DS	34 SBSC 25 T	-23.7667	4.50925	-5.27065	<0.0001
34 VBSC 25 A	34 RT 50 T	-24.5	4.50925	-5.43328	<0.0001
34 VBSC 50 DS	34 RT 50 T	-28.2333	4.50925	-6.2612	<0.0001

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix M

Analysis of Additional Safety Metrics

This appendix presents additional results and analyses for safety metrics not covered in Chapter 5.

M.1 Results for Duration of Tertiary Halo Incursions due to Aircraft (DTHIA)

Another view of safety looks at the total time during which halo incursions occur. It was previously hypothesized that while System-Based Human Supervisory Control (SBSC) would have some of the highest counts of Tertiary Halo Incursions due to Aircraft (THIA), it was expected to have some of the lowest values for DTHIA because of the smaller expected Launch event Duration (LD). Figure M-1 provides a Pareto plot of DTHIA versus LD for the 22 aircraft Density setting. The trend in the data is similar to the THIA data earlier, with Dynamic Separation providing some of the largest values (worst performance), while Temporal+Area provides the best overall. Kruskal-Wallis tests confirm this trend ($\chi^2(3) = 108.3789, p < 0.0001$) and CA ($\chi^2(5) = 736.3356, p < 0.0001$). For Safety Protocol (SP) settings, Steel-Dwass tests reveal similar performance to the THIA metric, with Temporal (mean rank = 596.74) and Temporal+Area (mean rank = 574.38) providing the best performance, followed by Area (mean rank = 665.52) and Dynamic Separation (mean rank = 837.29) measures, in line with prior expectations.

The trends in Control Architecture (CA) performance are also similar to that of the THIA measures, which can be better observed in Figure M-2. This figure shows only results for all control architectures for the Dynamic Separation protocol only. Manual Control (MC), Local Teleoperation (LT), and Remote Teleoperation (RT) provide the best performance in the DTHIA measures, followed by Gestural Control (GC), Vehicle-Based Human Supervisory Control (VBSC), and SBSC.

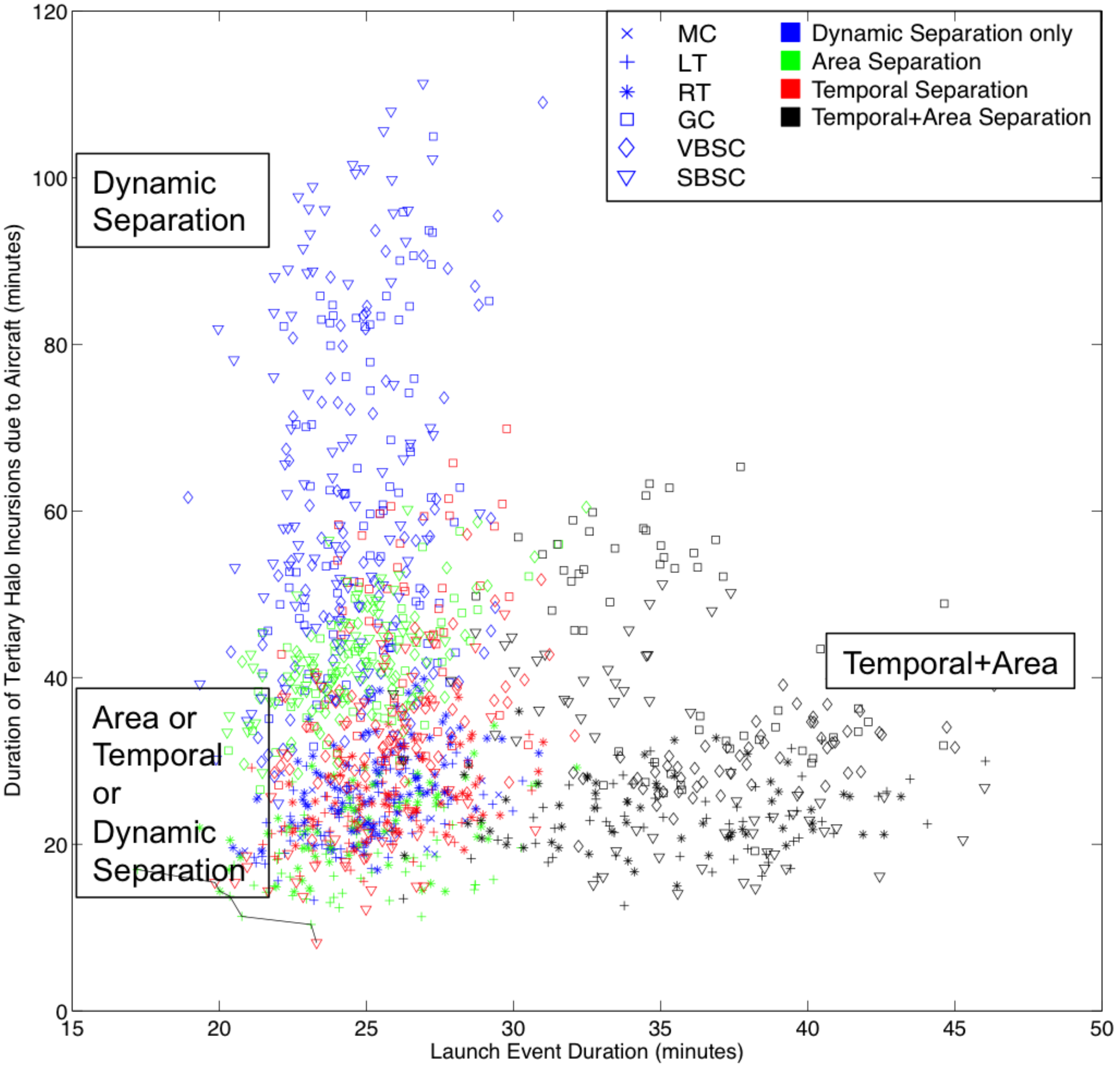


Figure M-1: Pareto frontier plot of Duration of Tertiary Halo Incursions due to Aircraft (DTHIA) versus LD for 22 aircraft missions.

A Steel-Dwass test shows that only three pairwise comparisons are *not* significant: MC (mean rank =311.7) vs. RT (424.8, $p = 0.0631$), MC vs. LT (287.9, $p = 0.9571$), and VBSC (893.6) vs. SBSC (859.1, $p = 0.8122$). All other comparisons are shown to be significant at $p < 0.0015$, with the "unmanned" CA performing worse than their manned counterparts and with the GC architecture (mean rank = 1029.3) again showing the worst performance overall.

This is in line with the earlier THIA results but against the earlier hypotheses on performance: it was expected that the SBSC cases would complete missions more quickly than other cases, thus reducing the total possible time that halo incursions could exist. However, as shown in Section 5.5.1, the benefits of the SBSC architecture on Launch event Duration (LD) were slight, so it is not surprising that no effects were seen. It is true, however, that the SBSC case shows both the best performance on this metric as well as some of the worst (although the GC architecture performs worst overall). These trends are largely the same at the 34 aircraft Density setting, whose Pareto chart may be found in Appendix N.

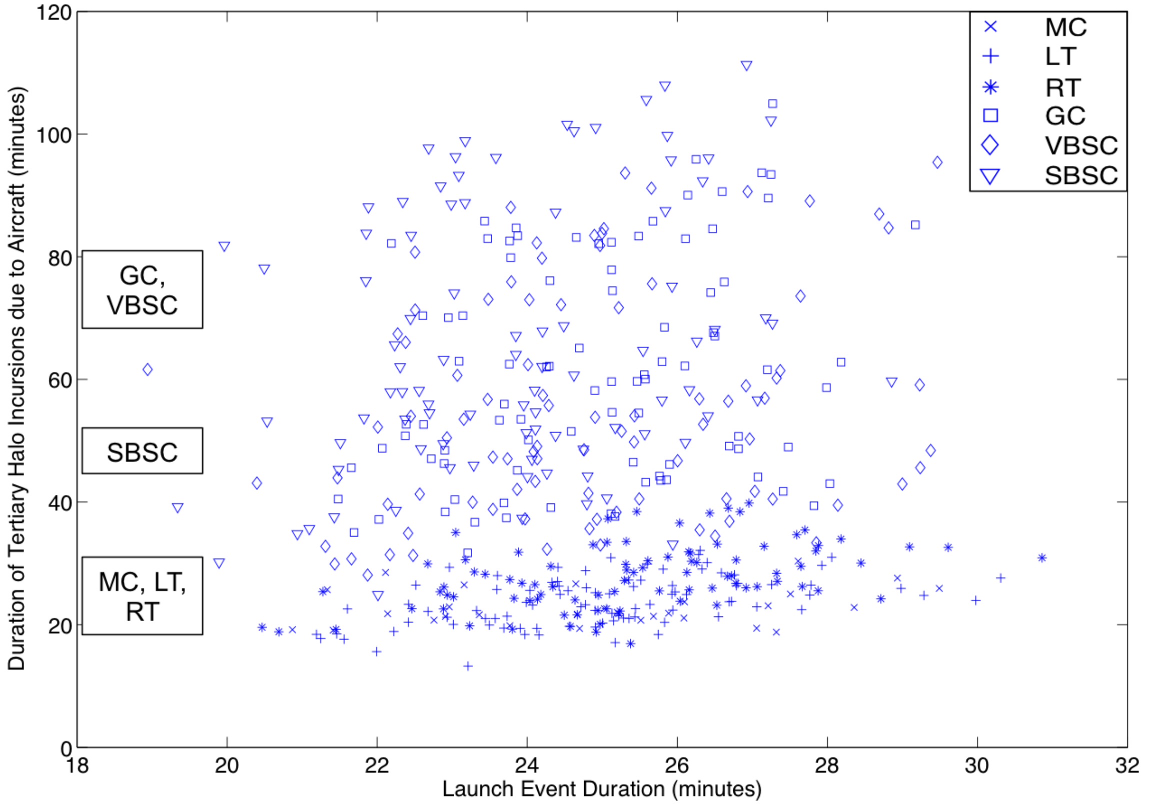


Figure M-2: Pareto frontier plot of Duration of Tertiary Halo Incursions due to Aircraft (DTHIA) versus LD for 22 aircraft Dynamic Separation missions.

M.2 Results for Primary Halo Incursion (PHI)

This section reviews results for the Primary Halo Incursion (PHI) metric of crew incursions near aircraft during operations. Because the Tertiary Halo Incursions (THI) metric reviewed in Section 5.5.2 considered the largest halos around vehicles, crew incursions at that level would likely involve no real danger to crew during operations. Reducing the size of the incursion radius to the Primary halo level should provide a better understanding of how often crew are placed in potentially dangerous situations. Whether or not these situations are truly dangerous and may lead to accidents is contingent on the exactly how crew are moving in and near to other aircraft on the flight deck at a level of complexity not currently contained in Multi-Agent Safety and Control Simulation (MASCS). As such, MASCS is not able to offer a truly detailed understanding of crew safety at this time. Future work could involve more complex models of crew motion in order to address these limitations.

Even so, this section reviews PHI results in the context of when crew are likely to work very near to aircraft on the flight deck. In Chapter 5, the performance of the THIA measures were shown to be in partial agreement with hypotheses regarding safety measures on the flight deck, but given that the safety protocols are not directly aimed at separating *crew* from unmanned vehicles and that the crew are integral parts of a functioning flight deck, those results may not be consistent here. This is apparent from the first set of data shown in this section; Figure M-3 shows the Pareto plot of PHI versus LD at the 22 aircraft case. For this data, the results for cases using the Temporal+Area Separation (T+A) safety protocols are excluded for the time being.

The first major trend in Figure M-3 is that there is a clear separation between the VBSC and SBSC cases and the other control architectures. Neither the VBSC nor the SBSC architecture are routed via the crew zone coverage topology. Instead, they receive waypoint commands from the Deck Handler via the centrally-networked system, providing these two architectures greater freedom in taxiing on the flight deck. This routed also included the use of crew “Escorts,” which appears to have generated significantly higher PHI values. However, improving the performance of these two architectures along the PHI metric would simply require adjusting the encoded Escort heuristics to be more aware of the aircraft “halos” when moving in and near aircraft. Alternatively, the Escorts are also optional and are not required for use on the flight deck; they could be entirely removed from the simulation model.

Within the remaining control architectures (MC, LT, RT, and GC) in the bottom of Fig. M-3, the choice of control architecture appears to have no effect on results. The stronger trends appears to be in terms of SPs: the Area Separation cases seem to provide the smallest PHI values, followed by Dynamic Separation, then with Temporal Separation providing the worst. A Kruskal-Wallis nonparametric one-way analysis of variance test applied to just this group of control architectures (MC, LT, RT, and GC) verifies that these differences along SPs settings are significant ($\chi^2(2) =$

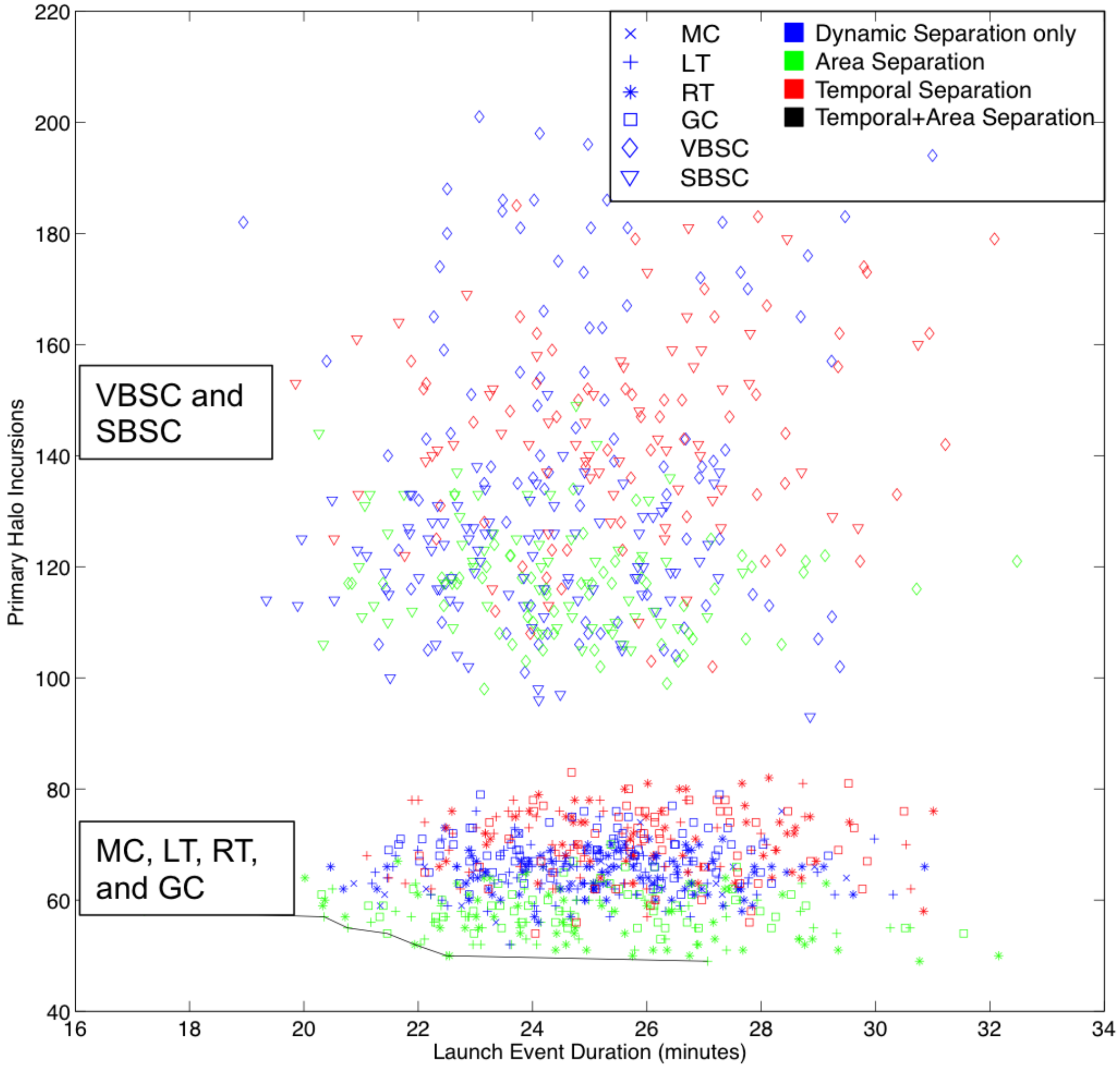


Figure M-3: Pareto frontier plot of Primary Halo Incursion (PHI) versus LD for 22 aircraft missions, excluding Temporal+Area cases.

$121.737, p < 0.0001$). Area Separation provides the best performance (mean rank = 125.51), then Dynamic Separation only (362.37) and Temporal (482.39). A Steel-Dwass multiple comparisons test shows that significant differences exists between each pairwise comparison at the $p < 0.0001$ level (full results of these tests are included in Appendix N).

The differences between the Area and Temporal cases, as noted previously, are in terms of the routing of traffic on the flight deck. Aircraft operating under the Area protocols typically move in an aft-to-forward direction. Additionally, the layout of the areas limits how many aircraft parked aft in the Fantail are taxied to the forward catapults. The Temporal cases result in a different pattern, taxiing Manned aircraft from the forward area of deck to both the forward and aft catapults in order to clear space in the Street area of the deck. Dynamic Separation results in patterns somewhere in between the two: some forward to aft transitions do occur, but the general flow of traffic is aft to forward.

Given these trends, this change in aircraft routing that results from the changes in Deck Handler heuristics appear to be the primary influence in this lower cluster of results. If the source of these variations are the transitions between areas, then the performance of the T+A cases should show performance roughly average of the Temporal and Area cases; under this protocol, no manned aircraft transition from forward to aft, but the remaining Unmanned Aerial Vehicles (UAVs) must do so in order to launch. Figure M-4 shows the same data as Figure M-3 but now includes the Temporal+Area Separation protocols (black markers on the right side of the figure). Figure M-5 shows this data again broken down for only the Remote Teleoperation control architecture.

Running statistical tests for this data for the non-supervisory control cases, a Steel-Dwass multiple comparisons test (Appendix N provides the full list of significant comparisons) demonstrates that the T+A cases are significantly different from the all other cases and containing the largest mean rank values (mean rank = 649.03; Temporal mean rank = 547.47). This data suggests that the key issue with PHI values is not transitions between the forward and aft areas of deck, this data, but simply the emphasis on using the forward area of the flight deck in general — the T+A case includes no transitions between the forward and aft areas of the flight deck whatsoever.

The same general results appear at the 34 aircraft case, as shown in Figure M-6. A clear separation once again exists between the VBSC and SBSC cases and the other and non-Human Supervisory Control (HSC) architectures. Within the non-HSC architectures (MC, LT, RT, GC), the trends are generally the same, although the differences between the Area and Dynamic Separation results are not as significant as in the 22 aircraft results. This is likely explained by the way in which 34 aircraft mission evolve: because of the increased number of aircraft on the deck and the blocking of one of the forward catapults, the forward area of the flight deck is cleared more slowly than the aft under the both the Dynamic and Area Separation cases. This means that even the Area Separation case results in aircraft parked forward being assigned to the aft catapults. However, as would be expected, the Temporal and Temporal+Area cases both still provide the worst performance for PHI values at the 34 aircraft Density level. A Kruskal-Wallis one-way comparison test for the lower cluster of data (MC, LT, RT, GC) reveals significant differences across SP settings ($\chi^2(3) = 611.815, p < 0.0001$), with a Steel-Dwass multiple comparisons test demonstrating that significant differences exist in all possible pairwise comparisons. Area (mean rank = 191.17) provides the lowest average values, followed by Dynamic Separation (mean rank = 282.855), Temporal+Area (577.175), and Temporal Separation (722.564).

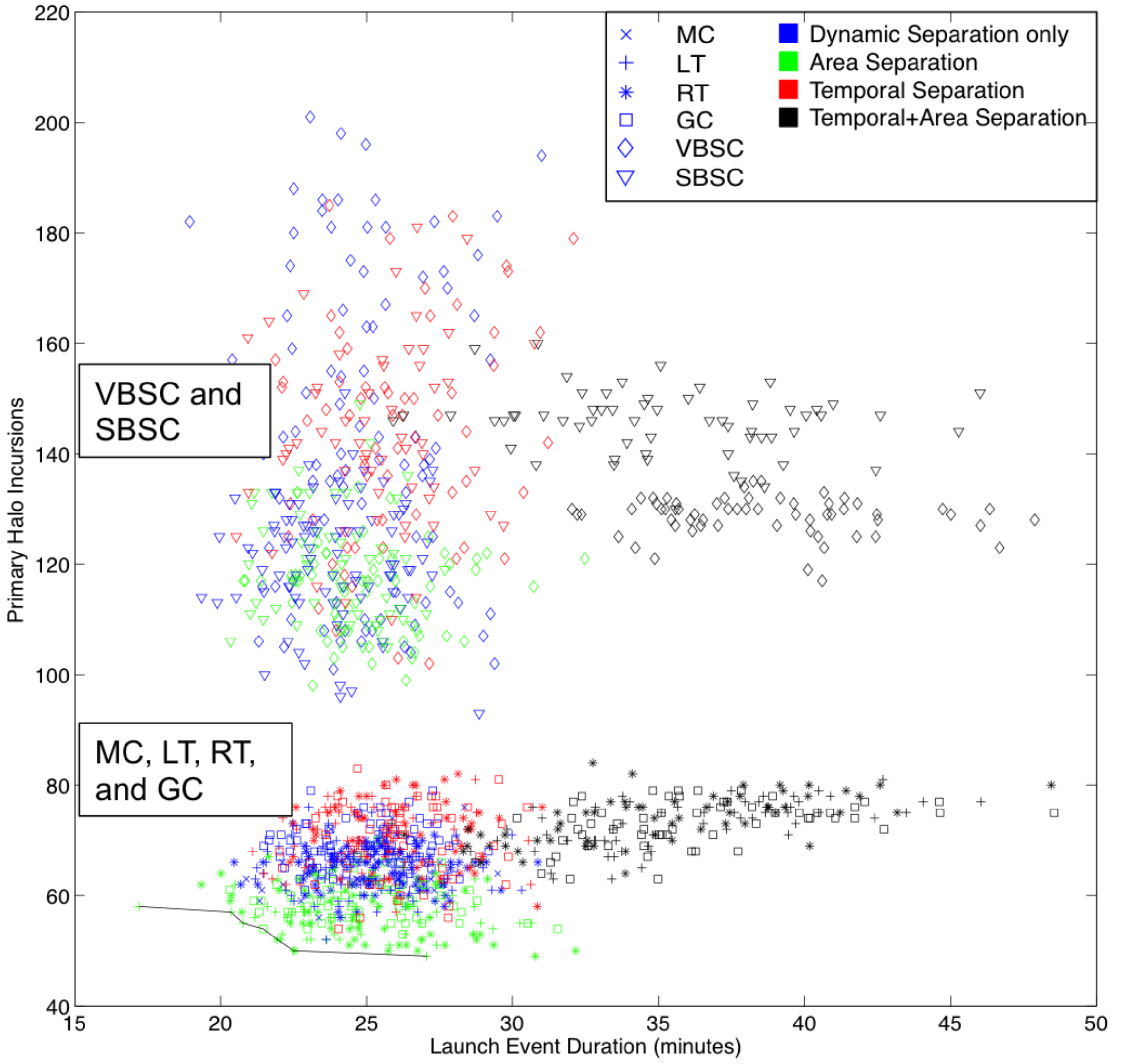


Figure M-4: Pareto frontier plot of Primary Halo Incursion (PHI) versus LD for 22 aircraft missions.

M.3 Results for Duration of Primary Halo Incursions (DPHI)

Viewing these measures in terms of the total Duration of Primary Halo Incursions (DPHI) demonstrates different trends in performance. Despite the variations seen in the count values, there is very little variation in terms of durations, as seen for the 34 aircraft data in Figure M-7. For the data in the figure, it appears that the choice of Safety Protocol has no difference on performance, while the choice of Control Architecture only matters if it is the VBSC or GC cases. Similar performance is observed at the 22 aircraft level (the Pareto chart can be found in Appendix N), although there only the VBSC case exhibits poorer performance. These effects show that the trends hypothesized

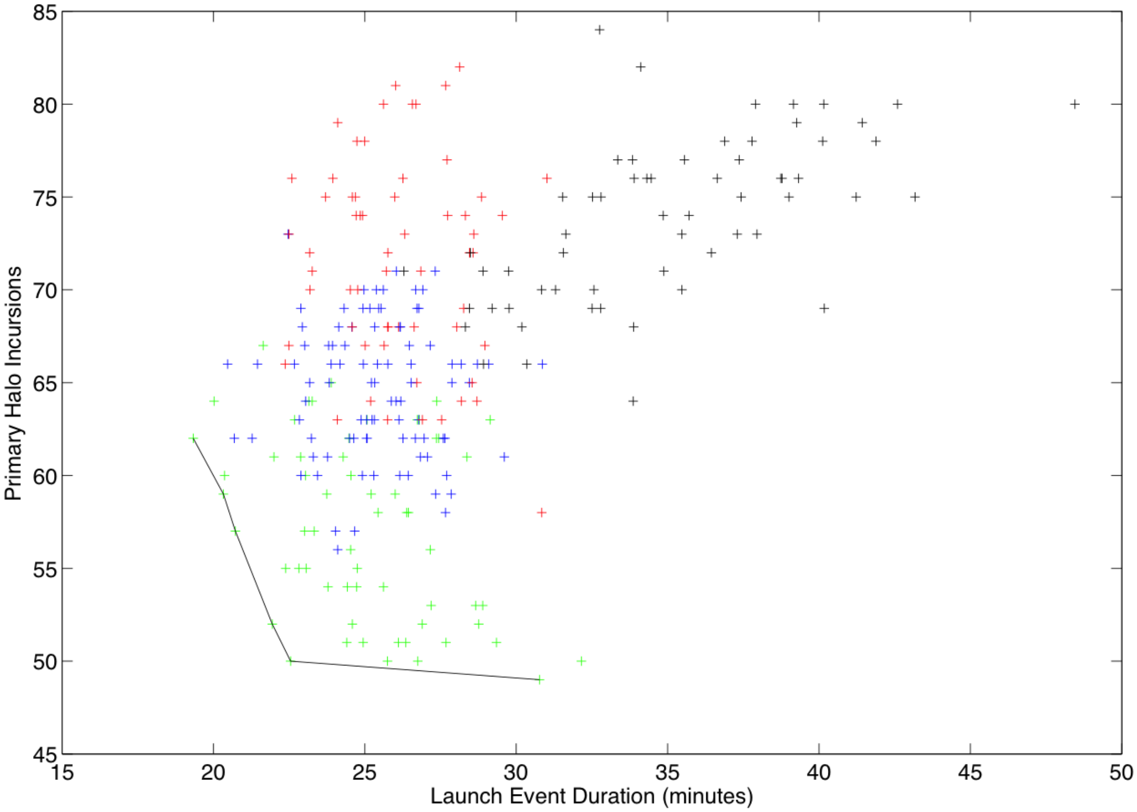


Figure M-5: Pareto frontier plot of Primary Halo Incursion (PHI) versus LD for 22 aircraft Remote Teleoperation missions, including Temporal+Area cases.

to appear in the Duration measures do not appear, although it is notable that the VBSC and GC cases (which suffer the greatest delays in operations) provide the worst performance. In general, however, the differences in results for DPHI and PHI are significant for future research: while it is not known currently which is the larger driver of risk in operations, the two do not appear to be entirely correlated with one another in the MASCS model and do not produce the same patterns of performance. This, in turn, suggests that reducing the number of halo incursions that occur during a given operation may not necessarily translate into a reduction of the duration of those incursions. Understanding which is the more important feature in terms of actual accidents on the flight deck is an important aspect of future research.

A key factor in the results for PHI and THIA has been differences in how the crew are utilized. In both cases, differences in the routing topology (either using or not using the zone coverage Aircraft Director crew) had significant effects on results, with some additional effects based on the number of high-level constraints applied to operations. In other cases, the choice of Safety Protocol and it interactions with the zone coverage routing topology produced the most significant effects. What results is a series of tradeoffs between safety metrics. Utilizing the combined Temporal+Area protocol reduces the number of THIA occurrences while increasing the number of PHI occurrences,

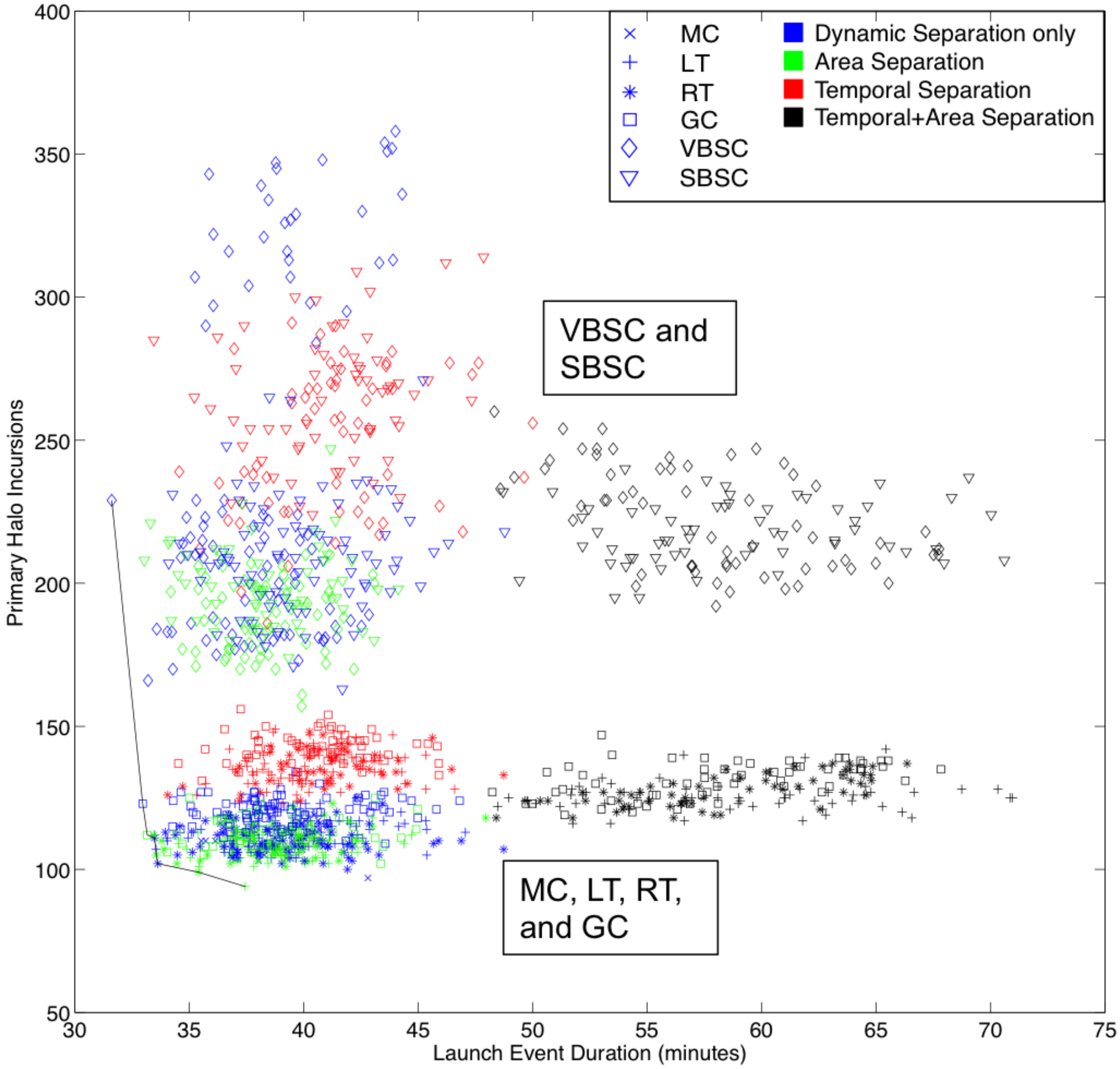


Figure M-6: Pareto frontier plot of Primary Halo Incursion (PHI) versus LD for 34 aircraft missions.

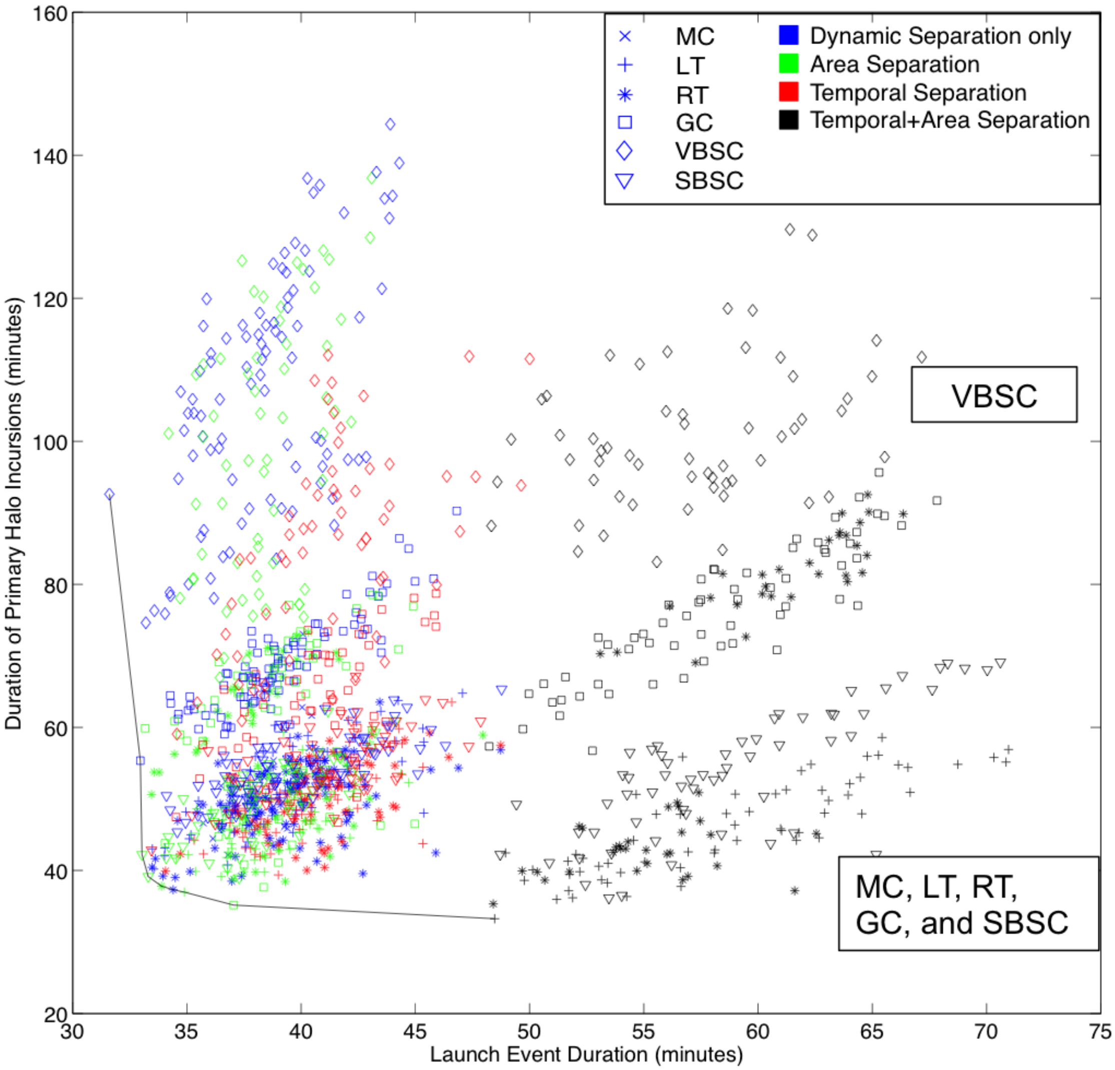


Figure M-7: Pareto frontier plot of Duration of Primary Halo Incursions (DPHI) versus LD for 34 aircraft missions.

due to the manner in which missions are planned and executed on the flight deck. Using Area Separation provides better results for PHI but slightly worse for THIA; using Temporal or Dynamic Separation provides worse performance in PHI values with performance on THIA ranging from moderately good to very poor. Just as with the earlier results on mission productivity, however, the key features driving behavior concern the role of crew on the flight deck, where they are located, and what tasks they perform. Differences in how the unmanned vehicle control architectures interact with the crew have helped identify the crew's role in influencing these safety measures, but further improvements may require rethinking the role of the crew in flight deck operations.

As example of this involves the crew "Escorts" defined for use with the VBSC and SBSC models in this work. Within the PHI measures, the VBSC and SBSC architectures both exhibited much higher values for crew incursions than the other cases. The model that described Escort behavior resulted in the crew continually moving and readjusting to aircraft, running in and out of the different halos over the course of operations. Altering Escort behavior so that they were more aware of halo distances and less active in moving in and around aircraft should reduce the number of these incursions. However, these crew were also optional: removing them entirely from operations should drop these incursion counts to zero for the SBSC architecture and significantly reduce them for VBSC (which still requires some Aircraft Director crew in operations). Similarly, when using the zone coverage Aircraft Director routing, reducing the number of crew or adjusting their spacing, could also be effective in promoting safe operations. These are questions that remain for future work, however.

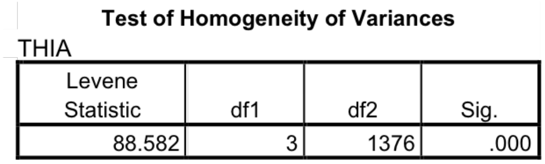
THIS PAGE INTENTIONALLY LEFT BLANK

Appendix N

Statistical Tests for Pareto Frontier Results

N.1 Supplemental statistical tests for Tertiary Halo Incursions (THI)

This section presents more detailed statistics on Tertiary Halo Incursions (THI) and Tertiary Halo Incursions due to Aircraft (THIA) measures as previously discussed in Chapter 5.

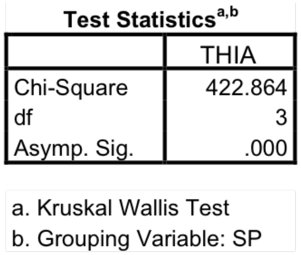


Test of Homogeneity of Variances

THIA

Levene Statistic	df1	df2	Sig.
88.582	3	1376	.000

Figure N-1: SPSS output for Levene test of equal variances for the 22 aircraft THI values against SPs.



Test Statistics[a,b]

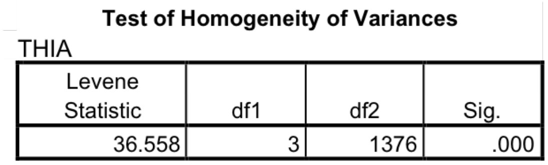
	THIA
Chi-Square	422.864
df	3
Asymp. Sig.	.000

a. Kruskal Wallis Test
b. Grouping Variable: SP

Figure N-2: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 22 aircraft THIA values against SPs.

Table N.1: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Tertiary Halo Incursions due to Aircraft (THIA) 22 aircraft cases versus Safety Protocols.

Level	-Level	Score Mean Difference	Std Err Dif	Z	p-Value
T+A	DS	-304.509	16.57664	-18.3698	<0.0001
T+A	A	-198.283	14.14334	-14.0196	<0.0001
T+A	T	-164.507	14.1445	-11.6304	<0.0001
T	DS	-157.95	16.57347	-9.5303	<0.0001
T	A	-30.347	14.13862	-2.1464	0.1386
DS	A	140.043	16.57096	8.4511	<0.0001

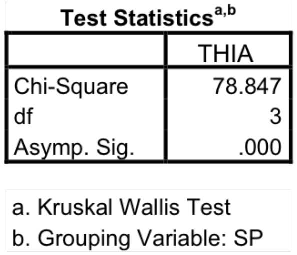


Test of Homogeneity of Variances

THIA

Levene Statistic	df1	df2	Sig.
36.558	3	1376	.000

Figure N-3: SPSS output for Levene test of equal variances for the 34 aircraft THIA values against SPs.



Test Statistics[a,b]

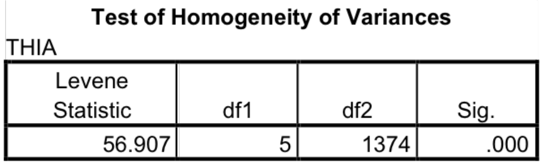
	THIA
Chi-Square	78.847
df	3
Asymp. Sig.	.000

a. Kruskal Wallis Test
b. Grouping Variable: SP

Figure N-4: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 34 aircraft THIA values against SPs.

Table N.2: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Tertiary Halo Incursions due to Aircraft (THIA) 34 aircraft cases versus Safety Protocols.

Level	-Level	Score Mean Difference	Std Err Dif	Z	p-Value
T+A	A	-85.68	14.1494	-6.05538	<0.0001
T	DS	-90.231	16.57937	-5.44235	<0.0001
T+A	DS	-131.297	16.57916	-7.91942	<0.0001
T	A	-49.207	14.14992	-3.47752	0.0028
T+A	T	-39.567	14.15114	-2.79601	0.0266
DS	A	32.256	16.57825	1.9457	0.209

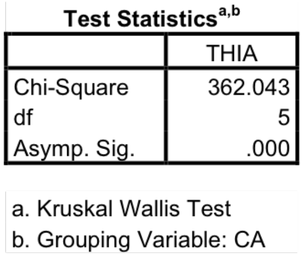


Test of Homogeneity of Variances

THIA

Levene Statistic	df1	df2	Sig.
56.907	5	1374	.000

Figure N-5: SPSS output for Levene test of equal variances for the 22 aircraft THIA values against CAs.



Test Statistics[a,b]

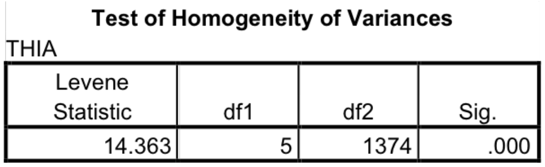
	THIA
Chi-Square	362.043
df	5
Asymp. Sig.	.000

a. Kruskal Wallis Test
b. Grouping Variable: CA

Figure N-6: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 22 aircraft THIA values against CAs.

Table N.3: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Tertiary Halo Incursions due to Aircraft (THIA) 22 aircraft cases versus Control Architectures.

Level	-Level	Score Mean Difference	Std Err Dif	Z	p-Value
SBSC	LT	157.2	13.42179	11.7123	<0.0001
VBSC	LT	150.2	13.41918	11.1929	<0.0001
SBSC	RT	139.004	13.42073	10.3574	<0.0001
VBSC	RT	124.993	13.41869	9.3148	<0.0001
SBSC	MC	79.315	16.68611	4.7533	<0.0001
VBSC	MC	62.296	16.68312	3.7341	0.0026
MC	LT	42.981	16.66496	2.5792	0.1024
RT	LT	42.767	13.40995	3.1892	0.0179
SBSC	GC	-4.5	13.42322	-0.3352	0.9994
RT	MC	-13.278	16.66759	-0.7966	0.9682
VBSC	SBSC	-35.463	13.42316	-2.6419	0.0874
VBSC	GC	-52.385	13.42134	-3.9031	0.0013
MC	GC	-97.444	16.68439	-5.8405	<0.0001
RT	GC	-179.056	13.41981	-13.3426	<0.0001
LT	GC	-199.641	13.42073	-14.8755	<0.0001

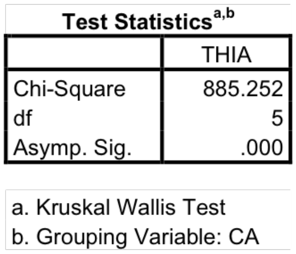


Test of Homogeneity of Variances

THIA

Levene Statistic	df1	df2	Sig.
14.363	5	1374	.000

Figure N-7: SPSS output for Levene test of equal variances for the 34 aircraft THIA values against CAs.



Test Statistics[a,b]

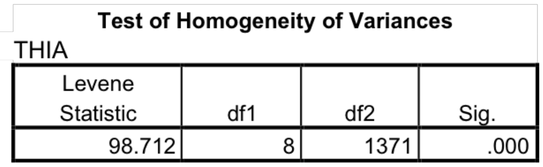
	THIA
Chi-Square	885.252
df	5
Asymp. Sig.	.000

a. Kruskal Wallis Test
b. Grouping Variable: CA

Figure N-8: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 34 aircraft THIA values against CAs.

Table N.4: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Tertiary Halo Incursions due to Aircraft (THIA) 34 aircraft cases versus Control Architectures.

Level	-Level	Score Mean Difference	Std Err Dif	Z	p-Value
VBSC	LT	263.081	13.42601	19.5949	<0.0001
SBSC	LT	240.696	13.42488	17.9291	<0.0001
VBSC	RT	240.415	13.42439	17.9088	<0.0001
SBSC	RT	172.874	13.42254	12.8794	<0.0001
VBSC	MC	147.852	16.6856	8.861	<0.0001
SBSC	MC	137.611	16.68821	8.246	<0.0001
RT	LT	136.948	13.42157	10.2036	<0.0001
VBSC	SBSC	80.67	13.42345	6.0097	<0.0001
RT	MC	70.556	16.68154	4.2296	0.0003
MC	LT	26.889	16.67997	1.612	0.5905
VBSC	GC	-49.107	13.42016	-3.6592	0.0034
SBSC	GC	-117.944	13.42366	-8.7863	<0.0001
MC	GC	-148.852	16.68544	-8.9211	<0.0001
RT	GC	-256.748	13.42533	-19.1242	<0.0001
LT	GC	-266.919	13.42617	-19.8805	<0.0001

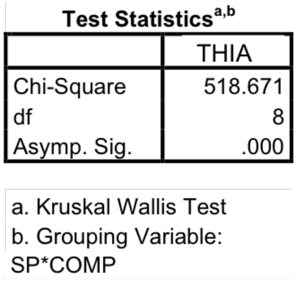


Test of Homogeneity of Variances

THIA

Levene Statistic	df1	df2	Sig.
98.712	8	1371	.000

Figure N-9: SPSS output for Levene test of equal variances for the 22 aircraft THIA values against SP-Composition pairs.

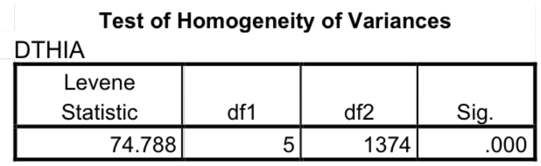


Test Statistics[a,b]

	THIA
Chi-Square	518.671
df	8
Asymp. Sig.	.000

a. Kruskal Wallis Test
b. Grouping Variable: SP*COMP

Figure N-10: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 22 aircraft THIA values against SP-Composition pairs.



Test of Homogeneity of Variances

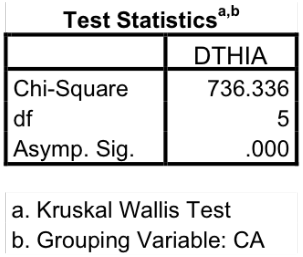
DTHIA

Levene Statistic	df1	df2	Sig.
74.788	5	1374	.000

Figure N-11: SPSS output for Levene test of equal variances for the 22 aircraft DTHIA values against CAs.

Table N.5: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Tertiary Halo Incursions due to Aircraft (THIA) 22 aircraft cases versus Safety Protocols-Composition pairs.

Level	-Level	Score Mean Difference	Std Err Dif	Z	p-Value
50 A	100 DS	-160.411	10.54764	-15.2082	<0.0001
50 A	25 T	-149.993	10.01665	-14.9744	<0.0001
50 A	25 T+A	-149.993	10.01665	-14.9744	<0.0001
50 A	25 DS	-149.913	10.01665	-14.9664	<0.0001
50 DS	25 T+A	-148.673	10.01665	-14.8426	<0.0001
50 A	25 A	-147.433	10.01665	-14.7188	<0.0001
50 T+A	25 T+A	-143.9	10.01665	-14.3661	<0.0001
50 DS	25 T	-143.82	10.01665	-14.3581	<0.0001
50 T	25 T+A	-97.78	10.01665	-9.7617	<0.0001
50 T	25 T	-76.54	10.01665	-7.6413	<0.0001
50 T+A	25 T	-72.993	10.01665	-7.2872	<0.0001
25 A	100 DS	-35.976	10.54764	-3.4108	0.0187
50 DS	25 DS	-27.82	10.01665	-2.7774	0.122
25 T+A	25 T	14.473	10.01665	1.4449	0.8804
50 DS	100 DS	23.729	10.54764	2.2497	0.3733
50 T+A	50 T	35.007	10.01665	3.4948	0.014
25 DS	100 DS	51.217	10.54764	4.8558	<0.0001
50 DS	25 A	53.647	10.01665	5.3557	<0.0001
25 DS	25 A	74.367	10.01665	7.4243	<0.0001
50 T	25 DS	96.193	10.01665	9.6033	<0.0001
50 T	50 DS	107.06	10.01665	10.6882	<0.0001
50 T	100 DS	124.221	10.54764	11.7771	<0.0001
50 T	25 A	125.527	10.01665	12.5318	<0.0001
50 T+A	25 DS	136.167	10.01665	13.594	<0.0001
50 T+A	50 DS	139.06	10.01665	13.8829	<0.0001
25 T	25 DS	141.66	10.01665	14.1424	<0.0001
50 T+A	25 A	146.9	10.01665	14.6656	<0.0001
50 T+A	100 DS	154.691	10.54764	14.6659	<0.0001
25 T	25 A	148.02	10.01665	14.7774	<0.0001
50 DS	50 A	149.473	10.01665	14.9225	<0.0001
25 T+A	25 DS	149.713	10.01665	14.9464	<0.0001
25 T+A	25 A	149.993	10.01665	14.9744	<0.0001
50 T	50 A	149.993	10.01665	14.9744	<0.0001
50 T+A	50 A	149.993	10.01665	14.9744	<0.0001
25 T	100 DS	159.433	10.54764	15.1155	<0.0001
25 T+A	100 DS	164.798	10.54764	15.6242	<0.0001



Test Statistics[a,b]

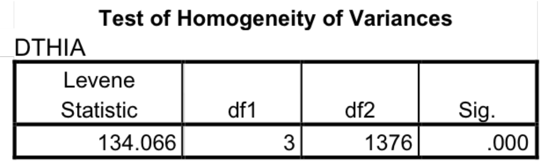
	DTHIA
Chi-Square	736.336
df	5
Asymp. Sig.	.000

a. Kruskal Wallis Test
b. Grouping Variable: CA

Figure N-12: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 22 aircraft DTHIA values against CAs.

Table N.6: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Duration of Tertiary Halo Incursions due to Aircraft (DTHIA) 22 aircraft cases versus Control Architectures.

Level	-Level	Score Mean Difference	Std Err Dif	Z	p-Value
VBSC	LT	253.485	13.42881	18.8762	<0.0001
VBSC	RT	222.481	13.42881	16.5675	<0.0001
SBSC	LT	188.511	13.42881	14.0378	<0.0001
SBSC	RT	163.915	13.42881	12.2062	<0.0001
VBSC	MC	142.093	16.69439	8.5114	<0.0001
SBSC	MC	99.407	16.6944	5.9545	<0.0001
RT	LT	95.667	13.42878	7.124	<0.0001
RT	MC	46.167	16.69434	2.7654	0.0631
MC	LT	14.259	16.69433	0.8541	0.9571
VBSC	SBSC	-16.781	13.42881	-1.2497	0.8122
SBSC	GC	-51.907	13.42881	-3.8654	0.0015
VBSC	GC	-81.43	13.42881	-6.0638	<0.0001
MC	GC	-147.389	16.6944	-8.8286	<0.0001
RT	GC	-249.985	13.42881	-18.6156	<0.0001
LT	GC	-264.737	13.42881	-19.7141	<0.0001

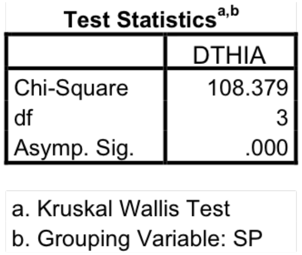


Test of Homogeneity of Variances

DTHIA

Levene Statistic	df1	df2	Sig.
134.066	3	1376	.000

Figure N-13: SPSS output for Levene test of equal variances for the 22 aircraft DTHIA values against SPs.



Test Statistics[a,b]

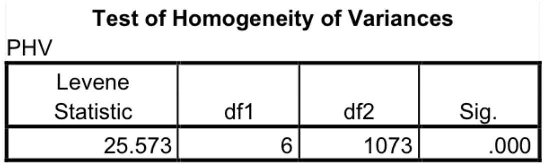
	DTHIA
Chi-Square	108.379
df	3
Asymp. Sig.	.000

a. Kruskal Wallis Test
b. Grouping Variable: SP

Figure N-14: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 22 aircraft DTHIA values against SPs.

Table N.7: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Duration of Tertiary Halo Incursions due to Aircraft (DTHIA) 22 aircraft cases versus Safety Protocols.

Level	-Level	Score Mean Difference	Std Err Dif	Z	p-Value
DS	A	105.88	16.58242	6.38505	<0.0001
T+A	T	-13.043	14.1539	-0.92154	0.7933
T	A	-36.967	14.1539	-2.61177	0.0446
T+A	A	-43.39	14.1539	-3.06559	0.0117
T	DS	-132.92	16.58242	-8.01569	<0.0001
T+A	DS	-142.84	16.58242	-8.61395	<0.0001

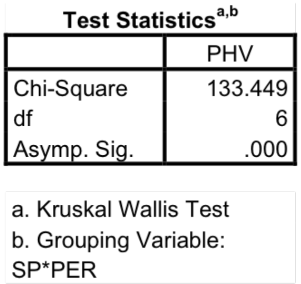


Test of Homogeneity of Variances

PHV

Levene Statistic	df1	df2	Sig.
25.573	6	1073	.000

Figure N-15: SPSS output for Levene test of equal variances for the 22 aircraft PHI values against SP-Composition pairs, excluding T+A safety protocols.



Test Statistics[a,b]

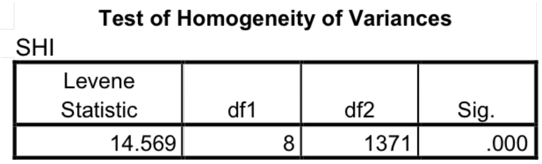
	PHV
Chi-Square	133.449
df	6
Asymp. Sig.	.000

a. Kruskal Wallis Test
b. Grouping Variable:
SP*PER

Figure N-16: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 22 aircraft PHI values against SP-Composition pairs, excluding T+A safety protocols.

Table N.8: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Primary Halo Incursion (PHI) 22 aircraft cases versus Safety Protocols-Composition pairs, excluding T+A cases.

Level	-Level	Score Mean Difference	Std Err Dif	Z	p-Value
50 A	25 T	-72.14	10.01379	-7.20407	<0.0001
50 A	100 DS	-71.115	10.54384	-6.7447	<0.0001
50 A	25 DS	-50.7267	10.01315	-5.066	<0.0001
50 A	25 A	-48.48	10.0107	-4.84282	<0.0001
50 DS	25 T	-44.8533	10.01264	-4.47967	0.0002
25 A	100 DS	-40.4617	10.5391	-3.83919	0.0024
50 T	25 T	-21.3533	10.01306	-2.13255	0.3333
25 DS	100 DS	9.1178	10.5399	0.86507	0.9776
50 DS	25 DS	13.5133	10.00922	1.35009	0.8281
50 DS	100 DS	20.0078	10.53917	1.89842	0.4813
50 T	50 DS	19.0733	10.01156	1.90513	0.4768
50 T	25 DS	28.9067	10.01181	2.88726	0.0595
50 T	100 DS	30.5861	10.54213	2.90132	0.0572
25 DS	25 A	36.98	10.01086	3.69399	0.0041
50 DS	25 A	52.34	10.01088	5.22831	<0.0001
25 T	100 DS	57.7683	10.54343	5.47908	<0.0001
25 T	25 DS	55.1667	10.01291	5.50956	<0.0001
50 T	25 A	58.1067	10.01187	5.80377	<0.0001
50 DS	50 A	65.62	10.01342	6.55321	<0.0001
25 T	25 A	68.3467	10.01373	6.8253	<0.0001
50 T	50 A	73.84	10.0138	7.37383	<0.0001

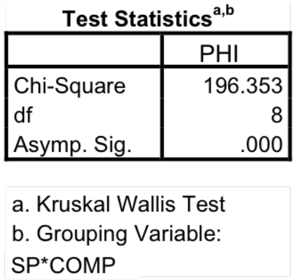


Test of Homogeneity of Variances

SHI

Levene Statistic	df1	df2	Sig.
14.569	8	1371	.000

Figure N-17: SPSS output for Levene test of equal variances for the 22 aircraft SHI values against SPs-Composition interactions.



Test Statistics^{a,b}

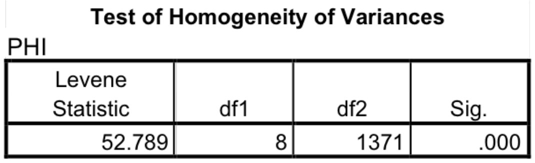
	PHI
Chi-Square	196.353
df	8
Asymp. Sig.	.000

a. Kruskal Wallis Test
b. Grouping Variable: SP*COMP

Figure N-18: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 22 aircraft PHI values against SPs-Composition pairs.

Table N.9: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Tertiary Halo Incursions due to Aircraft (THIA) 22 aircraft cases versus Safety Protocols-Composition pairs.

Level	-Level	Score Mean Difference	Std Err Dif	Z	p-Value
50 A	25 T+A	-76.3133	10.01159	-7.6225	<0.0001
50 A	25 T	-72.14	10.01379	-7.20407	<0.0001
50 A	100 DS	-71.115	10.54384	-6.7447	<0.0001
50 DS	25 T+A	-56.5667	10.01164	-5.65009	<0.0001
50 A	25 DS	-50.7267	10.01315	-5.066	<0.0001
50 A	25 A	-48.48	10.0107	-4.84282	<0.0001
50 DS	25 T	-44.8533	10.01264	-4.47967	0.0003
50 T+A	25 T+A	-44.76	10.00948	-4.47176	0.0003
25 A	100 DS	-40.4617	10.5391	-3.83919	0.0039
50 T	25 T+A	-32.8867	10.01064	-3.28517	0.0283
50 T+A	25 T	-22.6667	10.01045	-2.2643	0.3641
50 T	25 T	-21.3533	10.01306	-2.13255	0.4508
50 T+A	50 T	5.9333	10.01156	0.59265	0.9996
25 DS	100 DS	9.1178	10.5399	0.86507	0.9947
50 DS	25 DS	13.5133	10.00922	1.35009	0.916
50 DS	100 DS	20.0078	10.53917	1.89842	0.6149
50 T	50 DS	19.0733	10.01156	1.90513	0.6102
25 T+A	25 T	21.5733	10.00572	2.1561	0.4348
50 T	25 DS	28.9067	10.01181	2.88726	0.0917
50 T	100 DS	30.5861	10.54213	2.90132	0.0883
50 T+A	50 DS	34.68	10.00976	3.46462	0.0156
25 DS	25 A	36.98	10.01086	3.69399	0.0068
50 T+A	100 DS	50.7283	10.54188	4.81208	<0.0001
50 T+A	25 DS	48.3133	10.01022	4.8264	<0.0001
50 DS	25 A	52.34	10.01088	5.22831	<0.0001
25 T	100 DS	57.7683	10.54343	5.47908	<0.0001
25 T	25 DS	55.1667	10.01291	5.50956	<0.0001
50 T	25 A	58.1067	10.01187	5.80377	<0.0001
50 DS	50 A	65.62	10.01342	6.55321	<0.0001
25 T+A	100 DS	69.245	10.54296	6.56789	<0.0001
25 T	25 A	68.3467	10.01373	6.8253	<0.0001
50 T+A	25 A	69.56	10.01299	6.94697	<0.0001
25 T+A	25 A	71.76	10.01184	7.16751	<0.0001
25 T+A	25 DS	72.7267	10.01199	7.26396	<0.0001
50 T	50 A	73.84	10.0138	7.37383	<0.0001
50 T+A	50 A	76.2	10.01328	7.60989	<0.0001

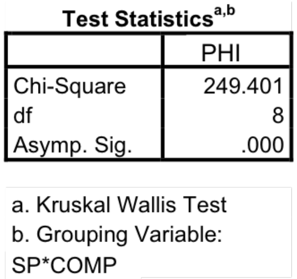


Test of Homogeneity of Variances

PHI

Levene Statistic	df1	df2	Sig.
52.789	8	1371	.000

Figure N-19: SPSS output for Levene test of equal variances for the 34 aircraft PHI values against SP-Composition pairs.



Test Statistics[a,b]

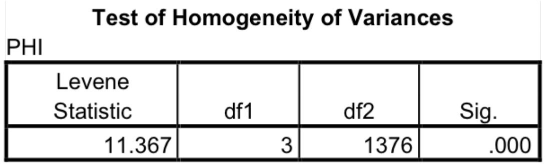
	PHI
Chi-Square	249.401
df	8
Asymp. Sig.	.000

a. Kruskal Wallis Test
b. Grouping Variable: SP*COMP

Figure N-20: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 34 aircraft PHI values against SP-Composition pairs, excluding T+A safety protocols.

Table N.10: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Primary Halo Incursion (PHI) 34 aircraft cases versus Safety Protocols-Composition pairs.

Level	-Level	Score Mean Difference	Std Err Dif	Z	p-Value
50 A	25 T	-76.8333	10.01511	-7.67174	<0.0001
50 T+A	50 T	-74.5267	10.0147	-7.44173	<0.0001
50 A	25 T+A	-73.4933	10.01486	-7.33843	<0.0001
50 DS	25 T	-68.74	10.01548	-6.86338	<0.0001
50 T+A	25 T	-50.0933	10.01375	-5.00245	<0.0001
50 DS	25 T+A	-44.8067	10.01508	-4.47392	0.0003
50 A	100 DS	-38.7689	10.54312	-3.67718	0.0073
25 T+A	25 T	-35.92	10.01365	-3.5871	0.0101
50 A	25 DS	-23.4	10.01318	-2.33692	0.3197
50 A	25 A	-21.46	10.01231	-2.14336	0.4434
50 T+A	25 T+A	-17.6	10.01332	-1.75766	0.7103
25 A	100 DS	-9.3256	10.54328	-0.8845	0.9938
25 DS	100 DS	2.585	10.54457	0.24515	1
50 DS	100 DS	9.5211	10.54374	0.90301	0.9929
25 DS	25 A	10.8933	10.01347	1.08787	0.9761
50 DS	25 DS	18.0067	10.01369	1.79821	0.6835
50 DS	25 A	26.1933	10.01299	2.61593	0.1797
50 T	25 T	39.12	10.01371	3.90665	0.003
50 DS	50 A	53.58	10.01276	5.35117	<0.0001
50 T+A	50 DS	54.28	10.01412	5.42035	<0.0001
50 T+A	100 DS	62.5167	10.54502	5.92855	<0.0001
25 T+A	100 DS	64.3317	10.54612	6.10003	<0.0001
50 T+A	25 DS	63.9133	10.01394	6.38244	<0.0001
50 T	25 T+A	65.1867	10.01377	6.5097	<0.0001
25 T+A	25 A	69.22	10.01525	6.91146	<0.0001
25 T	100 DS	73.7428	10.54638	6.99224	<0.0001
50 T+A	25 A	70.5533	10.01449	7.04513	<0.0001
25 T+A	25 DS	70.7	10.01523	7.05925	<0.0001
50 T	100 DS	77.5439	10.54628	7.35272	<0.0001
25 T	25 DS	75.6867	10.01553	7.55693	<0.0001
25 T	25 A	75.92	10.01552	7.58023	<0.0001
50 T+A	50 A	76	10.01455	7.58896	<0.0001
50 T	50 DS	77.1733	10.01542	7.70545	<0.0001
50 T	25 DS	77.8	10.01546	7.76799	<0.0001
50 T	25 A	77.9867	10.01533	7.78673	<0.0001
50 T	50 A	77.9933	10.01492	7.78772	<0.0001

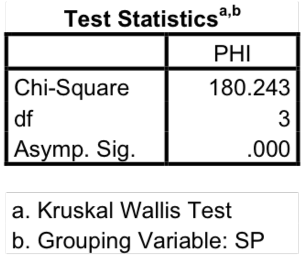


Test of Homogeneity of Variances

PHI

Levene Statistic	df1	df2	Sig.
11.367	3	1376	.000

Figure N-21: SPSS output for Levene test of equal variances for the 22 aircraft PHI values against SP.



Test Statistics[a,b]

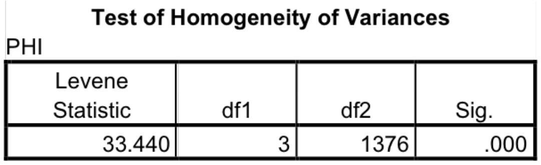
	PHI
Chi-Square	180.243
df	3
Asymp. Sig.	.000

a. Kruskal Wallis Test
b. Grouping Variable: SP

Figure N-22: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 22 aircraft PHI values against SP.

Table N.11: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Primary Halo Incursion (PHI) 22 aircraft cases versus Safety Protocols.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
DS	A	197.1511	13.05863	15.09738	<.0001
T+A	DS	183.4178	13.05781	14.0466	<.0001
T+A	A	177.9444	10.96205	16.23278	<.0001
T	A	163.5389	10.96168	14.91914	<.0001
T	DS	112.1778	13.05428	8.59318	<.0001
T+A	T	49.8222	10.94888	4.55044	<.0001

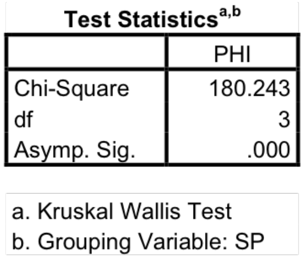


Test of Homogeneity of Variances

PHI

Levene Statistic	df1	df2	Sig.
33.440	3	1376	.000

Figure N-23: SPSS output for Levene test of equal variances for the 34 aircraft PHI values against SP.



Test Statistics[a,b]

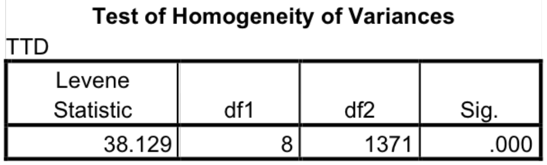
	PHI
Chi-Square	180.243
df	3
Asymp. Sig.	.000

a. Kruskal Wallis Test
b. Grouping Variable: SP

Figure N-24: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 34 aircraft PHI values against SP.

Table N.12: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Primary Halo Incursion (PHI) 22 aircraft cases versus Safety Protocols.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
T	DS	237.973	13.0727	18.2038	<.0001
T+A	DS	211.622	13.07027	16.1911	<.0001
T	A	179.894	10.96536	16.4057	<.0001
T+A	A	175.567	10.96457	16.0122	<.0001
DS	A	82.547	13.06124	6.32	<.0001
T+A	T	-126.75	10.9616	-11.5631	<.0001

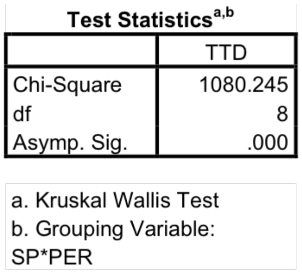


Test of Homogeneity of Variances

TTD

Levene Statistic	df1	df2	Sig.
38.129	8	1371	.000

Figure N-25: SPSS output for Levene test of equal variances for the 22 aircraft TTD values against SP-Composition pairs.

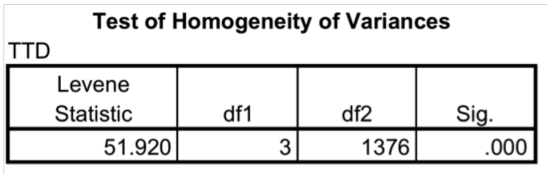


Test Statistics[a,b]

	TTD
Chi-Square	1080.245
df	8
Asymp. Sig.	.000

a. Kruskal Wallis Test
b. Grouping Variable: SP*PER

Figure N-26: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 22 aircraft TTD values against SP-Composition pairs.



Test of Homogeneity of Variances

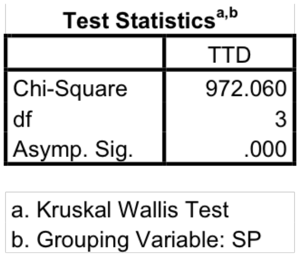
TTD

Levene Statistic	df1	df2	Sig.
51.920	3	1376	.000

Figure N-27: SPSS output for Levene test of equal variances for the 22 aircraft TTD values against SP.

Table N.13: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Total Taxi Distance (TTD) 22 aircraft cases versus SP-Composition pairs.

Level	-Level	Score Mean Difference	Std Err Dif	Z	p-Value
50 A	100 DS	-160.411	10.54764	-15.2082	<0.0001
50 A	25 T	-149.993	10.01665	-14.9744	<0.0001
50 A	25 T+A	-149.993	10.01665	-14.9744	<0.0001
50 A	25 DS	-149.913	10.01665	-14.9664	<0.0001
50 DS	25 T+A	-148.673	10.01665	-14.8426	<0.0001
50 A	25 A	-147.433	10.01665	-14.7188	<0.0001
50 T+A	25 T+A	-143.9	10.01665	-14.3661	<0.0001
50 DS	25 T	-143.82	10.01665	-14.3581	<0.0001
50 T	25 T+A	-97.78	10.01665	-9.7617	<0.0001
50 T	25 T	-76.54	10.01665	-7.6413	<0.0001
50 T+A	25 T	-72.993	10.01665	-7.2872	<0.0001
25 A	100 DS	-35.976	10.54764	-3.4108	0.0187
50 DS	25 DS	-27.82	10.01665	-2.7774	0.122
25 T+A	25 T	14.473	10.01665	1.4449	0.8804
50 DS	100 DS	23.729	10.54764	2.2497	0.3733
50 T+A	50 T	35.007	10.01665	3.4948	0.014
25 DS	100 DS	51.217	10.54764	4.8558	<0.0001
50 DS	25 A	53.647	10.01665	5.3557	<0.0001
25 DS	25 A	74.367	10.01665	7.4243	<0.0001
50 T	25 DS	96.193	10.01665	9.6033	<0.0001
50 T	50 DS	107.06	10.01665	10.6882	<0.0001
50 T	100 DS	124.221	10.54764	11.7771	<0.0001
50 T	25 A	125.527	10.01665	12.5318	<0.0001
50 T+A	25 DS	136.167	10.01665	13.594	<0.0001
50 T+A	50 DS	139.06	10.01665	13.8829	<0.0001
25 T	25 DS	141.66	10.01665	14.1424	<0.0001
50 T+A	25 A	146.9	10.01665	14.6656	<0.0001
50 T+A	100 DS	154.691	10.54764	14.6659	<0.0001
25 T	25 A	148.02	10.01665	14.7774	<0.0001
50 DS	50 A	149.473	10.01665	14.9225	<0.0001
25 T+A	25 DS	149.713	10.01665	14.9464	<0.0001
25 T+A	25 A	149.993	10.01665	14.9744	<0.0001
50 T	50 A	149.993	10.01665	14.9744	<0.0001
50 T+A	50 A	149.993	10.01665	14.9744	<0.0001
25 T	100 DS	159.433	10.54764	15.1155	<0.0001
25 T+A	100 DS	164.798	10.54764	15.6242	<0.0001



Test Statistics[a,b]

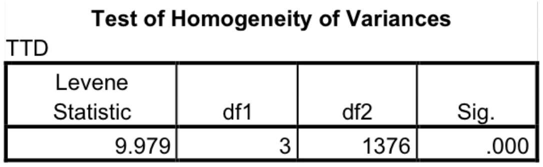
	TTD
Chi-Square	972.060
df	3
Asymp. Sig.	.000

a. Kruskal Wallis Test
b. Grouping Variable: SP

Figure N-28: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 22 aircraft TTD values against SP.

Table N.14: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Total Taxi Distance (TTD) 22 aircraft cases versus SP.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
T+A	DS	374.6356	16.58243	22.59232	<.0001
T	DS	324.2715	16.58243	19.55512	<.0001
T+A	A	298.45	14.15392	21.08604	<.0001
T	A	286.7767	14.15392	20.2613	<.0001
DS	A	260.6798	16.58243	15.72024	<.0001
T+A	T	37.1367	14.15392	2.62377	0.0432

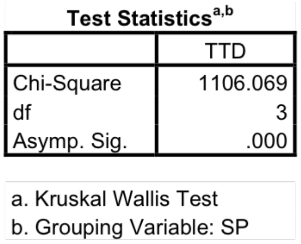


Test of Homogeneity of Variances

TTD

Levene Statistic	df1	df2	Sig.
9.979	3	1376	.000

Figure N-29: SPSS output for Levene test of equal variances for the 34 aircraft TTD values against SP.



Test Statistics[a,b]

	TTD
Chi-Square	1106.069
df	3
Asymp. Sig.	.000

a. Kruskal Wallis Test

b. Grouping Variable: SP

Figure N-30: SPSS output for the non-parametric Kruskal-Wallis one-way analysis of variance for the 34 aircraft TTD values against SP.

Table N.15: JMP PRO 10 output for Steel-Dwass Multiple Comparisons tests for Total Taxi Distance (TTD) 34 aircraft cases versus SP.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
T	DS	389.932	16.58243	23.5148	<.0001
T+A	DS	372.978	16.58243	22.4924	<.0001
T	A	299.997	14.15392	21.1953	<.0001
T+A	A	297.55	14.15392	21.0225	<.0001
DS	A	218.018	16.58243	13.1475	<.0001
T+A	T	-244.097	14.15392	-17.2459	<.0001

N.2 Full Color Pareto Charts

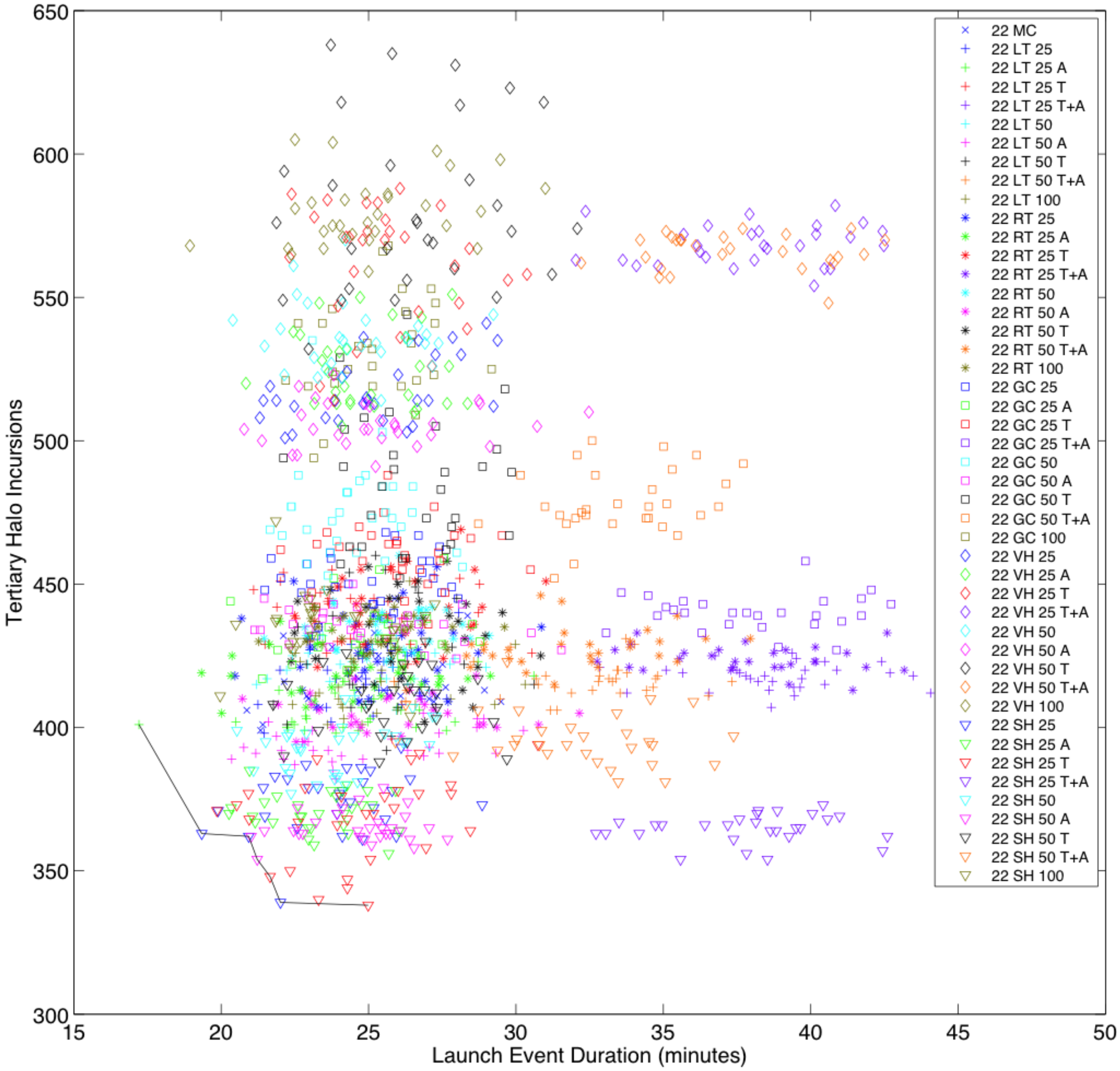


Figure N-31: Pareto frontier plot of Tertiary Halo Incursions (THI) versus LD for 22 aircraft missions.

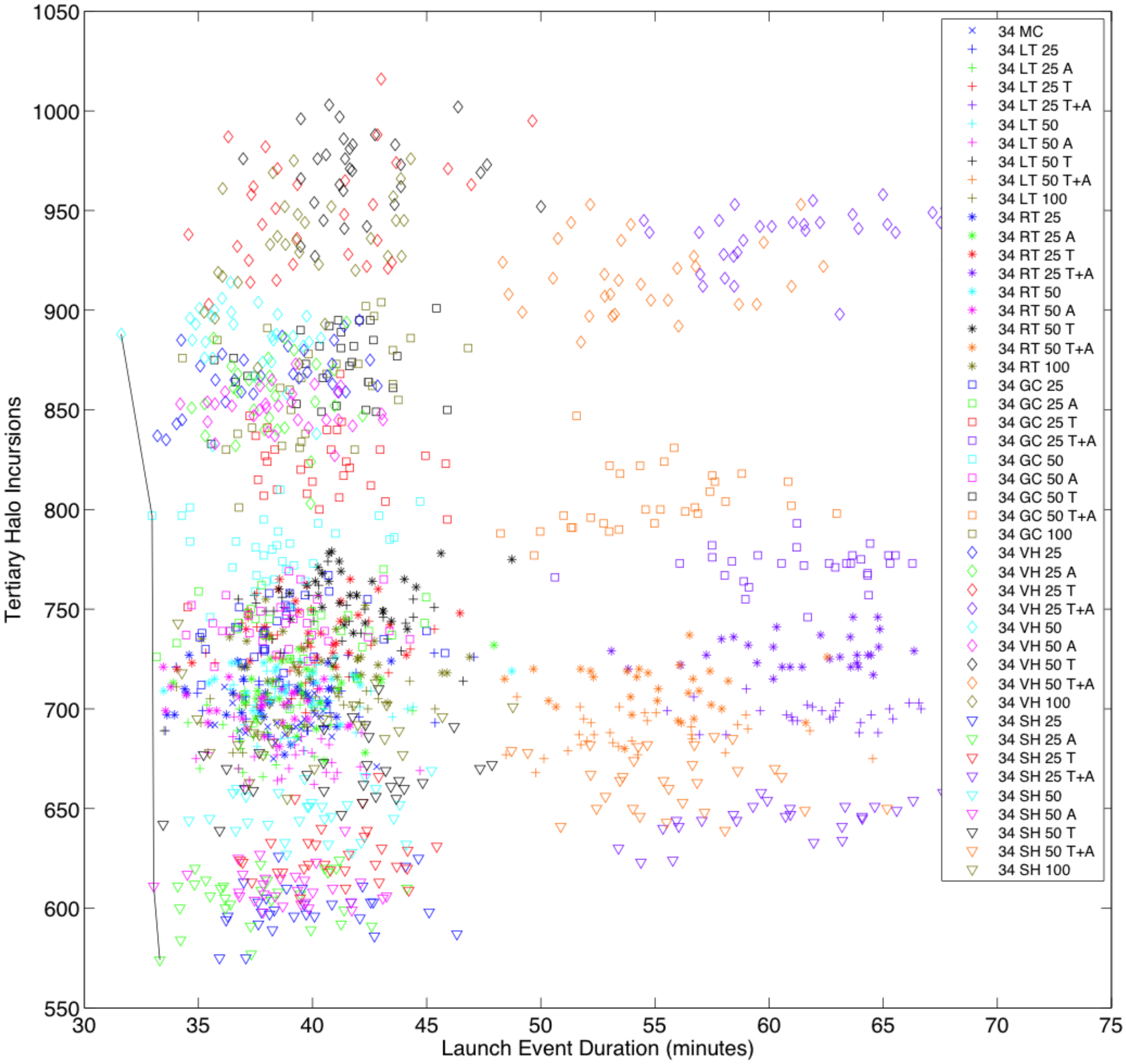


Figure N-32: Pareto frontier plot of Tertiary Halo Incursions (THI) versus LD for 34 aircraft missions.

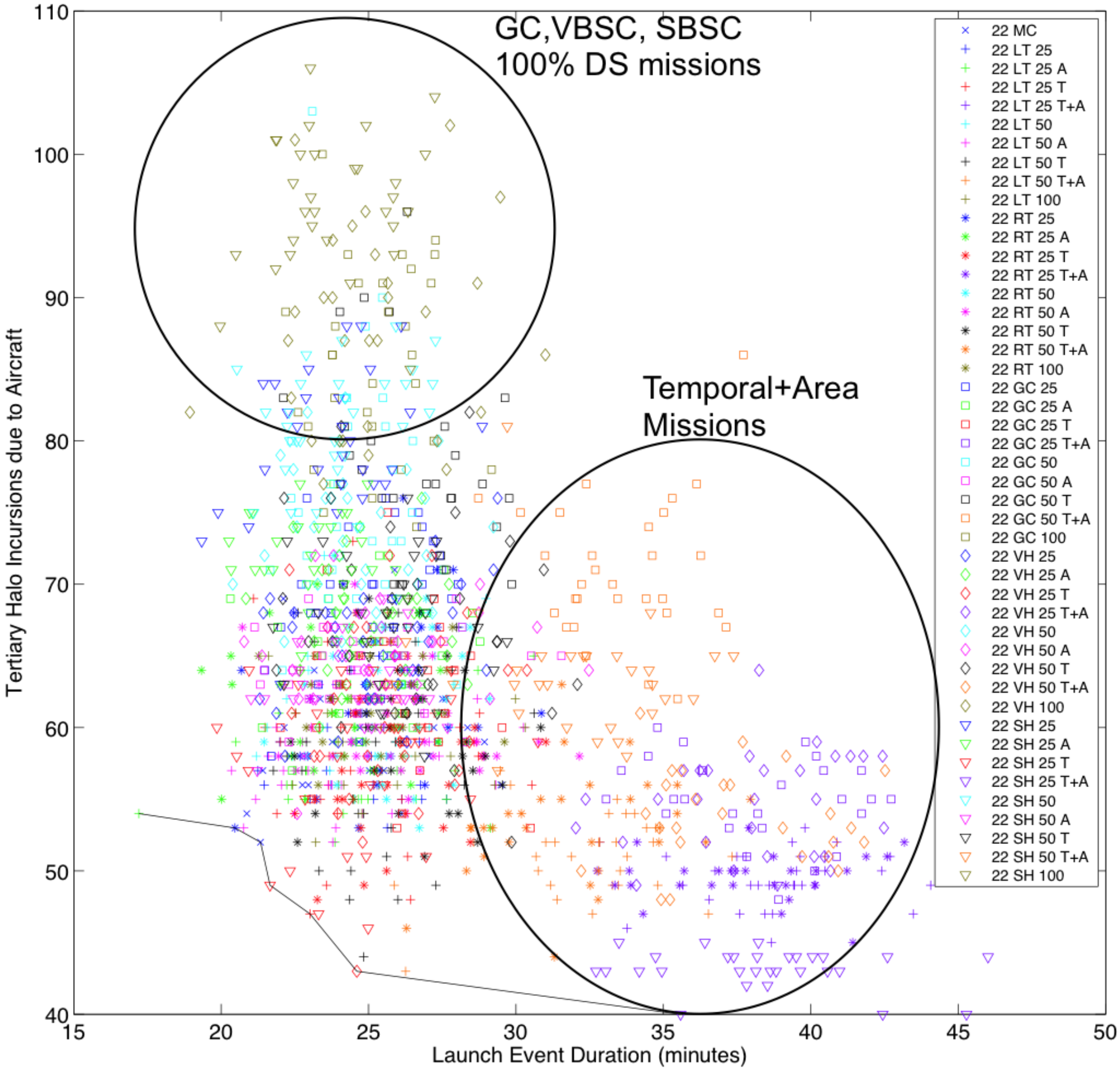


Figure N-33: Pareto frontier plot of Tertiary Halo Incursions due to Aircraft (THIA) versus LD for 22 aircraft missions.

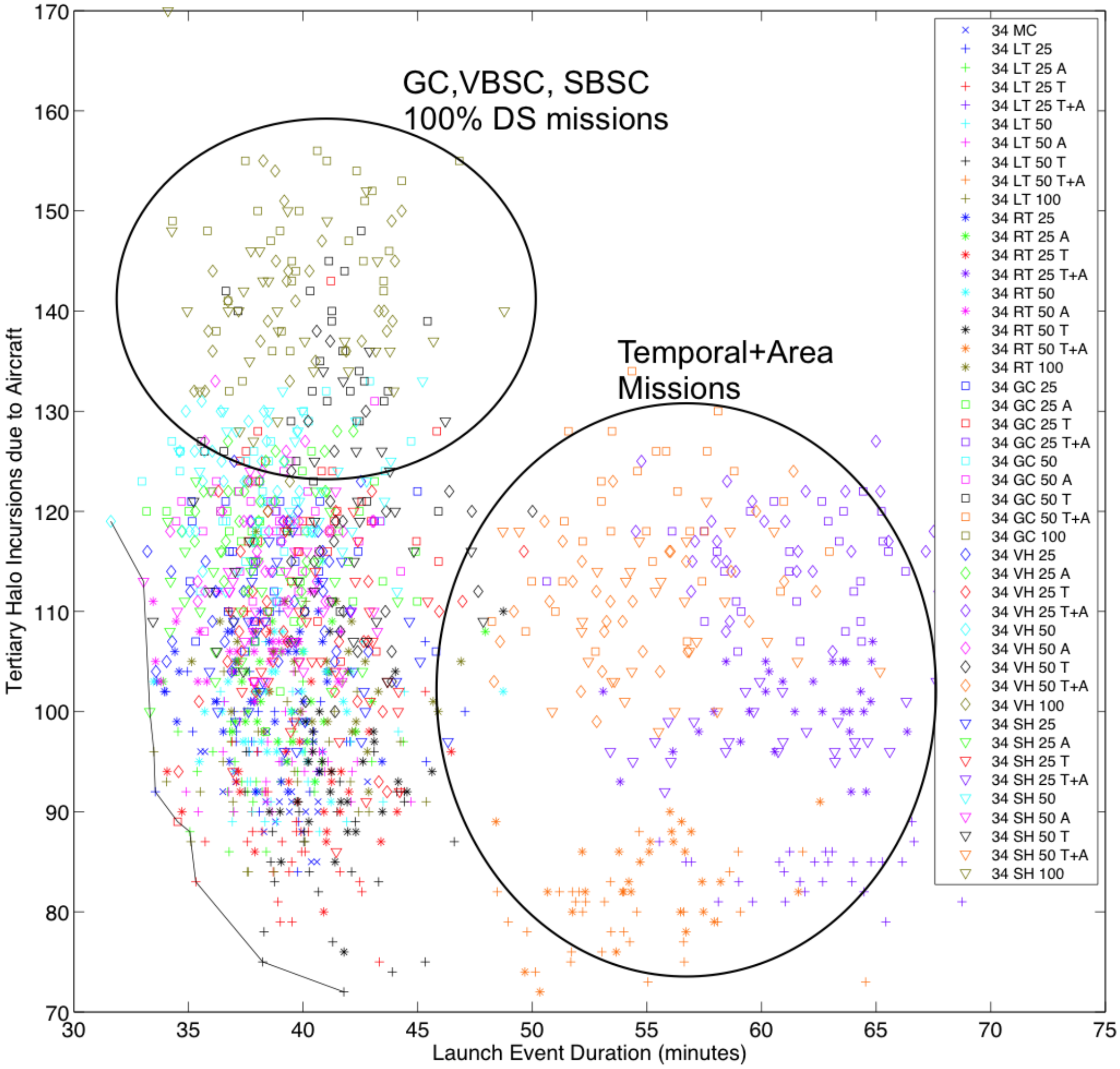


Figure N-34: Pareto frontier plot of Tertiary Halo Incursions due to Aircraft (THIA) versus LD for 34 aircraft missions.

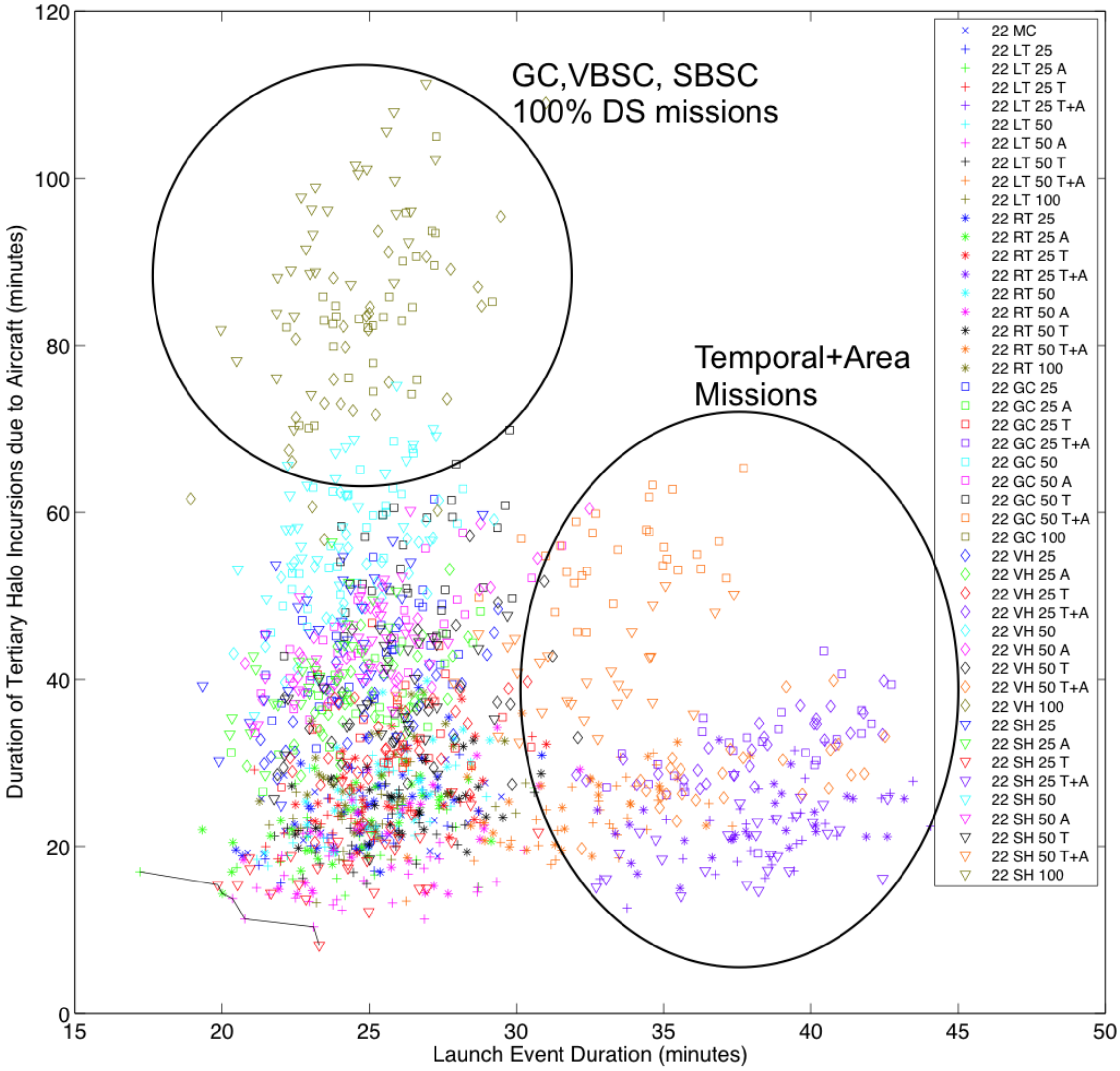


Figure N-35: Pareto frontier plot of Duration of Tertiary Halo Incursions due to Aircraft (DTHIA) versus LD for 22 aircraft missions.

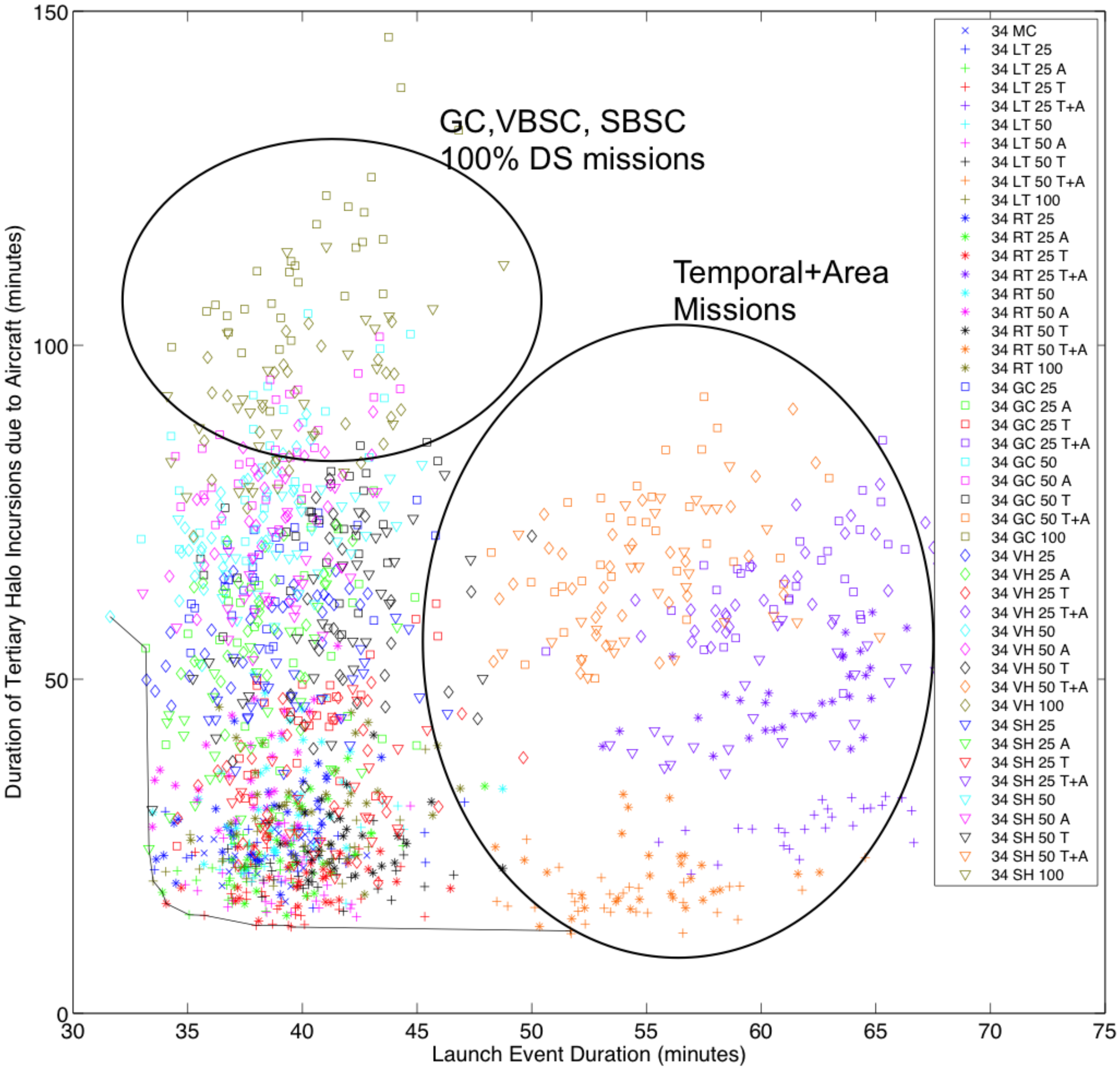


Figure N-36: Pareto frontier plot of Duration of Tertiary Halo Incursions due to Aircraft (DTHIA) versus LD for 34 aircraft missions.

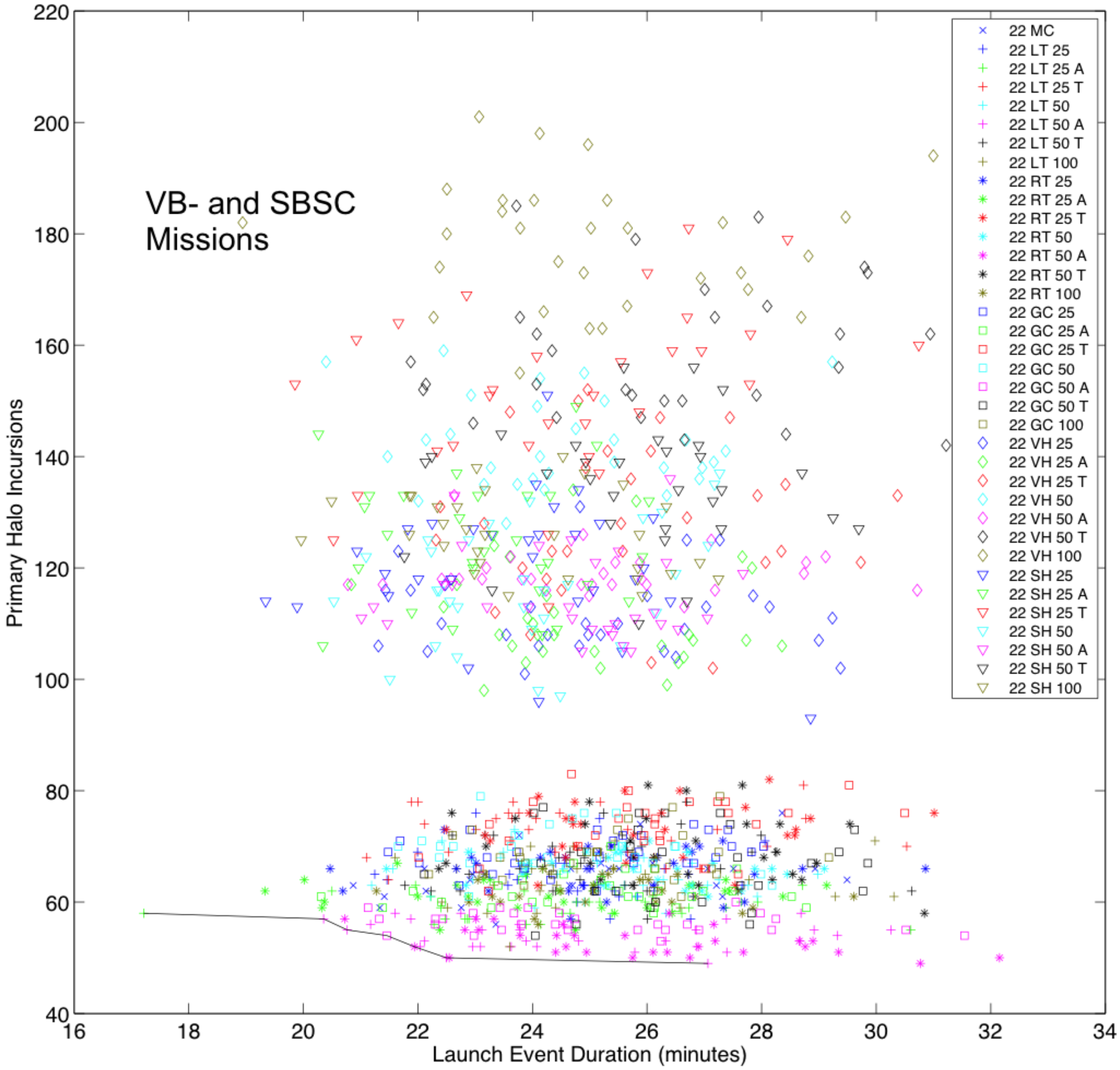


Figure N-37: Pareto frontier plot of Primary Halo Incursion (PHI) versus LD for 22 aircraft missions, excluding Temporal+Area cases.

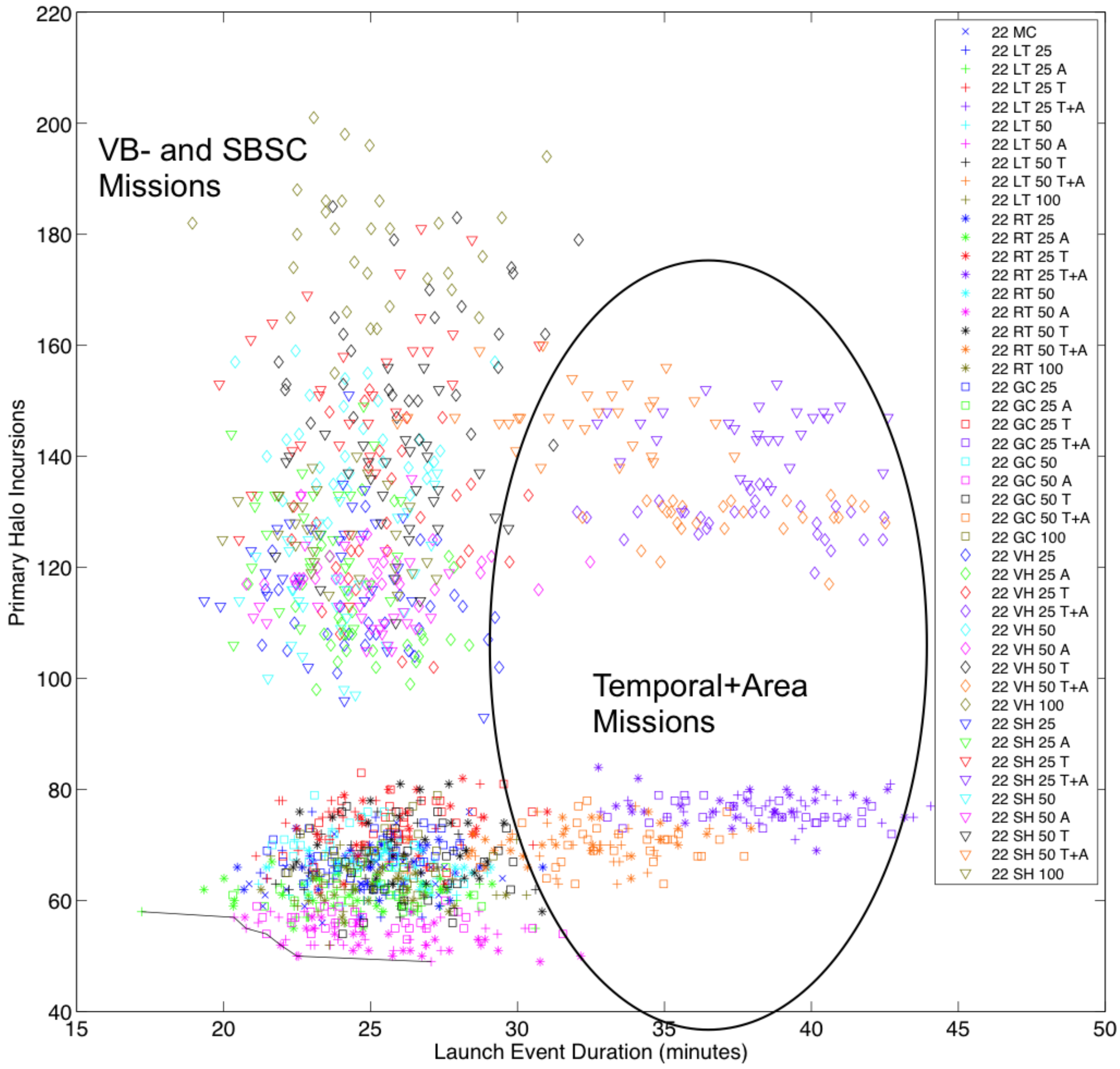


Figure N-38: Pareto frontier plot of Primary Halo Incursion (PHI) versus LD for 22 aircraft missions.

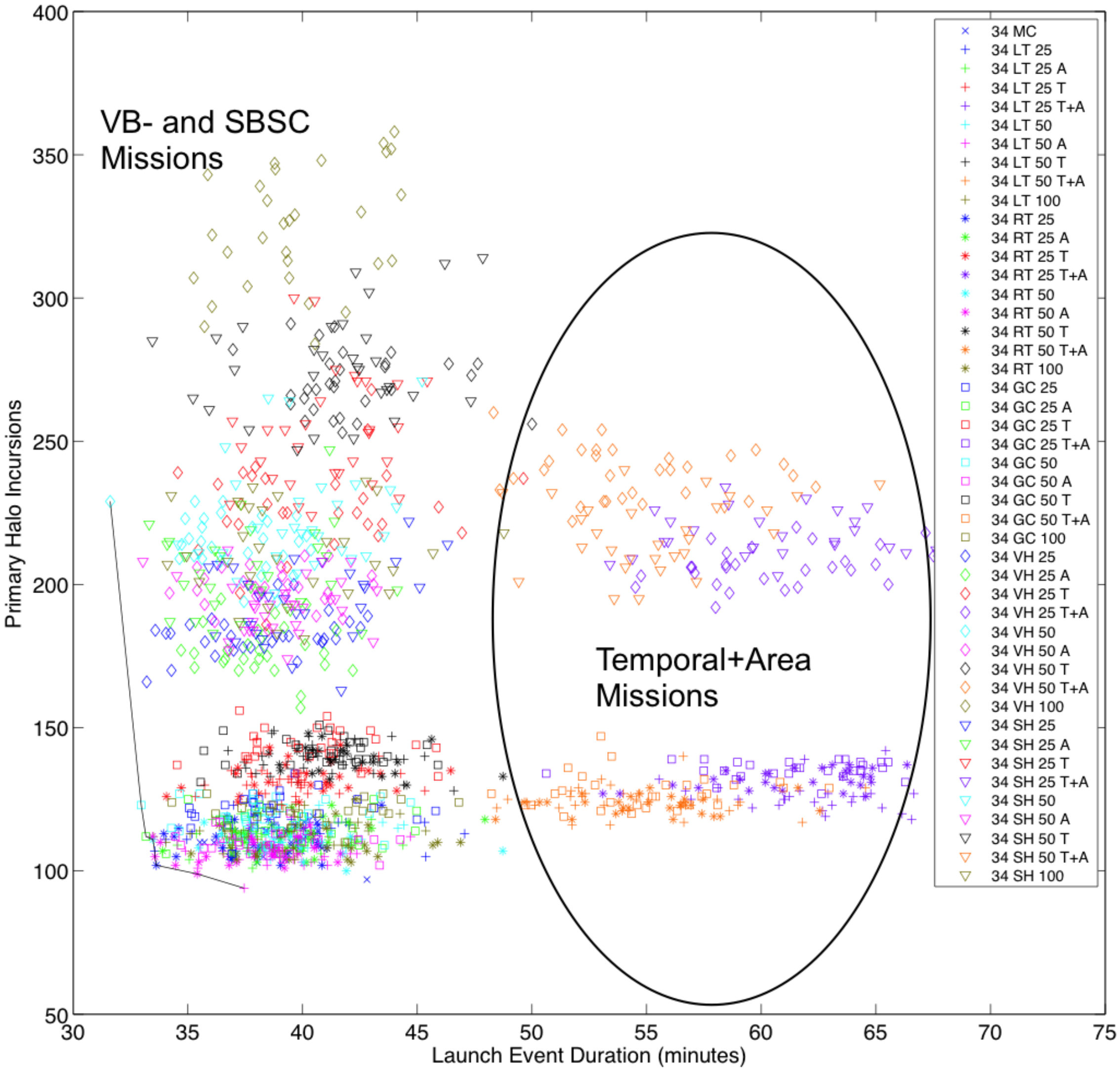


Figure N-39: Pareto frontier plot of Primary Halo Incursion (PHI) versus LD for 34 aircraft missions.

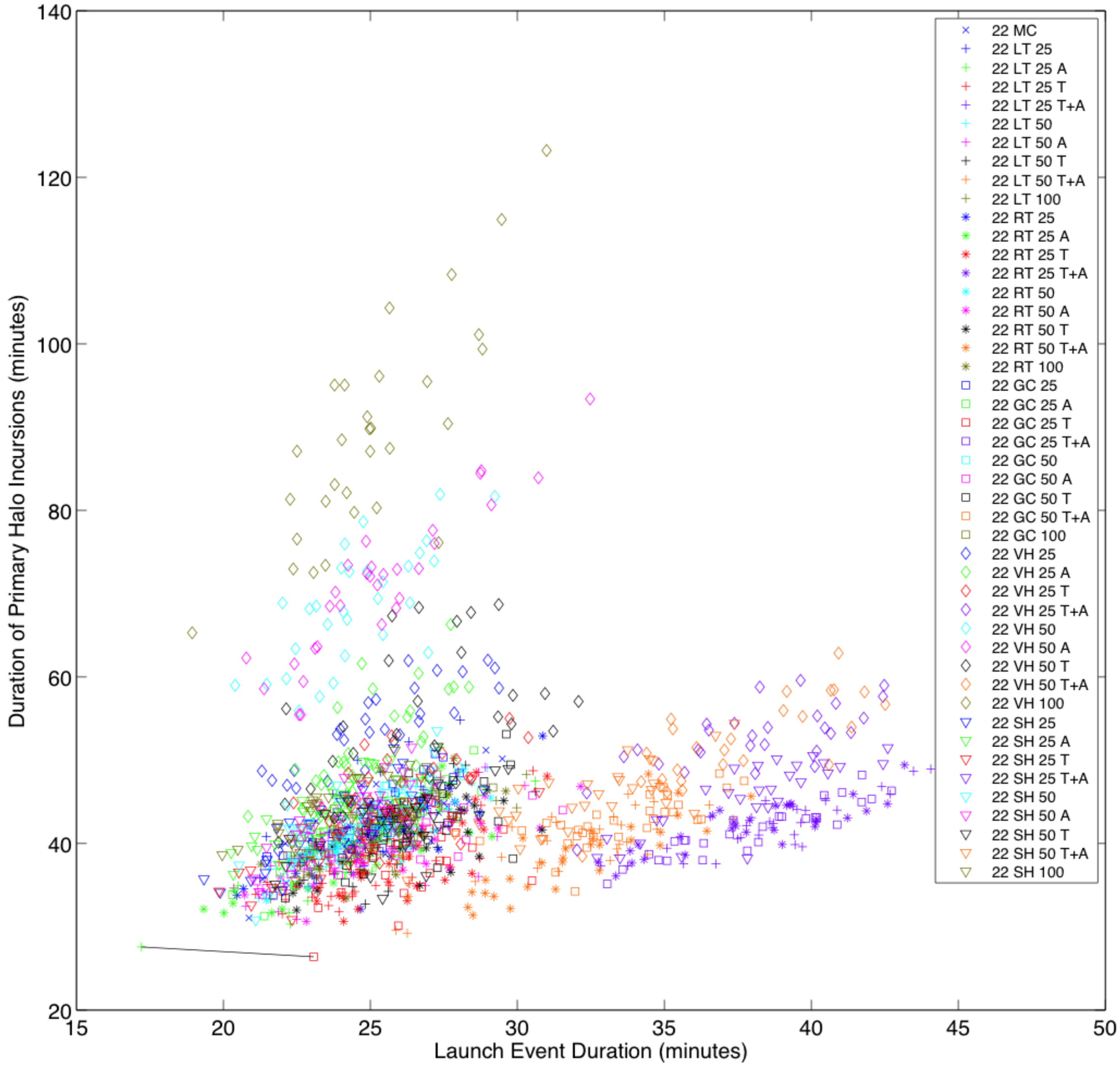


Figure N-40: Pareto frontier plot of Duration of Primary Halo Incursions (DPHI) versus LD for 22 aircraft missions.

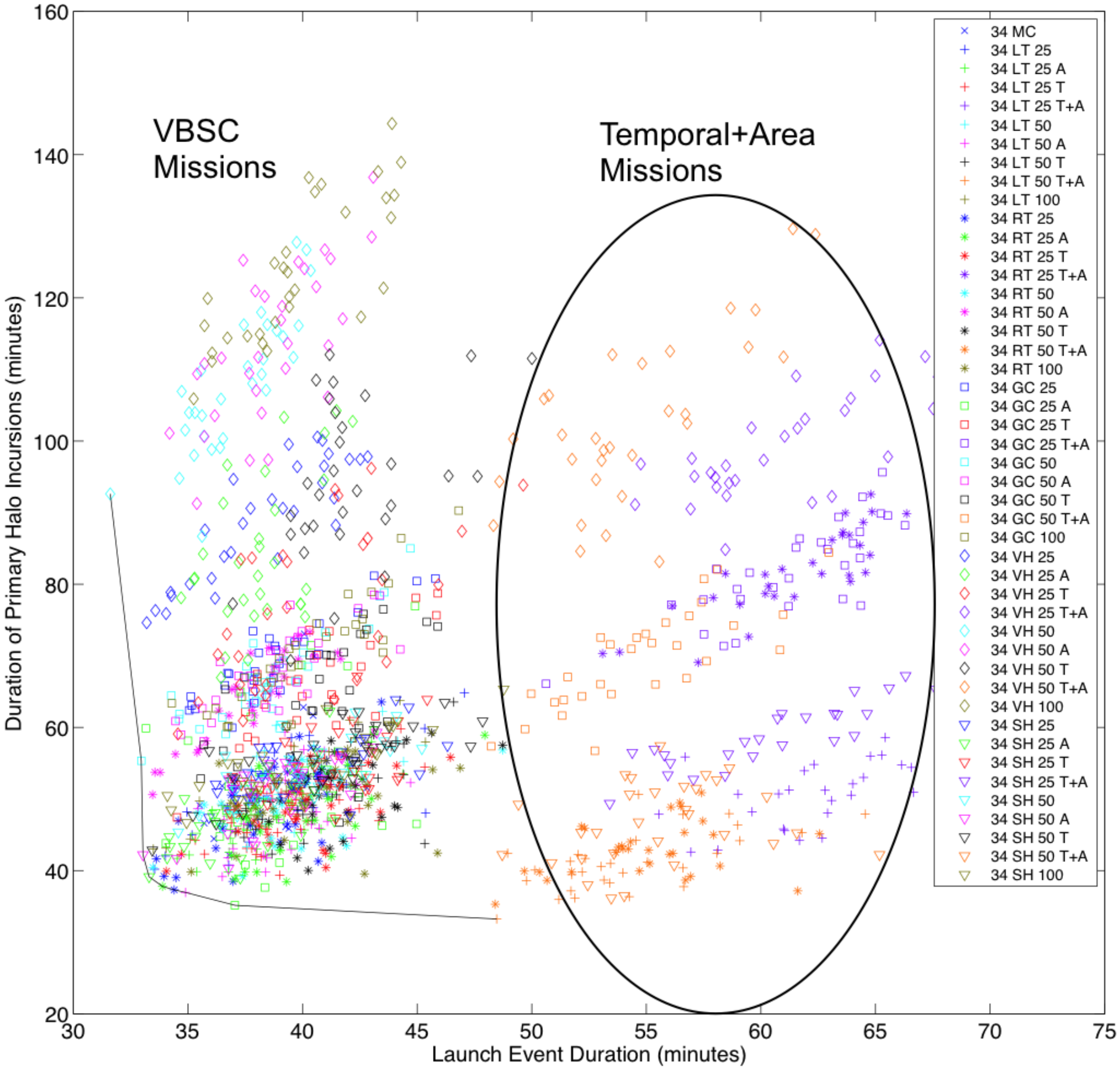


Figure N-41: Pareto frontier plot of Duration of Primary Halo Incursions (DPHI) versus LD for 34 aircraft missions.

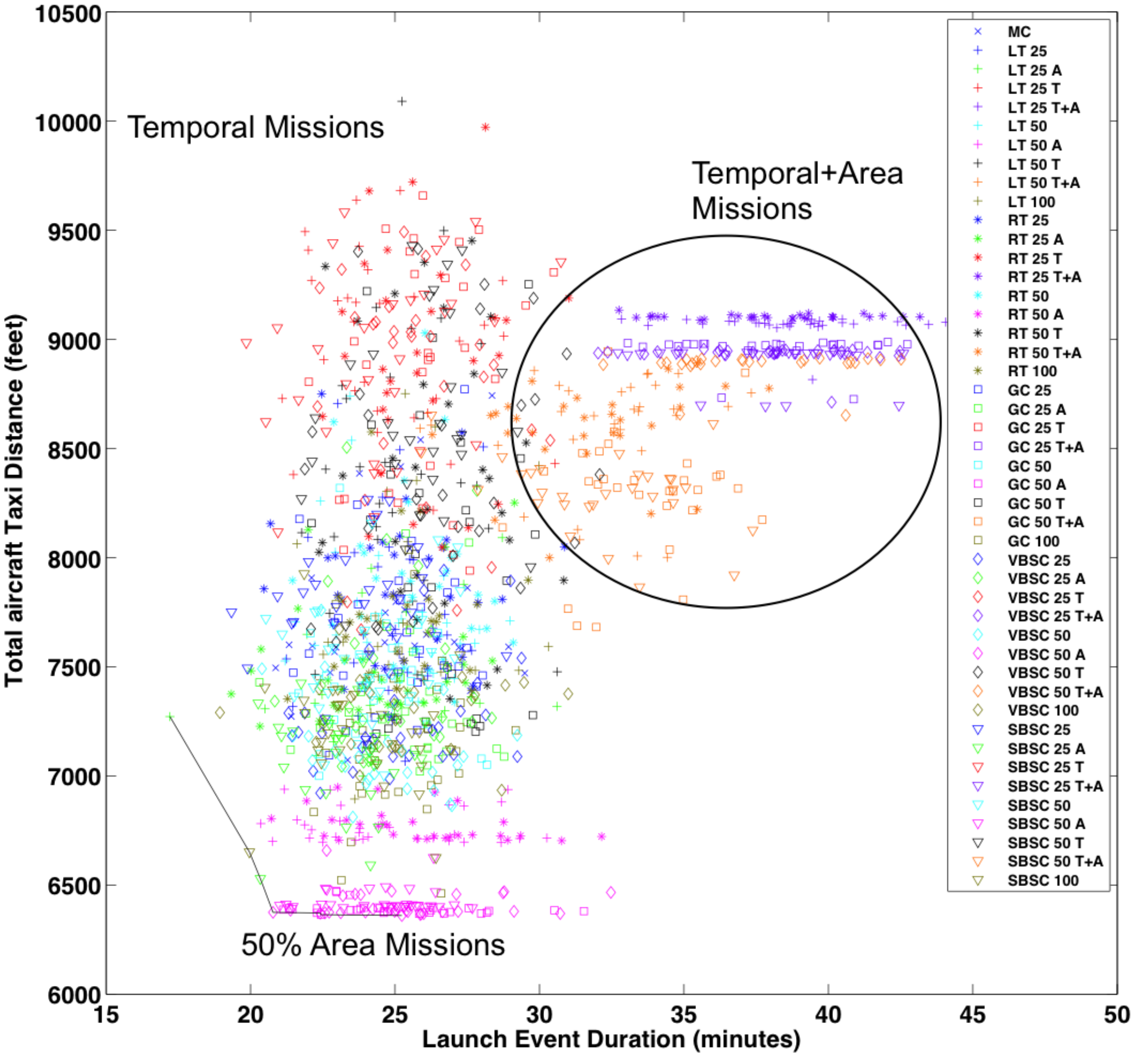


Figure N-42: Pareto frontier plot of Total Taxi Distance (TTD) versus LD for 22 aircraft missions.

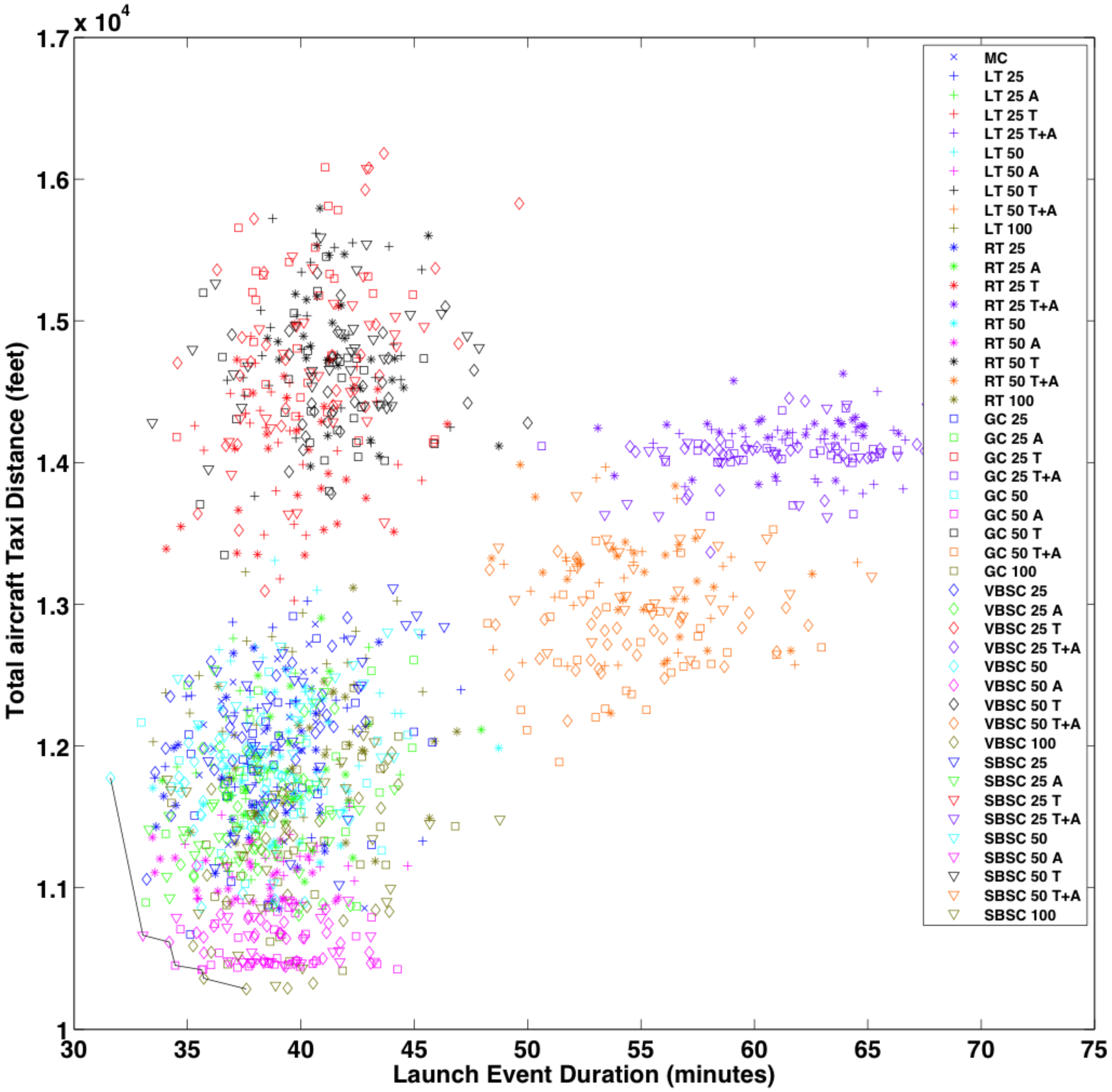


Figure N-43: Pareto frontier plot of Total Taxi Distance (TTD) versus LD for 34 aircraft missions.

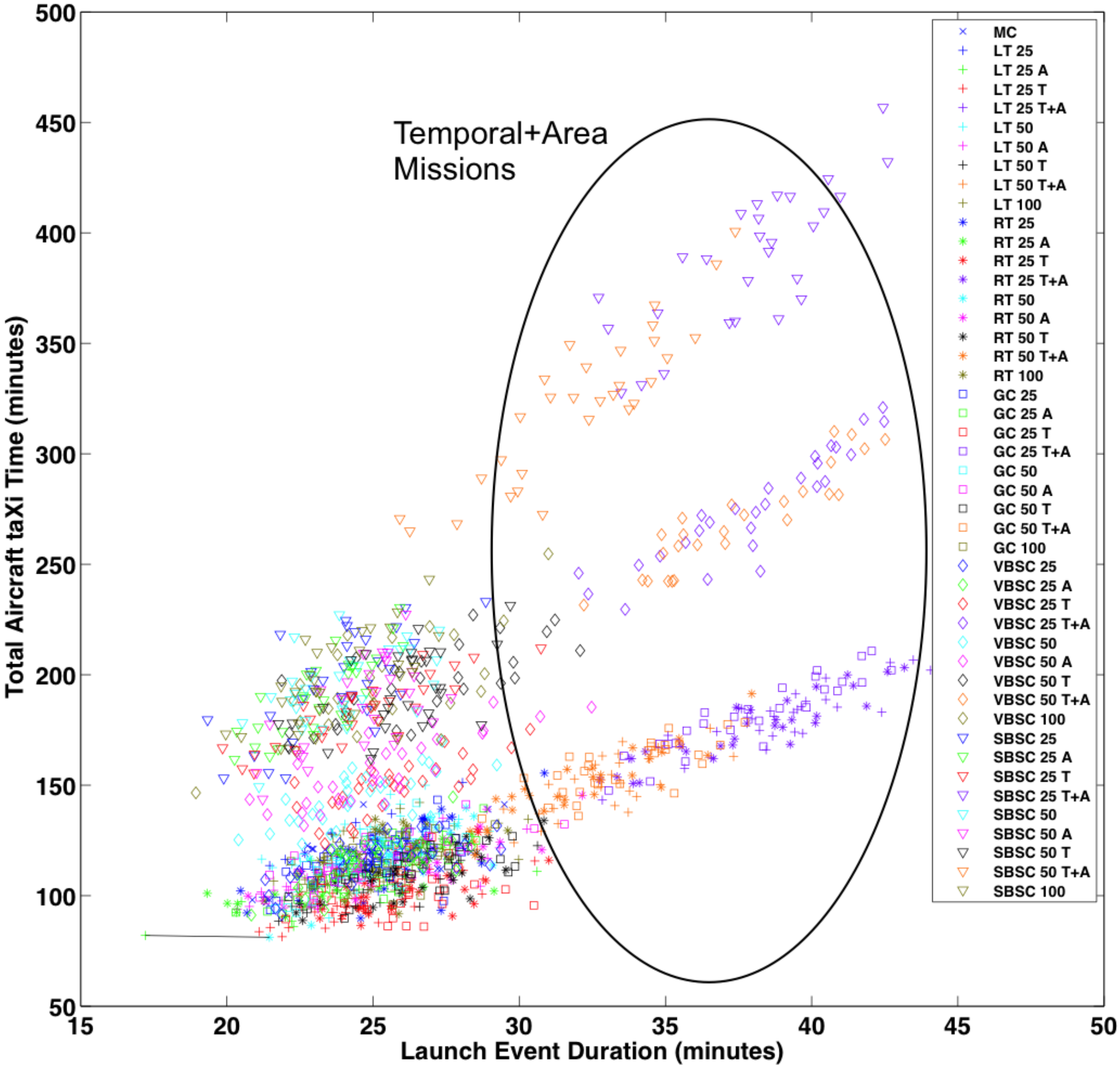


Figure N-44: Pareto frontier plot of Total Aircraft taXi Time (TAXT) versus LD for 22 aircraft missions.

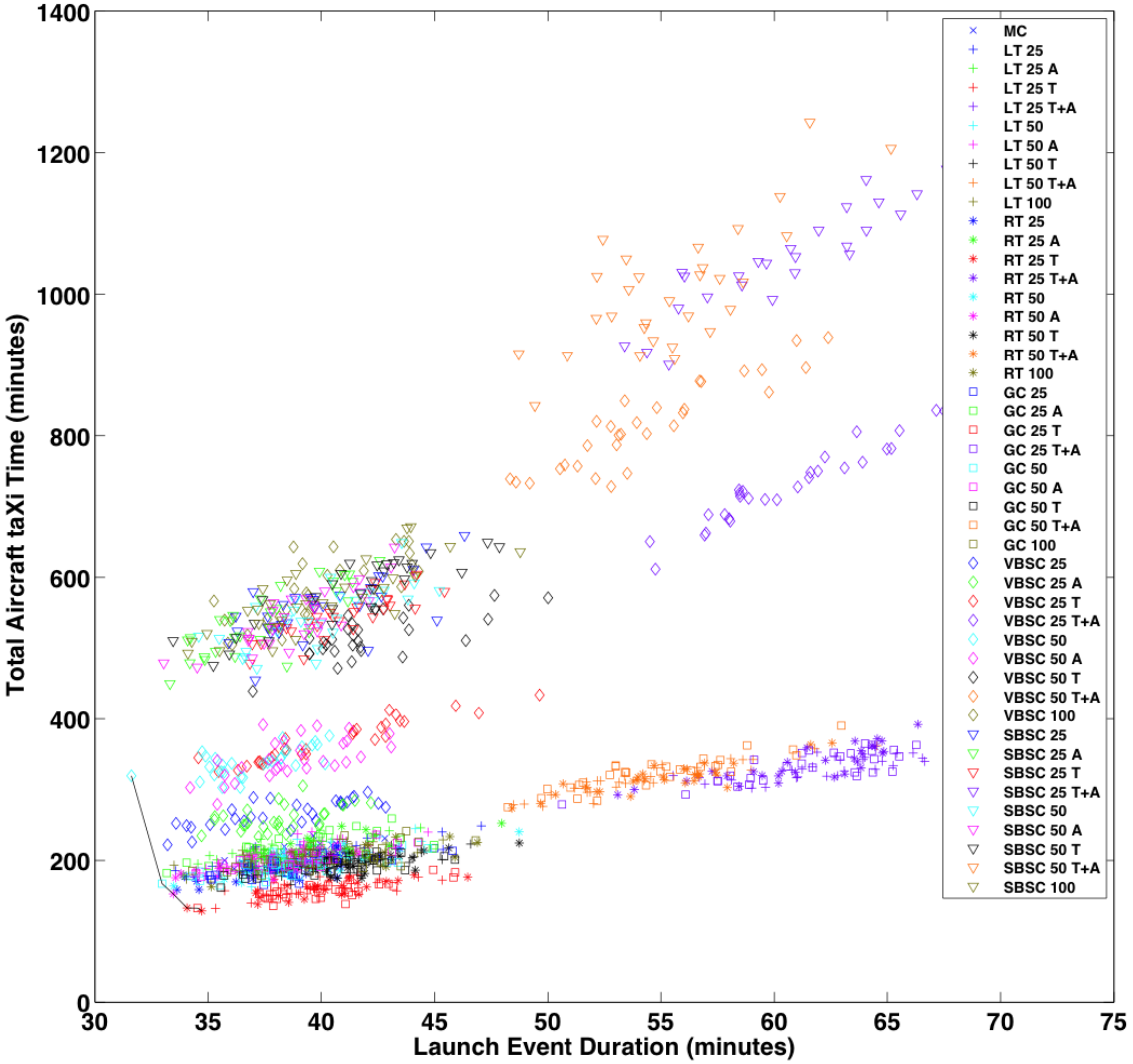


Figure N-45: Pareto frontier plot of Total Aircraft taXi Time (TAXT) versus LD for 34 aircraft missions.

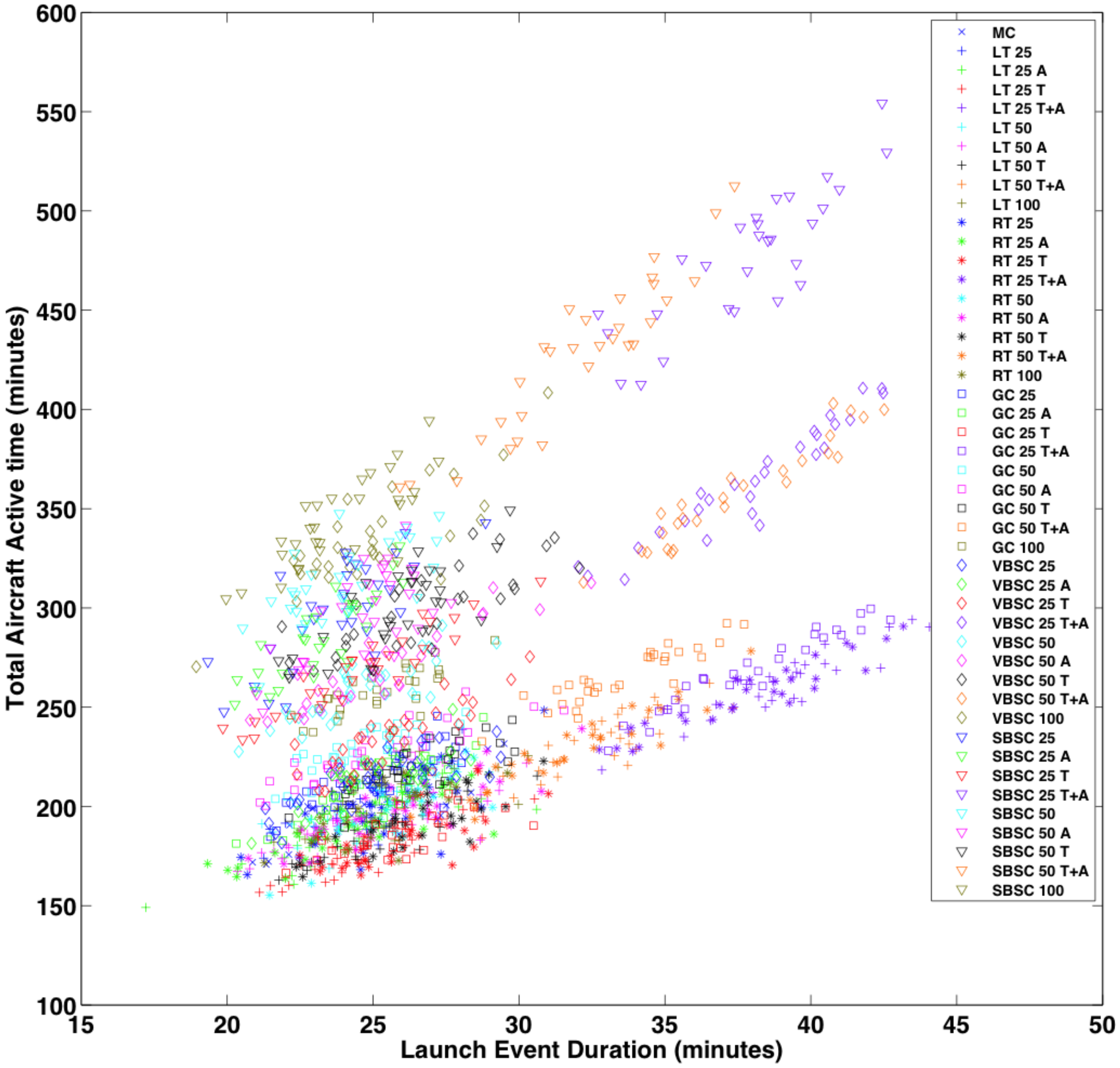


Figure N-46: Pareto frontier plot of Total Aircraft Active Time (TAAT) versus LD for 22 aircraft missions.

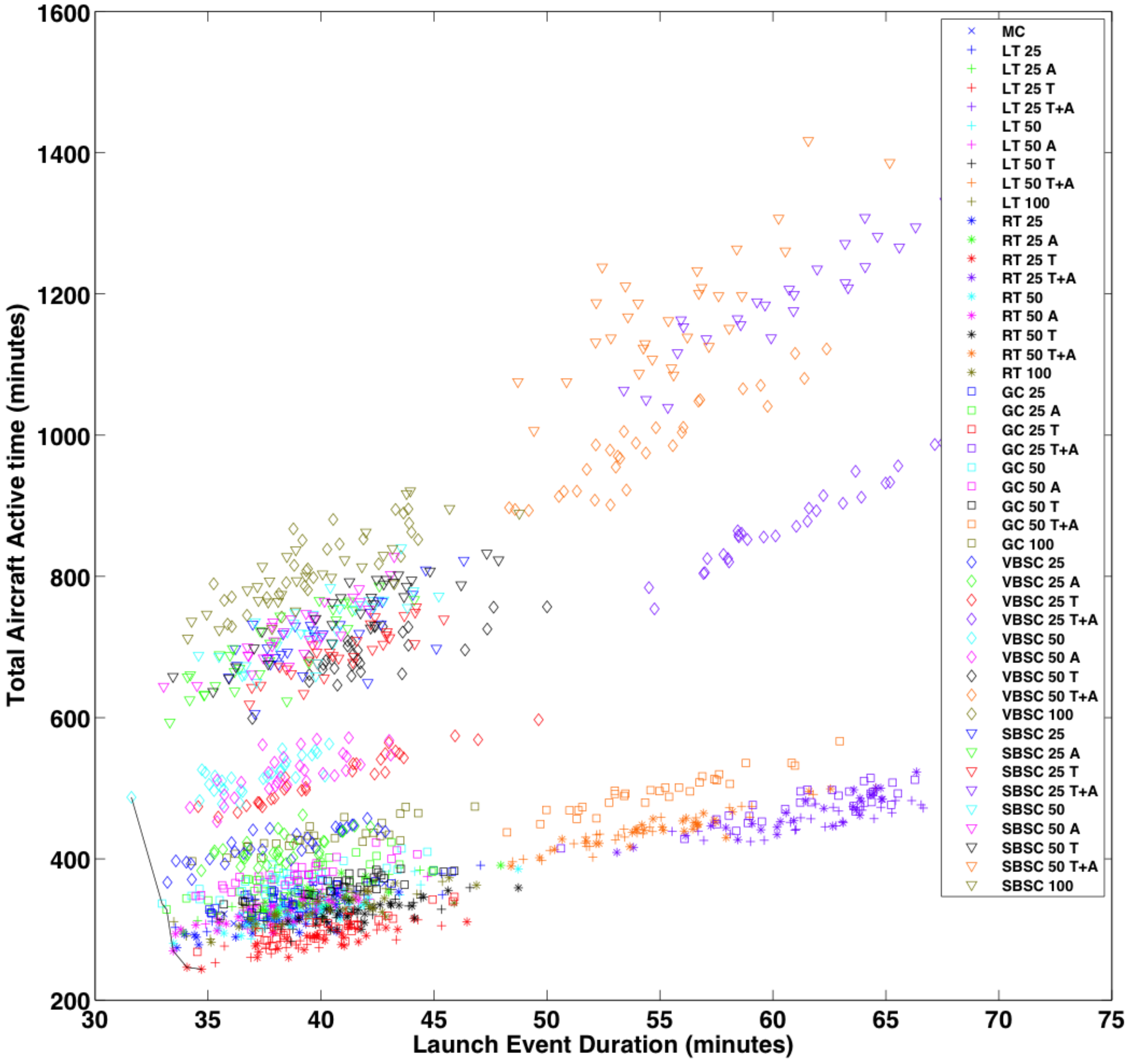


Figure N-47: Pareto frontier plot of Total Aircraft Active Time (TAAT) versus LD for 34 aircraft missions.

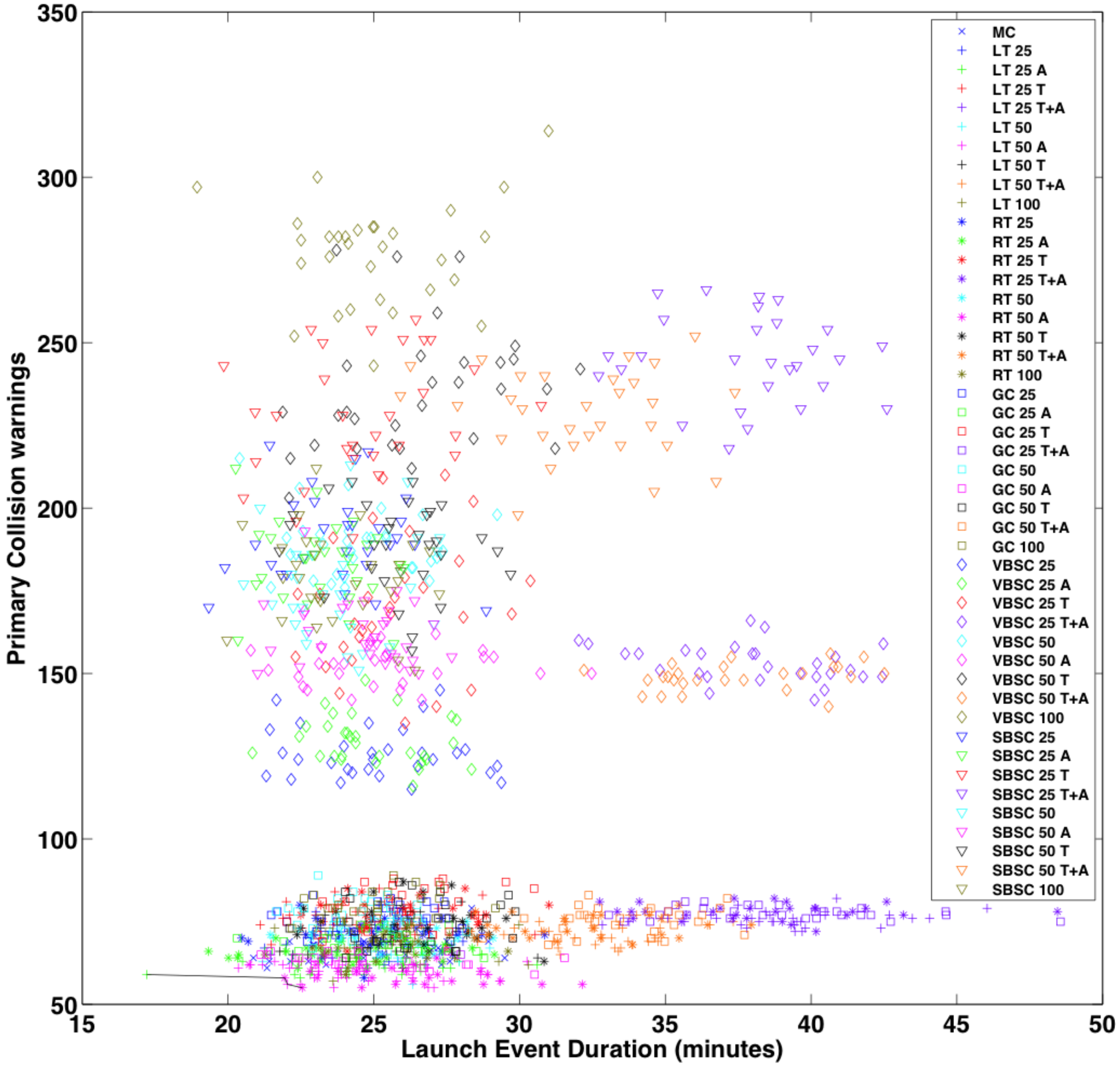


Figure N-48: Pareto frontier plot of Primary Collision warnings (PC) versus LD for 22 aircraft missions.

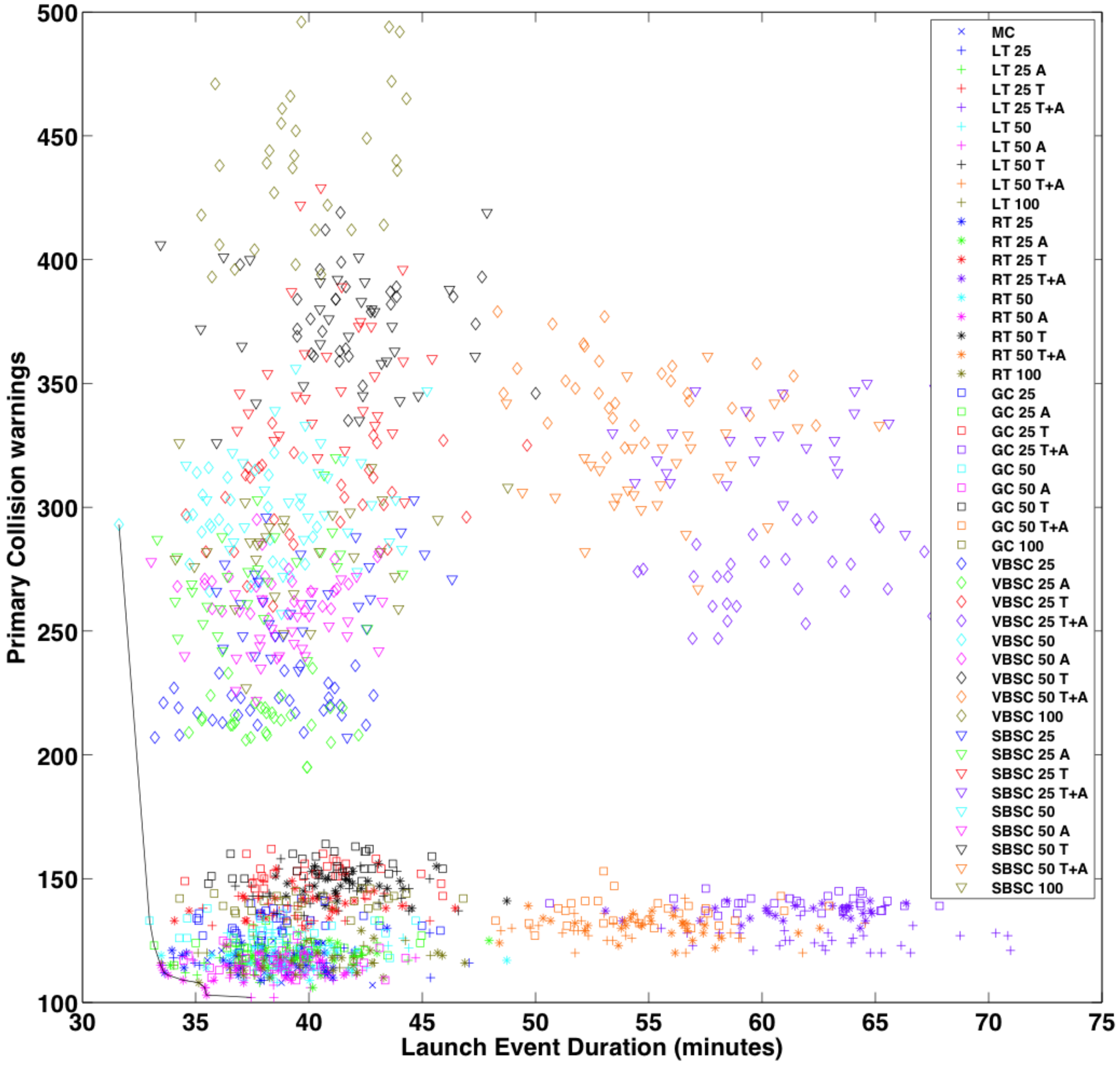


Figure N-49: Pareto frontier plot of Primary Collision warnings (PC) versus LD for 34 aircraft missions.

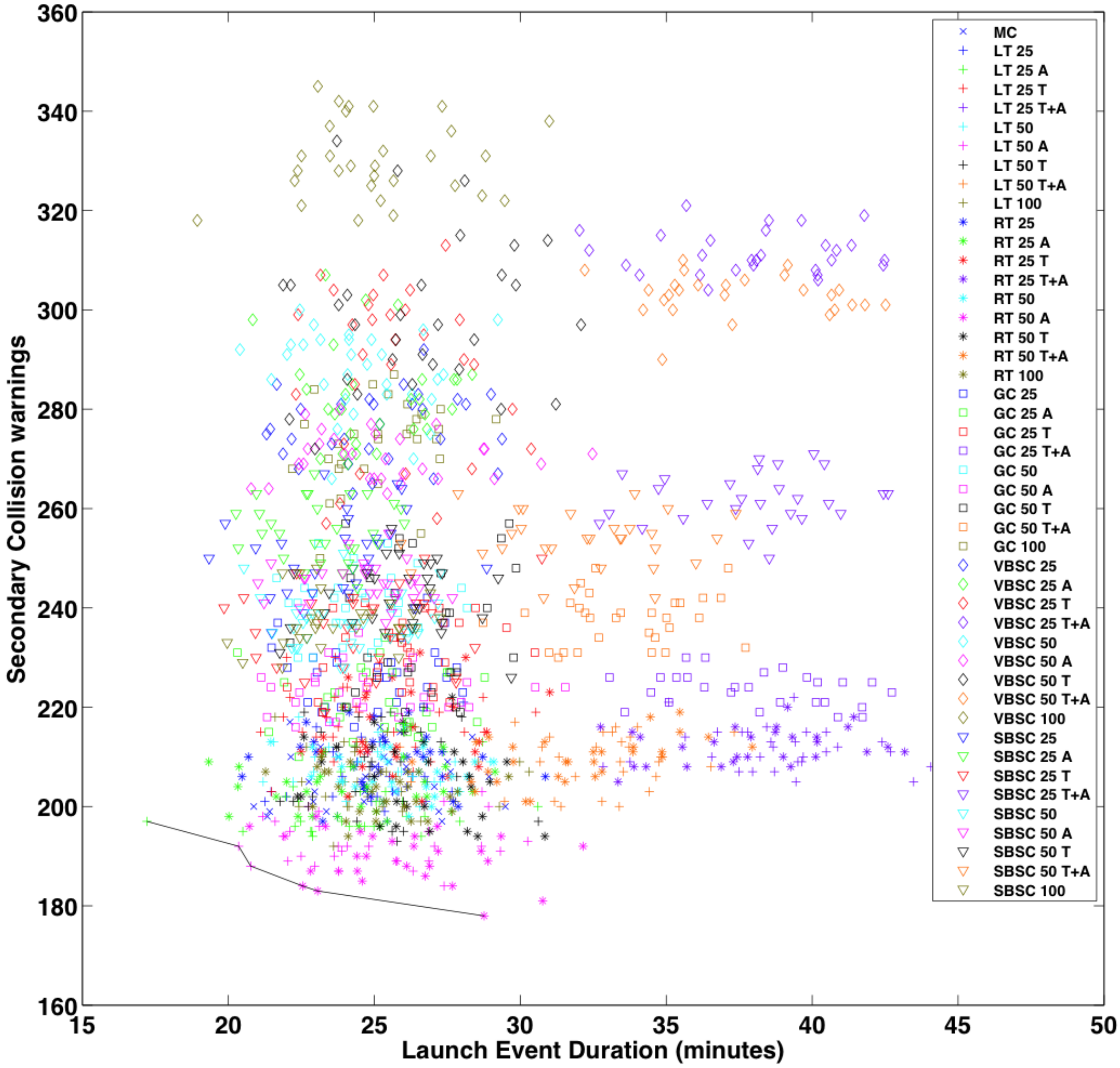


Figure N-50: Pareto frontier plot of Secondary Collision warnings (SC) versus LD for 22 aircraft missions.

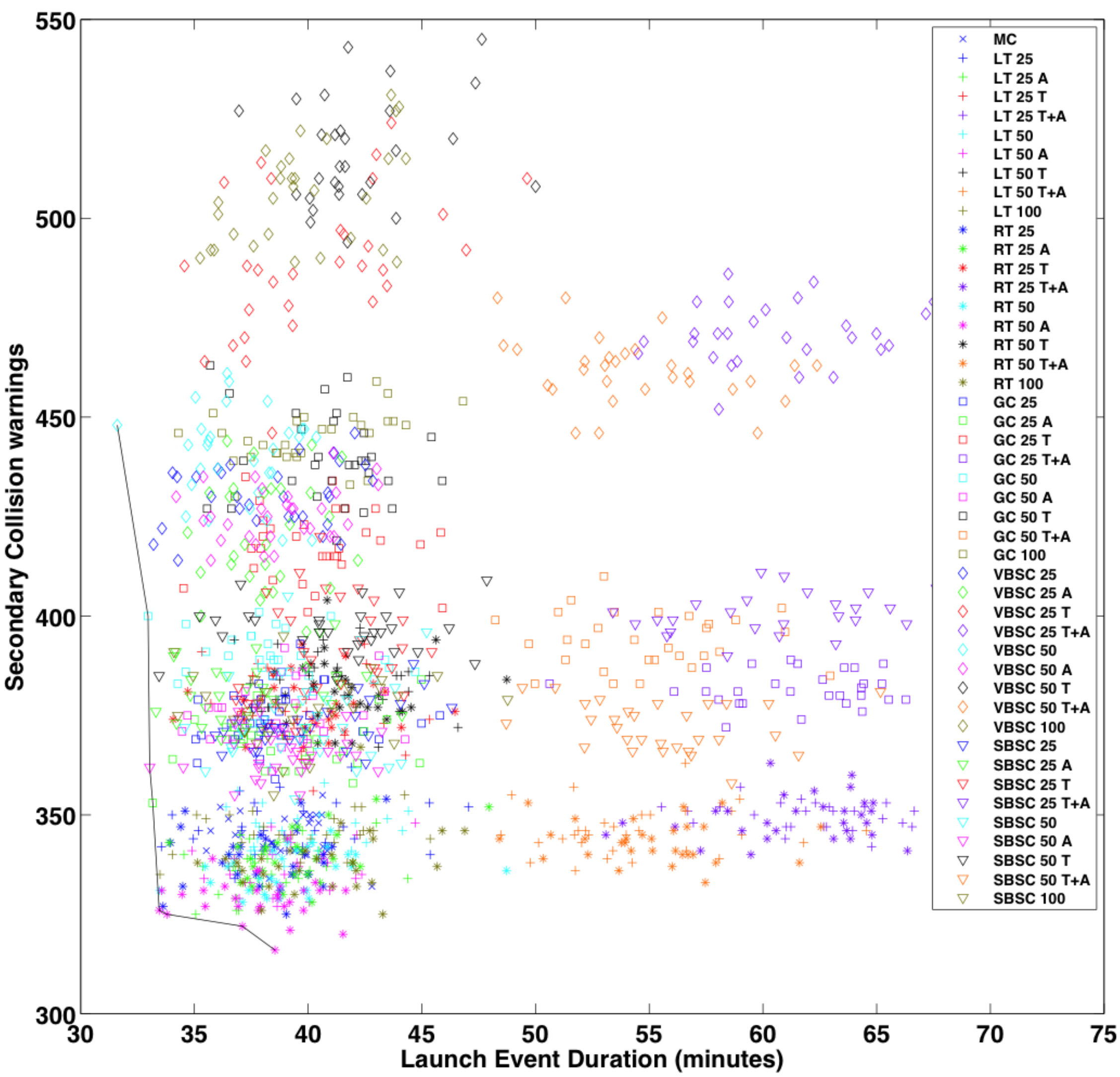


Figure N-51: Pareto frontier plot of Secondary Collision warnings (SC) versus LD for 34 aircraft missions.

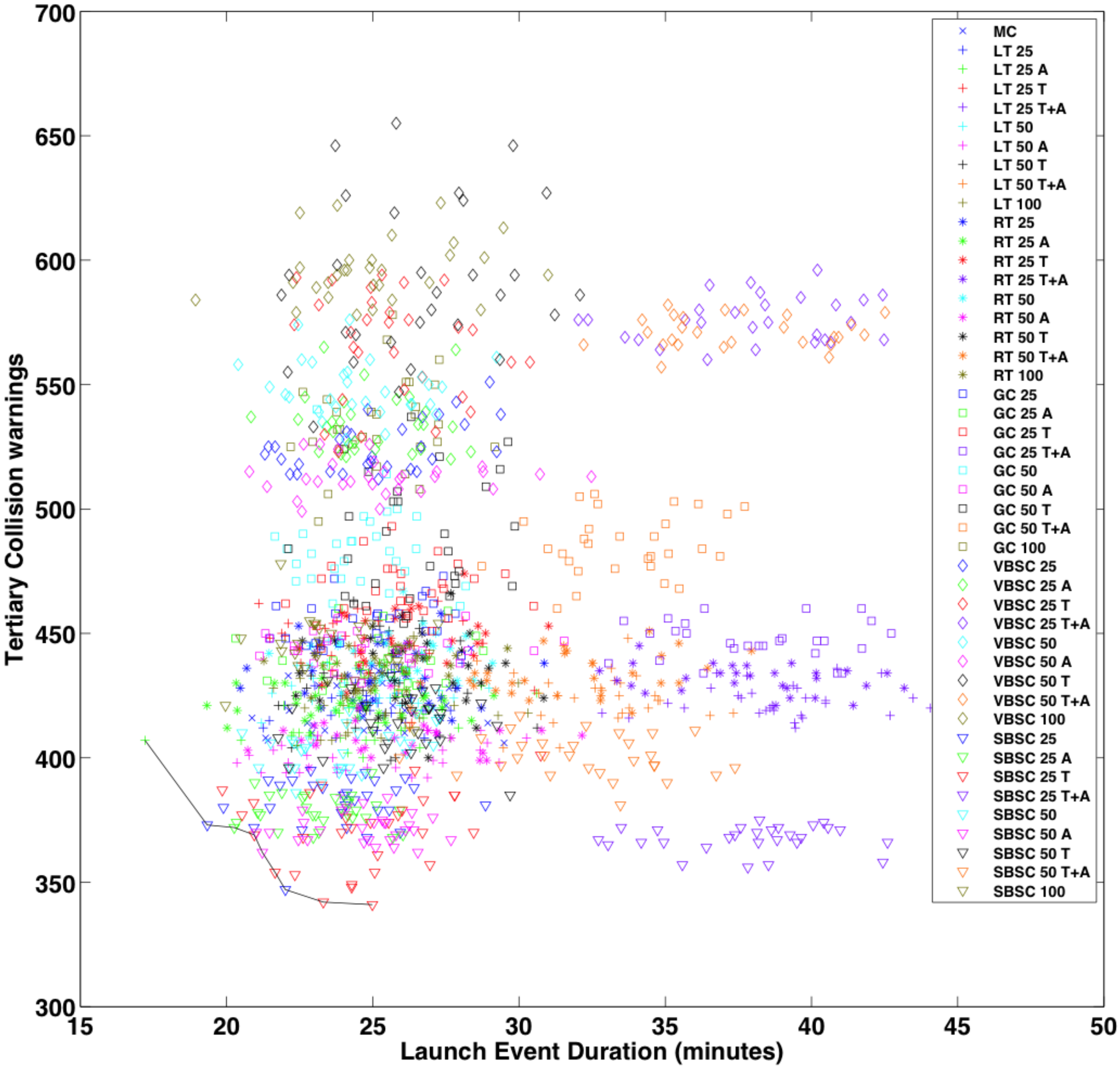


Figure N-52: Pareto frontier plot of Tertiary Collision warnings (TC) versus LD for 22 aircraft missions.

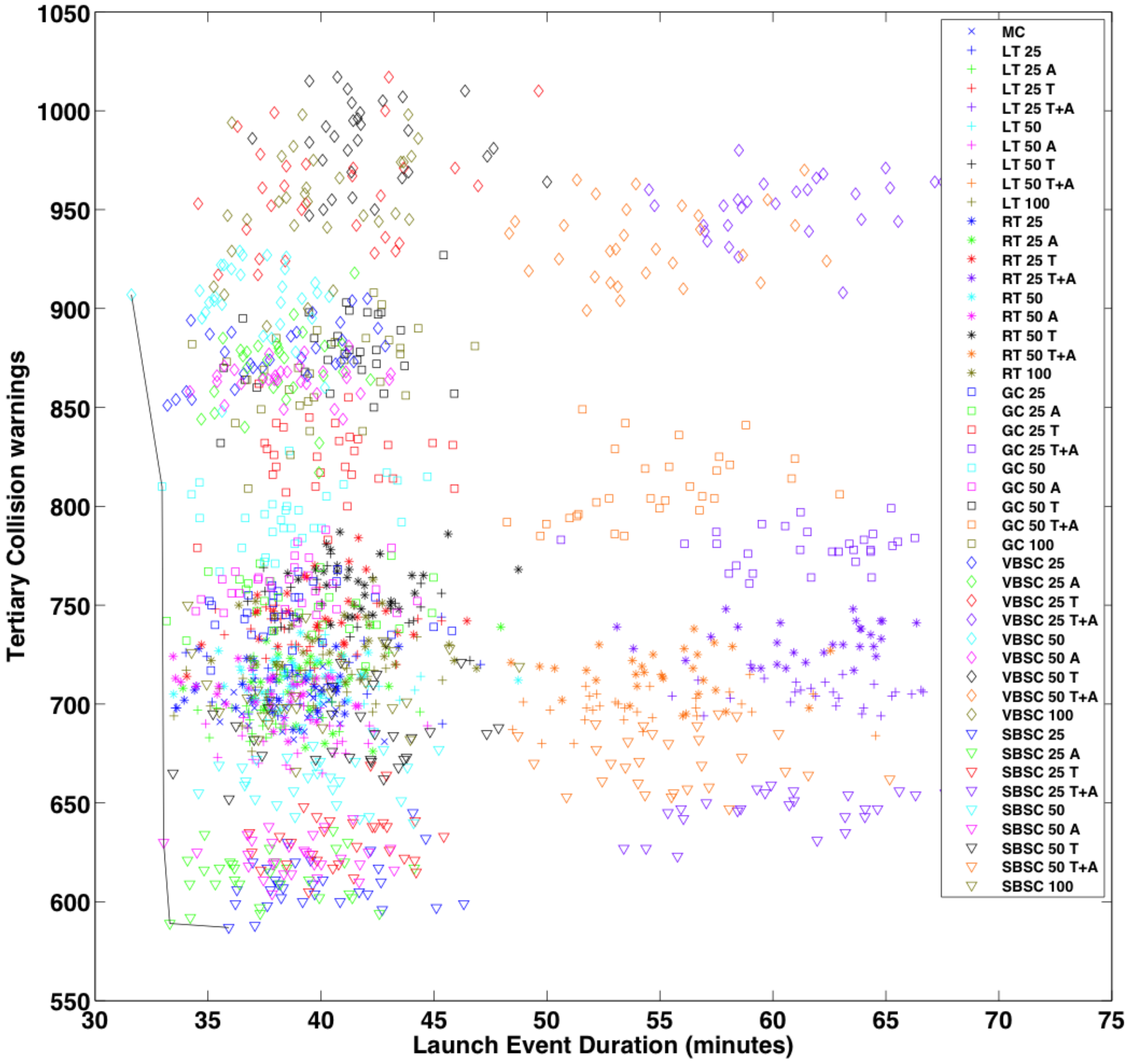


Figure N-53: Pareto frontier plot of Tertiary Collision warnings (TC) versus LD for 34 aircraft missions.

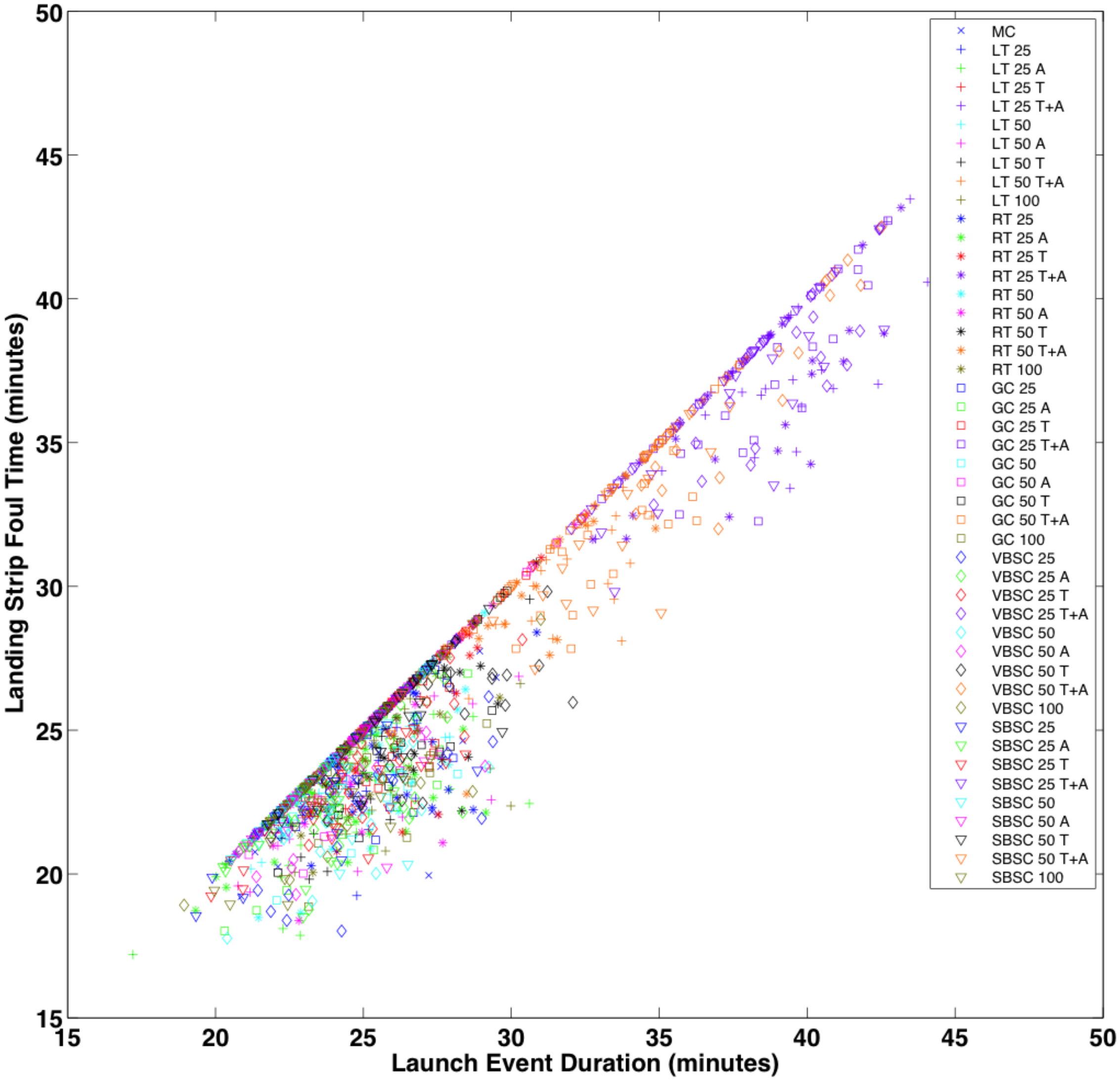


Figure N-54: Pareto frontier plot of Landing Strip Foul Time (LSFT) versus LD for 22 aircraft missions.

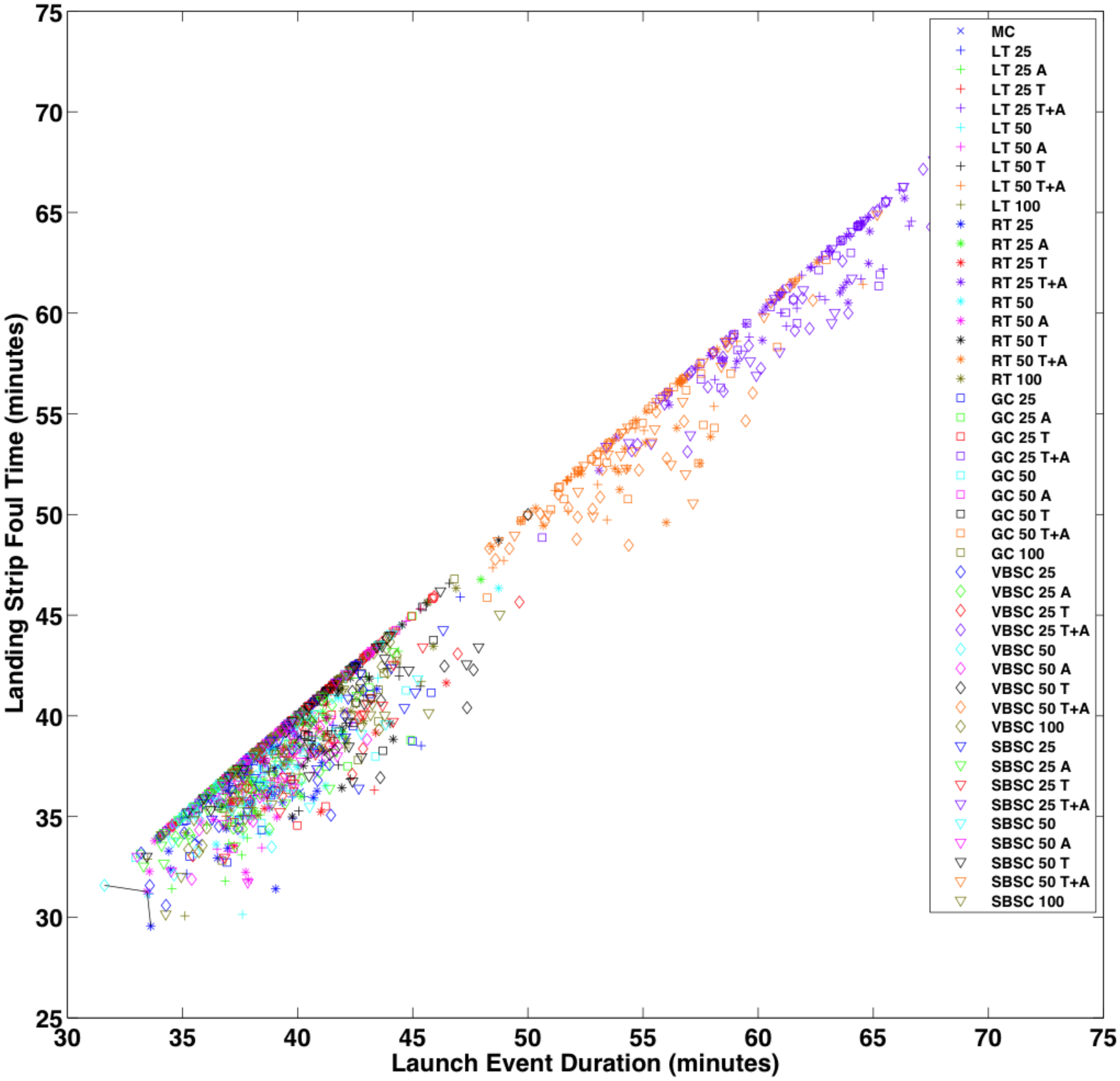


Figure N-55: Pareto frontier plot of Landing Strip Foul Time (LSFT) versus LD for 34 aircraft missions.

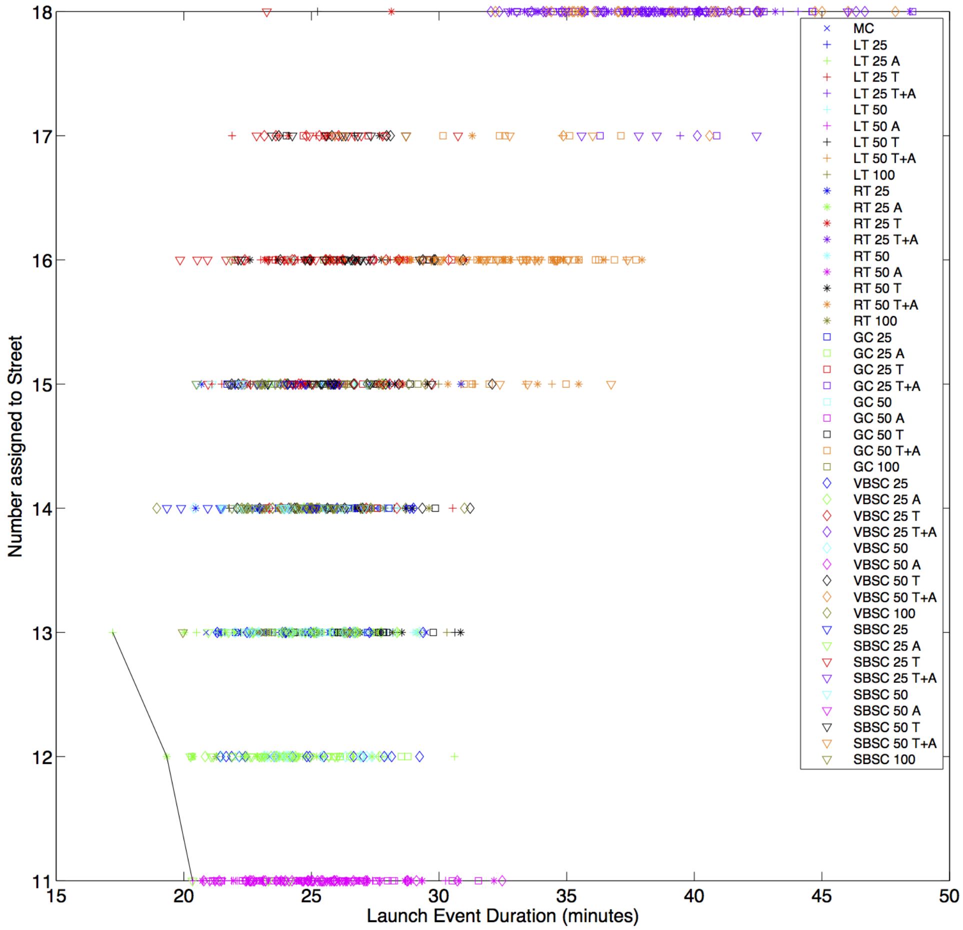


Figure N-56: Pareto frontier plot of Number assigned to Street (NS) versus LD for 22 aircraft missions.

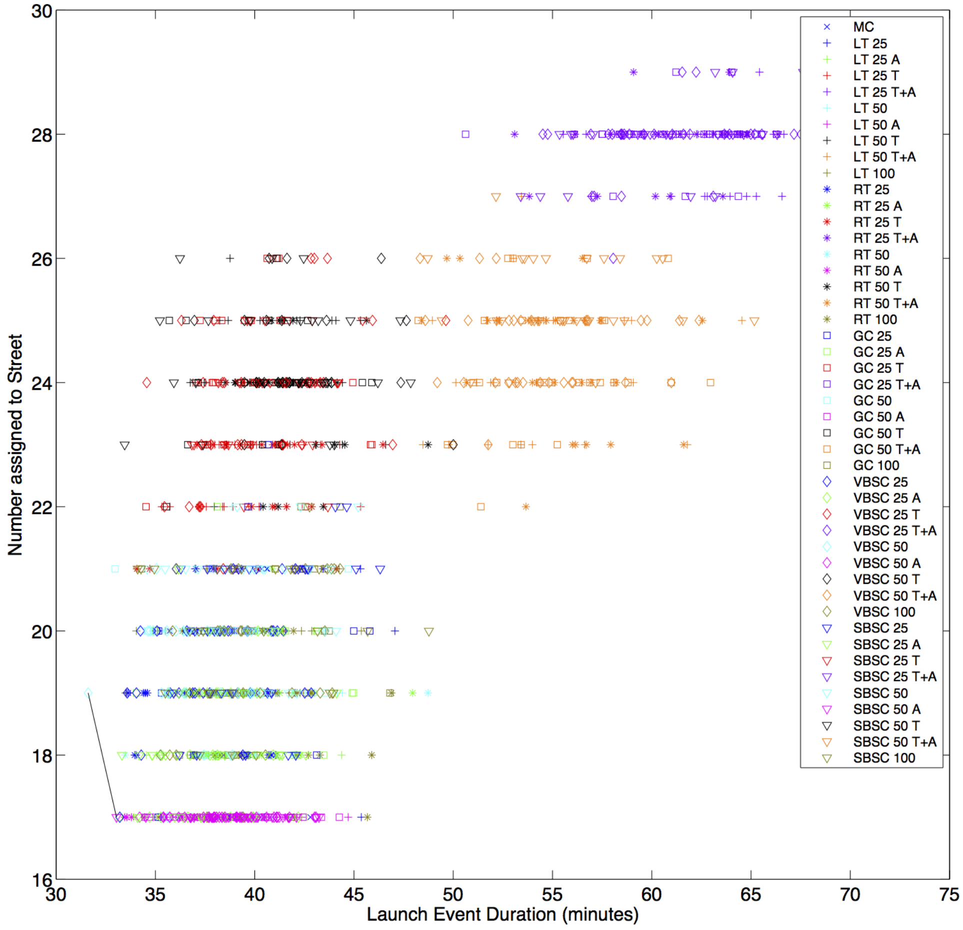


Figure N-57: Pareto frontier plot of Number assigned to Street (NS) versus LD for 34 aircraft missions.

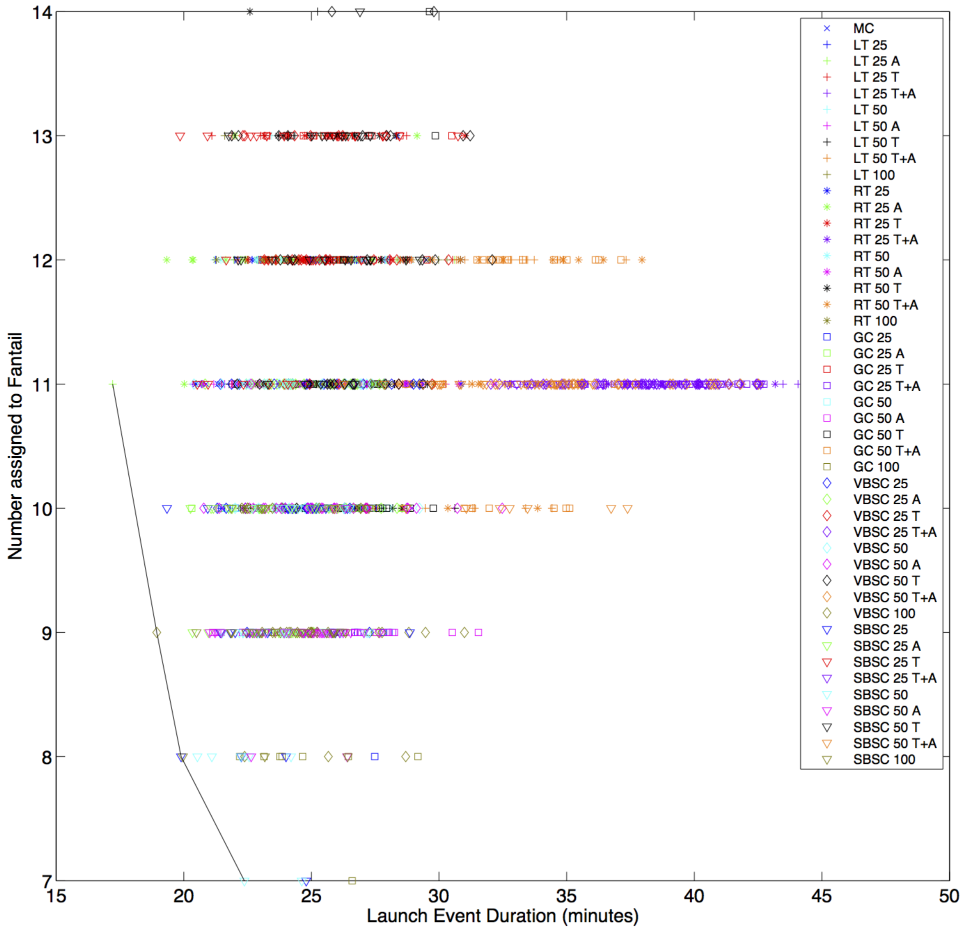


Figure N-58: Pareto frontier plot of Number assigned to Fantail (NF) versus LD for 22 aircraft missions.

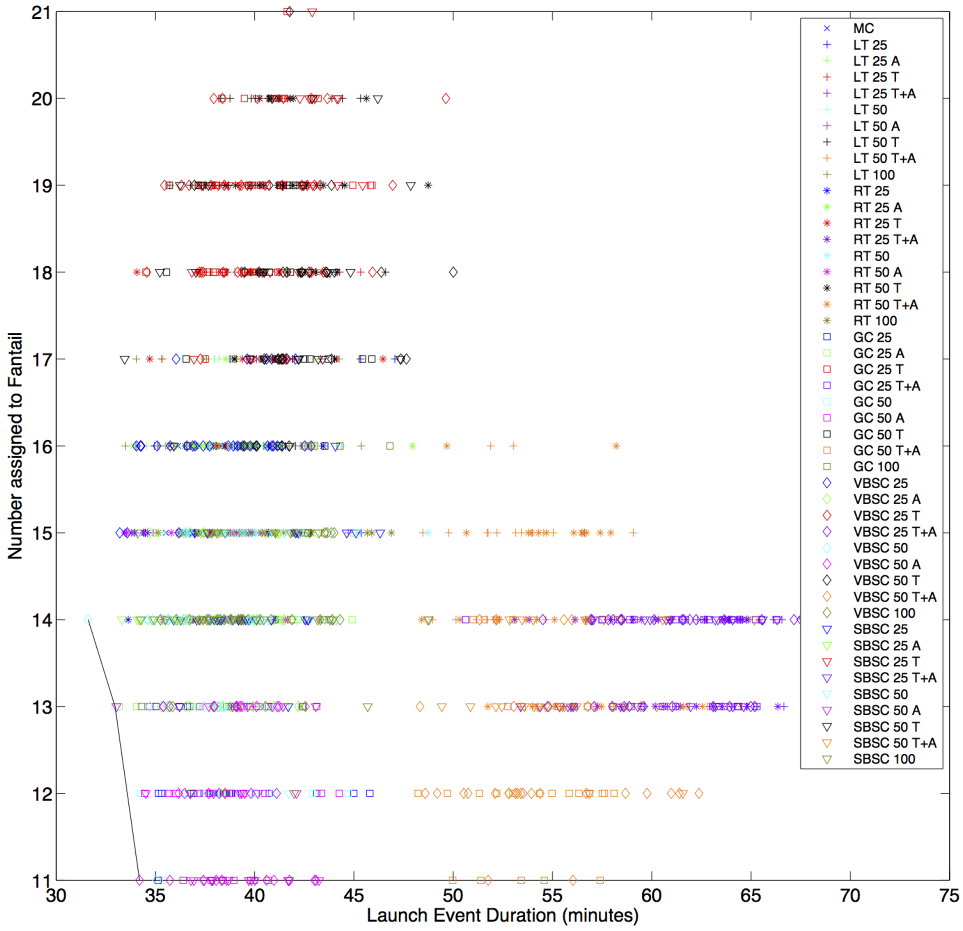


Figure N-59: Pareto frontier plot of Number assigned to Fantail (NF) versus LD for 34 aircraft missions.

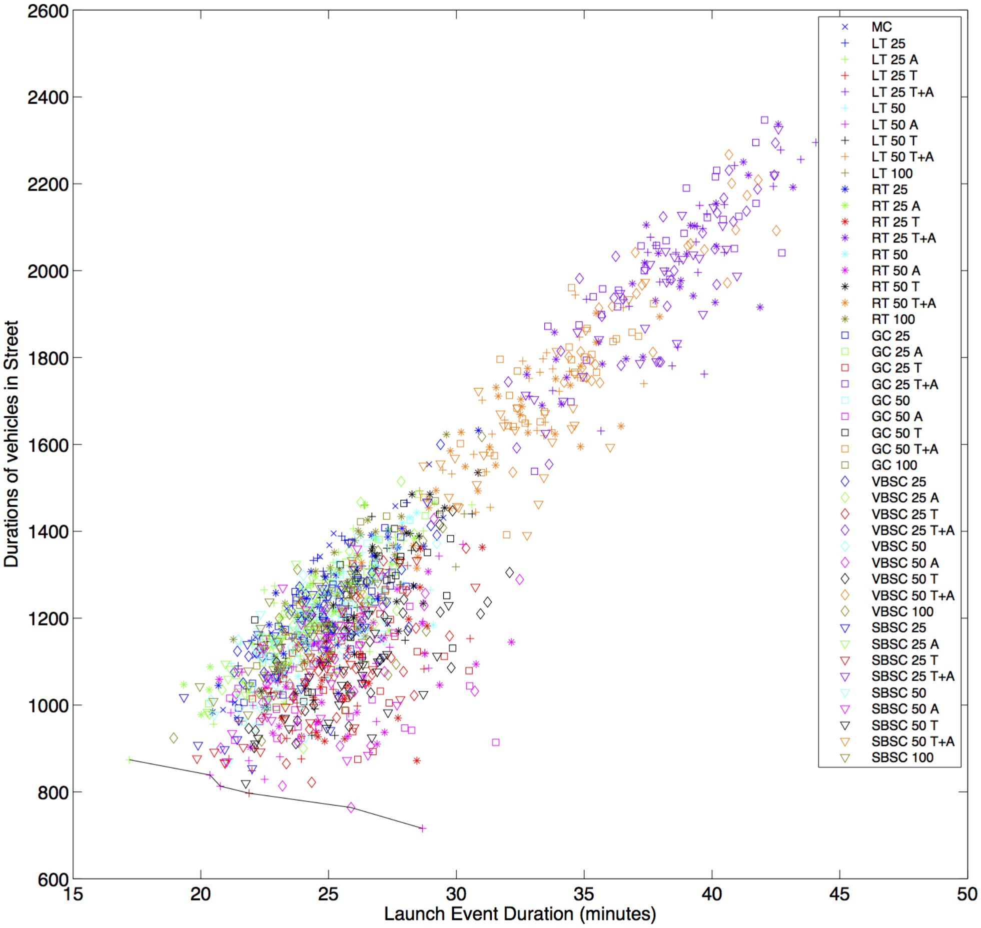


Figure N-60: Pareto frontier plot of Duration of Vehicles in Street (SD) versus LD for 22 aircraft missions.

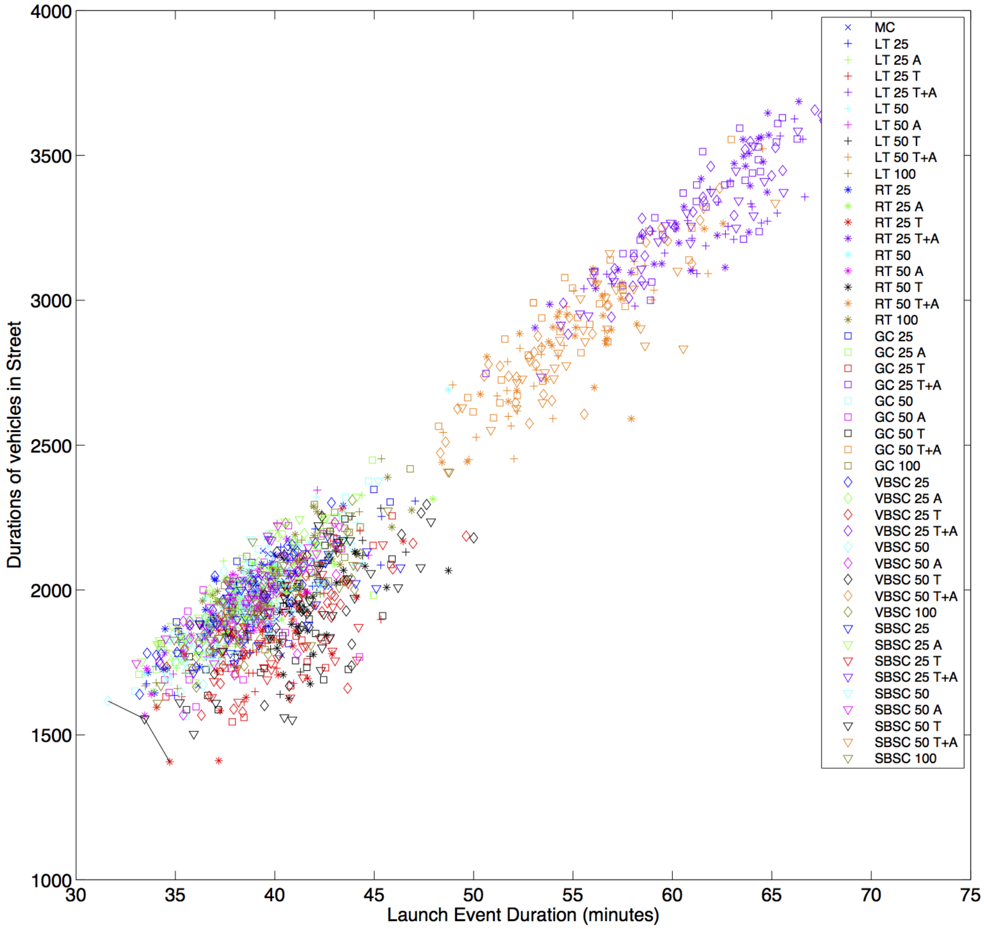


Figure N-61: Pareto frontier plot of Duration of Vehicles in Street (SD) versus LD for 34 aircraft missions.

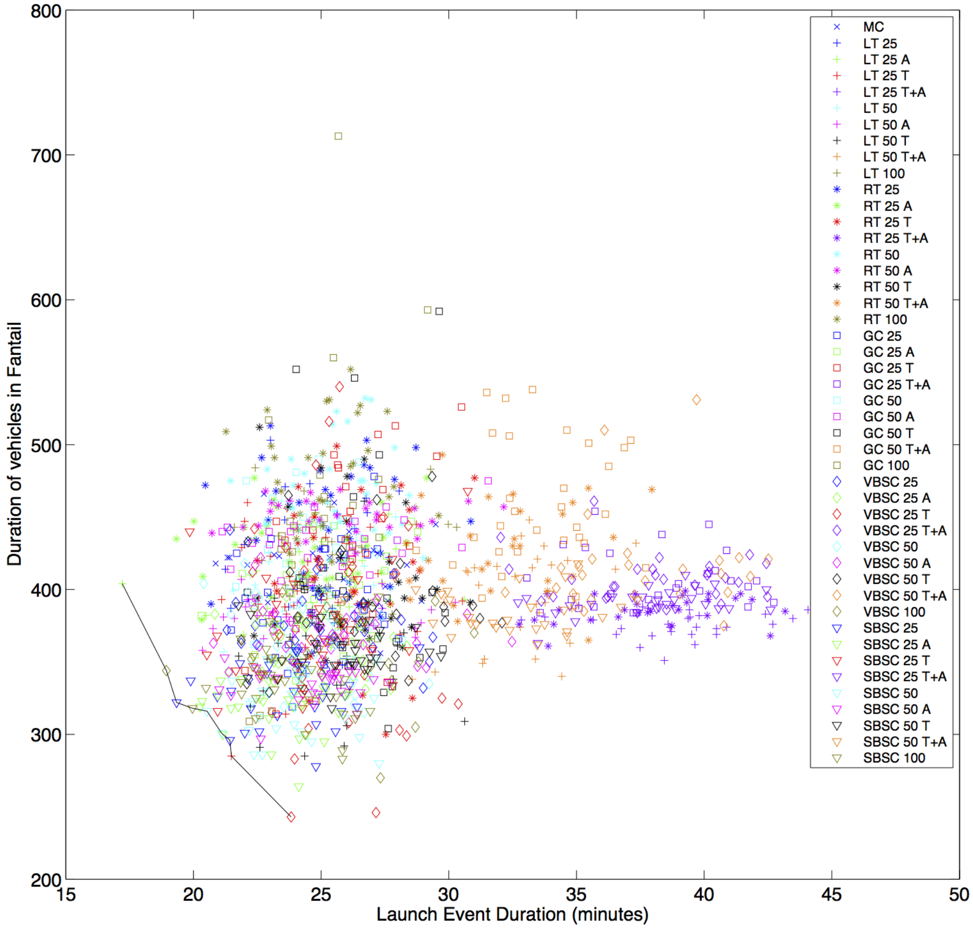


Figure N-62: Pareto frontier plot of Duration of Vehicles in Fantail (FD) versus LD for 22 aircraft missions.

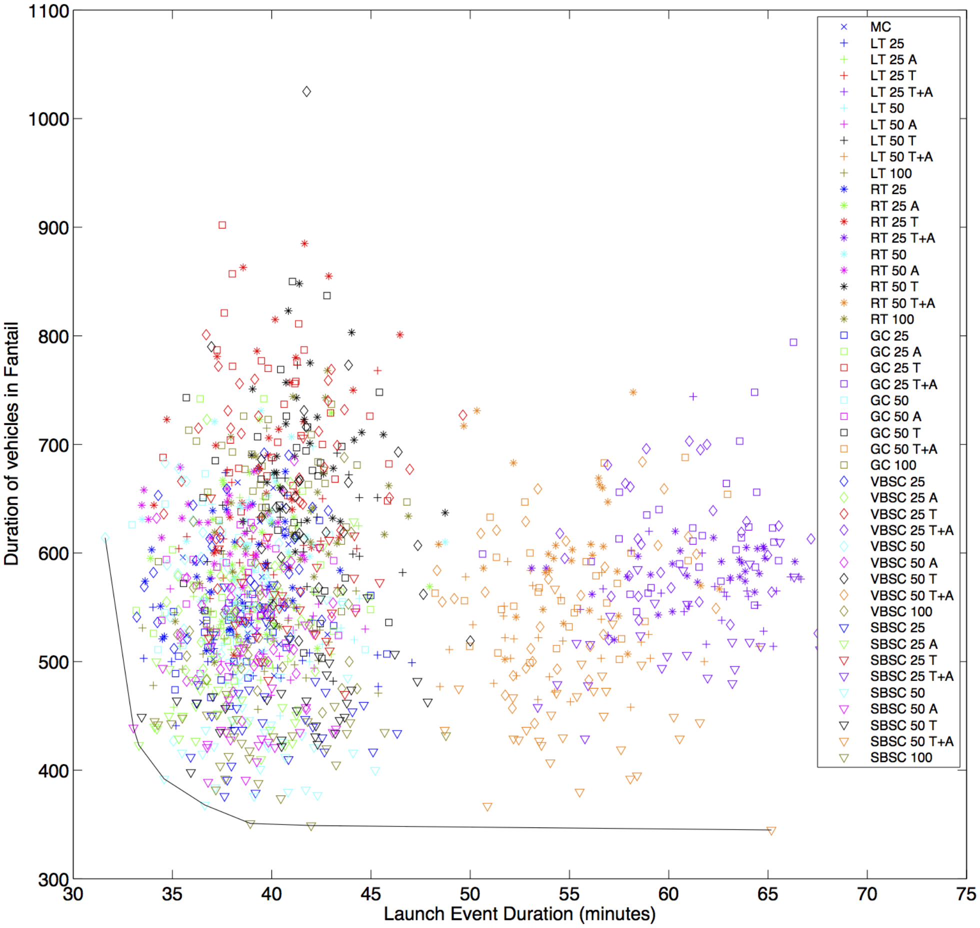
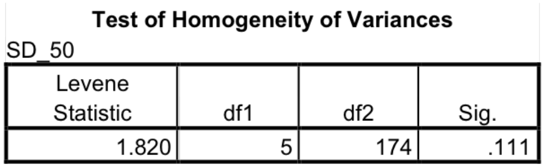


Figure N-63: Pareto frontier plot of Duration of Vehicles in Fantail (FD) versus LD for 34 aircraft missions.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix O

Statistical Tests for Design Exploration

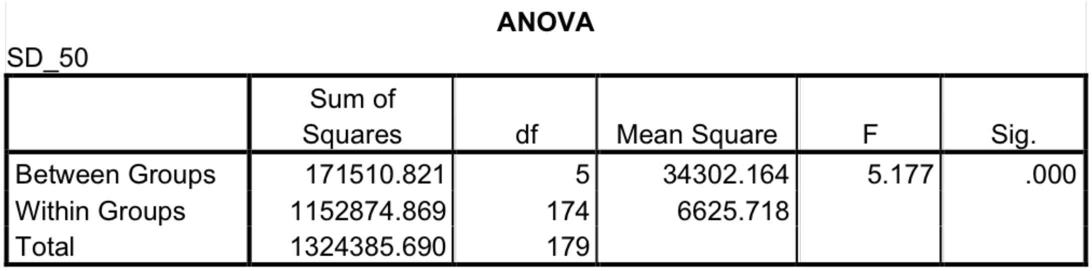


Test of Homogeneity of Variances

SD_50

Levene Statistic	df1	df2	Sig.
1.820	5	174	.111

Figure O-1: SPSS output for Levene test of equal variances for LD values for all control architectures at 22 aircraft under the 50% standard deviation launch preparation time.

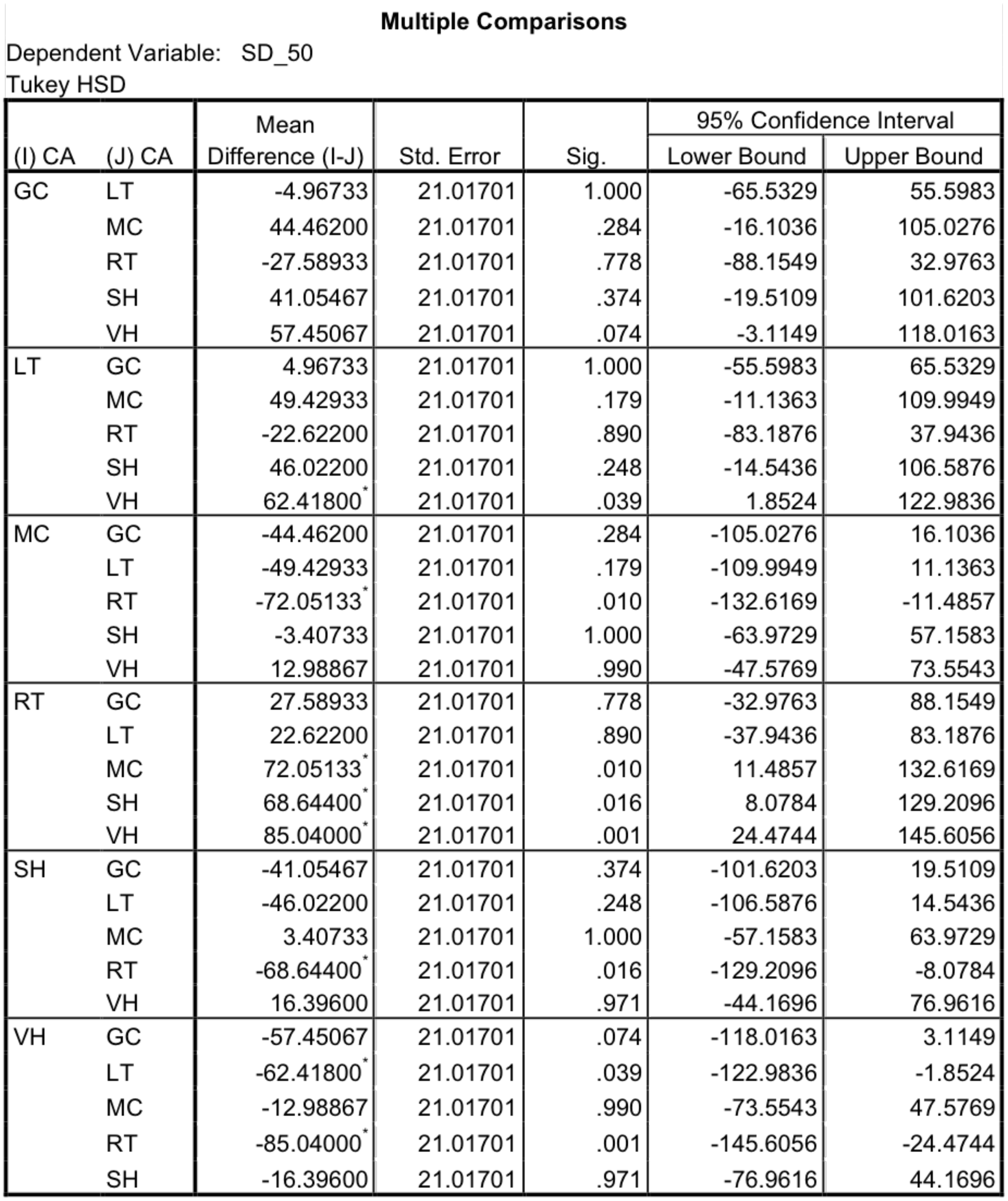


ANOVA

SD_50

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	171510.821	5	34302.164	5.177	.000
Within Groups	1152874.869	174	6625.718		
Total	1324385.690	179			

Figure O-2: SPSS output for an ANOVA of LD values for all control architectures at 22 aircraft under the 50% standard deviation launch preparation time.



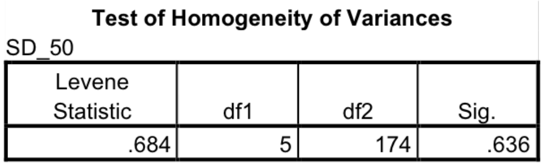
Multiple Comparisons

Dependent Variable: SD_50

Tukey HSD

(I) CA	(J) CA	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
GC	LT	-4.96733	21.01701	1.000	-65.5329	55.5983
	MC	44.46200	21.01701	.284	-16.1036	105.0276
	RT	-27.58933	21.01701	.778	-88.1549	32.9763
	SH	41.05467	21.01701	.374	-19.5109	101.6203
	VH	57.45067	21.01701	.074	-3.1149	118.0163
LT	GC	4.96733	21.01701	1.000	-55.5983	65.5329
	MC	49.42933	21.01701	.179	-11.1363	109.9949
	RT	-22.62200	21.01701	.890	-83.1876	37.9436
	SH	46.02200	21.01701	.248	-14.5436	106.5876
	VH	62.41800*	21.01701	.039	1.8524	122.9836
MC	GC	-44.46200	21.01701	.284	-105.0276	16.1036
	LT	-49.42933	21.01701	.179	-109.9949	11.1363
	RT	-72.05133*	21.01701	.010	-132.6169	-11.4857
	SH	-3.40733	21.01701	1.000	-63.9729	57.1583
	VH	12.98867	21.01701	.990	-47.5769	73.5543
RT	GC	27.58933	21.01701	.778	-32.9763	88.1549
	LT	22.62200	21.01701	.890	-37.9436	83.1876
	MC	72.05133*	21.01701	.010	11.4857	132.6169
	SH	68.64400*	21.01701	.016	8.0784	129.2096
	VH	85.04000*	21.01701	.001	24.4744	145.6056
SH	GC	-41.05467	21.01701	.374	-101.6203	19.5109
	LT	-46.02200	21.01701	.248	-106.5876	14.5436
	MC	3.40733	21.01701	1.000	-57.1583	63.9729
	RT	-68.64400*	21.01701	.016	-129.2096	-8.0784
	VH	16.39600	21.01701	.971	-44.1696	76.9616
VH	GC	-57.45067	21.01701	.074	-118.0163	3.1149
	LT	-62.41800*	21.01701	.039	-122.9836	-1.8524
	MC	-12.98867	21.01701	.990	-73.5543	47.5769
	RT	-85.04000*	21.01701	.001	-145.6056	-24.4744
	SH	-16.39600	21.01701	.971	-76.9616	44.1696

Figure O-3: SPSS output for Tukey HSC test of LD values for all control architectures at 22 aircraft under the 50% standard deviation launch preparation time.

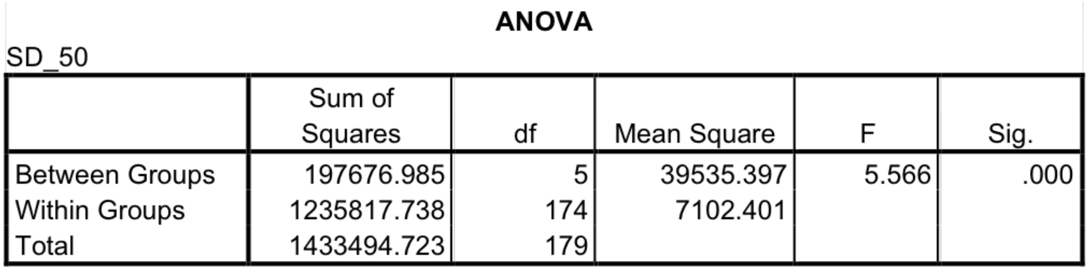


Test of Homogeneity of Variances

SD_50

Levene Statistic	df1	df2	Sig.
.684	5	174	.636

Figure O-4: SPSS output for Levene test of equal variances for LD values for all control architectures at 34 aircraft under the 50% standard deviation launch preparation time.

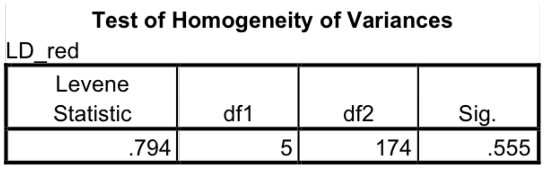


ANOVA

SD_50

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	197676.985	5	39535.397	5.566	.000
Within Groups	1235817.738	174	7102.401		
Total	1433494.723	179			

Figure O-5: SPSS output for an ANOVA of LD values for all control architectures at 34 aircraft under the 50% standard deviation launch preparation time.



Test of Homogeneity of Variances

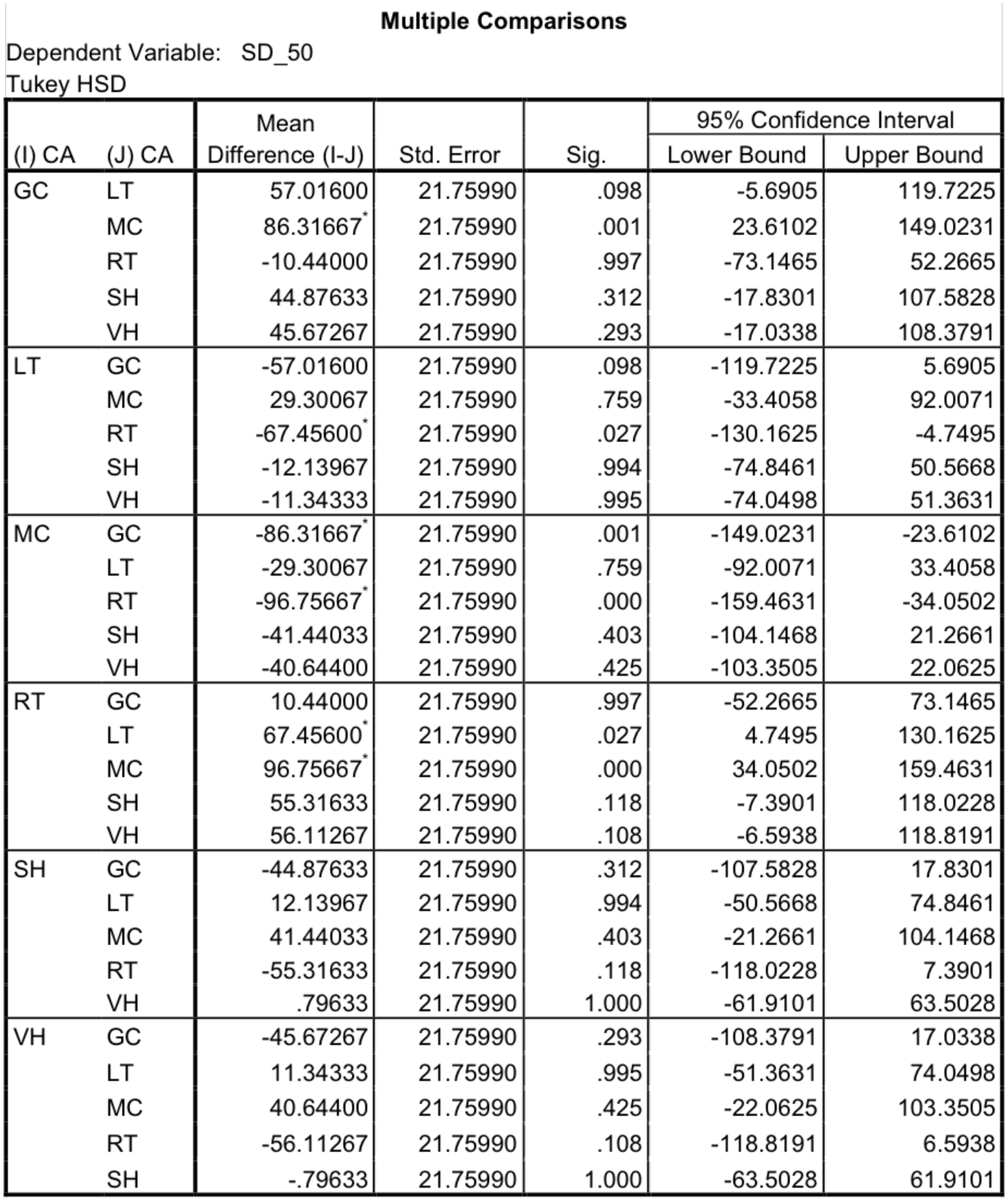
LD_red

Levene Statistic	df1	df2	Sig.
.794	5	174	.555

Figure O-6: SPSS output for Levene test of equal variances for LD values for all control architectures at 22 aircraft under the launch preparation time model with 50% reduction in mean and 50% reduction in standard deviation.

Table O.1: JMP PRO 10 output for a Steel-Dwass non-parametric simultaneous comparisons test of LD values for all control architectures for 34 aircraft under the launch preparation time model with 50% reduction in mean and standard deviation.

RT -50	LT -50	26.2333	4.50925	5.81767	<.0001
RT -50	MC -50	24.4333	4.50925	5.41849	<.0001
VH -50	SH -50	22.5667	4.50925	5.00453	<.0001
VH -50	LT -50	20.7667	4.50925	4.60535	<.0001
VH -50	MC -50	19.4333	4.50925	4.30966	0.0002
MC -50	LT -50	-0.9	4.50925	-0.19959	1
RT -50	GC -50	-0.9667	4.50925	-0.21437	0.9999
SH -50	MC -50	-4.4333	4.50925	-0.98316	0.9235
SH -50	LT -50	-6.5	4.50925	-1.44148	0.7015
VH -50	GC -50	-8.7333	4.509187	-1.93679	0.3795
VH -50	RT -50	-9.9	4.50925	-2.19549	0.2397
MC -50	GC -50	-23.6333	4.50925	-5.24108	<.0001
LT -50	GC -50	-24.9	4.50925	-5.52198	<.0001
SH -50	GC -50	-25.7	4.50925	-5.6994	<.0001
SH -50	RT -50	-26.3667	4.50925	-5.84724	<.0001

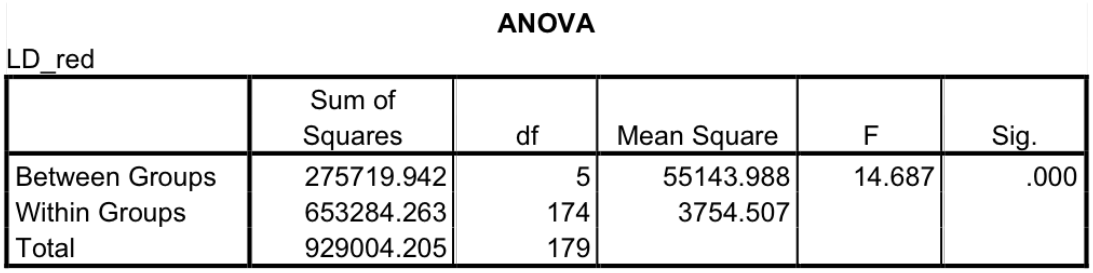


Multiple Comparisons

Dependent Variable: SD_50
Tukey HSD

(I) CA	(J) CA	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
GC	LT	57.01600	21.75990	.098	-5.6905	119.7225
	MC	86.31667*	21.75990	.001	23.6102	149.0231
	RT	-10.44000	21.75990	.997	-73.1465	52.2665
	SH	44.87633	21.75990	.312	-17.8301	107.5828
	VH	45.67267	21.75990	.293	-17.0338	108.3791
LT	GC	-57.01600	21.75990	.098	-119.7225	5.6905
	MC	29.30067	21.75990	.759	-33.4058	92.0071
	RT	-67.45600*	21.75990	.027	-130.1625	-4.7495
	SH	-12.13967	21.75990	.994	-74.8461	50.5668
	VH	-11.34333	21.75990	.995	-74.0498	51.3631
MC	GC	-86.31667*	21.75990	.001	-149.0231	-23.6102
	LT	-29.30067	21.75990	.759	-92.0071	33.4058
	RT	-96.75667*	21.75990	.000	-159.4631	-34.0502
	SH	-41.44033	21.75990	.403	-104.1468	21.2661
	VH	-40.64400	21.75990	.425	-103.3505	22.0625
RT	GC	10.44000	21.75990	.997	-52.2665	73.1465
	LT	67.45600*	21.75990	.027	4.7495	130.1625
	MC	96.75667*	21.75990	.000	34.0502	159.4631
	SH	55.31633	21.75990	.118	-7.3901	118.0228
	VH	56.11267	21.75990	.108	-6.5938	118.8191
SH	GC	-44.87633	21.75990	.312	-107.5828	17.8301
	LT	12.13967	21.75990	.994	-50.5668	74.8461
	MC	41.44033	21.75990	.403	-21.2661	104.1468
	RT	-55.31633	21.75990	.118	-118.0228	7.3901
	VH	.79633	21.75990	1.000	-61.9101	63.5028
VH	GC	-45.67267	21.75990	.293	-108.3791	17.0338
	LT	11.34333	21.75990	.995	-51.3631	74.0498
	MC	40.64400	21.75990	.425	-22.0625	103.3505
	RT	-56.11267	21.75990	.108	-118.8191	6.5938
	SH	-.79633	21.75990	1.000	-63.5028	61.9101

Figure O-7: SPSS output for a Tukey HSD test of LD values for all control architectures at 34 aircraft under the 50% standard deviation launch preparation time.

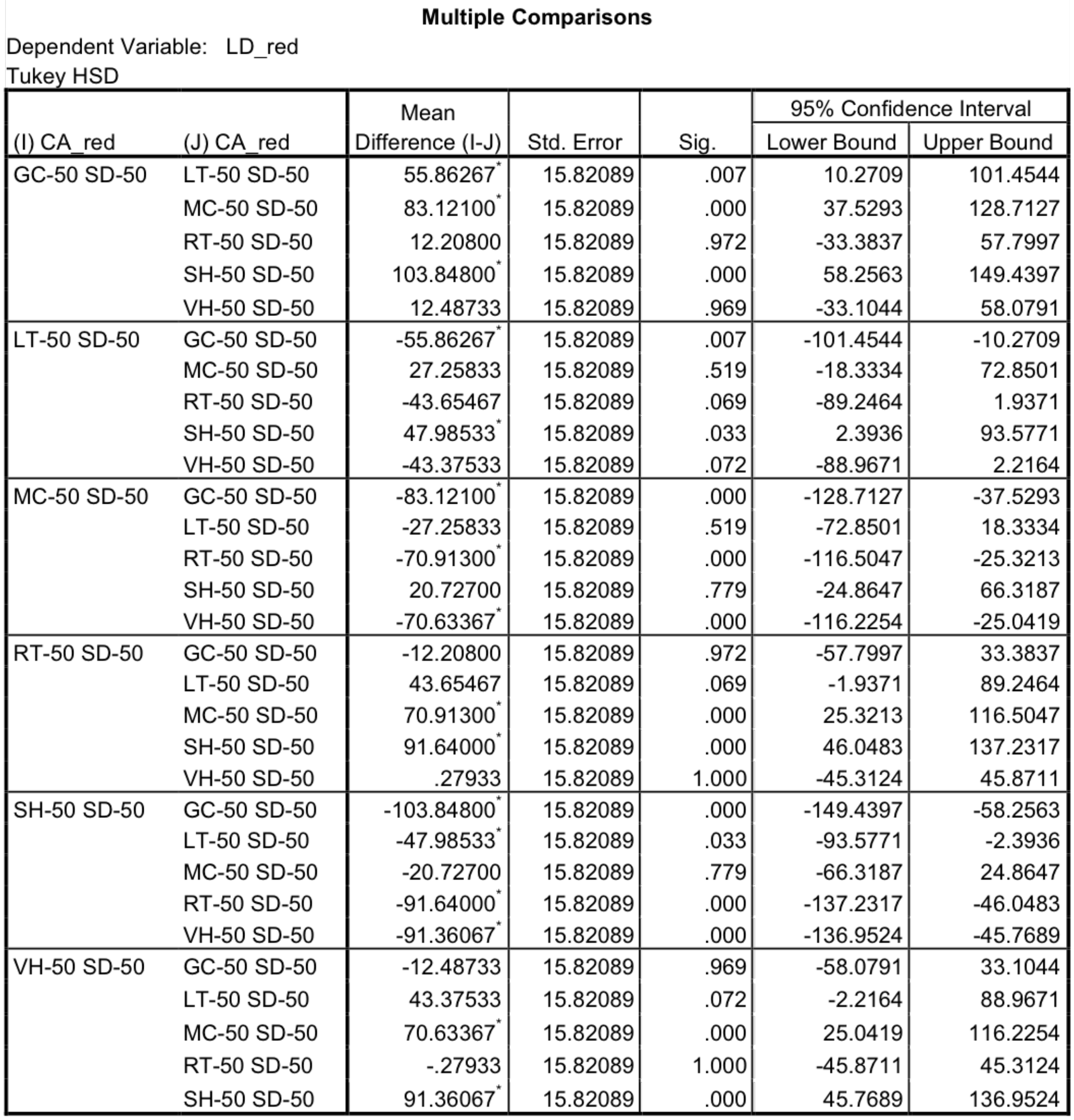


ANOVA

LD_red

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	275719.942	5	55143.988	14.687	.000
Within Groups	653284.263	174	3754.507		
Total	929004.205	179			

Figure O-8: SPSS output for an ANOVA test for LD values for all control architectures at 22 aircraft under the launch preparation time model with 50% reduction in mean and 50% reduction in standard deviation.

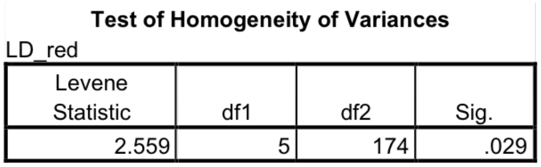


Multiple Comparisons

Dependent Variable: LD_red
Tukey HSD

		Mean			95% Confidence Interval	
(I) CA_red	(J) CA_red	Difference (I-J)	Std. Error	Sig.	Lower Bound	Upper Bound
GC-50 SD-50	LT-50 SD-50	55.86267*	15.82089	.007	10.2709	101.4544
	MC-50 SD-50	83.12100*	15.82089	.000	37.5293	128.7127
	RT-50 SD-50	12.20800	15.82089	.972	-33.3837	57.7997
	SH-50 SD-50	103.84800*	15.82089	.000	58.2563	149.4397
	VH-50 SD-50	12.48733	15.82089	.969	-33.1044	58.0791
LT-50 SD-50	GC-50 SD-50	-55.86267*	15.82089	.007	-101.4544	-10.2709
	MC-50 SD-50	27.25833	15.82089	.519	-18.3334	72.8501
	RT-50 SD-50	-43.65467	15.82089	.069	-89.2464	1.9371
	SH-50 SD-50	47.98533*	15.82089	.033	2.3936	93.5771
	VH-50 SD-50	-43.37533	15.82089	.072	-88.9671	2.2164
MC-50 SD-50	GC-50 SD-50	-83.12100*	15.82089	.000	-128.7127	-37.5293
	LT-50 SD-50	-27.25833	15.82089	.519	-72.8501	18.3334
	RT-50 SD-50	-70.91300*	15.82089	.000	-116.5047	-25.3213
	SH-50 SD-50	20.72700	15.82089	.779	-24.8647	66.3187
	VH-50 SD-50	-70.63367*	15.82089	.000	-116.2254	-25.0419
RT-50 SD-50	GC-50 SD-50	-12.20800	15.82089	.972	-57.7997	33.3837
	LT-50 SD-50	43.65467	15.82089	.069	-1.9371	89.2464
	MC-50 SD-50	70.91300*	15.82089	.000	25.3213	116.5047
	SH-50 SD-50	91.64000*	15.82089	.000	46.0483	137.2317
	VH-50 SD-50	.27933	15.82089	1.000	-45.3124	45.8711
SH-50 SD-50	GC-50 SD-50	-103.84800*	15.82089	.000	-149.4397	-58.2563
	LT-50 SD-50	-47.98533*	15.82089	.033	-93.5771	-2.3936
	MC-50 SD-50	-20.72700	15.82089	.779	-66.3187	24.8647
	RT-50 SD-50	-91.64000*	15.82089	.000	-137.2317	-46.0483
	VH-50 SD-50	-91.36067*	15.82089	.000	-136.9524	-45.7689
VH-50 SD-50	GC-50 SD-50	-12.48733	15.82089	.969	-58.0791	33.1044
	LT-50 SD-50	43.37533	15.82089	.072	-2.2164	88.9671
	MC-50 SD-50	70.63367*	15.82089	.000	25.0419	116.2254
	RT-50 SD-50	-.27933	15.82089	1.000	-45.8711	45.3124
	SH-50 SD-50	91.36067*	15.82089	.000	45.7689	136.9524

Figure O-9: SPSS output for Tukey HSD test for LD values for all control architectures at 22 aircraft under the launch preparation time model with 50% reduction in mean and 50% reduction in standard deviation.

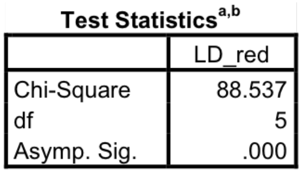


Test of Homogeneity of Variances

LD_red

Levene Statistic	df1	df2	Sig.
2.559	5	174	.029

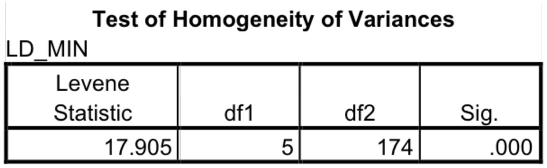
Figure O-10: SPSS output for Levene test of equal variances for LD values for all control architectures for 34 aircraft under the launch preparation time model with 50% reduction in mean and standard deviation.



Test Statistics[a,b]

	LD_red
Chi-Square	88.537
df	5
Asymp. Sig.	.000

Figure O-11: SPSS output for a Kruskal-Wallis test of LD values for all control architectures for 34 aircraft under the launch preparation time model with 50% reduction in mean and standard deviation.

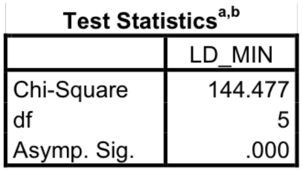


Test of Homogeneity of Variances

LD_MIN

Levene Statistic	df1	df2	Sig.
17.905	5	174	.000

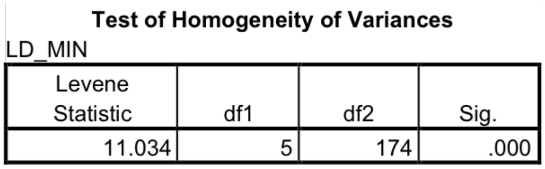
Figure O-12: SPSS output for Levene test of equal variances for LD values for all control architectures at 22 aircraft under the "automated" launch preparation time.



Test Statistics[a,b]

	LD_MIN
Chi-Square	144.477
df	5
Asymp. Sig.	.000

Figure O-13: SPSS output for Kruskal-Wallis nonparametric analysis of variance for LD values for all control architectures at 22 aircraft under the "automated" launch preparation time.



Test of Homogeneity of Variances

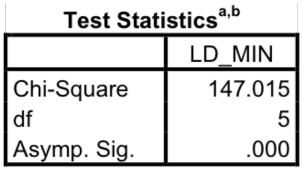
LD_MIN

Levene Statistic	df1	df2	Sig.
11.034	5	174	.000

Figure O-14: SPSS output for Levene test of equal variances for LD values for all control architectures at 34 aircraft under the "automated" launch preparation time.

Table O.2: JMP Pro 10 output for Steel-Dwass nonparametric simultaneous comparisons test of LD values for all control architectures at 22 aircraft under the "automated" launch preparation time.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
RT	MC	29.9667	4.50925	6.6456	< .0001
VH	MC	29.9667	4.50925	6.6456	< .0001
VH	LT	29.7667	4.50925	6.60125	< .0001
RT	LT	29.5667	4.50925	6.55689	< .0001
VH	SH	27.3	4.50925	6.05422	< .0001
SH	MC	27.1667	4.50925	6.02465	< .0001
SH	LT	24.1667	4.50925	5.35935	< .0001
VH	RT	22.7667	4.50925	5.04888	< .0001
VH	GC	4.0333	4.509187	0.89447	0.948
MC	LT	-6.9	4.50925	-1.53019	0.6447
SH	RT	-15.5	4.50925	-3.43738	0.0077
RT	GC	-22.9	4.509187	-5.07852	< .0001
SH	GC	-27.5	4.509187	-6.09866	< .0001
LT	GC	-29.9	4.509187	-6.63091	< .0001
MC	GC	-29.9667	4.509187	-6.64569	< .0001



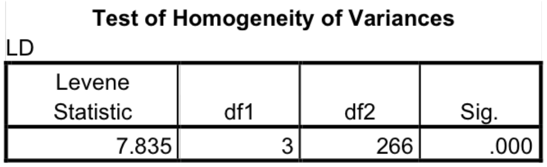
Test Statistics[a,b]

	LD_MIN
Chi-Square	147.015
df	5
Asymp. Sig.	.000

Figure O-15: SPSS output for Kruskal-Wallis nonparametric analysis of variance for LD values for all control architectures at 34 aircraft under the "automated" launch preparation time.

Table O.3: JMP Pro 10 output for Stell-Dwass nonparametric simultaneous comparisons test of LD values for all control architectures at 34 aircraft under the "automated" launch preparation time.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
RT	LT	29.9667	4.50925	6.6456	< .0001
RT	MC	29.9667	4.50925	6.6456	< .0001
VH	SH	29.3	4.50925	6.49775	< .0001
RT	GC	18.8333	4.50925	4.1766	0.0004
VH	MC	0.9	4.50925	0.19959	1
VH	LT	-2.4333	4.50925	-0.53963	0.9945
MC	LT	-3.3	4.50925	-0.73183	0.9781
SH	MC	-27.0333	4.50925	-5.99508	< .0001
SH	LT	-28.9	4.50925	-6.40905	< .0001
LT	GC	-29.3667	4.50925	-6.51254	< .0001
MC	GC	-29.3667	4.50925	-6.51254	< .0001
VH	GC	-29.3667	4.50925	-6.51254	< .0001
SH	GC	-29.9667	4.50925	-6.6456	< .0001
SH	RT	-29.9667	4.50925	-6.6456	< .0001
VH	RT	-29.9667	4.50925	-6.6456	< .0001

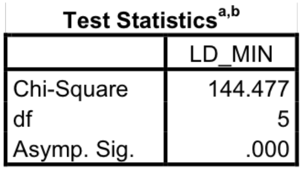


Test of Homogeneity of Variances

LD

Levene Statistic	df1	df2	Sig.
7.835	3	266	.000

Figure O-16: SPSS output for Levene test of equal variances for LD values for the GC architecture across all safety protocols at 22 aircraft under the “automated” launch preparation time.



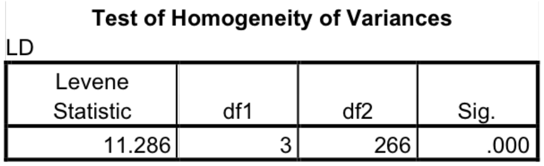
Test Statistics[a,b]

	LD_MIN
Chi-Square	144.477
df	5
Asymp. Sig.	.000

Figure O-17: SPSS output for Kruskal-Wallis non-parametric analysis of variance for LD values for the GC architecture across all safety protocols at 22 aircraft under the “automated” launch preparation time.

Table O.4: SPSS output for Steel-Dwass non-parametric simultaneous comparisons tests for LD values for the GC architecture across all safety protocols at 22 aircraft under the “automated” launch preparation time.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
T+A	DS	73.45833	7.240888	10.14493	< .0001
T	DS	68.43056	7.240888	9.45057	< .0001
T+A	A	59.98333	6.350853	9.44493	< .0001
T	A	59.71667	6.350853	9.40294	< .0001
DS	A	27.76389	7.240888	3.83432	0.0007
T+A	T	21.15	6.350853	3.33026	0.0048

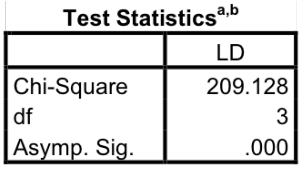


Test of Homogeneity of Variances

LD

Levene Statistic	df1	df2	Sig.
11.286	3	266	.000

Figure O-18: SPSS output for Levene test of equal variances for LD values for the GC architecture across all safety protocols at 34 aircraft under the “automated” launch preparation time.



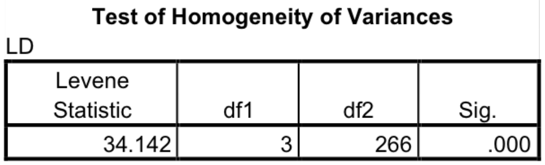
Test Statistics[a,b]

	LD
Chi-Square	209.128
df	3
Asymp. Sig.	.000

Figure O-19: SPSS output for Kruskal-Wallis non-parametric analysis of variance for LD values for the GC architecture across all safety protocols at 34 aircraft under the “automated” launch preparation time.

Table O.5: SPSS output for Steel-Dwass non-parametric simultaneous comparisons tests for LD values for the GC architecture across all safety protocols at 34 aircraft under the “automated” launch preparation time.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
T+A	DS	74.98611	7.240895	10.35592	< .0001
T	DS	74.70833	7.240895	10.31756	< .0001
T+A	A	59.98333	6.350853	9.44493	< .0001
T	A	59.91667	6.350853	9.43443	< .0001
T+A	T	41.01667	6.350853	6.45845	< .0001
DS	A	18.45833	7.240895	2.54918	0.0527

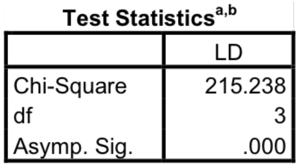


Test of Homogeneity of Variances

LD

Levene Statistic	df1	df2	Sig.
34.142	3	266	.000

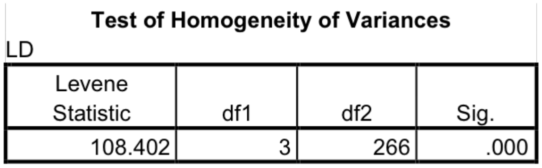
Figure O-20: SPSS output for Levene test of equal variances for LD values for the SBSC architecture across all safety protocols at 22 aircraft under the “automated” launch preparation time.



Test Statistics[a,b]

	LD
Chi-Square	215.238
df	3
Asymp. Sig.	.000

Figure O-21: SPSS output for Kruskal-Wallis non-parametric analysis of variance for LD values for the SBSC architecture across all safety protocols at 22 aircraft under the “automated” launch preparation time.



Test of Homogeneity of Variances

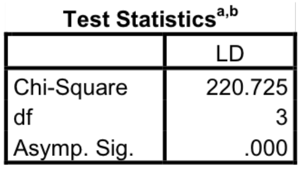
LD

Levene Statistic	df1	df2	Sig.
108.402	3	266	.000

Figure O-22: SPSS output for Levene test of equal variances for LD values for the SBSC architecture across all safety protocols at 34 aircraft under the “automated” launch preparation time.

Table O.6: SPSS output for Steel-Dwass non-parametric simultaneous comparisons tests for LD values for the SBSC architecture across all safety protocols at 22 aircraft under the "automated" launch preparation time.

Level	**- Level**	**Score Mean Difference**	**Std Err Dif**	**Z**	**p-Value**
T	DS	74.98611	7.240895	10.35592	< .0001
T+A	DS	74.98611	7.240895	10.35592	< .0001
T	A	59.98333	6.350853	9.44493	< .0001
T+A	A	59.98333	6.350853	9.44493	< .0001
T+A	T	49.25	6.350853	7.75486	< .0001
DS	A	26.125	7.240895	3.60798	0.0018



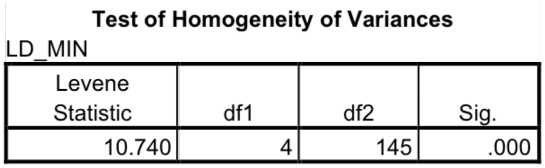
Test Statistics[a,b]

	LD
Chi-Square	220.725
df	3
Asymp. Sig.	.000

Figure O-23: SPSS output for Kruskal-Wallis non-parametric analysis of variance for LD values for the SBSC architecture across all safety protocols at 34 aircraft under the "automated" launch preparation time.

Table O.7: SPSS output for Steel-Dwass non-parametric simultaneous comparisons tests for LD values for the SBSC architecture across all safety protocols at 34 aircraft under the "automated" launch preparation time.

Level	**- Level**	**Score Mean Difference**	**Std Err Dif**	**Z**	**p-Value**
T	DS	74.9861	7.240895	10.3559	< .0001
T+A	DS	74.9861	7.240895	10.3559	< .0001
T	A	59.9833	6.350853	9.4449	< .0001
T+A	A	59.9833	6.350853	9.4449	< .0001
T+A	T	16.45	6.350853	2.5902	0.0473
DS	A	-58.375	7.240895	-8.0618	< .0001

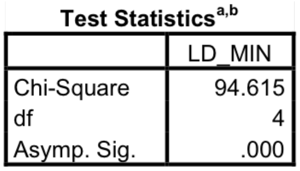


Test of Homogeneity of Variances

LD_MIN

Levene Statistic	df1	df2	Sig.
10.740	4	145	.000

Figure O-24: SPSS output for Levene test of equal variances for LD values for the GC architecture across different parameter settings at 22 aircraft under the "automated" launch preparation time.



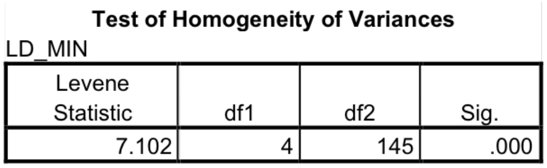
Test Statistics[a,b]

	LD_MIN
Chi-Square	94.615
df	4
Asymp. Sig.	.000

Figure O-25: SPSS output for Kruskal-Wallis non-parametric analysis of variance for LD values for the GC architecture across different parameter settings at 22 aircraft under the "automated" launch preparation time.

Table O.8: JMP PRO 10 output for Steel-Dwass non-parametric simultaneous comparisons test of LD values for the GC architecture across different parameter settings at 22 aircraft under the "automated" launch preparation time.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
LowLatency	LowFailure	-0.3	4.50925	-0.06653	1
Manual Control	LowLat+Fail	-13.0667	4.509187	-2.89779	0.0308
LowLat+Fail	LowLatency	-18.6333	4.50925	-4.13225	0.0003
LowLatency	GC Base	-19.1667	4.509187	-4.25058	0.0002
LowFailure	GC Base	-19.6333	4.509187	-4.35407	0.0001
LowLat+Fail	LowFailure	-20.4333	4.50925	-4.53143	< .0001
Manual Control	LowLatency	-27.3	4.50925	-6.05422	< .0001
Manual Control	LowFailure	-28.2333	4.50925	-6.2612	< .0001
LowLat+Fail	GC Base	-28.6333	4.509187	-6.35	< .0001
Manual Control	GC Base	-29.9667	4.509187	-6.64569	< .0001

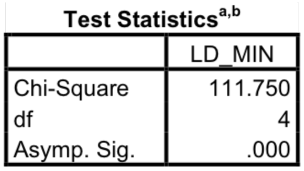


Test of Homogeneity of Variances

LD_MIN

Levene Statistic	df1	df2	Sig.
7.102	4	145	.000

Figure O-26: SPSS output for Levene test of equal variances for LD values for the GC architecture across different parameter settings at 34 aircraft under the "automated" launch preparation time.



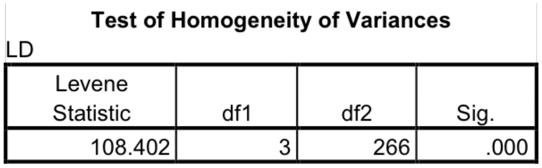
Test Statistics[a,b]

	LD_MIN
Chi-Square	111.750
df	4
Asymp. Sig.	.000

Figure O-27: SPSS output for Kruskal-Wallis non-parametric analysis of variance for LD values for the GC architecture across different parameter settings at 34 aircraft under the "automated" launch preparation time.

Table O.9: JMP PRO 10 output for Steel-Dwass non-parametric simultaneous comparisons test of LD values for the GC architecture across different parameter settings at 34 aircraft under the "automated" launch preparation time.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
LowLatency	LowFailure	15.4333	4.50925	3.42259	0.0056
Manual Control	LowLat+Fail	2.0333	4.50925	0.45092	0.9915
LowLatency	GC Base	-12.6333	4.50925	-2.80165	0.0407
LowFailure	GC Base	-25.1667	4.50925	-5.58112	< .0001
Manual Control	LowFailure	-25.7	4.50925	-5.6994	< .0001
LowLat+Fail	LowFailure	-28.4333	4.50925	-6.30556	< .0001
Manual Control	LowLatency	-28.5667	4.50925	-6.33513	< .0001
Manual Control	GC Base	-29.3667	4.50925	-6.51254	< .0001
LowLat+Fail	LowLatency	-29.7	4.50925	-6.58646	< .0001
LowLat+Fail	GC Base	-29.7667	4.50925	-6.60125	< .0001

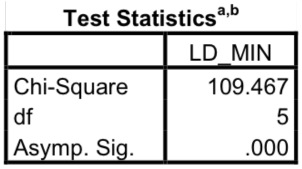


Test of Homogeneity of Variances

LD

Levene Statistic	df1	df2	Sig.
108.402	3	266	.000

Figure O-28: SPSS output for Levene test of equal variances for LD values for the GC architecture across different parameter settings at 22 aircraft under the “automated” launch preparation time, including the Low Penalty case.



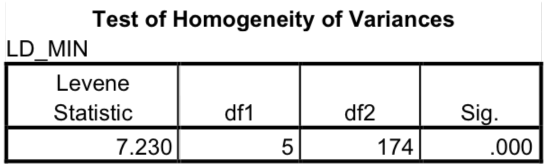
Test Statistics[a,b]

	LD_MIN
Chi-Square	109.467
df	5
Asymp. Sig.	.000

Figure O-29: SPSS output for Kruskal-Wallis non-parametric analysis of variance for LD values for the GC architecture across different parameter settings at 22 aircraft under the “automated” launch preparation time, including the Low Penalty case.

Table O.10: JMP PRO 10 output for Steel-Dwass non-parametric simultaneous comparisons test of LD values for the GC architecture across different parameter settings at 22 aircraft under the “automated” launch preparation time, including the Low Penalty case.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
LowLatency	LowFailure	-0.3	4.50925	-0.06653	1
Manual Control	LowPen	-2.5667	4.508247	-0.56933	0.993
LowPen	LowLat+Fail	-5.9667	4.508811	-1.32333	0.7722
Manual Control	LowLat+Fail	-13	4.507746	-2.88392	0.0454
LowLat+Fail	LowLatency	-18.6667	4.508999	-4.13987	0.0005
LowLatency	GC Base	-19.2333	4.508999	-4.26554	0.0003
LowFailure	GC Base	-19.6667	4.509062	-4.36159	0.0002
LowLat+Fail	LowFailure	-20.4667	4.509124	-4.53894	< .0001
LowPen	LowLatency	-21.5667	4.509124	-4.78289	< .0001
LowPen	LowFailure	-22.1	4.509124	-4.90117	< .0001
Manual Control	LowLatency	-27.3667	4.50831	-6.07027	< .0001
Manual Control	LowFailure	-28.2667	4.508435	-6.26973	< .0001
LowLat+Fail	GC Base	-28.6333	4.509062	-6.35018	< .0001
LowPen	GC Base	-29.5	4.508999	-6.54247	< .0001
Manual Control	GC Base	-29.9667	4.508373	-6.64689	< .0001

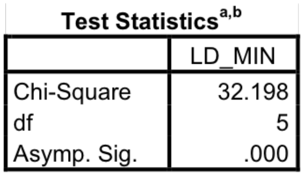


Test of Homogeneity of Variances

LD_MIN

Levene Statistic	df1	df2	Sig.
7.230	5	174	.000

Figure O-30: SPSS output for Levene test of equal variances for LD values for the GC architecture across different failure rates at 22 aircraft under the “automated” launch preparation time, using Low Latency and Low Penalty.



Test Statistics[a,b]

	LD_MIN
Chi-Square	32.198
df	5
Asymp. Sig.	.000

Figure O-31: SPSS output for Kruskal-Wallis non-parametric analysis of variance for LD values for the GC architecture across different failure rates at 22 aircraft under the “automated” launch preparation time, using Low Latency and Low Penalty.

Table O.11: JMP PRO 10 output for Steel-Dwass non-parametric simultaneous comparisons test for LD values for the GC architecture across different failure rates at 22 aircraft under the “automated” launch preparation time, using Low Latency and Low Penalty.

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value
LowLat+Pen 3%	LowLat+Pen 2%	15.8333	4.50925	3.5113	0.0059
LowLat+Pen 3%	LowLat+Pen 1%	15.4333	4.50925	3.42259	0.0082
LowLat+Pen 5%	LowLat+Pen 2%	8.6333	4.50925	1.91458	0.393
LowLat+Pen 4%	LowLat+Pen 2%	8.5667	4.50925	1.8998	0.4022
LowLat+Pen 5%	LowLat+Pen 1%	8.3	4.50925	1.84066	0.4394
LowLat+Pen 4%	LowLat+Pen 1%	7.2333	4.50925	1.60411	0.5958
LowLat+Pen 5%	LowLat+Pen 4%	3.2333	4.50925	0.71704	0.98
LowLat+Pen 2%	LowLat+Pen 1%	0.5	4.50925	0.11088	1
MC	LowLat+Pen 1%	-2.5667	4.508498	-0.5693	0.993
MC	LowLat+Pen 2%	-6.4333	4.508498	-1.42693	0.7106
LowLat+Pen 5%	LowLat+Pen 3%	-6.8333	4.50925	-1.5154	0.6543
LowLat+Pen 4%	LowLat+Pen 3%	-9.2333	4.509124	-2.0477	0.3153
MC	LowLat+Pen 5%	-14.0333	4.508498	-3.11264	0.0228
MC	LowLat+Pen 4%	-15.9	4.508498	-3.52667	0.0056
MC	LowLat+Pen 3%	-23.7	4.508498	-5.25674	$< .0001$