



VCU

Virginia Commonwealth University
VCU Scholars Compass

Theses and Dissertations


Graduate School

2016

Design & Analysis of a Computer Experiment for an Aerospace Conformance Simulation Study

Ryan W. Gryder
Virginia Commonwealth University

Follow this and additional works at: <https://scholarscompass.vcu.edu/etd>

 Part of the [Aerospace Engineering Commons](#), [Applied Statistics Commons](#), [Statistical Methodology Commons](#), and the [Statistical Models Commons](#)

© The Author

Downloaded from

<https://scholarscompass.vcu.edu/etd/4208>

This Thesis is brought to you for free and open access by the Graduate School at VCU Scholars Compass. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of VCU Scholars Compass. For more information, please contact libcompass@vcu.edu.

©Ryan Gryder, May 2016

All Rights Reserved.

DESIGN & ANALYSIS OF A COMPUTER EXPERIMENT FOR AN AEROSPACE
CONFORMANCE SIMULATION STUDY

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of
Science at Virginia Commonwealth University.

by

RYAN GRYDER

August 2014 to May 2016

Director: Dr. David Edwards,

Associate Professor of Statistics, Department of Statistical Sciences & Operations Research

Virginia Commonwealth University

Richmond, Virginia

May, 2016

Acknowledgements

I would like thank my mentors at NASA Langley Research Center, Dr. Sara Wilson and Kurt Swieringa, for letting me join their team and work on this interesting problem. Also, many thanks and gratitude to Dr. David Edwards for his help and support throughout this research project.

TABLE OF CONTENTS

Chapter	Page
Acknowledgements	i
Table of Contents	ii
List of Tables	ii
List of Figures	iii
1 Introduction	1
1.1 Interval Management within ATD-1	1
1.2 Motivation for Computer Experiment	3
1.3 Research Questions & Chosen Statistical Methods	4
2 Literature Review	7
2.1 ATD-1, Interval Management, and ASTAR Spacing Algorithms	7
2.2 Computer Experiments and Space-filling Designs	10
3 Methodology	14
3.1 Simulation Environment	14
3.1.1 Delayed Trajectory Construction	18
3.1.2 Emulated ASTAR12 Algorithm and Simulation of Aircraft Flights	24
3.2 Experimental Design	27
3.3 Response Metrics	35
3.4 Modeling	40
4 Results	48
4.1 Response Visualizations	48
4.2 Model Results	52
5 Discussion	69
References	71
Vita	75

LIST OF TABLES

Table		Page
1	The average $\min d(b_i, b_j)$ over 1000 space-filling samples using each of our space-filling methods for two wind sets.	34
2	Variable types and descriptions for computer experiment.	39
3	Logistic regression model fitting results for <i>Interrupt</i> response. * Significant at 0.05 level.	53
4	Logistic regression model fitting results for <i>SpaceViol</i> response. * Significant at 0.05 level.	54
5	Multiple linear regression model fitting results for <i>CPA</i> response. * Significant at 0.05 level.	56
6	Gamma regression model fitting results for <i>Interrupt_DTG</i> response. * Significant at 0.05 level.	58
7	Gamma regression model fitting results for <i>SpaceViol_DTG</i> response. * Significant at 0.05 level.	59

LIST OF FIGURES

Figure	Page
1 The specific airspace configuration that is modeled in this simulation study. . . .	2
2 The <i>CAS</i> route attribute vector for the BOSS2 arrival. * Meter fix ** Merge point.	16
3 The five nested factors (left) that result in the three delay differentials (right) through an unknown function f	30
4 An enumeration of the delay differential space for $Wind_1$ in three dimensions (a) and projected onto each two-dimensional plane (b)-(d). Delay differentials are in seconds.	32
5 The 30 selected delay differential points for $Wind_1$ in three dimensions (a) and projected onto each two-dimensional plane (b)-(d). Delay differentials are in seconds.	36
6 Percentage of Spacing Violations and IM Interruptions for each wind set across all levels of IM knowledge.	49
7 Percentage of Spacing Violations and IM Interruptions for different levels/combinations of IM knowledge across all wind sets.	49
8 Distribution of CPA response over the entire data set. CPA is in nmi.	50
9 Distribution of the DTG of spacing violations where they first occur across the entire data set.	51
10 Distribution of the DTG of IM interruptions where they first occur across the entire data set.	52
11 The predicted <i>CPA</i> response surface versus $TGT_delaydiff$ (X_1) and $LED_delaydiff$ (X_2) for $Wind_1$ (a), $Wind_2$ (b), $Wind_3$ (c), and $Wind_4$ (d). The delay differentials are standardized to be within 0 and 1.	63
12 The predicted <i>CPA</i> response surface versus $TGT_delaydiff$ (X_1) and $LED_delaydiff$ (X_2) for $Wind_5$ (a), $Wind_6$ (b), $Wind_7$ (c), and $Wind_8$ (d). The delay differentials are standardized to be within 0 and 1.	64

- 13 The predicted *CPA* response surface versus *TGT_delaydiff* (X_1) and *TRL_delaydiff* (X_2) for *Wind*₁ (a), *Wind*₂ (b), *Wind*₃ (c), and *Wind*₄ (d). The delay differentials are standardized to be within 0 and 1. 65
- 14 The predicted *CPA* response surface versus *TGT_delaydiff* (X_1) and *TRL_delaydiff* (X_2) for *Wind*₅ (a), *Wind*₆ (b), *Wind*₇ (c), and *Wind*₈ (d). The delay differentials are standardized to be within 0 and 1. 66
- 15 The predicted *CPA* response surface versus *LED_delaydiff* (X_1) and *TRL_delaydiff* (X_2) for *Wind*₁ (a), *Wind*₂ (b), *Wind*₃ (c), and *Wind*₄ (d). The delay differentials are standardized to be within 0 and 1. 67
- 16 The predicted *CPA* response surface versus *LED_delaydiff* (X_1) and *TRL_delaydiff* (X_2) for *Wind*₅ (a), *Wind*₆ (b), *Wind*₇ (c), and *Wind*₈ (d). The delay differentials are standardized to be within 0 and 1. 68

Abstract

Within NASA's Air Traffic Management Technology Demonstration # 1 (ATD-1), Interval Management (IM) is a flight deck tool that enables pilots to achieve or maintain a precise in-trail spacing behind a target aircraft. Previous research has shown that violations of aircraft spacing requirements can occur between an IM aircraft and its surrounding non-IM aircraft when it is following a target on a separate route. This research focused on the experimental design and analysis of a deterministic computer simulation which models our airspace configuration of interest. Using an original space-filling design and Gaussian process modeling, we found that aircraft delay assignments and wind profiles significantly impact the likelihood of spacing violations and the interruption of IM operations. However, we also found that implementing two theoretical advancements in IM technologies can potentially lead to promising results.

CHAPTER 1

INTRODUCTION

1.1 Interval Management within ATD-1

As a part of the National Aeronautics and Space Administration's (NASA's) Airspace Operations and Safety Program, the objective of the Air Traffic Management Technology Demonstration # 1 (ATD-1) is to demonstrate the operational capabilities of three new integrated research technologies developed by NASA. The three integrated technologies are an effort to address the need for fuel efficiency, increased aircraft throughput into high-density airports, greater schedule reliability while in the presence of large aircraft delay assignments, and adequate aircraft spacing during Trajectory-Based Operations (Baxley et al., 2013). These three technologies consist of the following:

- TMA-TM: *Traffic Management Advisor with Terminal Metering* produces precise time-based schedules for an aircraft's arrival to the runway within terminal airspace.
- CMS: *Controller-Managed Spacing* is a suite of decision support tools that provide controllers with advisories to help meet the schedule generated by TMA-TM.
- FIM: *Flight-deck Interval Management* is a system of aircraft avionic technologies as well as flight crew procedures for achieving or maintaining aircraft spacing requirements.

The goal of ATD-1 is to accelerate the implementation of NASA scheduling and spacing technologies enabling aircraft to use speed control to fly more efficient arrivals during high-density operations. This research is concerned with assessing the feasibility of the last ATD-1 technology, FIM.

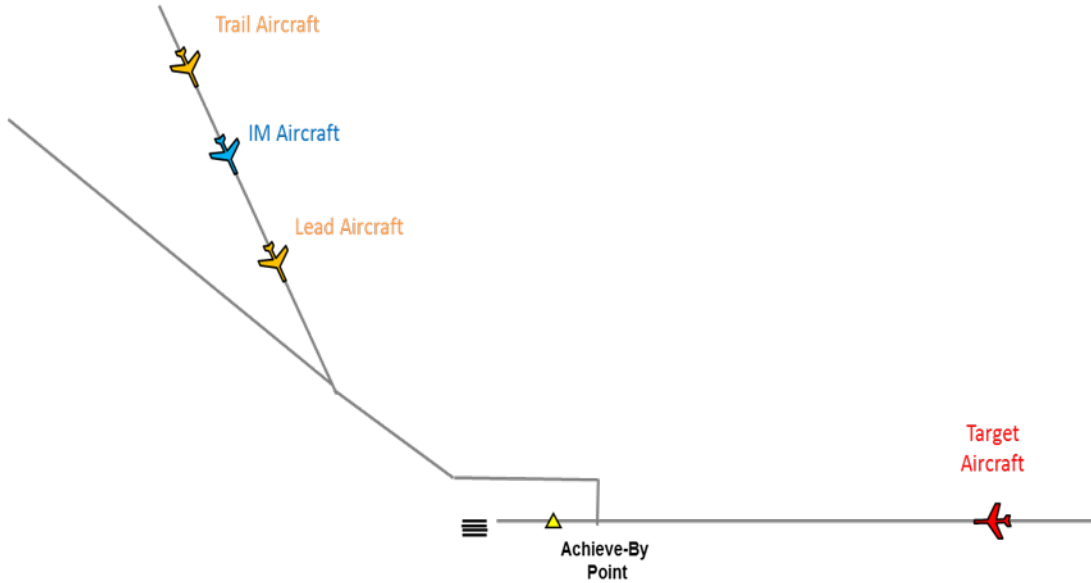


Fig. 1. The specific airspace configuration that is modeled in this simulation study.

Within ATD-1, Interval Management (IM) is a new speed control technology that will allow an IM-equipped aircraft (FIM) to achieve a desired spacing goal behind a target aircraft (TGT) at the final approach fix (FAF) of the arrival route into a high-density airport. The FAF is the last scheduling waypoint prior to the arrival airport on an aircraft's route where waypoints are simply points along a route where trajectory speeds and altitudes have specific requirements. We are interested in the specific scenario where FIM is following a TGT on a different route and whether this situation will cause spacing violations between FIM and the non-IM equipped lead aircraft in front (LED) or the trail aircraft behind (TRL) on the same route. In this case, the FAF is the merge waypoint between the FIM and TGT route. In other words, we are dealing with a scenario where an aircraft (FIM) is being controlled to meet a spacing requirement behind another aircraft (TGT) at the moment the two aircrafts' routes merge together. Figure 1 below displays our aircraft configuration of interest.

NASA has seen behavior in human-in-the-loop (HITL) simulations, which is an experiment involving human subjects, where spacing violations occur between FIM and its surrounding non-IM aircraft on the same route when following a TGT on a different route.

We review the literature on this in the next chapter. The following factors and their influence on spacing violations are of primary interest for this research: winds, the FIM aircraft's knowledge of delayed trajectories, the FIM aircraft's knowledge of TGT specific winds, and the difference in delay assignments between FIM and all other aircraft. Throughout this thesis, a 'trajectory' is an arrival flight into an airport, and 'delay' is a time amount that is added to an aircraft's total trajectory flight time that needs to be absorbed. The computer experiment and statistical analysis described in this paper are intended to shed light on initial airspace/aircraft conditions that can possibly lead to aircraft spacing violations while IM operations are implemented.

1.2 Motivation for Computer Experiment

The goal of IM is to improve the precision of inter-aircraft spacing to increase the fuel efficiency of trajectories as well as the throughput for arrivals into high-density airports. However, we do not expect that all consecutive aircraft in a given string will be equipped with IM technology. If IM technologies are to be safely implemented in the future, there is a need for a computer simulation model with a high level of fidelity to examine the interaction between IM and non-IM aircraft flying in a consecutive sequence while the TGT aircraft is on a different route. A computer simulation allows for the analysis of a wider range of airspace conditions that can be examined in much quicker succession than HITL simulations can allow. While there is clearly some sacrifice in fidelity and accuracy when performing computer simulations over HITL simulations of IM operations, we are confident that the wider range of test conditions and conclusions coupled with statistical rigor in the analysis more than balance out the cost. We discuss the fidelity of the computer model in the Simulation Verification and Validation section in the Methods chapter.

One of the main benefits of computer simulation studies is the ability to test theoretical ideas and concepts that might not be able to be physically tested or even physically exist. The current state of IM operations limits the FIM aircraft's knowledge of the TGT aircraft's

delayed trajectory as well as winds encountered by the TGT aircraft on its separate route. In the near-term national airspace system, the FIM aircraft is only expected to have access to the TGT aircraft's nominal trajectory data; FIM also assumes that TGT is encountering the same wind profile as FIM despite flying on different routes with possibly very different wind profiles. Therefore, we are interested in whether implementing FIM access to the TGT aircraft's actual delayed trajectory and route-specific wind profile will lead to considerably fewer spacing violations. While the IM technology does not currently exist to allow for this level of sophistication, a computer simulation gives us the opportunity to investigate whether implementing these advancements could prove worthwhile in terms of decreasing spacing violations and improving IM technology.

The speed-control algorithm that enables an aircraft to achieve a desired spacing goal behind a TGT aircraft by an achieve-by point is known as Airborne Spacing for Terminal Arrival Routes - Version 12 (ASTAR12). A description of this algorithm and its emulation within the computer model can be found in Chapters 2 and 3. Additional motivation behind the use of a computer experiment here is the relative ease in mathematically formulating the ASTAR12 speed control algorithm. A computer experiment allows us to investigate the performance of IM operations under control of an emulated trajectory-based ASTAR12 spacing algorithm given a variety of initial airspace conditions and in much less required time than a HITL simulation study.

1.3 Research Questions & Chosen Statistical Methods

Our main research questions are motivated by the current limitations of IM operations as well as factors that are known to have some influence on them as evident by recent HITL studies. The research questions are as follows:

1. To what extent do the following four factors contribute to spacing violations between a FIM aircraft and its surrounding non-IM aircraft (LED and TRL) when FIM is following a TGT on a different route?

- (a) Winds
 - (b) Difference in delay assignments between FIM and TGT, LED, and TRL
 - (c) FIM aircraft's knowledge of delayed trajectories
 - (d) FIM aircraft's knowledge of TGT aircraft's route-specific winds
2. Given a set of initial airspace/aircraft conditions expressed in the factors above, what is the probability that a spacing violation will occur at some point along the trajectory?
 3. Given a set of initial airspace/aircraft conditions expressed in the factors above, what is the Closest Point of Approach (CPA) between FIM and its surrounding non-IM aircraft, and where along the route do losses of appropriate separation occur?

In order to answer these questions, we coded a computer simulation in MATLAB that models our aircraft configuration of interest with four aircraft and simulates their flown trajectory. The simulation is deterministic in that there is no random component relating the input to the output. The same initial aircraft conditions such as winds, initial distance from the runway, and delay assignment will lead to the exact same flown trajectory of each aircraft. This removes a level of complexity in the design of an appropriate experiment in that repeated trials are unnecessary; however, it adds a level of complexity in terms of statistical analysis and building predictive models when quantifying error becomes challenging. A lot of difficulty in this deterministic computer experiment originated from manipulating the delay differential factor: the difference in assigned delay between FIM and TGT, LED, and TRL. This 3-dimensional continuous delay differential space cannot be completely explored in terms of running the experiment at every single point; therefore, we created an original *space-filling design* to strategically select a finite number of design points to run the experiment. From recorded responses at these selected design points, we interpolated response surfaces via *Gaussian stochastic process* modeling by formulating one run of the deterministic computer experiment as a realization of a stochastic process that is the actual physical experiment

being emulated. For our other responses, we used classical statistical models such as multiple linear regression, logistic regression, and Gamma regression.

The following Chapter 2 is a literature review of recent NASA research on IM technologies and ASTAR12 within ATD-1 as well as a literature review on the statistical tools used in the design and analysis of this computer experiment. In particular, the review discusses computer experiments in general and space-filling designs. Chapter 3 describes the methodology used. It includes a detailed description of the computer model and the design and analysis of the experiment. Chapter 4 presents our results, and the final Chapter 5 is a discussion along with future work.

CHAPTER 2

LITERATURE REVIEW

2.1 ATD-1, Interval Management, and ASTAR Spacing Algorithms

With the large growth in aviation and number of airborne passengers predicted over the next 20 years, there is a need to revise current aircraft arrival procedures (FAA, 2011). During inclement weather and peak periods of air travel, current arrival procedures often lead to high demand on ATC, increased delays, and airports operating at reduced capacity. ATD-1 provides a proposed solution to some of these problems through an integrated system of airborne and ground-based technologies and decision support tools. Lab simulation studies are currently being conducted to prepare for a field prototype and operational demonstration of the three integrated ATD-1 technologies. In preparation for operational implementation, each individual piece within ATD-1 needs to be further studied (Prevot et al., 2011).

TMA-TM, within ATD-1, allows for precise automatic time-based scheduling of arriving aircraft. HITL simulation studies have concluded that this new scheduling procedure leads to a 10% increase in airport throughput during maximum capacity (Swenson et al., 2011). CMS is a set of tools that provides advisories to air traffic controllers to help meet the schedule produced by TMA-TM. Recent simulation results suggest that experienced air traffic controllers using these new CMS tools for the first time can handle them effectively within ATD-1 operations (Callantine et al., 2014).

CMS and TMA-TM are both ground-based tools, but what about the airborne aspect of ATD-1 operations? FIM is a set of avionics and procedures that allow an equipped aircraft to achieve or maintain a spacing goal behind a target aircraft. Interest in airborne spacing concepts have been renewed due to enabling technology known as Automatic Dependent Surveillance Broadcast (ADS-B). This ADS-B technology allows for the broadcast

and reception of accurate current trajectory state information of surrounding aircraft by an IM-equipped aircraft so it can achieve or maintain spacing behind a target (Abbott, 2015).

The NASA algorithm that enables an IM-equipped aircraft to achieve a spacing goal behind a target aircraft is ASTAR. The ASTAR algorithm is a trajectory-based IM algorithm specifically designed to be used in a Next Generation Air Transportation System (NextGen) environment where accurate information about a target aircraft's predicted trajectory with a delay assignment is available to the IM aircraft via ADS-B. The algorithm uses the predicted trajectories of the IM and target aircraft to calculate the time-to-go for each aircraft to an achieve-by point. A *spacing error* is calculated by comparing the difference between the time-to-go of the IM and target aircraft with the spacing goal calculated by TMA-TM. This formulation of the *spacing error*, along with other aspects of ASTAR, allows the IM aircraft to follow a target on a different route. HITL and batch simulation studies have demonstrated that the ASTAR algorithm is precise in achieving spacing goals and also produces acceptable aircraft speeds. These simulation studies assumed that ADS-B would allow for the reception of detailed information of the target aircraft's intended trajectory. However, NASA does not foresee the implementation of these detailed data link broadcasts in the near future. Instead, IM clearances will be provided via voice communications from ATC on the ground, and the predicted trajectories of the IM and target aircraft will be assumed to be the published Standard Terminal Arrival Routes (STARs). The STARs are the ideal nominal trajectory speeds when an aircraft is not assigned any delay amount; therefore, they are rarely the speeds that an aircraft actually flies. In order to have ATD-1 technologies operating as soon as possible, recent simulation studies have been conducted to examine the integration of FIM with TMA-TM and CMS without the ADS-B data link communication. The previous version of ASTAR, ASTAR11, did not perform well with TMA-TM and CMS within ATD-1. This was due to ASTAR using published STARs to estimate the target aircraft's intended trajectory without an updated delay assignment while TMA-TM produces an accurate schedule including aircraft delay assignments; the speeds expected by ASTAR

and TMA-TM did not always match. ASTAR11 was not designed to handle large differences between the target’s actual speed and published speeds when predicting Estimated Times of Arrival (ETAs). This led to spacing violations when the target aircraft was absorbing a large amount of delay (Swieringa, 2015).

A recent update to the ASTAR algorithm, ASTAR12, is intended to alleviate these problems when the target aircraft is flying a delayed trajectory and allow for improved integration of FIM with TMA-TM and CMS. The main modification is a *groundspeed term* that compensates for discrepancies between the target aircraft’s actual delayed speeds and published speeds. This new *groundspeed term* enables the IM aircraft to match the target’s speed deviation and proceed to correct for the *spacing error* using a proportional control term. HITL and batch simulation studies have confirmed the acceptability of the ASTAR12 algorithm; in general, ASTAR12 exhibits good behavior when the IM aircraft is following a target aircraft on a different route with an unknown delayed trajectory (Swieringa et al., 2015). But it is not enough that ASTAR12 can achieve or maintain a precise spacing goal when the IM aircraft is following a delayed target aircraft on a separate route. What about the interaction between the IM aircraft and its immediate surrounding non-IM aircraft when flying in a string? In the near future, it is not expected that all aircraft will be equipped with the necessary IM avionics. Instead, IM is expected to fit within a larger arrival management context developed to support a much broader mixed-equipage environment (Levitt et al., 2014). Recent ATD-1 HITL simulation study results indicate that controllers needed to interrupt or terminate FIM operations due to unexpected compression of the IM aircraft with its preceding or leading in-trail aircraft that are not controlled by IM (Robinson, 2014). This brings us to this research; there is a need to investigate the interaction between an IM-equipped aircraft controlled by ASTAR12 to achieve a spacing goal behind a delayed target on a separate route and the surrounding non-IM aircraft controlled to meet Scheduled Times of Arrival (STAs) at scheduling waypoints on the IM aircraft’s shared route.

2.2 Computer Experiments and Space-filling Designs

With the rise in computing efficiency and storage capacity, computer experiments have increased in popularity among statisticians as well as engineers as a way to emulate physical processes and experiments via codified mathematical models. A computer experiment is computer code that executes the emulation of a physical scientific experiment where factors, responses, and even randomness are quantified and simulated via code inputs and outputs. Computer experiments can be beneficial when the physical experiment is overly time-consuming or costly, and there is a need to run the experiment more frequently than what is physically feasible. Computer experiments can also be useful when researchers want to investigate new theoretical technologies or physical procedures but first would like to get an idea if the theoretical implementation would even be beneficial and worth creating. This is a primary motivator for the use of a computer experiment in this research: to investigate the expansion of certain IM capabilities and address some of its limitations.

Scientists have been conducting computer experiments throughout the history of computers. Some of the earliest accounts of computer experiments were of physicists simulating the nonlinear interaction of atoms in a crystal (Fermi et al., 1955) as well as the fluid-dynamics of a flame via a complex system of partial differential equations (Kee et al., 1985). The earliest examples of statisticians analyzing the output of computer experiments also involved physics models such as the analysis of nuclear reactor safety codes to understand various accident scenarios and potential consequences in a nuclear power plant. Statistical analysis of these nuclear reactor safety codes represented a shift towards using sampling of the input space of a computer model to gain an understanding of the consequences of different accident scenarios. In these early stages of the analysis of computer model output, substantial computing resources were needed for one execution of a piece of computer code. This led to the emergence of a rich statistical research area for developing statistical sampling, experimental design, and analysis methods for computer experiments.

Computer experiments can either be stochastic or deterministic. In a stochastic com-

puter experiment, there is an element of randomness within the simulated experiment. The randomness can originate from stochastic differential equations, random number generators, or any use of random variables within the simulation. A deterministic computer experiment has no random component; the same set of inputs will return the exact same outputs under repeated runs. Most computer experiments are deterministic, and this research is centered around the analysis of a deterministic computer experiment. Throughout this paper, our focus will be on the deterministic type of computer experiments. It is important to point out the differences between deterministic computer experiments and physical experiments (despite the obvious) that motivate the methods of analysis. The usual facets of a good experimental design such as blocking, replication, and randomization have no meaning since the experiment has no random component; every experimental combination of factors need only be tested once. Another major difference between these two kinds of experiments that is related to the lack of randomness is the source of variation within the data. In a computer experiment, all variation is contained within the data set and dependent on the complexity of the computer code. With these differences present and realized, how can we quantify the uncertainty in a deterministic computer experiment? How do we interpret usual metrics of uncertainty stemming from least squares residuals? If we think of the computer experiment as a realization of a stochastic process that is the physical experiment being emulated, then we have a statistical framework for analyzing the data and building predictive models (Sacks et al., 1989). We will discuss the mathematical formulation for analyzing a deterministic computer experiment as a realization of a stochastic process in the Methods chapter as it is not pertinent for the reader until then. In this section, however, we will give a brief overview of the literature pertaining to the design of deterministic computer experiments with a particular emphasis on *space-filling designs*.

Since computer experiments are most often used when there are complex interrelationships among the variables and responses, the difficulty in designing an appropriate experiment lies in picking design points within the complicated and irregular experimental region.

Since it is more often than not impossible to completely fill the experimental region of interest, the common strategy is to use a *space-filling design* that fills the experimental region, or *space*, by spreading out a small number of selected design points evenly and uniformly. *Space-filling designs* are useful in these situations since predictive models are usually based on interpolating a response surface over the entire design region. Evenly spaced design points are also advantageous since the model prediction error at any given point is relative to the distance to the closest design point. Unevenly selected design points can lead to poor predictions in sparsely sampled portions of the experimental region.

The most common *space-filling design* is a *Latin Hypercube Design*, or LHD, first introduced by McKay et al. (1979). LHDs are the multi-dimensional generalization of the *Latin Square Design* where there is exactly one run per row/column combination of two-factor treatments. An LHD of n runs for k factors is represented by an $n \times k$ matrix where each column is a permutation of n equally spaced intervals. The n levels are $-(n-1)/2, -(n-3)/2, \dots, (n-3)/2, (n-1)/2$. Given an $n \times k$ Latin hypercube $L = (l_{ij})$, a corresponding design $D \in (0, 1)^k$ can be generated as an $n \times k$ matrix with the (i, j) entry being

$$d_{ij} = \frac{l_{ij} + (n-1)/2 + \mu_{ij}}{n}, \quad i = 1, \dots, n; \quad j = 1, \dots, k \quad (2.1)$$

where each μ_{ij} is an independent random number in $(0,1)$. LHDs have one-dimensional uniformity; that is, for each factor, there is exactly one point in each of the n intervals. Along with this attribute, LHDs are popular due to their reduction on variance of model estimates. Let $\hat{\mu}_{srs}$ and $\hat{\mu}_{lhs}$ be the mean estimates of a predictive model where the design points are chosen by a simple random sample and Latin hypercube sample, respectively, both of size n . McKay (1979) showed that when the response is monotonic in the each of the input factors, then $Var(\hat{\mu}_{lhs}) \leq Var(\hat{\mu}_{srs})$.

Even with such desirable qualities, randomly generated LHDs do not always perform well with respect to some criteria such as space-filling or orthogonality. When projected onto

two factors, for example, design points from a random LHD could lie on the diagonal leaving a large area of the design space left untested. To combat this, many modifications to the basic LHD have been proposed. Proposed by Johnson et al. (1990), the maximin distance criteria seeks to maximize the minimum distance between any two design points so that no two points are too close. A maximin distance design spreads design point evenly throughout the entire experimental region. Morris and Mitchell (1995) combined the maximin criteria with LHDs and proposed *maximin Latin hypercube designs*.

As the number of input dimensions increases, it becomes more difficult to select a small number of design points that effectively cover the entire high-dimensional input space. An easier approach is to construct a design that is space-filling in low-dimensional projections. Moon (2011) proposed designs that are space-filling in two dimensions while simultaneously satisfying the maximin criteria in high dimensions. Other designs that are space-filling in low dimensions are randomized orthogonal arrays (Owen, 1994) and orthogonal array based LHDs (Tang, 1993). As an alternative to space-filling designs, there are designs that perform well with respect to model-dependent criteria such as the minimum integrated mean square error (Sacks et al., 1989) and maximum entropy (Shewry and Wynn, 1987). However, these alternatives to space-filling designs require prior knowledge of the model, but some Bayesian approaches have been proposed (Leatherman, 2014).

CHAPTER 3

METHODOLOGY

3.1 Simulation Environment

A computer simulation was coded using the MATLAB programming language and environment to run the experiment and produce data. The simulation models the arrival flights of four aircraft into the Denver International Airport via the arrival routes known as BOSS2 and ANCHR2. The four modeled aircraft consist of:

- FIM: The IM equipped aircraft that is given an achieve-by clearance to achieve a particular spacing goal behind a target aircraft at the Final Approach Fix (FAF); in our specific scenario, the FAF is the Merge point where the FIM and TGT routes join. The FIM aircraft's speeds are controlled by the ASTAR12 algorithm. FIM flies the BOSS2 route. Note: for the remainder of this thesis, FIM will refer to the IM-equipped aircraft within the simulation rather than *Flight-deck Interval Management* as one of the three research technologies within ATD-1.
- LED: The lead non-IM aircraft immediately in front of FIM on the same BOSS2 arrival route whose speed is controlled by Air Traffic Control (ATC) on the ground to meet its predetermined schedule with a delay assignment.
- TRL: The trail non-IM aircraft immediately behind FIM on the same BOSS2 route whose speed is controlled by ATC on the ground to meet its predetermined schedule with a delay assignment.
- TGT: The target non-IM aircraft that FIM must achieve a spacing goal behind at the Merge point. The TGT is on a different arrival route, ANCHR2, than the other

three aircraft. Its speed is controlled by ATC on the ground to meet its predetermined schedule with a delay assignment.

While TRL, FIM, and LED start flying on a separate route from TGT, the two routes merge at a Merge waypoint towards the end of the arrival flight. At this Merge point, the TGT starts flying in-trail behind LED and in front of FIM. Therefore, the arrival order of the modeled aircraft is as follows: LED, TGT, FIM, TRL. The simulation is designed to model this specific aircraft configuration. Modifications are needed if it is desired to model more than four aircraft on more than two routes. However, any pair of routes that eventually merge can be used in the simulation. We discuss how the simulation can and cannot be generalized to model other airspace scenarios in the final Chapter 5 where we discuss future work.

The simulation starts by assigning a route and wind profile to the TRL, FIM, and LED aircraft and a separate route and wind profile to the TGT. Route information is stored as a group of vectors where each vector represents a different trajectory attribute such as speed or distance. Each vector entry represents a different waypoint along the route at a particular distance-to-go (DTG) to the runway. A waypoint is just a physical location where the nominal/published route has required speeds and altitudes. Each route attribute vector is the same length; positions of vector entries correspond to one another. For example, the first entry of each vector represents all of the trajectory attributes at the furthest waypoint along the route. The routes can be divided into two disjoint trajectory segments (starting with furthest from the runway): Center Air Space and TRACON. The Meter Fix waypoint separates the Center Air Space from the TRACON; this is the waypoint where an aircraft's nominal calculated air speed first reaches 250 knots. In our chosen pair of arrival routes into Denver International Airport, the Merge waypoint for the two routes is the FAF within the TRACON, and it is the third-to-last waypoint. The following is a list of all of the initial route attribute vectors (*italicized for the remainder of this thesis*) along with an illustration of the two disjoint route segments given the calculated air speed vector (Figure 2):

$$CAS = \begin{bmatrix} 280 \\ 280 \\ 280 \\ 280 \\ 280 \\ 250^* \\ 210 \\ 210 \\ 210 \\ 210 \\ 210 \\ 170^{**} \\ 135 \\ 135 \end{bmatrix}$$

} Center Air Space
} TRACON

Fig. 2. The *CAS* route attribute vector for the BOSS2 arrival. * Meter fix ** Merge point.

DTG = distance to go (nmi) at all waypoints,

alt = altitude (ft) at all waypoints,

CAS = calculated air speed (knots) at all waypoints,

CAS_{min} = minimum allowable calculated air speed (knots) at all waypoints,

track_angle = route track angle (degrees) with respect to the runway at all waypoints.

For the remainder of this thesis, when any of these route attribute vectors is followed by (*j*) where *j* is an integer, such as *DTG(j)*, we are referring to the route attribute value at the *j* waypoint which is the *j* entry of the vector. After we assign a route to an aircraft,

we assign it a route specific wind profile. A wind profile consists of two wind grids: a *Truth* and *Forecast* wind grid. The *Forecast* wind grid is what ATC predicts the winds to be at the time of flight. These wind values are used in the delayed trajectory construction as well as in the ASTAR12 algorithm emulation. The *Truth* wind grid is what an aircraft will actually encounter while flying. These wind values are only used for the computation of each aircraft's actual flown speed and distance along the route as the simulation is run. The term 'grid' is used because the winds originate in the form of two matrices; one matrix has wind speeds (knots) while the other has wind heading directions (degrees) with respect to the front of the aircraft. Each matrix column represents a different waypoint DTG along the particular route while each row represents a different altitude. The altitudes range from the altitude at the runway to 36000 feet in increments of 1000 feet. After a *Truth* and *Forecast* wind grid of wind speeds and heading directions is assigned to each aircraft, we compute the following additional route attribute vectors to complete the route information:

$$\begin{aligned}
 Wind_Spd_T &= Truth \text{ wind speed (knots) interpolated from grid based on } alt, \\
 Wind_Dir_T &= Truth \text{ wind heading direction (degrees) interpolated from grid based on } alt, \\
 Wind_Spd_F &= Forecast \text{ wind speed (knots) interpolated from grid based on } alt, \\
 Wind_Dir_F &= Forecast \text{ wind heading direction (degrees) interpolated from grid based on } alt, \\
 HW_T &= Truth \text{ headwind (knots) } = Wind_Spd_T * \cos(Wind_Dir_T - track_angle), \\
 HW_F &= Forecast \text{ headwind (knots) } = Wind_Spd_F * \cos(Wind_Dir_F - track_angle), \\
 TAS &= \text{true air speed (knots) calculated from } CAS \text{ and } alt \text{ at each waypoint,} \\
 GS &= \text{ground speed (knots) at each waypoint } = HW_F + TAS, \\
 TTG &= \text{time to go (sec) to runway calculated from } GS \text{ and } DTG \text{ at each waypoint.}
 \end{aligned}$$

After all route information is computed for each aircraft, the computer simulation can be divided into two main sections which we will explain individually:

1. Delayed Trajectory Construction: In this section, the initial DTG of each aircraft, DTG_i , is assigned, scheduling requirements at scheduling waypoints are assigned to

meet spacing requirements, and an aircraft’s delayed trajectory is constructed to satisfy its schedule.

2. ASTAR12 Emulation and Simulation of Aircraft Arrival: In this section, we actually simulate the arrival flight of all of the aircraft. The FIM aircraft’s speed and trajectory is controlled by an emulated ASTAR12 algorithm implementation while the LED, TRL, and TGT aircrafts’ speeds and trajectories are controlled by ATC on the ground according to their respective delayed trajectories.

3.1.1 Delayed Trajectory Construction

The first half of the simulation involves constructing each aircraft’s delayed trajectory. In most scenarios, an aircraft’s nominal trajectory (fastest speeds) is not sufficient in ensuring appropriate spacing between consecutive aircraft at scheduling waypoints. Therefore, an aircraft is assigned a delayed trajectory which consists of slower speeds in order to meet spacing requirements. We refer to the nominal trajectories as *nomRte* and delayed trajectories as *dRte*. In this section, we will explain the procedure that the simulation uses to construct each aircraft’s delayed trajectory. Generally, it involves assigning an initial DTG for an aircraft, computing Estimated Times of Arrival (ETA’s) and Scheduled Times of Arrival (STA’s) at in-trail waypoints to meet spacing requirements, and assigning delay accordingly to satisfy that schedule. We note that the aircraft are assigned a delayed trajectory in the order that they arrive: LED, TGT, FIM, and TRL. Also, in order to mimic how this process is done in real time, each aircraft’s schedule and delayed trajectory is first constructed using the *Forecast* wind profiles since this is what ATC has access to prior to flight. Then a second delayed trajectory is constructed using the same schedule but with the *Truth* wind profile. This is because when an aircraft is actually flying, it needs to meet the same predetermined schedule but with the actual encountered winds. We will call these two delayed trajectories *dRte_F* and *dRte_T*.

LED Aircraft

Since the LED is the first to arrive, we deal with its initial DTG, schedule, and delayed trajectory first. We begin by setting the minimum and maximum value for the LED initial DTG and select a value within the range as such:

$$\begin{aligned}\min DTG_i &= \text{last waypoint in Center Air Space,} \\ \max DTG_i &= \text{furthest waypoint to ensure minimum spacing for FIM and TRL,} \\ DTG_i &\in (\min DTG_i, \max DTG_i).\end{aligned}$$

The $\min DTG_i$ ensures that we are modeling scenarios where all of the aircraft start in the Center Air Space. We will explain how we decide the value of DTG_i in the following Experimental Design section. Then, we compute the maximum amount of delay that the LED aircraft can absorb during its flight (starting at its $DTG_i = DTG(1)$) and select a delay amount less than that specified by the experimental design. In order to compute this maximum delay, we use the CAS_{min} or the slowest allowable calculated air speed profile along with the DTG vector to compute the time to fly the slowest speed profile, TTF_{slow} as such:

$$\begin{aligned}TTF_{nom} &= TTG(1) = \text{time to fly the nominal speed profile starting at } DTG(1), \\ \max Delay &= TTF_{slow} - TTF_{nom}, \\ Delay &\in (0, \max Delay).\end{aligned}$$

We will explain how we decide the value of the LED aircraft $Delay$ in the following Experimental Design section. Now that we have chosen the LED aircraft's delay amount, we need to distribute it throughout the LED's trajectory. We explain this procedure later at the end of this section where we discuss it for every aircraft. Assuming we have the LED aircraft's $dRte_F$, we now need to compute its STA's at each waypoint; the other aircraft will need it in their $dRte_F$ construction. The following Pseudocode 1 is for computing an aircraft's STA at all waypoints (not just LED).

```

    STA(1) = 0
    for j = 2 : number of waypoints
        STA(j) = TTG(j - 1) - TTG(j) + STA(j - 1)
    end

```

Pseudocode 1: How to compute an aircraft's STA.

Now that we are done with the LED aircraft, we move on to the remaining three aircraft which all share the same delayed trajectory construction process that is slightly different from that of LED.

TGT/FIM/TRL Aircraft

For the remaining three aircraft, the first thing to do in their delayed trajectory construction process is to compute the min DTG_i and max DTG_i as such:

min $DTG_i = DTG_i$ that ensures adequate spacing at FAF when aircraft flies at slowest speeds,

max $DTG_i = DTG_i$ that ensures adequate spacing at FAF when aircraft flies at nominal speeds,

$DTG_i \in (\min DTG_i, \max DTG_i)$.

The min and max DTG_i are computed by finding the DTG_i that corresponds to a time-to-fly that ensures adequate time-based spacing at the Merge point for each aircraft. The Merge point spacings are based on the standard FAF distance spacing of 3.3 nmi. For the TGT aircraft, the bounds on its DTG_i are determined by having a desired time-to-fly (TTF_{des}) to the Merge point that is the LED's Merge waypoint STA (s below) plus a distance-to-time converted spacing requirement (τ below) as such:

$$s = LED.STA(Merge),$$

$$\tau = 3600 * (3.3/TGT.GS(MERGE)),$$

$$TTF_{des} = s + \tau.$$

After we locate the $\min DTG_i$ and $\max DTG_i$ that correspond to a TTF_{des} to the Merge when the TGT is flying its slowest and nominal speed profile, respectively, we select the TGT DTG_i to be a value within the bounds. We explain our method of choosing this DTG_i in the Experimental Design section. We now have to construct the TGT aircraft's schedule of ETA's and STA's. The ETA's are computed from its nominal speed profile, and at non in-trail waypoints where the aircraft is not flying directly in front of or behind another aircraft, the STA's are the same since there are no other (modeled) aircraft to enforce spacing requirements. However, at the TGT in-trail waypoints behind LED from the Merge to the runway, the TGT STA needs to be at least the LED STA plus a time-based spacing requirement. Since all of the TGT aircraft's in-trail waypoints are from the Merge to the runway and after the FAF, these time spacings are all based on a distance spacing of 3.3 nmi. The following Pseudocode 2 computes the TGT aircraft's ETA and STA at all waypoints.

```

TGT.ETA(1) = 0
for j = 2 : number of waypoints
    TGT.ETA(j) = TGT.TTG(j - 1) - TGT.TTG(j) + TGT.ETA(j - 1)
    if waypoint j is an in-trail waypoint
         $\tau = 3600 * (3.3 / TGT.GS(j))$ 
        if TGT.ETA(j) < LED.STA(j) +  $\tau$ 
            TGT.STA(j) = LED.STA(j) +  $\tau$ 
        else
            TGT.STA(j) = TGT.ETA(j)
        end
    else
        TGT.STA(j) = TGT.ETA(j)
    end
end

```

end

Pseudocode 2: How to compute the TGT aircraft's ETA and STA at all waypoints while adhering to constraints induced by the LED aircraft at in-trail waypoints.

The min DTG_i , max DTG_i , ETA's, and STA's for the FIM and TRL aircraft are computed similarly. The only differences arise from the aircraft that FIM and TRL are flying in-trail behind and at which route segments. For FIM, its min DTG_i and max DTG_i are computed based on the TGT aircraft's STA at the Merge point and achieving the same STA plus an additional time-based spacing buffer. Based on the modeled trajectory, the FIM aircraft's STA's are computed based on maintaining adequate spacing behind the LED aircraft in Center Air Space and the TRACON but switches to maintaining spacing behind the TGT aircraft from the FAF, or Merge point, to the runway. For TRL, its min DTG_i and max DTG_i are computed based on the FIM aircraft's STA at the Merge point and achieving the same STA plus an additional time-based spacing buffer. Based on the modeled trajectory, the TRL aircraft's STA's are computed based on maintaining adequate spacing behind the FIM aircraft throughout the entire trajectory since it is always flying in-trail behind FIM. We have established that the schedule time spacings are based on distance spacings of 3.3 nmi from the FAF to the runway, but for the Center Air Space and TRACON portions of the route, the required distance spacings are different. For the Center Air Space, we used a constant distance spacing of 6 nmi. In the TRACON, we used a distance spacing that linearly decreases from 6 nmi at the Meter Fix (start of TRACON) to 3.3 nmi at the Merge point. In order to give the aircraft some breathing room when starting the simulation and to prevent spacing violations occurring from the get go, we implemented a hard constraint for the FIM and TRL min DTG_i so that the string of three aircraft (TRL, FIM, LED) start at least 6.5 nmi apart from one another.

Constructing Delayed Trajectories from Schedules

Now that we have each aircraft’s schedule of ETA’s and STA’s, we are ready to build their delayed trajectories. This process consists of determining the *CAS* profile that will satisfy the difference between an aircraft’s ETA and STA at each waypoint. The difference between an aircraft’s ETA and STA at a single waypoint is the amount of time delay that needs to be absorbed prior to that waypoint. We start by satisfying the needed delay for the first waypoint beyond an aircraft’s DTG_i and work our way towards the runway. It is an iterative procedure where for a given waypoint, we decrease all previous waypoint *CAS* speeds by 1 knot until the delay absorbed from the DTG_i to the current waypoint is within 1 second of what is desired based on the difference between the ETA and STA at that waypoint. This procedure works for the TGT, FIM and TRL aircraft; however, for the LED aircraft, this situation is slightly different. Since this aircraft is the first to arrive at the runway and there is no (modeled) aircraft flying in front of it to enforce any STA requirements, we were able to directly assign it a total delay amount. Because of this, the LED aircraft simply needs to absorb its total delay amount evenly throughout its entire route rather than satisfy specific delay amounts for every single waypoint like the other three aircraft. Therefore, we decrease the entire LED *CAS* speed profile by 1 knot until the delay absorbed is within 1 second of the delay value we selected for LED. After we compute each aircraft’s new slow *CAS* profile, which we call CAS_{slow} , we use the same DTG , alt , $track_angle$, and HW_F vectors from the $nomRte$ to compute new TAS_{slow} , GS_{slow} , and TTG_{slow} vectors for the $dRte_F$. We note that the only trajectory state attribute vectors that are different between the nominal and delayed trajectories are those that have to do with speed and time in their calculation. The distance, altitude, and wind trajectory state attribute vectors remain the same since they have nothing to do with the aircraft’s speed. We also remind the reader that after the $dRte_F$ is constructed using the *Forecast* winds, the $dRte_T$ is constructed using the *Truth* winds by replacing HW_F with HW_T in the GS_{slow} calculation. In the simulation of each aircraft’s flight which we describe in the next section, the $dRte_T$ is used because each

aircraft needs to fly according to its predetermined schedule but with the actual encountered winds.

Since one of the primary factors of interest in our computer experiment is the difference in delay between FIM and the other aircraft, we need to establish where these delay amounts come from. For the LED aircraft, we are at liberty to explicitly select this delay value; however, for the other three aircraft, the delay amount is the difference in time it takes to fly the nominal trajectory versus the delayed trajectory we have constructed as such:

$$TGT.Delay = TGT.STA(end) - TGT.ETA(end),$$

$$FIM.Delay = FIM.STA(end) - FIM.ETA(end),$$

$$TRL.Delay = TRL.STA(end) - TRL.ETA(end).$$

3.1.2 Emulated ASTAR12 Algorithm and Simulation of Aircraft Flights

After a DTG_i , $dRte_T$, and $dRte_F$ is computed for each of our four modeled aircraft, we are ready to simulate their actual flights. We do this by tracking the actual flown trajectory attributes of each aircraft in 1 second increments. This process is different between FIM and the other three aircraft. For this reason, we divide this section accordingly. When one run of the simulation is done, it returns the actual flown trajectory attribute vectors for CAS , GS , and DTG which we label as CAS_{act} , GS_{act} , and DTG_{act} , respectively. Unlike the trajectory attribute vectors for $nomRte$, $dRte_T$, and $dRte_F$ where each vector entry represents a different waypoint, each vector entry of CAS_{act} , GS_{act} , and DTG_{act} represents a 1 second time step of each aircraft's actual flight. These vectors have a length of the total number of seconds it takes the aircraft to complete their flown trajectory. A run of the simulation stops when the TGT aircraft crosses the Merge point as this is the point where FIM must achieve its spacing goal behind it. The ASTAR12 algorithm is only meant to control the speeds of FIM until this point.

LED, TRL, TGT Simulated Flights

We simulate the actual flown trajectories of LED, TRL, and TGT via the following procedure:

1. Using $DTG_i = DTG(1)$ and $GS(1)$ from the $dRte_T$ and an increment of 1 second, we compute the actual distance-to-go of each aircraft after one second of trajectory flight time. This is $DTG_{act}(1)$.
2. Using $DTG_{act}(1)$, we interpolate each aircraft's current HW_T , alt , and CAS based on values from their $dRteT$ structure vectors.
3. From each aircraft's current flown HW_T , alt , and CAS , we compute its actual ground speed after 1 second of trajectory of flight time. This is $GS_{act}(1)$.
4. From $DTG_{act}(1)$, $GS_{act}(1)$, and an increment of 1 second, we compute the actual distance-to-go of each aircraft after 1 more second (2 seconds total) of trajectory flight time. This is $DTG_{act}(2)$.
5. Repeat steps 1-4 until the TGT aircraft's DTG_{act} is less than that of the Merge point.

FIM Simulated Flights via ASTAR12

We use a similar procedure to simulate the flight of the FIM aircraft by computing its DTG_{act} and GS_{act} in 1 second increments, but the difference lies in how we compute the FIM aircraft's actual CAS which is used in the GS_{act} and DTG_{act} calculation. We label this as CAS_{act} , and it is computed as

$$CAS_{act} = CAS_{nom} + u \tag{3.1}$$

where CAS_{nom} is interpolated from the FIM aircraft's $nomRte$ and u is the ASTAR12 speed control term. We use the $nomRte$ because in the current state of IM operations, the FIM aircraft only has knowledge of nominal published speeds. We will discuss changing this later. The ASTAR12 spacing algorithm computes the speed control term, u , as

$$u = kp \cdot \epsilon + kgs \cdot (TGT.GS_{act} - TGT.GS_{exp}) \quad (3.2)$$

where ϵ is the spacing error term, kp is the proportional gain term for the spacing error that changes based on how close FIM is to the Merge point, kgs is the ground speed compensation term, $TGT.GS_{act}$ is the actual ground speed of TGT computed using the previous steps, and $TGT.GS_{exp}$ is the expected ground speed of TGT by the ASTAR12 algorithm interpolated from TGT's *nomRte*. The spacing error term, ϵ , is the difference between the current time-based spacing of FIM and TGT and the desired time-based spacing of FIM and TGT at the Merge point. Each of the two individual terms in Equation 3.2 serves its own purpose. The first term is meant to gradually close the gap between FIM and TGT based on the difference between their current spacing and desired spacing. The second term is what makes ASTAR12 different from previous versions of the algorithm. This term is specifically meant to handle discrepancies between the TGT aircraft's nominal published speeds (which FIM only has access to) and the TGT aircraft's actual speeds as it flies according to its delayed trajectory. For this research, we decided to set the $kgs = 1$ due to undesired spacing error term values at certain points along the trajectory induced by other methods of computing kgs . For more information on this choice of kgs value, see Swieringa et al. (2015). For more detailed information on the ASTAR12 algorithm and its parameters, see Abbott (2015).

Under the current state of IM operations, the FIM aircraft only has knowledge of nominal published speed profiles for itself and TGT, and it also assumes that the TGT aircraft is experiencing the same winds rather than its own route-specific winds. If we want to change these, all we have to do is alter Equations 3.1 and 3.2 accordingly. To give the FIM aircraft knowledge of delayed trajectories, we change CAS_{nom} in Equation 3.1 to $CAS_{delayed}$ which is interpolated from the FIM aircraft's $dRte_F$ rather than its *nomRte*. We also need to change the way we compute $TGT.GS_{exp}$ in Equation 3.2. Rather than interpolated from the TGT aircraft's *nomRte*, we calculate its expected ground speed by interpolating from its $dRte_F$. To give the FIM aircraft knowledge of the TGT aircraft's route-specific winds, all we have

to do is use the TGT's HW_F rather than FIM's HW_F in the calculation of $TGT.GS_{exp}$ and ϵ . These are exactly the changes we make between experimental runs as the level of FIM knowledge is one of our independent variables of interest. We explain this further in the next section.

3.2 Experimental Design

In order to design an appropriate and efficient experiment to examine the effect that our factors of interest have on aircraft spacing violations during IM operations, we need to look at each factor individually and how best to manipulate them.

FIM Knowledge of Delayed Trajectories

In the current state of IM operations, the IM-equipped aircraft, FIM, assumes that all aircraft are flying their nominal trajectories without any delay. This can inevitably lead to spacing violations; if TGT is assigned a large amount of delay, it will arrive at the Merge point at a later time than what is assumed by ASTAR12 when controlling the speed of FIM. The simulation provides a means of determining whether implementing IM knowledge of delayed trajectories could lead to significantly fewer spacing violations. We will call this binary variable *IM_route_knowledge* and run the entire experiment at both levels since it is easy to change within the simulation.

FIM Knowledge of TGT Winds

Also within the current state of IM operations, FIM assumes that the TGT aircraft is encountering the same winds as FIM despite flying on a different route. This can also inevitably lead to spacing violations; if the TGT's route-specific wind profile consists of higher headwind speeds than FIM's route-specific wind profile, TGT could potentially arrive at the Merge point at a later time than what is assumed by ASTAR12 when controlling FIM. Using a computer simulation, we can see if implementing FIM knowledge of TGT route-specific winds is worth future research in terms of reducing spacing violations. We will call this

binary variable *IM_wind_knowledge* and run the entire experiment at both levels.

Wind Profiles

If IM technology and operations are going to be successfully incorporated into commercial air travel, it needs to be robust enough to handle a wide range of weather conditions. A simple and common way to distinguish between weather conditions is by winds. However, difficulty lies in deciding what specific metric(s) to use in manipulating wind profiles. Do we want certain headwind intensities at certain route locations such as the Meter Fix or Merge point? What about the difference between the *Truth* and *Forecast* wind profiles? For simplification, we decided to use eight *Forecast* and *Truth* wind profile sets for our two routes of interest that were used in a recent batch study (Swieringa, 2015). In the batch study, the eight wind sets were specifically chosen because they provided a range of the following metrics:

1. BOSS2 ETA Difference: This is the difference in nominal ETA at the FAF between the *Forecast* and *Truth* wind profiles for the BOSS2 arrival route into Denver International Airport. This is achieved via different headwind intensities between the two wind profiles throughout the route.
2. ANCHR2 ETA Difference: This is the same as above but for the ANCHR2 arrival route into Denver International Airport.
3. FAF HW Speed: This is specifically the difference in headwind intensity at the FAF between the *Forecast* and *Truth* wind profiles for both arrival routes.

Our *Winds* factor will be a categorical variable with eight levels; we will run the entire experiment at all levels. We label each *Winds* category as $Wind_1, Wind_2, \dots, Wind_8$.

Delay Differential Between FIM and LED, TRL, TGT

It has been observed in previous HITL simulation studies that different combinations of delay amounts between aircraft can lead to an increased rate of spacing violations and IM

operation disturbances (Swieringa et al., 2015). Assigning delay to an aircraft is an essential aspect of arrival scheduling and therefore, must be carefully examined when implementing IM technology. The majority of this research concerned how best to manipulate the delay differential variable; that is, the difference in delay between FIM and TGT, FIM and LED, and FIM and TRL. However, as has been shown in the previous section describing the simulation procedure, the delay that is assigned to an aircraft (other than the LED) is not something that we are at liberty to explicitly select. Instead, it is a quantity that is the result of a lengthy and complex delayed trajectory construction process that depends on the value we select for the following five quantities: LED DTG_i , LED delay, TGT DTG_i , FIM DTG_i , and TRL DTG_i . There is also an additional layer of difficulty in that these five continuous quantities are ‘nested’ in the sense that they are ordered; their bounds depend on the value selected for the previous quantity as well as the specific wind profiles. Figure 3 illustrates the space of the five nested factors, A , and how they relate to our delay differential space, B , through an unknown function, f as such:

$$f : A \rightarrow B$$

$$A \subset \mathbb{R}^5; \text{Nested factor space}$$

$$B \subset \mathbb{R}^3; \text{Delay differential space.}$$

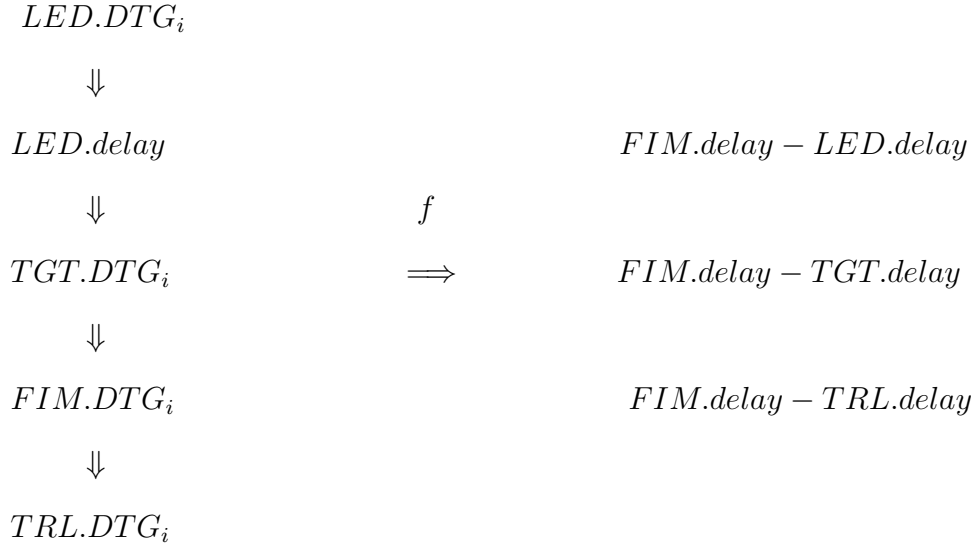


Fig. 3. The five nested factors (left) that result in the three delay differentials (right) through an unknown function f .

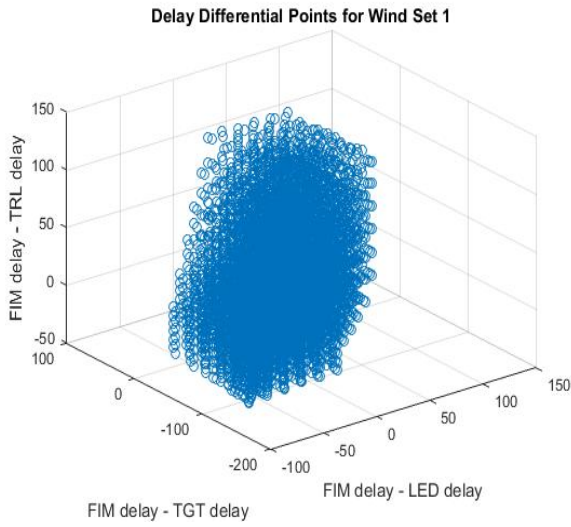
Our big question is: **How do we sample points** $a_i = (a_{i1}, a_{i2}, a_{i3}, a_{i4}, a_{i5})^T \in A$ **to get points** $b_i = (b_{i1}, b_{i2}, b_{i3})^T \in B$ **that efficiently fill** B ? We also need to keep in mind that the delay differential space is unique for each wind. What we need is an appropriate *space-filling design* for the delay differential factor since the entire continuous space, B , cannot be fully explored. As was shown in the Literature Review chapter, *space-filling designs* for deterministic computer experiments have been thoroughly studied as there are many techniques for creating good designs; however, little research has been done on the situation where the design space is dependent on continuous nested factors. The first natural thought is to enumerate A exhaustively up to a certain resolution (effectively *not* using a space-filling method). That is, we can pick every combination of the five nested factors up to a certain resolution and use the resulting points in the delay differential space, B . This ‘completely’ fills the space to a certain degree, but that many simulation runs (at least 10^5) will take too much time. Nonetheless, we can still use this method to visualize the space B for each wind set and get an idea of the boundaries of the space that we are trying to fill. This also provides us with some intuition on the nature of the f function from $A \rightarrow B$. The following 3-D plot (Figure 4 (a)) and 2-D projection plots (Figures 4 (b)-(d)) display a coarse enumeration of

the delay differential space for $Wind_1$ which we label B_1 . We only show plots for $Wind_1$ because they look very similar among all winds.

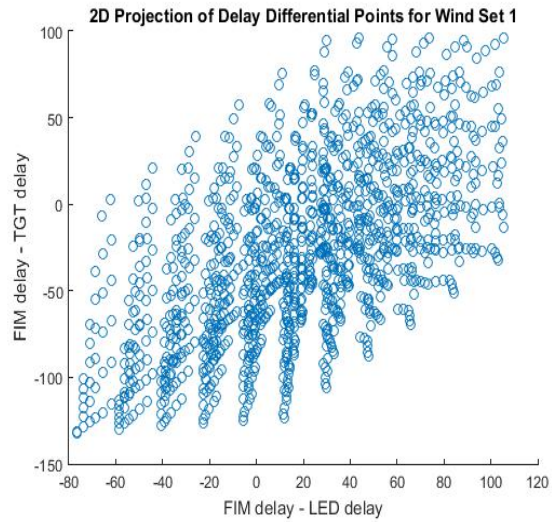
From the 3-D plot, we can see that some sections of the delay differential space are not explored by our enumeration. This is either because our enumeration was not performed at a coarse enough resolution, or some combinations of delay differentials are not possible within the simulation. From the 2-D projection plots, we see that a lot of unexplored space lies in combinations of extreme values of delay differentials near their maximums and minimums. Even though it seems likely that our simulation just can't provide us with certain extreme delay differential combinations, we would like to predict what would happen if we could test points there. This is the motivation for seeking a modeling technique that uses interpolation to create a predicted response surface based on a select group of tested points and their responses in order to see what a response would be at an extreme delay differential combination.

Since the simulation takes a fair amount of time to complete one run, we need a method for picking a small number of uniformly distributed delay differential points to run the experiment; then, we will use a *Gaussian process model* to interpolate the entire delay differential response surface from these select experimental runs. But how many runs is sufficient? We will use the informal rule that the number of runs in a computer experiment should be around 10 times the input dimension (Loeppky et al., 2009). Since our delay differential input is in three dimensions, we will use 30 runs.

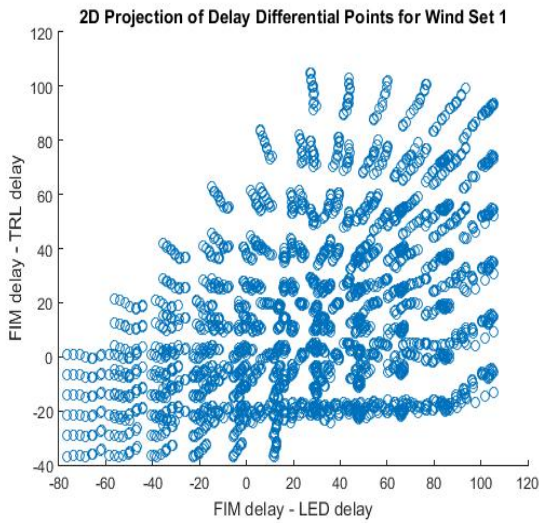
Now we need to come up with a space-filling method that samples 30 points from the nested factor space, A , that correspond with 30 points in the delay differential space, B , under the operation of f . Intuitively, it makes sense that the top factor in our five nested factor scheme, the $LED.DTG_i$, has a significant amount of influence on the bounds for the remaining four factors while its own bounds are only changed depending on the wind set. Therefore, we focus our efforts on finding the best way to select the value for this parameter in our pursuit of a good delay B -space-filling method. Below is a list of four space-filling



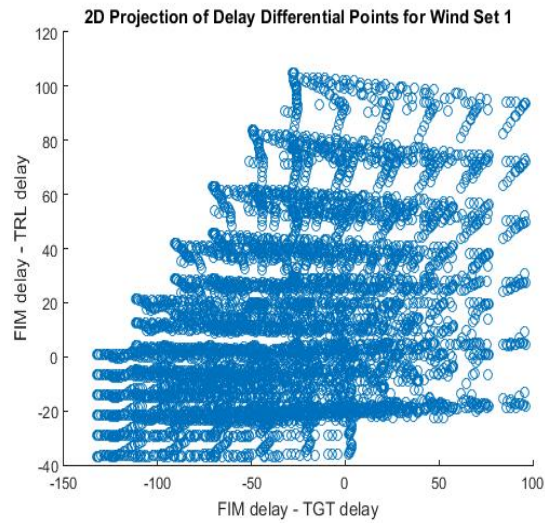
(a)



(b)



(c)



(d)

Fig. 4. An enumeration of the delay differential space for $Wind_1$ in three dimensions (a) and projected onto each two-dimensional plane (b)-(d). Delay differentials are in seconds.

methods that we proposed based on a different level of randomness in selecting the value for $LED.DTG_i$. We note that these four methods only differ in the procedure for choosing the value for this top level parameter.

1. *Random_Whole*: First, select a value for $LED.DTG_i$ randomly. Then, continue to select values for the remaining four nested factors randomly.
2. *Random_Half*: First, divide the $LED.DTG_i$ range in half (within each wind set). Then, pick a value randomly from the first half of $LED.DTG_i$ and proceed to pick values for the remaining factors randomly. This will be the sampling method for 15 of the 30 delay differential points. Repeat with the second half of $LED.DTG_i$ to get the other 15 delay differential points.
3. *Random_Third*: First, divide the $LED.DTG_i$ range in thirds (within each wind set). Then, pick a value randomly from the first third of $LED.DTG_i$ and proceed to pick values for the remaining factors randomly. This will be the sampling method for 10 of the 30 delay differential points. Repeat with the second third and last third of $LED.DTG_i$ to get the other 20 delay differential points.
4. *Not_Random*: First, select $LED.DTG_i$ to be its minimum (within each wind set). Then, select values for the remaining factors randomly; this is the first delay differential point. For the second delay differential point, select $LED.DTG_i$ to be $(1/29)$ within its range and proceed to pick values for the remaining factors randomly. Repeat while incrementing $LED.DTG_i$ by $(1/29)$ for each point. The procedure ends when the last delay differential point is selected by starting with $LED.DTG_i$ at its maximum (for that particular wind set).

In order to compare these methods and select the ‘best’ one in terms of sampling points that are uniformly spread out over the entire design region, we evaluated them using the *maximin criteria* for *space-filling designs*. This criteria seeks a space-filling method that

maximizes the minimum distance between every pair of design points. The *maximin criteria* seeks a space-filling method that maximizes the following:

$$\min_{b_i, b_j \in B; i \neq j} d(b_i, b_j) \quad (3.3)$$

where d is some measure of distance. For this research, we used the usual Euclidean measure of distance between two vectors in \mathbb{R}^3 :

$$d(b_i, b_j) = \|b_i - b_j\| = \sqrt{(b_{i1} - b_{j1})^2 + (b_{i2} - b_{j2})^2 + (b_{i3} - b_{j3})^2}. \quad (3.4)$$

In order to pick the best space-filling method, we randomly selected two wind sets, generated 1000 batches of 30 delay differential points using each of the four space-filling methods for both winds, computed the $\min d(b_i, b_j)$ for each of the 1000 batches, and compared the average $\min d(b_i, b_j)$ over the 1000 batches within each of the methods for both winds. If one method clearly performs well for both of the randomly selected wind sets in terms of the *maximin criteria*, we will assume that method will perform well for all eight wind sets. In Table 1, we present the average $\min d(b_i, b_j)$ for each of our original delay differential space-filling methods for $Wind_1$ and $Wind_3$.

Table 1. The average $\min d(b_i, b_j)$ over 1000 space-filling samples using each of our space-filling methods for two wind sets.

Space-filling Method	$Wind_1$	$Wind_3$
<i>Random_Whole</i>	3.0459	3.5897
<i>Random_Half</i>	3.0480	3.4723
<i>Random_Third</i>	3.2473	4.0475
<i>Not_Random</i>	2.9947	3.7805

From Table 1, we see that the *Random_Third* space-filling method performs the best for both wind sets according to the *maximin criteria*. Therefore, this will be our space-filling

method of choice for picking 30 delay differential test points within each wind set. In Figure 5, we plot the 30 chosen delay differential points for $Wind_1$ in three dimensions as well as each two-dimensional projection onto each delay differential plane.

If we compare these two-dimensional projection plots with those presented in Figure 4 of the entire delay differential space for $Wind_1$, we see that they share the same overall shape and the same unexplored regions of extreme combinations of delay differential values. The two-dimensional projection plots display delay differential points that are spread out uniformly to some degree. This was achieved by using the *maximin criteria* in selecting a good space-filling method. Responses from these 30 points will allow us to create a predicted response surface over the whole delay differential space for each wind set.

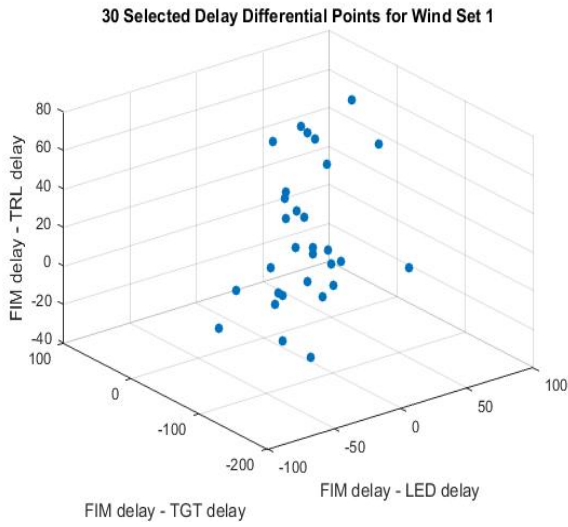
The experiment was repeated within each of the eight wind sets: for each wind, we ran the simulation at all 30 of its delay differential points crossed with the two FIM knowledge binary factors. Since the simulation is deterministic, there is no need for replicates. Therefore, we have 120 simulation runs per wind set, or 960 total runs.

3.3 Response Metrics

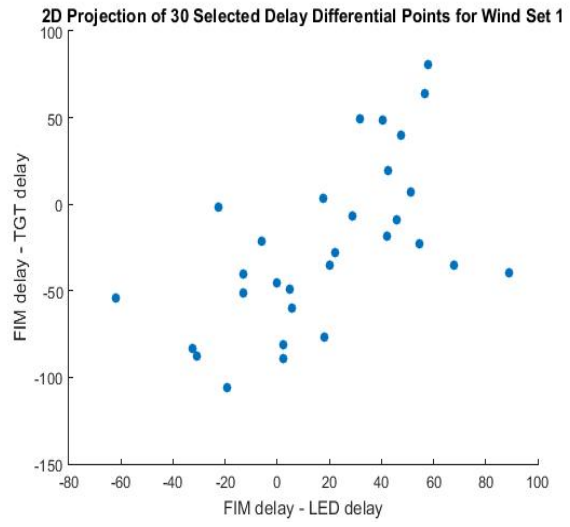
The goal of this research is to determine the initial airspace conditions that lead to separation violations between FIM and non-IM aircraft flying the same arrival route while FIM is following a TGT on a different arrival route. In order to answer this question completely, we need experimental responses that classify whether a separation violation has happened for a given run of the simulation. We are also interested in the severity of spacing violations when they do occur, as well as where along the FIM aircraft's trajectory they occur. The following is a discussion of our five chosen response metrics for this computer experiment.

Spacing Violation & Interruption of IM Operations

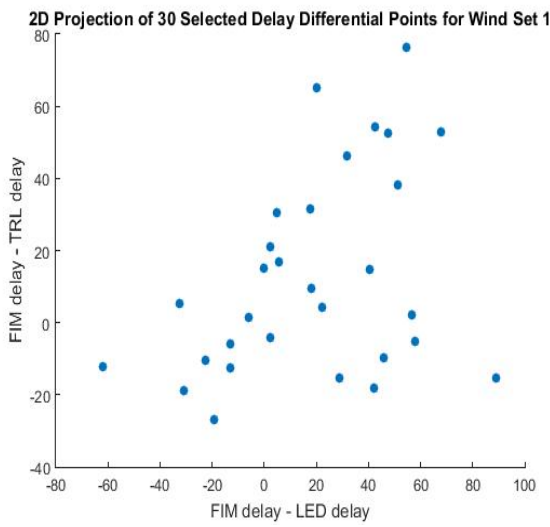
For a given run of the simulation (arrival of four modeled aircraft), we say a spacing violation has occurred when either the distance separation of FIM behind LED or TRL



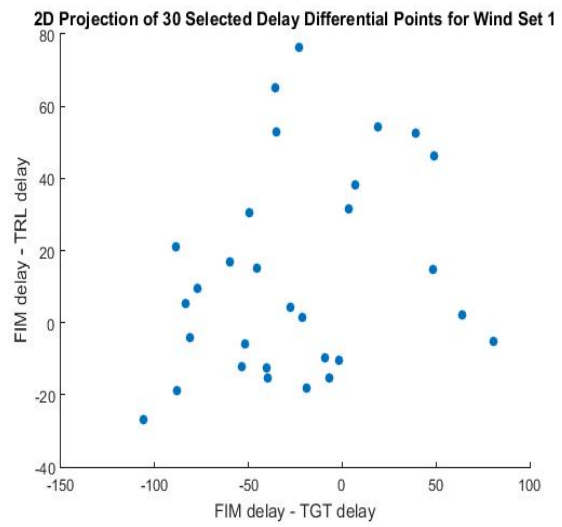
(a)



(b)



(c)



(d)

Fig. 5. The 30 selected delay differential points for $Wind_1$ in three dimensions (a) and projected onto each two-dimensional plane (b)-(d). Delay differentials are in seconds.

behind FIM breaches the minimum requirement. Similarly, we say that there has been an interruption of IM operations (a situation where ATC will take over the FIM arrival flight and IM operations will cease) when the distance separation of FIM behind LED or TRL behind FIM breaches a spacing buffer that is added on to the minimum spacing requirement. While FIM is in the Center Air Space, the minimum spacing requirement is 5 nmi with a buffer of 1 nmi. While FIM is in the TRACON the minimum spacing requirement is 3 nmi with a buffer of 0.1 nmi. We note that these distance requirements are different than those used in the scheduling procedure. This is because when an aircraft’s schedule is computed, ATC usually gives the aircraft some breathing room so as not to be scheduled at minimum spacings for the entirety of a flight. Based on our construction, an interruption has occurred when the spacing between FIM and LED or TRL breaches 6 nmi in Center Air Space or 3.1 nmi in the TRACON. We call these two binary response variables *Interrupt* and *SpaceViol*. For a given simulation run, *Interrupt* will be 1 if an IM interruption has occurred at any point during the modeled trajectory. The same holds for *SpaceViol*. We note that by construction, $Interrupt = 1$ whenever $SpaceViol = 1$; this is accurate since one is a more severe version of the other.

Closest Point of Approach (CPA)

In addition to determining whether a spacing violation or IM interruption has occurred given a particular set of airspace conditions, we want to know how severe the loss of appropriate spacing was. We can measure this by computing the Closest Point of Approach (CPA); that is, how close did FIM fly to LED and TRL over the course of an entire flight. The natural way to do this would be to take the smaller of the following two quantities: the minimum flown spacing between FIM and LED and the minimum flown spacing between FIM and TRL. However, a distance spacing of 4 nmi is worse if it occurs in Center Air Space as opposed to the TRACON portion of the trajectory due to the change in minimum spacing and buffer requirements between these two route segments. Therefore, we will determine the CPA by computing the minimum difference between the FIM aircraft’s actual spacing and

its required minimum spacing over the entire trajectory. In the following Pseudocode 3 for determining the *CPA* response value for a given simulation run, the function *minspace()* takes as input a DTG and returns the minimum required spacing at that DTG according to our scheme described previously.

for each recorded second i of a particular simulation run

$$FIMspaceTRL(i) = |FIM.DTG_{act}(i) - TRL.DTG_{act}(i)| - minspace(FIM.DTG_{act}(i))$$

$$FIMspaceLED(i) = |FIM.DTG_{act}(i) - LED.DTG_{act}(i)| - minspace(FIM.DTG_{act}(i))$$

end

$$CPA = \min\left(\min_{i \in \{1, 2, \dots\}} FIMspaceTRL(i), \min_{i \in \{1, 2, \dots\}} FIMspaceLED(i)\right)$$

Pseudocode 3: How to compute the *CPA* response for a given simulation run.

By construction, the continuous *CPA* response can take on negative and positive values. A negative *CPA* corresponds to a spacing violation, and how negative this *CPA* is gives us a measure of the severity of the loss of separation. We look to maximize the *CPA* response in the statistical modeling and analysis of the data.

DTG of IM Interruption & Spacing Violation

In addition to the occurrence and severity of spacing violations and IM interruptions, we also want to model where along the route these distance separation losses occur. Do more spacing violations and IM interruptions occur in the TRACON or Center Air Space? At the FAF (Merge waypoint) or near where the aircraft start their trajectory? To measure this, we create two additional continuous responses: *Interrupt_DTG* and *SpaceViol_DTG*. They are simply the FIM DTG where spacing violations and IM interruptions first occur. When a spacing violation or IM interruption does not occur during a given run of the simulation, these response variables will have a missing value. We note that these two responses are strictly positive as a DTG cannot be negative by definition; we will need to keep this in mind in order to build accurate predictive models. Table 2 summarizes our factors/responses.

Table 2. Variable types and descriptions for computer experiment.

Variable Name	Type	Description
<i>IM_wind_know</i>	Binary factor	Whether FIM assumes TGT is experiencing its own route-specific wind profile (1) or not (0)
<i>IM_route_know</i>	Binary factor	Whether FIM assumes aircraft are flying delayed trajectories (1) or nominal trajectories (0)
<i>Winds</i>	Categorical factor	8 sets of ANCHR2 and BOSS2 route wind profiles
<i>LED_delaydiff</i>	Continuous factor	Difference between FIM delay and LED delay (sec)
<i>TGT_delaydiff</i>	Continuous factor	Difference between FIM delay and TGT delay (sec)
<i>TRL_delaydiff</i>	Continuous factor	Difference between FIM delay and TRL delay (sec)
<i>Interrupt</i>	Binary response	Whether an IM interruption occurs during simulated flight (1) or not (0)
<i>SpaceViol</i>	Binary response	Whether a spacing violation occurs during simulated flight (1) or not (0)
<i>CPA</i>	Continuous response	Closest point of approach between FIM and surrounding aircraft (nmi)
<i>Interrupt_DTG</i>	Continuous (> 0) response	DTG along route where IM interruption occurs (nmi)
<i>SpaceViol_DTG</i>	Continuous (> 0) response	DTG along route where spacing violation occurs (nmi)

3.4 Modeling

In this section, we describe our procedure for analyzing this data set. Our methods are dictateded by our main research questions which we restate below:

1. To what extent do the following four factors contribute to the interruption of IM operations as well as spacing violations when FIM and TGT are flying different routes?
 - (a) Winds
 - (b) Difference in delay assignments between FIM and non-IM aircraft including TGT
 - (c) FIM aircraft knowledge of delayed trajectories
 - (d) FIM knowledge of TGT route winds
2. Given a set of initial airspace/aircraft conditions expressed in the factors above, what is the probability that IM operations will need to be interrupted or a spacing violation will occur at some point along the trajectory?
3. Given a set of initial airspace/aircraft conditions expressed in the factors above, what is the Closest Point of Approach (CPA) between FIM and its surrounding non-IM aircraft, and where along the route do losses of appropriate separation occur?

In order to answer these questions, we need to carefully consider our experimental design and each of its individual pieces. Our experimental design has many elements that each need to be considered when building predictive models from the data set:

- Within each wind set we have a 2^k full factorial design with $k = 2$ in the binary factors *IM_route_knowledge* and *IM_wind_knowledge*.
- The entire experiment is repeated within each of the eight wind sets; therefore, we can think of the *Winds* factor as a ‘blocking’ factor. However, this is not true blocking in the traditional sense because there is no random component whatsoever in the

experiment that we are attempting to control for by repeating the experiment. If we think of the computer experiment as emulating a real-time physical experiment, then the *Winds* factor can account for day-to-day changes in environmental conditions when an aircraft arrives under IM operations. We have just deterministically quantified this wind component within the computer experiment.

- The three delay differential factors are sampled from a space-filling design originating from five continuous nested factors. When we talk about nested factors, we usually mean that a discrete factor has different (explicit and knowable) levels depending on the level of a different factor. However, in our case, we have repeated nesting for five continuous factors where we do not know their specific continuous ranges and how explicitly they depend on the other factors.
- Our responses vary in type: positive and negative continuous, strictly positive continuous, and binary. We need to consider these differences when modeling and predicting the responses.

Our strategy is to build a separate model for each of the responses versus all of the factors (and some selected interactions) as well as *Gaussian stochastic process* models for the continuous responses versus the three delay differential factors. Here we describe each of the models used in detail before moving on to the Results chapter.

Interrupt & SpaceViol vs. All Factors

For each of our two binary responses, *SpaceViol* and *Interrupt*, we will build a logistic regression model via the logit link function against all of our factors with some additional interaction terms. This allows us to predict the probability of a spacing violation or IM interruption given certain factor values. The model we are trying to estimate is

$$\vec{Y} = \frac{1}{1 + \exp(-\vec{\beta}\mathbf{X}) + \vec{\epsilon}} \tag{3.5}$$

where \vec{Y} is the *SpaceViol* or *Interrupt* response vector, \mathbf{X} is the data matrix, $\vec{\beta}$ is the logistic regression coefficients vector, and $\vec{\epsilon}$ is the error vector term. In the model expressed in Equation 3.5, we will estimate interaction coefficients for the interactions between *IM_route_knowledge*, *IM_wind_knowledge*, and *Winds* because it is suspected that different weather conditions might have influence over the effectiveness of advancements in IM operations. Also, it is logical to suspect some interaction between the two different types of theoretical IM technology advancements that we modeled and whether having both of them operating together could lead to fewer spacing violations than if either of them were implemented in isolation. We note that because of the way we decided to code the *Winds* categorical variable via seven binary indicator variables, all of the indicators equaling zero corresponds to *Wind*₁.

CPA vs. All Factors

For our continuous response, *CPA*, we will use a multiple linear regression model against all of our factors plus the same interaction terms described previously. In this model, we also code the categorical *Winds* variable in the same manner as the logistic regression models. Since the *CPA* response can be positive or negative, we do not need to make any special considerations. The model we aim to estimate is

$$\vec{Y} = \vec{\beta}\mathbf{X} + \vec{\epsilon} \tag{3.6}$$

where \vec{Y} is the *CPA* response vector, \mathbf{X} is the data matrix, $\vec{\beta}$ is the regression coefficients vector, and $\vec{\epsilon}$ is the error vector term.

Interrupt_DTG & SpaceViol_DTG vs. All Factors

For each of our two strictly positive and continuous responses, *Interrupt_DTG* and *SpaceViol_DTG*, we will use a Gamma regression model against all of our factors plus the same interaction terms described previously. When we say ‘Gamma regression’, we mean a generalized linear model (GLM) with a log link function; this type of GLM is a common

method for handling a strictly positive and continuous skewed right response. Here we will present a brief mathematical overview of the GLM with a canonical link function.

Let us assume that we have a response vector, \vec{Y} , that has a distribution belonging to the Exponential Family of Distributions (EFD). If we assume that \vec{Y} can be modeled as the process mean, $\vec{\mu}$, plus an additional error term, $\vec{\epsilon}$, where $\vec{\mu}$ is related to our data matrix of regressors, \mathbf{X} , via a link function, g , then we have

$$E(\vec{Y}|\mathbf{X}) = \vec{\mu}$$

$$\vec{Y} = \vec{\mu} + \vec{\epsilon} \tag{3.7}$$

$$g(\vec{\mu}) = \mathbf{X}\vec{\beta} \tag{3.8}$$

$$\vec{Y}, \vec{\epsilon} \sim \text{EFD}$$

and after solving for $\vec{\mu}$ in Equation 3.8, we can substitute it into Equation 3.7 as

$$\vec{\mu} = g^{-1}(\mathbf{X}\vec{\beta}) \tag{3.9}$$

$$\vec{Y} = g^{-1}(\mathbf{X}\vec{\beta}) + \vec{\epsilon} \tag{3.10}$$

and proceed. If \vec{Y} follows a Gamma distribution, then the link function g is the log function.

Therefore, we can use Equations 3.8, 3.9, and 3.10 to derive

$$g(\vec{\mu}) = \log(\vec{\mu}) = \mathbf{X}\vec{\beta} \tag{3.11}$$

$$\vec{\mu} = g^{-1}(\mathbf{X}\vec{\beta}) = e^{\mathbf{X}\vec{\beta}} \tag{3.12}$$

$$\vec{Y} = e^{\mathbf{X}\vec{\beta}} + \vec{\epsilon} \tag{3.13}$$

which is the Gamma regression model. Using this GLM with a log link function and assuming that our *Interrupt_DTG* and *SpaceViol_DTG* responses follow a Gamma distribution, we

can fit a Gamma regression model to predict these positive continuous responses.

CPA vs Wind Delay Differentials via Gaussian Process Model

Since our three-dimensional delay differential space, B , is continuous and the simulation is computationally intensive, our strategy was to select a small group of 30 delay differential points within each wind set to run the experiment via a space-filling sampling method. From these 30 select points and their recorded CPA responses, we want to estimate the CPA response at many untested delay differential points to form an estimated CPA response surface. In the following formulation for estimating the CPA at untested delay differential points, we use B even though in practice, we must specify the particular delay differential space B_1, B_2, \dots, B_8 for a specific wind set $Wind_1, Wind_2, \dots, Wind_8$.

The following formulation follows Do (2007) and MacDonald et al. (2015). Let y be a function from our delay differential space B to our CPA response space C as such:

$$y : B \rightarrow C,$$

$$B \subset \mathbb{R}^3,$$

$$C \subset \mathbb{R}.$$

For $x_1, x_2, \dots, x_{30} \in B$ where $x_i = (x_{i1}, x_{i2}, x_{i3})^T$ we know $y(x_1), y(x_2), \dots, y(x_{30})$. But what about $y(x^*)$ for some untested $x^* \in B$? We can accomplish this via *Gaussian stochastic process regression*.

A *stochastic process* is a set of random variables $\{h(x) : x \in B\}$ indexed by elements from an index set B . Let B here be our delay differential space as before. A *Gaussian process* is a *stochastic process* such that any finite subcollection of random variables has a multivariate Gaussian distribution. That is, a finite collection of random variables $\{h(x) : x \in B\}$ is distributed as a *Gaussian process* with mean function $m(\cdot)$ and covariance function $k(\cdot, \cdot)$ if for any set of points $x_1, x_2, \dots, x_n \in B$, the set of random variables $h(x_1), h(x_2), \dots, h(x_n)$

has the following distribution:

$$\begin{bmatrix} h(x_1) \\ \vdots \\ h(x_n) \end{bmatrix} \sim N \left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{bmatrix} \right),$$

which we can also write as $h(\cdot) \sim GP(m(\cdot), k(\cdot, \cdot))$. Now are there any restrictions on the mean and covariance functions? For $m(\cdot)$, no. But $k(\cdot, \cdot)$ needs to be such that the covariance matrix of any collection of points $x_1, x_2, \dots, x_n \in B$ must be positive semidefinite.

To relate this back to our problem, we can model the *CPA* deterministic response as a realization of the following stochastic process:

$$y(x_i) = \mu + h(x_i) \tag{3.14}$$

$$i = 1, 2, \dots, 30$$

where x_i is one of our 30 delay differential points, $h(\cdot) \sim GP(m(\cdot), k(\cdot, \cdot))$ as defined previously, and μ is the true overall mean. For simplification, let's assume $m(h(x_i)) = E(h(x_i)) = 0$. This is a common assumption when using a Gaussian process model, and it makes sense in our case since *CPA* values are centered around 0 by construction. We will also assume that $k(h(x_i), h(x_i)) = Var(h(x_i)) = \sigma^2$ and $k(h(x_i), h(x_j)) = Cov(h(x_i), h(x_j)) = \sigma^2 R_{ij}$. If we make a 30 x 30 data matrix \mathbf{X} out of our 30 delay differential points $x_1, x_2, \dots, x_{30} \in B$ then $Y = y(\mathbf{X}) = (y(x_1), y(x_2), \dots, y(x_{30}))^T$ as the vector of 30 *CPA* responses has the following multivariate Gaussian distribution:

$$Y = y(\mathbf{X}) \sim N_{30}(\mathbf{J}_{30}\mu, \sigma^2 R)$$

where \mathbf{J}_{30} is a 30 x 1 vector of one's and R is a correlation matrix with elements R_{ij} . While there are many choices of correlation matrix R , we will choose the *Gaussian correlation*

function, or the *squared-exponential correlation* function, given by

$$R_{ij} = \prod_{k=1}^d \exp\{-\theta_k |x_{ik} - x_{jk}|^2\} \quad (3.15)$$

where in our case, $d = 3$ and $\theta = (\theta_1, \theta_2, \theta_3)$ is a vector of parameters that need estimation. We use this common correlation structure in Equation 3.15 since it has desirable properties. It is locally smooth, and nearby x_i 's will have a correlation ≈ 1 while far apart x_i 's will have a correlation ≈ 0 . Using the fact that $Y \sim N_{30}(\mathbf{J}_{30}\mu, \sigma^2 R)$, we can derive the Maximum Likelihood Estimators (MLE)

$$\hat{\mu}(\theta) = (\mathbf{J}_{30}^T R^{-1} \mathbf{J}_{30})^{-1} (\mathbf{J}_{30}^T R^{-1} Y) \quad (3.16)$$

$$\hat{\sigma}^2(\theta) = \frac{1}{30} \cdot (Y - \mathbf{J}_{30} \hat{\mu}(\theta))^T R^{-1} (Y - \mathbf{J}_{30} \hat{\mu}(\theta)) \quad (3.17)$$

for μ and σ^2 . The MLE estimates in Equations 3.16 and 3.17 are used to obtain the following negative log-likelihood function, or *deviance*:

$$-2 \log(L_\theta) \propto \log(|R|) + 30 \log[(Y - \mathbf{J}_{30} \hat{\mu}(\theta))^T R^{-1} (Y - \mathbf{J}_{30} \hat{\mu}(\theta))] \quad (3.18)$$

for estimating our θ parameters. Using the MLE technique again along with the estimates in Equations 3.16-18, we can obtain the best linear unbiased estimator (BLUE) of the *CPA* at an untested $x^* \in B$ and its mean squared error (MSE) as

$$\hat{y}(x^*) = \hat{\mu} + r^T R^{-1} (Y - \mathbf{J}_{30} \hat{\mu}) = \left[\frac{(1 - r^T R^{-1} \mathbf{J}_{30})}{\mathbf{J}_{30}^T R^{-1} \mathbf{J}_{30}} \mathbf{J}_{30}^T + r^T \right] R^{-1} Y = Z^T Y, \quad (3.19)$$

$$\begin{aligned} MSE(\hat{y}(x^*)) &= E[(\hat{y}(x^*) - y(x^*))^2] \\ &= \hat{\sigma}^2 (1 - 2Z^T r + Z^T R Z) = \hat{\sigma}^2 \left[1 - r^T R^{-1} r + \frac{(1 - \mathbf{J}_{30}^T R^{-1} r)^2}{\mathbf{J}_{30} R^{-1} \mathbf{J}_{30}} \right] \end{aligned} \quad (3.20)$$

where $r = (r_1(x^*), r_2(x^*), \dots, r_{30}(x^*))$ and $r_i(x^*) = \text{corr}(h(x^*), h(x_i))$. This is how we will estimate the *CPA* response at untested delay differential points and create an estimated *CPA* response surface within each wind set.

CHAPTER 4

RESULTS

4.1 Response Visualizations

Before we present the results from fitting each of our models, we need to examine and visualize the data. It is helpful to investigate what the responses look like before building good models to predict them. We will start with the *Interrupt* and *SpaceViol* responses. In Figure 6, we present the percentage of spacing violations ($SpaceViol = 1$) and IM interruptions ($Interrupt = 1$) for each wind set across all levels of IM knowledge.

After a quick glance, it is clear that there is a lot of variation between winds in terms of spacing violations and IM interruptions. The ‘worst’ wind set, $Wind_6$ has 55% IM interruptions and 33% spacing violations. On the other hand, the ‘best’ wind set, $Wind_8$, has only 8% IM interruptions and 7% spacing violations. So even if IM operations are conducted under severe problem-inducing wind conditions, we could potentially only see spacing violations happening about a third of the time. Now let us look at the percentage of spacing violations and IM interruptions across different levels of IM knowledge. In Figure 7, we present the percentage of IM interruptions and spacing violations for the four different combinations of our two IM knowledge variables, IM_route_know and IM_wind_know , across all eight wind sets.

From the bar graph, we can see that the smallest percentage of IM interruptions and spacing violations occurs when IM has maximum knowledge of delayed trajectories and winds while the largest percentage occurs when IM has minimal knowledge of such information. This is what we should expect. However, the most interesting aspect of this plot is that it reveals that implementing IM knowledge of TGT route-specific winds hardly seems to reduce the percentage of spacing violations and interruptions at all. On the other hand,

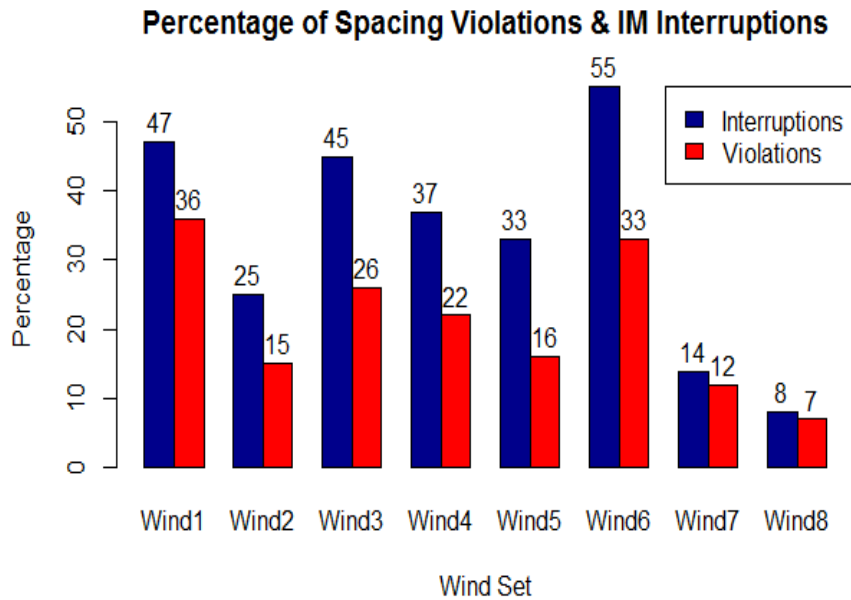


Fig. 6. Percentage of Spacing Violations and IM Interruptions for each wind set across all levels of IM knowledge.

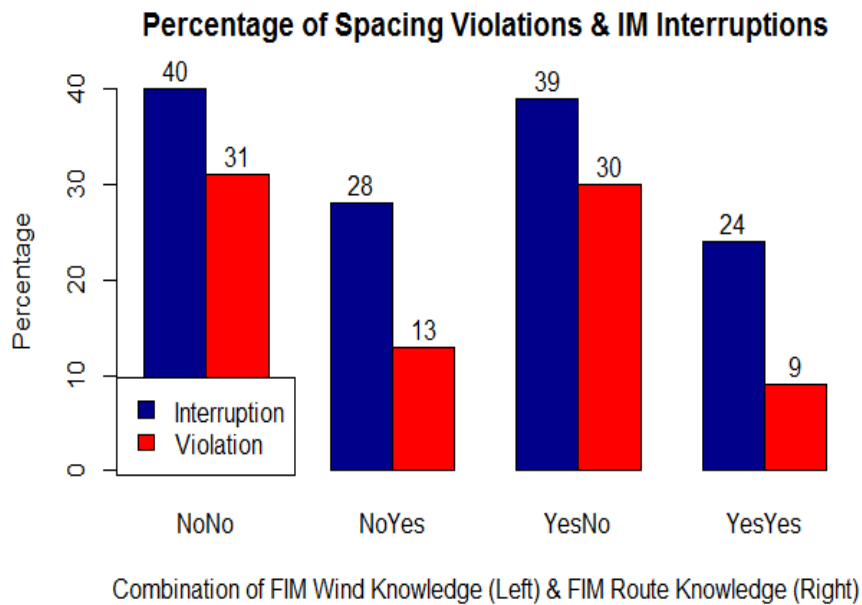


Fig. 7. Percentage of Spacing Violations and IM Interruptions for different levels/combinations of IM knowledge across all wind sets.

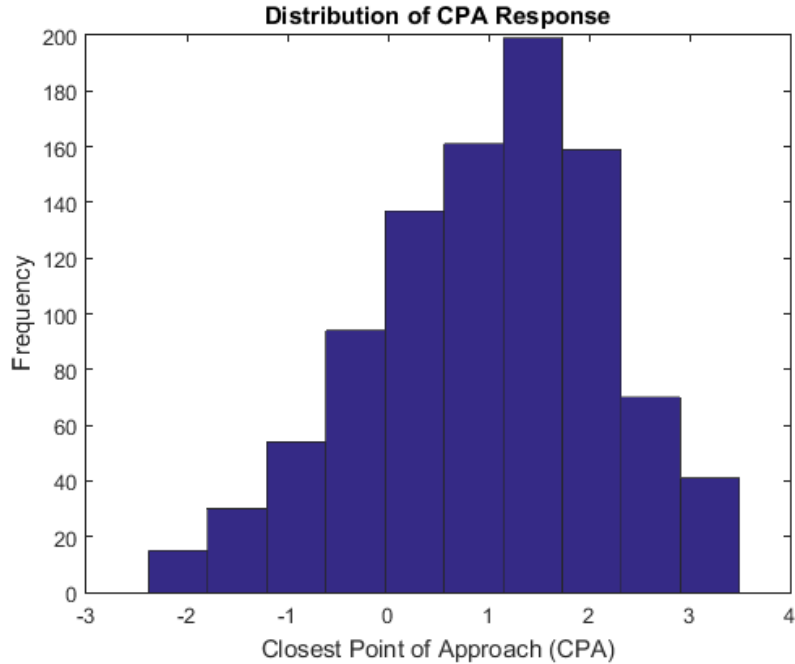


Fig. 8. Distribution of CPA response over the entire data set. CPA is in nmi.

implementing IM knowledge of delayed trajectories seems to reduce the percentage of IM interruptions and spacing violations by a decent amount. As for why it appears that giving the IM aircraft full wind knowledge of the TGT’s route does not lessen the occurrence of spacing violations and interruptions by a large amount, it could be that for each wind set, the wind profiles for the ANCHR2 and BOSS2 routes are not that different. The eight wind sets were chosen because they specifically displayed variation within the same arrival route and not necessarily variation between the two different arrival routes.

Now we will move on to the *CPA*. We recall that the *CPA* is a measure of the closest the FIM aircraft gets to its surrounding aircraft throughout its entire trajectory. The *CPA* for a given run can be positive or negative; a negative *CPA* signifies a spacing violation, and how negative the *CPA* becomes is a measure of the severity of a spacing violation. In the Figure 8 histogram, we display the distribution of *CPA* values for the entire data set.

The histogram appears to be slightly skewed-left with most *CPA* values leaning in the positive direction and the largest frequency of *CPA*’s being around +2 nmi. This is good;

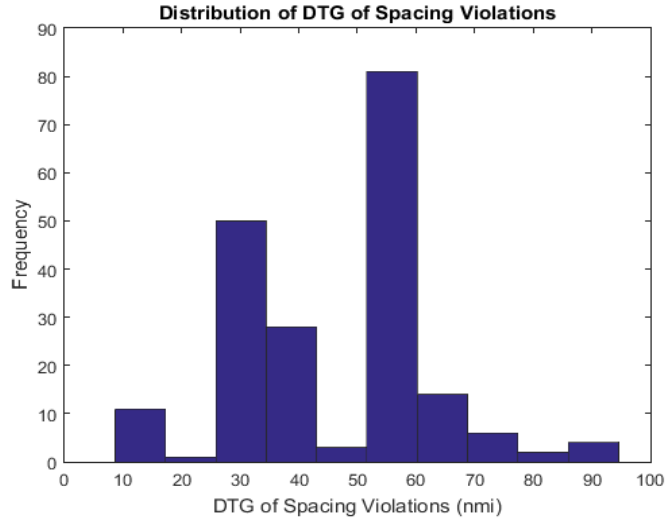


Fig. 9. Distribution of the DTG of spacing violations where they first occur across the entire data set.

more positive CPA values correspond with fewer spacing violations. We also see that the smallest CPA values are around -2 nmi; this means that the most severe spacing violations we observed were 2 nmi from the threshold of a spacing violation. Now let's look at the distributions of the $Interrupt_DTG$ and $SpaceViol_DTG$ responses. We recall that these responses are simply the DTG where spacing violations and interruptions occur when they do, in fact, occur. In Figures 9 and 10, we display the distributions of the $SpaceViol_DTG$ and $Interrupt_DTG$ responses, respectively.

As for commonalities between the two histograms, we see that most IM interruptions and spacing violations occur around 50-60 nmi. This is exactly where the Meter Fix is located: 54.5 nmi from the runway. So a lot of spacing violations and interruptions are occurring at the transition between the Center Air Space and TRACON portions of the route. This could be due to the change in minimum spacing requirements at this location, but it needs to be further investigated. As for differences between the two histograms, we see that the majority of interruptions occur at the Meter Fix and further away from the runway within the Center Air Space. This is what we want; interruptions of IM operations are less detrimental if they occur further away from the runway. On the other hand, the

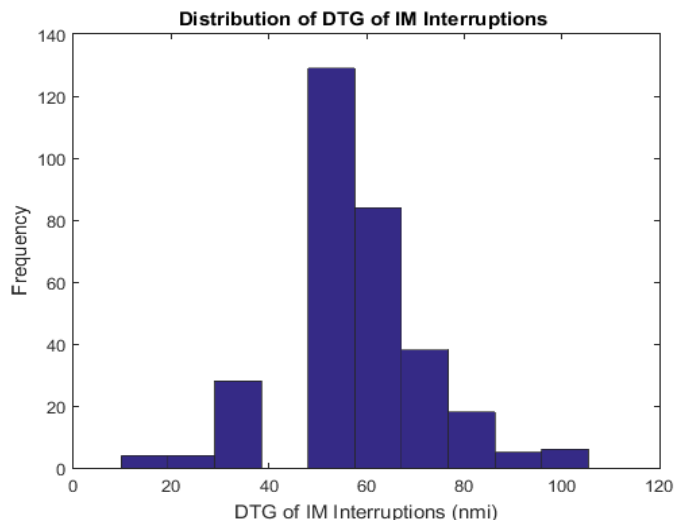


Fig. 10. Distribution of the DTG of IM interruptions where they first occur across the entire data set.

majority of spacing violations occur at the Meter fix and closer to the runway within the TRACON. It is rather curious that there is a spike in spacing violations around the 30-40 nmi DTG mark. There is nothing particularly notable about this section of the arrival route, and further investigation into this is needed.

4.2 Model Results

Here we present results from fitting our chosen models. Each model will be discussed in the order in which it was described in the previous section. All models were fit in R.

Interrupt & SpaceViol vs. All Factors

In Tables 3 and 4, we present the results from fitting a logistic regression model of the *Interrupt* and *SpaceViol* binary responses versus all of our factors plus some additional interaction terms. Residual analysis revealed no issues with model appropriateness.

Table 3.: Logistic regression model fitting results for *Interrupt* response. * Significant at 0.05 level.

Parameter	Estimate	Std. Error	z value	p-value
Intercept	3.179881	0.630718	5.042	4.61e-07
<i>LED_delaydiff</i>	0.036239	0.004986	7.268	3.65e-13*
<i>TRL_delaydiff</i>	-0.154837	0.011593	-13.356	< 2e-16*
<i>TGT_delaydiff</i>	0.023196	0.003584	6.472	9.67e-11*
<i>IM_route_know</i>	-1.443114	0.693488	-2.081	0.037438*
<i>IM_wind_know</i>	-1.856722	0.697845	-2.661	0.007799*
<i>Wind</i> ₂	-3.430612	0.814757	-4.211	2.55e-05*
<i>Wind</i> ₃	-0.048446	0.746174	-0.065	0.948233
<i>Wind</i> ₄	-2.658361	0.783179	-3.394	0.000688*
<i>Wind</i> ₅	-1.723421	0.799286	-2.156	0.031068*
<i>Wind</i> ₆	-1.241801	0.722211	-1.719	0.085534
<i>Wind</i> ₇	-4.144569	0.853671	-4.855	1.20e-06*
<i>Wind</i> ₈	-4.905486	0.888976	-5.518	3.43e-08*
<i>IM_route_know</i> * <i>IM_wind_know</i>	-0.420947	0.452858	-0.930	0.352612
<i>IM_route_know</i> * <i>Wind</i> ₂	-0.268830	0.901965	-0.298	0.765666
<i>IM_route_know</i> * <i>Wind</i> ₃	0.353213	0.863922	0.409	0.682651
<i>IM_route_know</i> * <i>Wind</i> ₄	1.651594	0.885987	1.864	0.062304
<i>IM_route_know</i> * <i>Wind</i> ₅	-0.643549	0.907961	-0.709	0.478458
<i>IM_route_know</i> * <i>Wind</i> ₆	0.616019	0.828238	0.744	0.457015
<i>IM_route_know</i> * <i>Wind</i> ₇	-1.715509	1.098834	-1.561	0.118475
<i>IM_route_know</i> * <i>Wind</i> ₈	-0.098703	1.125958	-0.088	0.930146
<i>IM_wind_know</i> * <i>Wind</i> ₂	2.056785	0.913022	2.253	0.024277*

$IM_wind_know * Wind_3$	3.367569	0.891042	3.779	0.000157*
$IM_wind_know * Wind_4$	1.364708	0.885389	1.541	0.123228
$IM_wind_know * Wind_5$	2.070839	0.918646	2.254	0.024182*
$IM_wind_know * Wind_6$	1.282937	0.832636	1.541	0.123362
$IM_wind_know * Wind_7$	2.240123	1.011905	2.214	0.026845*
$IM_wind_know * Wind_8$	1.635515	1.063693	1.538	0.124151

Table 4.: Logistic regression model fitting results for *SpaceViol* response. * Significant at 0.05 level.

Parameter	Estimate	Std. Error	z value	p-value
Intercept	1.305295	0.525184	2.485	0.012940
$LED_delaydiff$	0.039814	0.005080	7.837	4.63e-15*
$TRL_delaydiff$	-0.114018	0.009962	-11.445	< 2e-16*
$TGT_delaydiff$	0.011358	0.003213	3.535	0.000408*
IM_route_know	-1.327660	0.635843	-2.088	0.036795*
IM_wind_know	-1.653638	0.642265	-2.575	0.010033*
$Wind_2$	-2.493482	0.736835	-3.384	0.000714*
$Wind_3$	-0.438058	0.674198	-0.650	0.515855
$Wind_4$	-2.497553	0.714758	-3.494	0.000475*
$Wind_5$	-2.082686	0.744518	-2.797	0.005152*
$Wind_6$	-0.419932	0.653881	-0.642	0.520734
$Wind_7$	-2.808094	0.766564	-3.663	0.000249*
$Wind_8$	-3.588923	0.830629	-4.321	1.56e-05*
$IM_route_know * IM_wind_know$	-0.527580	0.484676	-1.089	0.276366
$IM_route_know * Wind_2$	-1.444940	1.029008	-1.404	0.160257

$IM_route_know * Wind_3$	0.318944	0.804739	0.396	0.691860
$IM_route_know * Wind_4$	0.418236	0.836330	0.500	0.617015
$IM_route_know * Wind_5$	-1.700625	1.022739	-1.663	0.096350
$IM_route_know * Wind_6$	-2.438763	0.863475	-2.824	0.004738*
$IM_route_know * Wind_7$	-1.062866	1.056562	-1.006	0.314432
$IM_route_know * Wind_8$	0.198047	1.083742	0.183	0.854999
$IM_wind_know * Wind_2$	1.762217	0.919720	1.916	0.055360
$IM_wind_know * Wind_3$	2.035426	0.813325	2.503	0.012329*
$IM_wind_know * Wind_4$	2.055912	0.843917	2.436	0.014844*
$IM_wind_know * Wind_5$	1.528806	0.937901	1.630	0.103096
$IM_wind_know * Wind_6$	1.194206	0.823986	1.449	0.147253
$IM_wind_know * Wind_7$	1.773520	0.957865	1.852	0.064093
$IM_wind_know * Wind_8$	1.806098	1.023390	1.765	0.077594

We see from the results tables that all three delay differentials are significant when predicting the probability of a spacing violation and IM interruption. The negative coefficient estimate for $TRL_delaydiff$ means that as FIM's delay increases above TRL's delay, the likelihood of an IM interruption or spacing violation decreases. However, the reverse relationship applies to LED and TGT. As FIM's delay increases above LED's delay and TGT's delay, the likelihood of an IM interruption or spacing violation increases. We also see that both the IM route and wind knowledge lead to significantly smaller probabilities of IM interruptions and spacing violations, but their interaction effect is not significant. Winds seem to have large significant effects on spacing violations and IM interruptions; some winds also appear to have a significant positive interaction coefficient estimate with FIM's wind knowledge. It is rather interesting that some wind sets coupled with FIM having detailed wind knowledge can actually be detrimental in terms of the likelihood of spacing violations and IM interruptions. This might be because for a given wind set, the wind profile for

FIM's arrival route could be less spacing-violation-inducing than the wind profile for TGT's arrival route. If this is the case, then by FIM actually assuming that TGT is encountering FIM's winds rather than its own, less spacing violations and IM interruptions occur. This relationship needs to be further investigated on a case-by-case basis for a given wind set. It is clear from the table that in general, spacing violations and IM interruptions are sensitive to winds.

CPA vs. All Factors

In Table 5, we present the results from fitting a multiple linear regression model of the continuous *CPA* response versus all of our factors plus some additional interaction terms. Residual analysis revealed no issues with model appropriateness.

Table 5.: Multiple linear regression model fitting results for *CPA* response. * Significant at 0.05 level.

Parameter	Estimate	Std. Error	z value	p-value
Intercept	-0.5442257	0.1196017	-4.550	6.06e-06
<i>LED_delaydiff</i>	-0.0105758	0.0008983	-11.773	< 2e-16*
<i>TRL_delaydiff</i>	0.0320669	0.0010395	30.849	< 2e-16*
<i>TGTdelay_diff</i>	-0.0050447	0.0006649	-7.587	7.93e-14*
<i>IM_route_know</i>	1.0162455	0.1386704	7.328	5.04e-13*
<i>IM_wind_know</i>	0.5928722	0.1386704	4.275	2.10e-05*
<i>Wind</i> ₂	1.0106198	0.1603153	6.304	4.47e-10*
<i>Wind</i> ₃	0.1859778	0.1609266	1.156	0.248113
<i>Wind</i> ₄	0.8819861	0.1601546	5.507	4.72e-08*
<i>Wind</i> ₅	0.7051860	0.1606954	4.388	1.27e-05*
<i>Wind</i> ₆	0.3948994	0.1607659	2.456	0.014217*
<i>Wind</i> ₇	0.7727770	0.1605713	4.813	1.74e-06*

$Wind_8$	0.9866280	0.1603996	6.151	1.14e-09*
$IM_route_know * IM_wind_know$	0.0531906	0.0924470	0.575	0.565184
$IM_route_know * Wind_2$	0.1348140	0.1848939	0.729	0.466098
$IM_route_know * Wind_3$	-0.1064143	0.1848939	-0.576	0.565063
$IM_route_know * Wind_4$	-0.4779604	0.1848939	-2.585	0.009887*
$IM_route_know * Wind_5$	0.0191784	0.1848939	0.104	0.917409
$IM_route_know * Wind_6$	-0.1151505	0.1848939	-0.623	0.533573
$IM_route_know * Wind_7$	-0.0556333	0.1848939	-0.301	0.763563
$IM_route_know * Wind_8$	-0.0976961	0.1848939	-0.528	0.597355
$IM_wind_know * Wind_2$	-0.6132222	0.1848939	-3.317	0.000946*
$IM_wind_know * Wind_3$	-0.9312502	0.1848939	-5.037	5.69e-07*
$IM_wind_know * Wind_4$	-0.6130130	0.1848939	-3.315	0.000950*
$IM_wind_know * Wind_5$	-0.5730873	0.1848939	-3.100	0.001996*
$IM_wind_know * Wind_6$	-0.4014490	0.1848939	-2.171	0.030165*
$IM_wind_know * Wind_7$	-0.6511415	0.1848939	-3.522	0.000450*
$IM_wind_know * Wind_8$	-0.5931959	0.1848939	-3.208	0.001381*

We remind the reader that for the *CPA* response, a larger value is preferred. Small negative *CPA* values correspond with spacing violations. From the results table, we see that all three delay differentials are very significant when it comes to predicting *CPA*. The positive sign on the *TRL_delaydiff* coefficient estimate means that as FIM's delay increases above TRL's delay, the *CPA* increases. The reverse relationship holds for LED and TGT. As FIM's delay increases above LED's delay and TGT's delay, the *CPA* decreases. This is a similar relationship to what we saw previously with the probability of IM interruptions and spacing violations if we equate their occurrence with smaller *CPA* values. We also see that the existence of both the IM wind and route knowledge advancements lead to significantly larger *CPA* values than if either were absent; however, their interaction is

not significant. Winds also seem to have a significant effect on *CPA*, and there is a lot of significant interaction between winds and FIM's wind knowledge. This means that depending on the wind set and whether FIM has full knowledge of such winds, the closest the FIM aircraft gets to the LED aircraft in front and the TRL aircraft behind varies significantly.

Interrupt_DTG & SpaceViol_DTG vs. All Factors

In Tables 6 and 7, we present the results from fitting a Gamma regression model of the strictly positive continuous *Interrupt_DTG* and *SpaceViol_DTG* responses versus all of our factors plus some additional interaction terms. Residual analysis revealed no issues with using a Gamma regression model for these responses.

Table 6.: Gamma regression model fitting results for *Interrupt_DTG* response. * Significant at 0.05 level.

Parameter	Estimate	Std. Error	z value	p-value
Intercept	4.2337035	0.0336390	125.857	< 2e-16
<i>LED_delaydiff</i>	0.0000976	0.0003323	0.294	0.769194
<i>TRL_delaydiff</i>	-0.0087822	0.0008045	-10.916	< 2e-16*
<i>TGT_delaydiff</i>	0.0009481	0.0002489	3.810	0.000170*
<i>IM_route_know</i>	0.0399261	0.0428185	0.932	0.351885
<i>IM_wind_know</i>	-0.2090156	0.0430459	-4.856	1.97e-06*
<i>Wind</i> ₂	-0.3457039	0.0535545	-6.455	4.58e-10*
<i>Wind</i> ₃	0.0069212	0.0504095	0.137	0.890890
<i>Wind</i> ₄	-0.0299476	0.0503725	-0.595	0.552629
<i>Wind</i> ₅	-0.1787085	0.0492536	-3.628	0.000337*
<i>Wind</i> ₆	-0.2215487	0.0433877	-5.106	5.98e-07*
<i>Wind</i> ₇	-0.7514998	0.0632958	-11.873	< 2e-16*
<i>Wind</i> ₈	-0.8606368	0.0774737	-11.109	< 2e-16*

$IM_route_know * IM_wind_know$	-0.0240927	0.0349850	-0.689	0.491594
$IM_route_know * Wind_2$	-0.0432304	0.0703537	-0.614	0.539388
$IM_route_know * Wind_3$	0.0238164	0.0576807	0.413	0.679987
$IM_route_know * Wind_4$	-0.0187974	0.0603759	-0.311	0.755768
$IM_route_know * Wind_5$	-0.0721409	0.0637438	-1.132	0.258688
$IM_route_know * Wind_6$	0.0013413	0.0544989	0.025	0.980382
$IM_route_know * Wind_7$	-0.7612838	0.1191732	-6.388	6.73e-10*
$IM_route_know * Wind_8$	-1.1555921	0.1259525	-9.175	< 2e-16*
$IM_wind_know * Wind_2$	0.2170100	0.0676867	3.206	0.001497*
$IM_wind_know * Wind_3$	0.2693450	0.0575975	4.676	4.49e-06*
$IM_wind_know * Wind_4$	0.0809403	0.0607000	1.333	0.183439
$IM_wind_know * Wind_5$	0.1699831	0.0620125	2.741	0.006506*
$IM_wind_know * Wind_6$	0.1471192	0.0545672	2.696	0.007428*
$IM_wind_know * Wind_7$	0.2118051	0.0833554	2.541	0.011579*
$IM_wind_know * Wind_8$	0.2321008	0.1077815	2.153	0.032114*

Table 7.: Gamma regression model fitting results for *SpaceViol_DTG* response. * Significant at 0.05 level.

Parameter	Estimate	Std. Error	z value	p-value
Intercept	4.1848344	0.0595910	70.226	< 2e-16
$LED_delaydiff$	-0.0005049	0.0006676	-0.756	0.450569
$TRL_delaydiff$	-0.0107573	0.0014897	-7.221	1.59e-11*
$TGT_delaydiff$	0.0019426	0.0005156	3.768	0.000226*
IM_route_know	-0.1912565	0.0758081	-2.523	0.012546*
IM_wind_know	-0.5246205	0.0767616	-6.834	1.37e-10*

$Wind_2$	-0.5048918	0.0957421	-5.273	3.98e-07*
$Wind_3$	-0.1504268	0.0917230	-1.640	0.102830
$Wind_4$	-0.2593822	0.0922284	-2.812	0.005489*
$Wind_5$	-0.3147737	0.0957742	-3.287	0.001229*
$Wind_6$	-0.3687661	0.0752886	-4.898	2.22e-06*
$Wind_7$	-0.8522274	0.1038453	-8.207	5.14e-14*
$Wind_8$	-0.8679335	0.1328947	-6.531	7.09e-10*
$IM_route_know * IM_wind_know$	-0.5077297	0.0769968	-6.594	5.05e-10*
$IM_route_know * Wind_2$	0.0640657	0.1842916	0.348	0.728541
$IM_route_know * Wind_3$	0.6932546	0.1125625	6.159	5.03e-09*
$IM_route_know * Wind_4$	0.2475709	0.1156796	2.140	0.033753*
$IM_route_know * Wind_5$	-0.3618888	0.1603354	-2.257	0.025262*
$IM_route_know * Wind_6$	0.1395829	0.1183575	1.179	0.239895
$IM_route_know * Wind_7$	-0.1909944	0.1865957	-1.024	0.307474
$IM_route_know * Wind_8$	-0.7719057	0.1976431	-3.906	0.000135*
$IM_wind_know * Wind_2$	0.5822546	0.1293836	4.500	1.25e-05*
$IM_wind_know * Wind_3$	0.7830226	0.1082322	7.235	1.48e-11*
$IM_wind_know * Wind_4$	0.5718573	0.1126556	5.076	9.93e-07*
$IM_wind_know * Wind_5$	0.4363660	0.1274145	3.425	0.000769*
$IM_wind_know * Wind_6$	0.3611706	0.1030258	3.506	0.000581*
$IM_wind_know * Wind_7$	0.6173167	0.1409481	4.380	2.06e-05*
$IM_wind_know * Wind_8$	0.6435919	0.1746588	3.685	0.000306*

For these *Interrupt_DTG* and *SpaceViol_DTG* responses, larger values are desired. It is easier for ATC to deal with an IM interruption or spacing violation if it occurs further out along an aircraft's trajectory rather than close to the runway. We can see from the results tables that only the *TRL_delaydiff* and *TGT_delaydiff* are significant delay differentials

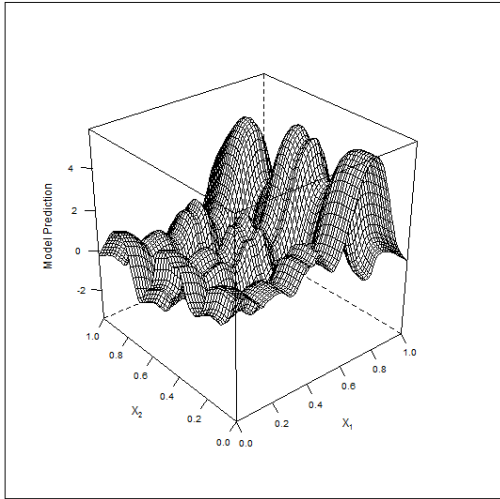
when it comes to predicting where a spacing violation or IM interruption will occur. The negative $TRL_delaydiff$ coefficient estimate means that as FIM's delay increases further above TRL's delay, IM interruptions and spacing violations will occur closer to the runway. The reverse relationship hold for $TGT_delaydiff$. As FIM's delay increases further above TGT's delay, IM interruptions and spacing violations will occur further away from the runway. The location of spacing violations is significantly impacted by both IM wind and route knowledge; however, the location of IM interruptions is only significantly impacted by IM wind knowledge. Even though IM knowledge advancements have been shown to have positive effects on other responses, here they appear to lead to IM interruptions and spacing violations occurring closer to the runway. This leads us to believe that granting FIM full knowledge of delayed trajectories and winds decreases the occurrence of spacing violations and IM interruptions by pushing forward their theoretical occurrence location beyond the runway at a DTG less than zero. Winds are also very significant for determining the location of IM interruptions and spacing violations. Many wind profiles have significant interactions with both IM wind and route knowledge. It is safe to say that in general, the location of losses of separation is affected by winds.

CPA vs Wind Delay Differentials via Gaussian Process Model

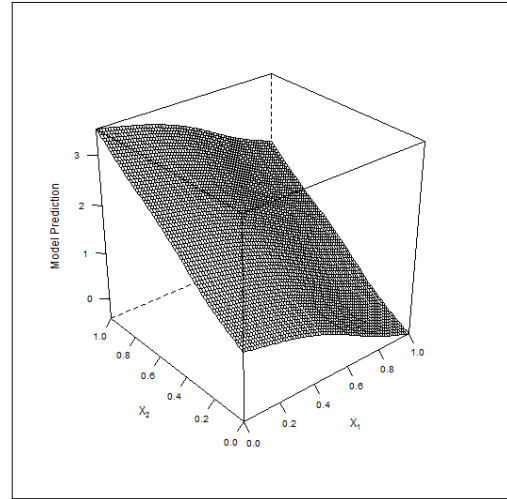
In this section, we present results from fitting Gaussian process models to create predicted CPA response surfaces versus the three delay differentials for each wind set. Since we cannot plot in more than three dimensions, we have three CPA surfaces for each wind set: one for each pair of delay differentials. In Figures 11 and 12, we have the CPA surface versus the $TGT_delaydiff$ and $LED_delaydiff$ for each wind set. In Figures 13 and 14, we have the CPA surface versus the $TGT_delaydiff$ and $TRL_delaydiff$ for each wind set. In Figures 15 and 16, we have the CPA surface versus the $LED_delaydiff$ and $TRL_delaydiff$ for each wind set. In order to estimate CPA response surfaces, the delay differentials needed to be standardized by their respective maximums and minimums. This is why the delay differential axes in Figures 11-16 range from 0 to 1. In our analysis of the response surfaces,

we will be looking for the delay differential values that lead to large CPA values.

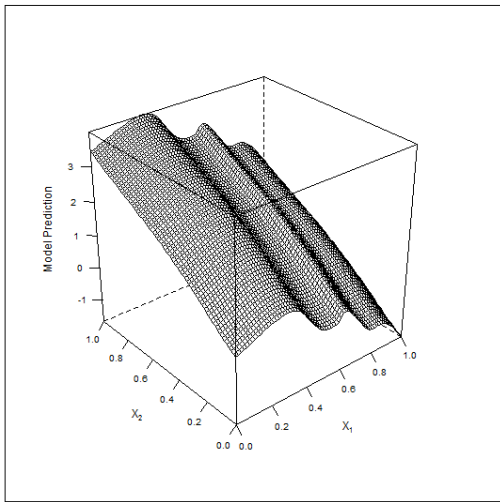
We will discuss Figures 11 and 12 first. For $Wind_2$, $Wind_3$, and $Wind_7$, large CPA values are achieved by a small $TGT_delaydiff$ and large $LED_delaydiff$. However, for $Wind_1$, $Wind_5$, and $Wind_6$, the reverse relationship holds; a large $TGT_delaydiff$ and small $LED_delaydiff$ result in large CPA values. The surfaces for $Wind_4$ and $Wind_8$ have many peaks and valleys; large CPA values can be achieved just about anywhere. Now let's move on to Figures 13 and 14. It is clear that the same relationship holds for all eight wind sets: the largest CPA can be achieved by a small $TGT_delaydiff$ and large $TRL_delaydiff$. This mirrors our results from the multiple linear regression model fitted to predict CPA which informed us that large $TRL_delaydiff$ values lead to large CPA values. Finally, we will discuss Figures 15 and 16. In these plots, the CPA surfaces for all eight wind sets seem to have the same overall shape with a few of them looking nearly identical. In these plots, the largest CPA values are achieved by a small $LED_delaydiff$ and large $TRL_delaydiff$. Looking at the plots as a whole, it appears that the CPA response surface is not terribly dependent on winds; the same overall patterns are shared among them.



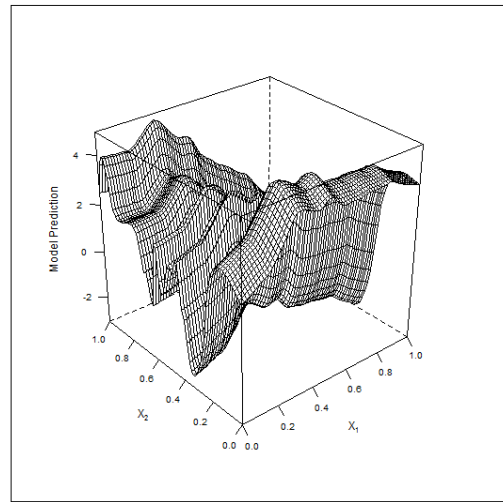
(a)



(b)

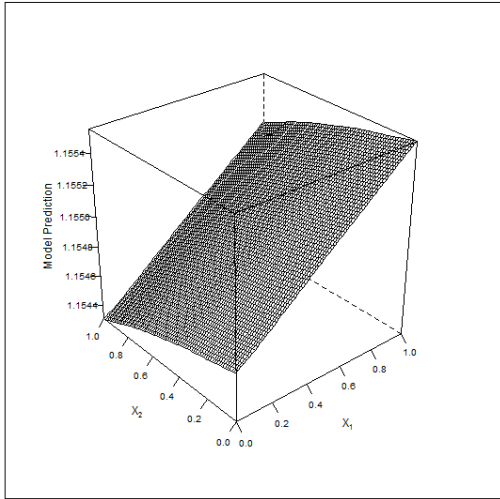


(c)

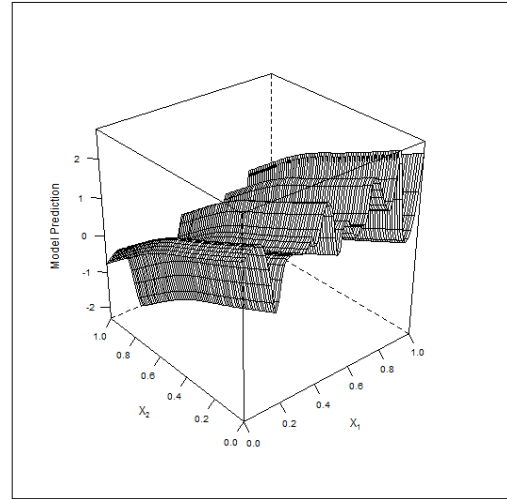


(d)

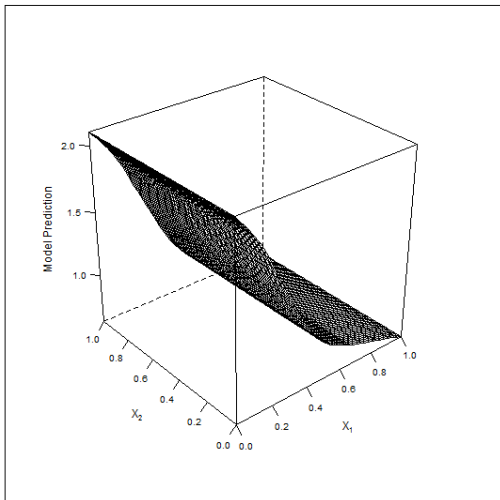
Fig. 11. The predicted *CPA* response surface versus *TGT_delaydiff* (X_1) and *LED_delaydiff* (X_2) for $Wind_1$ (a), $Wind_2$ (b), $Wind_3$ (c), and $Wind_4$ (d). The delay differentials are standardized to be within 0 and 1.



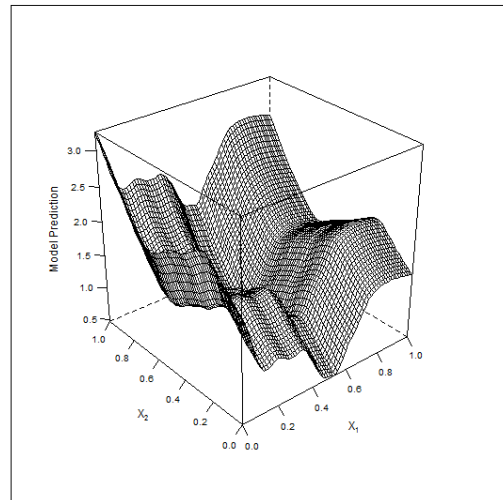
(a)



(b)

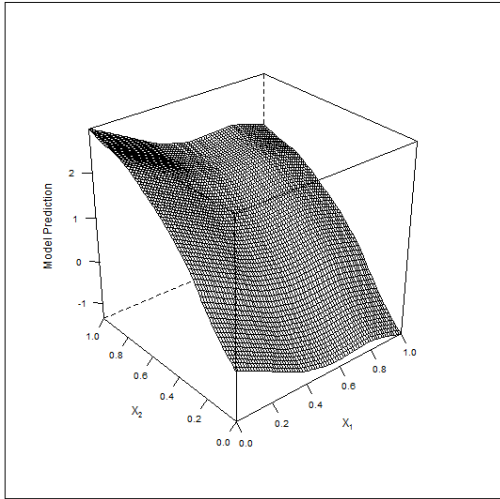


(c)

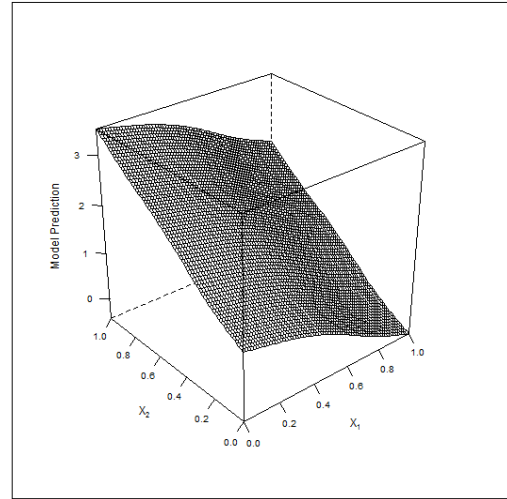


(d)

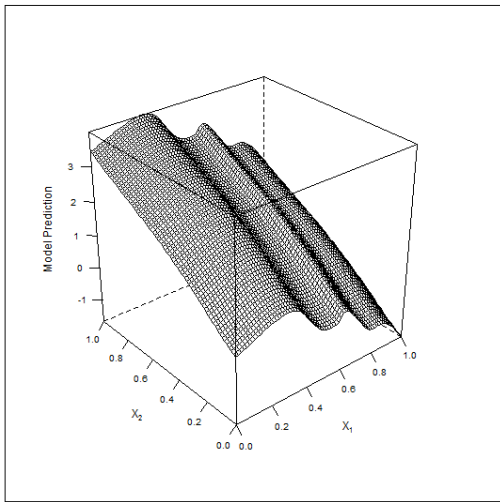
Fig. 12. The predicted *CPA* response surface versus *TGT_delaydiff* (X_1) and *LED_delaydiff* (X_2) for *Wind*₅ (a), *Wind*₆ (b), *Wind*₇ (c), and *Wind*₈ (d). The delay differentials are standardized to be within 0 and 1.



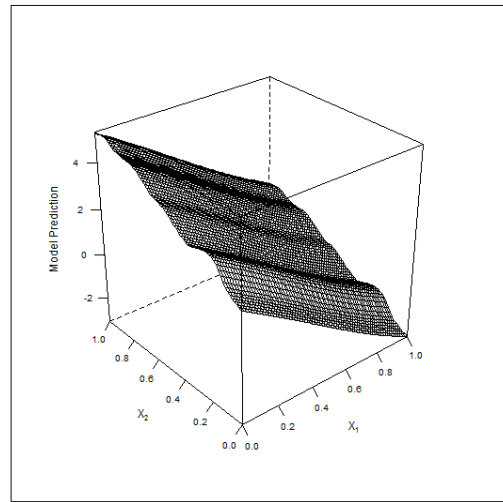
(a)



(b)

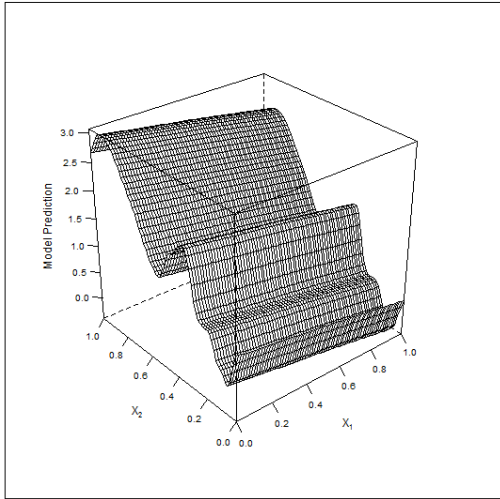


(c)

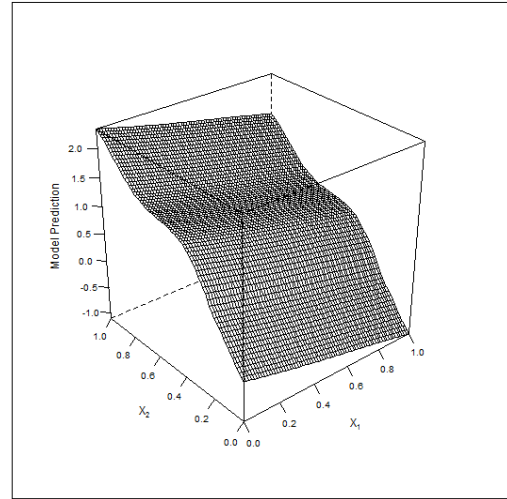


(d)

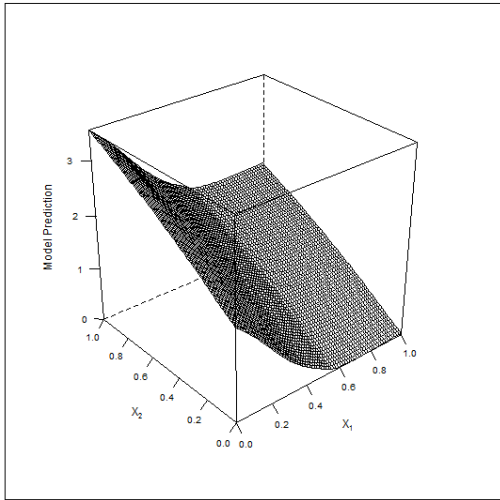
Fig. 13. The predicted *CPA* response surface versus *TGT_delaydiff* (X_1) and *TRL_delaydiff* (X_2) for *Wind*₁ (a), *Wind*₂ (b), *Wind*₃ (c), and *Wind*₄ (d). The delay differentials are standardized to be within 0 and 1.



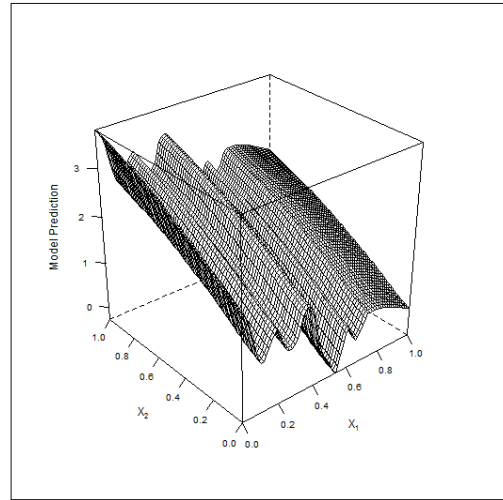
(a)



(b)

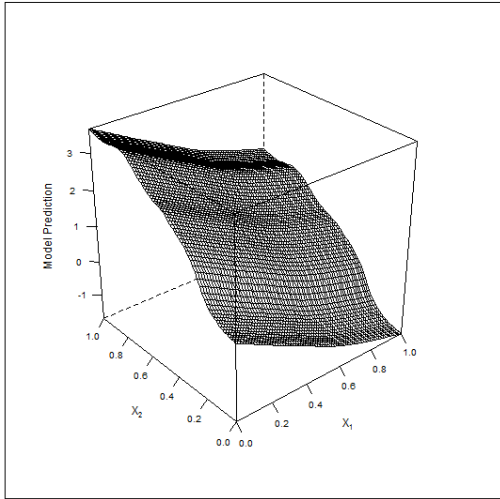


(c)

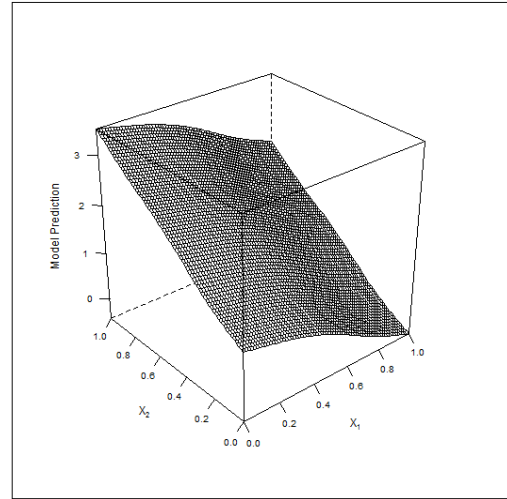


(d)

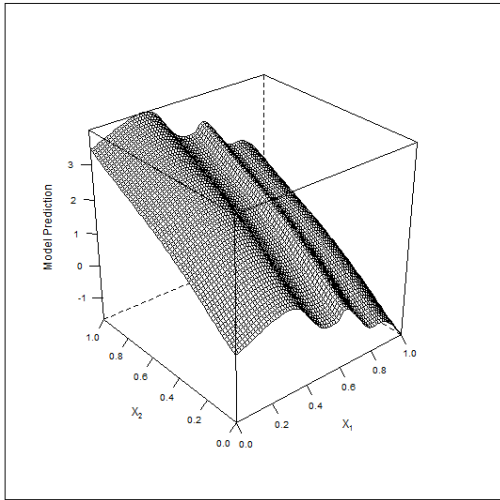
Fig. 14. The predicted *CPA* response surface versus *TGT_delaydiff* (X_1) and *TRL_delaydiff* (X_2) for *Wind*₅ (a), *Wind*₆ (b), *Wind*₇ (c), and *Wind*₈ (d). The delay differentials are standardized to be within 0 and 1.



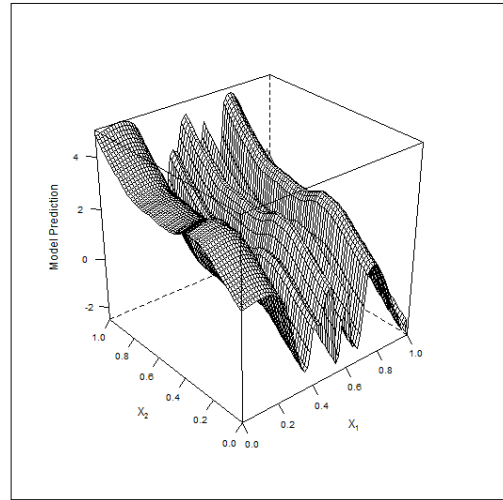
(a)



(b)

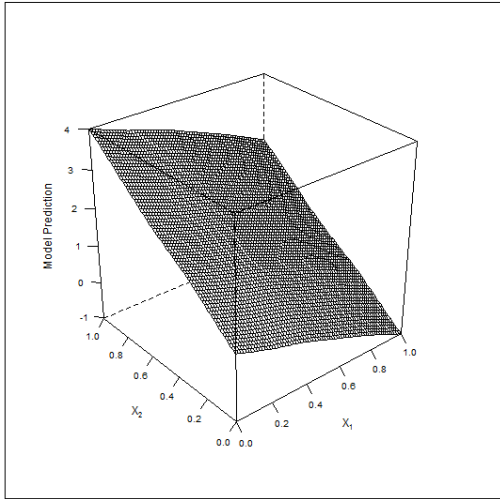


(c)

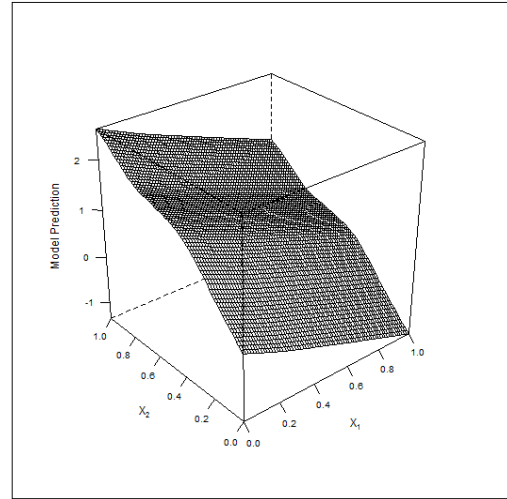


(d)

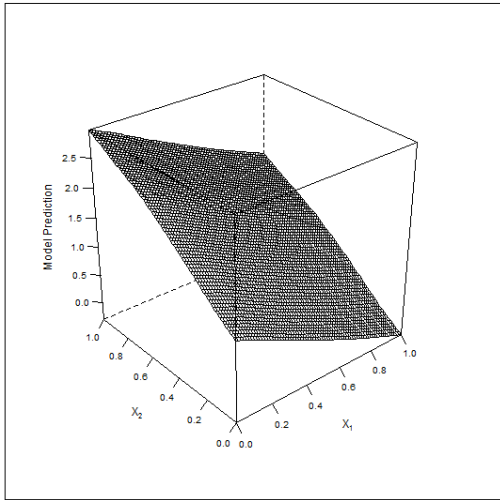
Fig. 15. The predicted *CPA* response surface versus *LED_delaydiff* (X_1) and *TRL_delaydiff* (X_2) for $Wind_1$ (a), $Wind_2$ (b), $Wind_3$ (c), and $Wind_4$ (d). The delay differentials are standardized to be within 0 and 1.



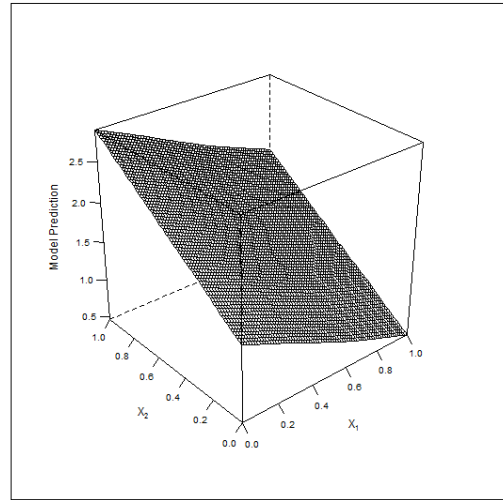
(a)



(b)



(c)



(d)

Fig. 16. The predicted *CPA* response surface versus *LED_delaydiff* (X_1) and *TRL_delaydiff* (X_2) for $Wind_5$ (a), $Wind_6$ (b), $Wind_7$ (c), and $Wind_8$ (d). The delay differentials are standardized to be within 0 and 1.

CHAPTER 5

DISCUSSION

This research was focused on Interval Management operations during the specific airspace scenario where an IM-equipped aircraft is following a target aircraft on a separate route while flying within a consecutive string of non-IM aircraft. In particular, we wanted to know what specific airspace conditions lead to increased probabilities of IM interruptions and spacing violations through the design and analysis of a computer experiment using deterministic computer simulation data. Our main factors of interest were winds, aircraft delay differentials, and two theoretical advancements in IM capabilities. These advancements are allowing the IM-equipped aircraft to have knowledge of delayed trajectories as well as the target aircraft's route-specific wind profile. The simulation emulated the arrival of four aircraft into Denver International Airport: an IM-equipped aircraft, a target aircraft on a separate route, a leading aircraft immediately in front of the IM-equipped aircraft, and a trailing aircraft immediately behind. The simulation included all of the main facets of IM operations upon aircraft arrival: aircraft scheduling, delay assignments, the ASTAR12 spacing algorithm, etc. In the design of the computer experiment to analyze the simulation data, an original space-filling design was proposed to sample the space of delay differentials; that is the difference in delay between the IM-equipped aircraft and the three other modeled aircraft. Our original space-filling design took advantage of the maximin criteria which seeks to maximize the minimum distance between every pair of design points in the sample. Once our sample of design points was chosen and tested on, we used Gaussian process modeling to interpolate a predicted response surface based on the responses at our small number of selected design points.

We found that delay differentials do significantly impact the likelihood of spacing violations and IM interruptions. If the IM-equipped aircraft has more delay assigned to it

than the trailing aircraft behind it, then spacing violations and interruptions are less likely to occur. The opposite relationship is true for the leading aircraft and target aircraft. We also found that the two theoretical IM technology advancements can decrease the number of spacing violations and interruptions immensely. Going from the current state of IM operations to the theoretical capability of the IM-equipped aircraft having full knowledge of delayed trajectories and winds, spacing violations decrease in occurrence from 31% to 9% while IM interruptions decrease in occurrence from 40% to 24%. However, further investigation is needed into the interaction between giving the IM-equipped aircraft detailed wind knowledge and the specific winds being encountered. The occurrence of spacing violations and IM interruptions is also very sensitive to encountered winds. Given certain wind conditions, IM interruptions can occur as often as 55% of the time or as little as 8% of the time. On the other hand, the closest point of approach between the IM-equipped aircraft and the leading/trailing aircraft is not very sensitive to wind conditions.

In the future, the simulation could be generalized to include different airspace configurations, more than a string of three aircraft, and possibly more than one IM-equipped aircraft. Further simulation studies could be conducted to investigate the sensitive dependence of IM performance on wind conditions. As another future endeavor, more work could be devoted to examining the feasibility of the two theoretical IM technology advancements of detailed delayed trajectory and wind knowledge. Our results indicate that they are worth further study.

REFERENCES

Baxley, B., Johnson, W., Swenson, H., Robinson, J., Prevot, T., Callantine, T., Scardina, J., & Greene, M. (2013). Air Traffic Management Technology Demonstration-1 Concept of Operations (ATD-1 ConOps), Version 2.0. *National Aeronautics and Space Administration*.

Federal Aviation Administration. FAA Aerospace Forecast, Fiscal Years 2011-2031; http://www.faa.gov/about/office_org/headquarters_offices/apl/aviation_forecasts/aerospace_forecasts/2011-2031/

Prevot, T., Baxley, B., Callantine, T., Johnson, W., Quon, L., Robinson, J., & Swenson, H. (2011). NASAs ATM Technology Demonstration-1: Transitioning Fuel Efficient, High Throughput Arrival Operations from Simulation to Reality. *National Aeronautics and Space Administration*.

Swenson, H., Thipphavong, J., Sadosky, A., Chen, L., Sullivan, C., & Martin, L. (2011). Design and Evaluation of the Terminal Area Precision Scheduling and Spacing System. *Ninth USA/Europe Air Traffic Management Research and Development Seminar*.

Callantine, T., Hunt, S., & Prevot, T. (2014). Simulation Evaluation of Controller-Managed Spacing Tools under Realistic Operational Conditions. *HCI-Aero*.

Abbott, T. (2015). An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Seventh Revision. *National Aeronautics and Space Administration*.

Swieringa, K. (2015). The String Stability of a Trajectory-Based Interval Management Al-

gorithm in the Midterm Airspace. *American Institute of Aeronautics and Astronautics*.

Swieringa, K., Underwood, M., Barmore, B., & Leonard, R. (2015) An Evaluation of a Flight Deck Interval Management Algorithm including Delayed Target Trajectories. *American Institute of Aeronautics and Astronautics*.

Levitt, I., Weitz, L., Barmore, B., & Castle, M. (2014). Feasibility Criteria for Interval Management Operations as Part of Arrival Management Operations. *American Institute of Aeronautics and Astronautics*.

Robinson, J. (2014). Calculation of Flight Deck Interval Management Assigned Spacing Goals Subject to Multiple Scheduling Constraints. *33rd Digital Avionics Systems Conference*.

Fermi E., Pasta J., & Ulam S. (1955). Studies of Nonlinear Problems I. *Los Alamos Report*.

Kee, R., Grcar, J., Smooke, M., Miller., & Meeks, E. (1985). PREMIX: A Fortran Program for Modeling Steady Laminar One-Dimensional Premixed Flames. *Sandia National Laboratories Report*.

Sacks, J., Welch, W., Mitchell, T., & Wynn, H. (1989). Design and Analysis of Computer Experiments. *Statistical Science*, 4(4), 409-423.

McKay, M., Beckman, R., & Conover, W. (1979). A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21(2), 239-245.

- Johnson, M., Moore, L., & Ylvisaker, D. (1990). Minimax and Maximin Distance Designs. *Journal of Statistical Planning and Interface*, 26, 131-148.
- Morris, M. & Mitchell, T. (1995). Exploratory Designs for Computer Experiments. *Journal of Statistical Planning and Inference*, 43, 381-402.
- Moon, H., Dean, A., & Santner, T. (2011) Algorithms for Generating Maximin Latin Hypercube and Orthogonal Designs. *Journal of Statistical Theory and Practice*, 5(1), 81-98.
- Owen, A. (1994). Controlling Correlations in Latin Hypercube Samples. *Journal of the American Statistical Association*, 89, 1517-1522.
- Tang, B. (1993). Orthogonal Array-Based Latin Hypercubes. *Journal of the American Statistical Association*, 88, 1392-1397.
- Shewry, M. & Wynn, H. (1987). Maximum Entropy Sampling. *Journal of Applied Statistics*, 14, 165-170.
- Leatherman, S. (2014). Adaptive Methods for Bayesian Time-to-Event Point-of-Care Clinical Trials. *Boston University Gradworks Dissertations & Theses*.
- Swieringa, K., Wilson, S., & Baxley, B. (2015). System Performance of an Integrated Airborne Spacing Algorithm with Ground Automation. *American Institute of Aeronautics and Astronautics*.
- Loeppky, J., Sacks, J., & Welch, W. (2009). Choosing the Sample Size of a Computer Experiment: A Practical Guide. *Technometrics*, 51(4), 366-376.

Do, C. (2007). Gaussian Processes. *Stanford University*.

MacDonald, B., Ranjan, P., & Chipman, H. (2015). GPfit: An R Package for Fitting a Gaussian Process Model to Deterministic Simulator Outputs. *Journal of Statistical Software*, 64(12).

VITA

Ryan Wayne Gryder was born in Richmond, Virginia, on April 1, 1992. After graduating from the Mathematics and Science High School at Clover Hill in Midlothian, Virginia, in 2010, he went to the College of William and Mary and received a Bachelor of Science in Mathematics in 2014. During his undergraduate studies, Ryan worked as a Teaching Assistant for an Ordinary Differential Equations class and Research Assistant in the Quantitative Biology Lab. After receiving his undergraduate degree, he proceeded to Virginia Commonwealth University to pursue a Master of Science in Mathematical Sciences with a concentration in Statistics and has been there for the past two years. During his graduate studies, Ryan tutored children at Mathnasium, was a Teaching Assistant and Lab Instructor for an Intro to Statistics class at VCU, and currently works as a Graduate Student Researcher at NASA Langley Research Center in the Statistical Engineering Team. Ryan lives in Richmond with his two cats, Nerd and Tiny.