OPTIMAL AIRCRAFT REROUTING
DURING SPACE LAUNCHES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND
ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Rachael Elizabeth Tompa
November 2019
© 2019 by Rachael Elizabeth Tompa. All Rights Reserved.
Re-distributed by Stanford University under license with the author.

This work is licensed under a Creative Commons Attribution-Noncommercial 3.0 United States License.
http://creativecommons.org/licenses/by-nc/3.0/us/

This dissertation is online at: http://purl.stanford.edu/kp546yk7234
I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Mykel Kochenderfer, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Juan Alonso

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Zachary Manchester

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Heinz Erzberger,

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.
Abstract

Commercial interest in space exploration and capabilities has seen steady growth. This growth has the potential to advance science, inspire innovation, introduce space tourism, and address global concerns such as geospatial monitoring and data access, universal data communications, and weather. One challenge facing space exploration is how to expand operations while minimizing the disruption to shared resources like airspace.

Currently, during a space launch or reentry, the Federal Aviation Administration prohibits air traffic within a large column of airspace around the launch trajectory. The prohibited airspace is often active for hours at a time, resulting in the rerouting of hundreds of flights at the cost of thousands of dollars to airlines per launch. If the current launch and reentry procedures do not change, they could limit technology advancement and space exploration or hurt airlines.

A new method of managing the airspace during space launches would need to maintain safety, maximize efficiency, and not increase air traffic controller workload. To address safety and efficiency, recent research has focused on leveraging new technology to make the prohibited airspace dynamic throughout the space launch and to limit the geographical extent. To do this successfully, the approach must accurately model the airspace, including both the aircraft and the dispersion of debris in case of an anomaly, and must capture the inherent uncertainty of launch anomalies to provide encompassing rerouting regions.

This thesis uses Markov decision processes to model the problem of aircraft rerouting during space launches and dynamic programming to solve for optimal rerouting policies. The resulting policies are run through simulated scenarios to measure their
effects on safety and efficiency. The results produce smaller prohibited regions and provide real-time reroutes for air traffic controllers to relay to pilots during a space launch. These outputs can be used as the foundation for a decision support tool that assists air traffic controllers in rerouting flights during space launches.
Acknowledgments

I am honored and humbled by the people that have gotten me to where I am today (submitting a PhD thesis at Stanford University)! With all of the space and time I could never thank and repay the favor enough but I will at least say a few words of gratitude.

My time at Stanford would not have been so amazing and rewarding without the mentorship, guidance, and friendship of my advisor, Mykel J. Kochenderfer. Thank you for everything you have done to enhance my research, academics, career, and me as a person. I am also thankful to the other members of my defense committee: Juan J. Alonso, Heinz Erzberger, and Zachary Manchester. I have valued all of our conversations and your insightful guidance.

I have some amazing friends that have helped make my time at Stanford amazing! I appreciate all of the friendships I have made in SISL over the years, including SISL Aftermath. These people have made going to lab enjoyable and have taught me so much! The best, unexpected gift of SISL and graduate school is my boyfriend, Jon. You are my biggest cheerleader both in the ocean and on a daily basis. Your confidence in me and constant support were pivotal in my completing this thesis, and I am so appreciative of everything you do. Since orientation, I have been lucky to have Anna by my side facing the twists and turns of graduate school. Thank you for being there for me! Tina, I am so lucky you were my Uganda roommate and have become an amazing study buddy and dearer friend. Thank you Jessica for being my forever travel buddy, I am so grateful for all of your support, family away from family, and friendship. I am so happy Christina moved out here while I was in graduate school. Thank you for providing a bit of graduate school escape and home while I finished
my dissertation.

Along my academic career I have been blessed with the best family support system. My little brother, Jason, thank you for being you and always adding new knowledge and humor to my life. My big sister, Sharon, thank you for always being just a phone call away even when I am stressed. These phone calls are something that I cherish while living away and have been really helpful in getting me to where I am today. There is no way to thank my Mom and Dad enough, but thank you for everything you have provided for me and the motivation you have instilled in me. There is no way I could be where I am today without your constant support and love. I appreciate everything you have done for me.

The work presented in this thesis would not have been possible without the generous financial support I received from MIT Lincoln Laboratory, the National Science Foundation, and the Federal Aviation Administration. Finally, getting into Stanford and specifically, the Aeronautics and Astronautics department was a dream come true. I must recognize the leadership and staff of the department who have created a remarkable environment for learning and the school for providing amazing opportunities, programs, and memories.
# Contents

Abstract

Acknowledgments

1 Introduction

1.1 Space Launches in the United States ............................................................... 1
   1.1.1 Current Space Launch Operations ......................................................... 3
   1.1.2 Dynamic Flight Restrictions ............................................................... 4
   1.1.3 Dynamic Rerouting ............................................................................. 5

1.2 Decision Making Under Uncertainty ............................................................... 6
   1.2.1 Optimization Objective ..................................................................... 6
   1.2.2 Markov Decision Process ................................................................... 7
   1.2.3 MDP Advantages and Disadvantages .................................................... 10

1.3 Contributions ................................................................................................. 10

2 MDP Formulation

2.1 Scenario .......................................................................................................... 12
   2.1.1 Debris Modeling ............................................................................... 13
   2.1.2 Debris Avoidance Methodologies ....................................................... 15

2.2 Heading Actions ............................................................................................ 16
   2.2.1 State Space ....................................................................................... 16
   2.2.2 Action Space ..................................................................................... 18
   2.2.3 Reward Model ............................................................................... 18
   2.2.4 Transition Model ........................................................................... 19
2.3 Metering Actions ................................................. 20
  2.3.1 State Space ............................................... 21
  2.3.2 Action Space ............................................. 22
  2.3.3 Reward Model ............................................ 22
  2.3.4 Transition Model ......................................... 23
2.4 Discussion ..................................................... 24

3 Solving MDPs ................................................... 26
  3.1 Rational Decision Making ................................. 26
  3.2 Solution Techniques ........................................ 27
    3.2.1 Exact Offline Solution Techniques .................. 27
    3.2.2 Approximate Offline Solution Techniques .......... 28
    3.2.3 Online Solution Techniques .......................... 28
    3.2.4 Offline vs. Online Tradeoff ........................... 29
  3.3 Value Iteration ............................................. 30
    3.3.1 Backward Induction Value Iteration ............... 30
  3.4 Adaptive Spatial Discretization .......................... 33
    3.4.1 Adaptive Spatial Discretization Algorithm ......... 33
    3.4.2 Utility Decomposition .................................. 35
  3.5 Discussion ................................................ 38

4 Simulation Framework and Results .......................... 41
  4.1 Utility and Policy Results ................................ 41
    4.1.1 Heading Action Solutions ............................ 41
    4.1.2 Metering Action Solution ............................ 49
  4.2 Simulation Framework ...................................... 50
    4.2.1 Domestic Flight Paths ................................ 52
    4.2.2 Oceanic Flight Trajectories ......................... 53
    4.2.3 Comparison Policies .................................. 56
    4.2.4 Safety vs. Efficiency Tradeoff Analysis .......... 56
  4.3 Simulation Results ......................................... 57
    4.3.1 Domestic Simulation Results ......................... 57
4.3.2 Oceanic Simulation Results ........................................ 61
4.4 Discussion ........................................................................ 63

5 Summary and Future Work ............................................... 66
5.1 Contributions ................................................................. 66
  5.1.1 MDP Formulations ..................................................... 67
  5.1.2 Implemented Solution Techniques ............................... 67
  5.1.3 Simulation Framework ............................................... 68
5.2 Future Work ...................................................................... 68
5.3 Potential Implementation .................................................. 71
List of Tables

2.1 Fixed discretization aircraft state space variables . . . . . . . . . . . . 18
2.2 Reward parameters . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
2.3 Reward parameters . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
4.1 Experimental $E$ and $N$ discretizations . . . . . . . . . . . . . . . . . 42
4.2 Domestic simulation results . . . . . . . . . . . . . . . . . . . . . . . 59
4.3 Domestic safety results for different thresholds . . . . . . . . . . . . . 61
4.4 Oceanic simulation results . . . . . . . . . . . . . . . . . . . . . . . . 62
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Graph of United States launch operations per year</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>A domestic TFR for the launch out of Kennedy Space Center on February 4, 2015</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>Example tradeoff curve for two objectives</td>
<td>8</td>
</tr>
<tr>
<td>1.4</td>
<td>Timestep update of MDP</td>
<td>9</td>
</tr>
<tr>
<td>1.5</td>
<td>Example MDP formulation</td>
<td>9</td>
</tr>
<tr>
<td>2.1</td>
<td>Modeled launch vehicle trajectory</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>Debris trajectories</td>
<td>14</td>
</tr>
<tr>
<td>2.3</td>
<td>Logarithmic histogram of potential debris locations</td>
<td>15</td>
</tr>
<tr>
<td>2.4</td>
<td>Example of heading actions avoiding debris in space</td>
<td>16</td>
</tr>
<tr>
<td>2.5</td>
<td>Example of metering actions avoiding debris in time</td>
<td>17</td>
</tr>
<tr>
<td>2.6</td>
<td>Potential flight waypoints near the coast of Florida</td>
<td>21</td>
</tr>
<tr>
<td>3.1</td>
<td>Example of solving for an optimal policy for a single piece of debris</td>
<td>32</td>
</tr>
<tr>
<td>3.2</td>
<td>Example evolution of $E$ and $N$ state space with ASD</td>
<td>36</td>
</tr>
<tr>
<td>3.3</td>
<td>Example updating $E_{\text{multi}}, N_{\text{multi}}$ grid with $E_{\text{single}}, N_{\text{single}}$</td>
<td>39</td>
</tr>
<tr>
<td>4.1</td>
<td>Utility for different discretizations over time</td>
<td>43</td>
</tr>
<tr>
<td>4.2</td>
<td>Policy for different discretizations over time</td>
<td>44</td>
</tr>
<tr>
<td>4.3</td>
<td>Utility for a single piece of debris over time</td>
<td>46</td>
</tr>
<tr>
<td>4.4</td>
<td>Policy for a single piece of debris over time</td>
<td>46</td>
</tr>
<tr>
<td>4.5</td>
<td>Policy for a single piece of debris over time with the least restrictive state space</td>
<td>48</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.6</td>
<td>ASD utility over time (s after launch)</td>
<td>48</td>
</tr>
<tr>
<td>4.7</td>
<td>ASD policy over time (s after launch)</td>
<td>48</td>
</tr>
<tr>
<td>4.8</td>
<td>Policy for a single path over time</td>
<td>51</td>
</tr>
<tr>
<td>4.9</td>
<td>Flight trajectories gathered from FACET</td>
<td>52</td>
</tr>
<tr>
<td>4.10</td>
<td>Randomly generated paths over debris locations</td>
<td>54</td>
</tr>
<tr>
<td>4.11</td>
<td>Single path with 33 simulation trajectories</td>
<td>55</td>
</tr>
<tr>
<td>4.12</td>
<td>Heading safety vs. efficiency lambda exploration</td>
<td>58</td>
</tr>
<tr>
<td>4.13</td>
<td>Metering safety vs. efficiency lambda exploration</td>
<td>59</td>
</tr>
<tr>
<td>4.14</td>
<td>Heading safety vs. efficiency results</td>
<td>60</td>
</tr>
<tr>
<td>4.15</td>
<td>Flight segment where trajectory interacts with debris</td>
<td>63</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Space exploration has been growing since the start of the 20th century. It has quickly advanced through large milestones, from the first long distance missile to the first artificial satellite, the first humans in space, orbit, and on the moon, and the development of the International Space Station. Recently, space exploration shifted from exclusively being a government practice to the focus of many commercial enterprises. There are many more players leading to an increased concern for safe and efficient practices.

This thesis focuses on the impact of increased space launches on air traffic management and proposes safe and efficient solutions. This chapter gives an introduction to the problem of rerouting aircraft during space launches as well as background on technical approaches that could be used to address this problem.

1.1 Space Launches in the United States

Historically, the majority of space launches within the United States have been conducted by the government to either deliver satellites into orbit or transport crew and cargo to the International Space Station. With the retirement of the space shuttle program, the number of government launches has decreased. Meanwhile, excluding 2015 when there were a series of mishaps, the actual and forecast commercial launch operations throughout the United States have been increasing as presented in Fig. 1.1
CHAPTER 1. INTRODUCTION

[1].

![Graph of United States launch operations per year](image)

This growth is due to technical innovation, commercial crew transportation contracts, rocket reusability, an interest in space tourism, and multiple government proposal announcements for industry participation (such as the call for Commercial Lunar Payload Services and DARPA Launch Challenge) [1]. In space tourism, there are multiple companies working on suborbital reusable vehicles (SRVs). The introduction of SRVs alone have the potential to increase the commercial space launch rate by a factor of 50 in the short term and could result in multiple daily launches. Further, the use of SRVs is expected to initiate additional launch facilities in dense air traffic regions. Specifically, there are proposed launch facilities in Alaska, California, Colorado, Texas, and Virginia [2].

The remainder of this section discusses current space launch operations, their effect on the airspace, an introduction to researched methods to improve these operations, and a vision for ideal future operations.
1.1.1 Current Space Launch Operations

Currently, the Federal Aviation Administration (FAA) Office of Commercial Space Transportation (AST) is responsible for licensing and regulating commercial space launches as well as promoting the commercial space transportation industry [1]. Part of AST’s regulations are the National Airspace System (NAS) operations, which ensure safe separation between aircraft and launch vehicles or, in case of an anomaly, potential pieces of debris. An anomaly is defined as a launch vehicle explosion that may disperse pieces of debris that are dangerous to aircraft in the NAS. Part of AST’s commercial space transportation promotions are researching ways for the industry to be efficiently integrated with current NAS procedures.

When launches were infrequent and carried out by the government, the FAA created the procedure of shutting down a large, infinitely tall column of airspace. This restricted region ensured safety in case of an anomaly that could result in thousands of pieces of debris. These restrictions were often marked on regional sectionals as Special Use Airspace (SUA) and were subject to Temporary Flight Restrictions (TFR) that were activated through Notices to Airmen (NOTAMs). NOTAMs were posted well before a launch and, depending on the launch schedule, could shut down a region for hours. An example TFR for domestic flights is shown in Fig. 1.2 for the launch out of Kennedy Space Center on February 4, 2015 [3].

Many flights are required to be rerouted to avoid these restrictions resulting in added airline expenses. For instance, on February 6, 2018, the SpaceX Falcon Heavy was launched out of Kennedy Space Center, disrupting 563 flights with an average delay of 8 min and an additional 62 nautical miles per flight [4]. It is predicted that on average, each disrupted flight incurred an additional expense of $547.84 due to fuel, labor, and aircraft expenses. Not including missed connections, a single commercial space launch costs commercial airlines more than $308,000 [5].

There is currently no proposed alternative to the highly disruptive procedures, and launches are becoming increasingly more frequent. Further, many of the launches are no longer carried out by the government, but the cost is still imposed on the airlines rather than the launch entity. Both airlines and commercial spaceflight companies
are aware of this issue. Commercial spaceflight companies brought it to the Senate’s attention [6] at the New Entrants in the National Airspace: Policy, Technology, and Security Issues for Congress hearing on May 8, 2019 to encourage the FAA to dynamically integrate space launches into the NAS [7].

1.1.2 Dynamic Flight Restrictions

To address the increase in airspace disruptions and minimize added airline costs, there is a desire to move from the static prohibited regions [8] towards a dynamic process. If the airspace restrictions were made dynamic and accounted for the estimated debris trajectories, the number of rerouted aircraft could be significantly decreased. Dynamic restrictions would allow the restricted area to change throughout the launch process and adapt to launch vehicle health. These dynamic and localized safety regions would limit disruptions while maintaining airspace safety, reducing added expenses, and minimizing delay.

There are two major approaches to this problem. The first uses compact envelopes
which are dynamic and vary shape with altitude. The compact envelopes use probabilistic risk analysis of off-nominal vehicle operations to create envelope boundaries that represent a quantifiable level of safety. This system relies on NextGen advancements to significantly decrease NAS disruption without compromising safety.

The second is space transition corridors. Similar to compact envelopes, these corridors are dynamic, vary with shape with altitude, and are designed based on probabilistic risk analysis of off-nominal vehicle operations. Additionally, space transition corridors are designed to adjust based on the condition of the launch vehicle.

1.1.3 Dynamic Rerouting

It is predicted that in the future, space launch operations will be fully integrated into the air traffic management system. The purpose of air traffic management is to manage the flow of aircraft in the NAS based on capacity and demand to provide safe, orderly, and expeditious flow of traffic. Air traffic management is conducted by Air Route Traffic Control Centers, selected terminal facilities, and the David J. Hurley Air Traffic Control System Command Center. At these locations, traffic management controllers input information from air traffic controllers and system customers into decision-making processes to relay system demand, rerouting, and Traffic Management Initiatives (TMI) to air traffic controllers.

Current rerouting procedures and TMIs are used for creating adequate separation between aircraft or addressing weather concerns. They include airborne holding, sequencing programs (departure, en route, and arrival), capping, tunneling, ground delay programs, time-based flow management (TBFM), traffic management advisor, airspace flow programs, integrated collaborative rerouting, the use of routes, the severe weather avoidance plan, and the route availability planning tool (RAPT). Similar to RAPT which aids air traffic managers during severe weather, a decision support tool should be created to reduce air traffic controller workload. Rather than defining restricted regions, this tool would provide air traffic controllers with commands for affected aircraft during a launch. The research presented in this thesis
provides the methodology to develop a decision support tool that would provide air traffic controllers commands to reroute air traffic during a launch.

While current rerouting procedures and TMIs are used for aircraft separation or weather, similar principles can be used for rerouting during space launches. The current methods can be divided into two rerouting methods: in space and in time. This thesis explores providing air traffic managers with a decision support tool that has commands from both methods.

1.2 Decision Making Under Uncertainty

In order to create a decision support tool, there first needs to be a decision making tool that takes inputs from the environment to provide actions that move the system towards a predetermined objective. After each action, there are updated inputs from the environment and the tool makes a new decision. Often, the environmental inputs and how the system responds to an action are uncertain, requiring a robust method to make decisions under uncertainty.

The remainder of this section discusses common space launch optimization objectives, introduces Markov decision processes (MDPs), and discusses the advantages and disadvantages of MDPs.

1.2.1 Optimization Objective

Annually, the FAA handles over 16,100,000 flights, roughly 44,000 flights a day, and at peak operational times, 5,000 aircraft in the air at once. On average, 2,789,971 passengers fly in and out of United States airports daily [15], which is about 0.85% of the United States’ population. With so many people flying, it is paramount that the FAA prioritizes safety as the main objective when designing new air traffic tools, systems, and procedures.

The airlines interacting with the FAA are safety focused but also profit driven. In 2017, the North American airline industry had a 15.6 billion dollar net profit [16]. To remain profitable, new air traffic tools, systems, and procedures should be efficient
to reduce unnecessary expenses. Some of these additional expenses are added fuel, added labor, additional maintenance, and missed connections. Besides the monetary concerns, environmental factors encourage efficiency.

With the dual demands of safety and efficiency, an air traffic decision making tool is designed with multi-objective optimization. Overall, both safety and efficiency will contribute to a reward (or in the negative, a cost) structure. In general, a safer system will be less efficient and a more efficient system will be less safe so a tradeoff has to be made to achieve the desired system performance. To understand the tradeoff, different reward weighting values can be implemented and tested on a validation data set to find the weighting that achieves the desired system performance. To visualize this, the safety and efficiency results can be plotted to create a tradeoff curve (sometimes referred to as a Pareto frontier) [17]. An example toy problem considers 100 simulations where safety and efficiency are binary variables. The aggregate results for those 100 simulations are the counts of how many simulations were safe and how many simulations were efficient. A representative tradeoff curve for this example is presented in Figure 1.3 where the key indicates a sweep of how the efficiency reward is weighted.

### 1.2.2 Markov Decision Process

In general, air traffic management is a sequential problem. When working with space launches there is an added component of uncertainty. For instance, whether or not the launch will have an anomaly and scatter pieces of debris across the NAS. Based on these parameters, the problem is modeled as a Markov decision process (MDP), which is ideal for a sequential decision making problem with stochasticity.

At each time step, an agent, chooses an action based on the observed state. Based on the current state and selected action, the state evolves probabilistically. The agent receives a reward based on the current state, selected actions, and sometimes the evolved state. An MDP is fully defined by the following:

- **S** is the state space, a set that contains all possible states.
- **A** is the action space, a set that contains all possible actions.
• $T$ is the transition model. Given the current state $s$ and action $a$, the probability of transitioning to some new state $s'$ is given by $T(s' \mid s, a)$. This transition model captures uncertainty in how the states will evolve.

• $R$ is the reward model. Given the current state $s$ and action $a$, the immediate reward is given by $R(s, a)$.

A dynamic Bayesian network representation of the MDP components is shown in Fig. 1.4. The round nodes indicate state variables, the square node represents the action variable, and the diamond node represents the reward variable. The arrows indicate direct (sometimes probabilistic) dependencies [18].

An example of a small MDP is shown in Fig. 1.5. This example has three states (1, 2, and 3) and two potential actions available at each state (A and B). For each state-action pair, the next state is determined probabilistically as denoted by the probability on the line connecting the action to next state such as 90% of the time taking action A from state 1 results in transitioning to state 2 and 10% of the time taking action B from state 1 results in transitioning to state 3.
results in staying in state 1. The values associated with the orange arrows denote the rewards (or costs) for transitioning between two states with a specific action such as the +1 reward from taking action A in state 2 and transitioning to state 1. If there is no orange arrow, the reward is 0. An optimal solution to an MDP is a policy that maximizes the accumulation of the expected rewards (or minimizes the accumulation of expected costs) when followed.

This generic framework can be adapted and used to solve a variety of problems including multi-objective problems such as rerouting aircraft during space launches,
CHAPTER 1. INTRODUCTION

which rewards both safety and efficiency.

1.2.3 MDP Advantages and Disadvantages

An MDP models a sequential problem with inherent stochasticity. Additionally, an MDP can be solved offline allowing the results to be stored and quickly queried in real time. If an anomaly occurs, it is important that a decision support tool is fast and able to aid an air traffic controller in rerouting all of the affected flights. Further, the FAA has rigorous algorithm approval processes. The FAA has experience working with safety critical systems derived from MDP formulations [19].

There are challenges with MDPs. Mainly, MDPs are subject to the curse of dimensionality: the solution space grows exponentially with the number and size of the state variables [20]. For this reason, continuous state space variables are often discretized and the discretization is driven by computational tractability. Further, the model needs to be accurate and have a specified reward. Finally, an MDP does not handle imperfect state space information, and therefore a partially observable Markov decision process (POMDP) should be used instead [21].

1.3 Contributions

This thesis presents rerouting aircraft during a space launch as an MDP to find a solution that could be implemented into a decision support tool for air traffic controllers to use during launches to safely reroute aircraft.

Chapter 2 describes the scenario and introduces the models. The scenario represents a launch that can be used to determine model parameters and used as a proof of concept to understand how the proposed results would affect the airspace. Two contributions are modeling the problem as an MDP where the optimal actions are heading changes and metering aircraft through speed changes.

MDP solution techniques are presented in Chapter 3. The chapter provides a brief overview of general solution techniques and discusses the implemented solution techniques. This chapter contributes an algorithm called adaptive spatial discretization
for backward induction value iteration that focuses on the state space around the region of interest. This approach allows a finer discretization in areas most relevant to determining the optimal action, often in safety critical regions.

Chapter 4 presents qualitative and quantitative results of the solution techniques. Utility plots, policy plots, and discussion are provided to validate the model solutions. A simulation framework was developed and used to generate aggregate metrics that quantify the model solutions performance. The simulation framework is the final thesis contribution.

The thesis concludes with Chapter 5 which presents a summary and future work. Much of the work in this thesis has been previously published in various venues [22]–[26].
Chapter 2

MDP Formulation

To assess the impact of dynamic aircraft rerouting during a space launch, a representative scenario is modeled as an MDP. This chapter defines the scenario, introduces rerouting methodologies, and describes how the scenario is modeled as an MDP for both heading and metering actions.

2.1 Scenario

This thesis focuses on a hypothetical scenario involving oceanic air traffic at 35,000 ft during a space launch from Kennedy Space Center. While this thesis focuses on a single launch scenario the methods and solution approaches are also applicable for other launch scenarios as well as reentries. For the remainder of this thesis, all coordinates are provided in an east, north, up reference frame centered at the Kennedy Space Center. The aircraft are modeled as Boeing 777-200 aircraft with a fixed cruise speed of Mach 0.84 [27]. This thesis focuses on aircraft traveling and interacting with debris only at 35,000 ft.

The launch vehicle follows a representative two-stage to orbit trajectory [28] shown in Fig. 2.1. The possibility of an anomaly and its resulting debris field is considered every ten seconds during the launch.
2.1.1 Debris Modeling

During an anomaly, the debris is modeled using a scaled version of the Columbia disaster debris catalog [29] and atmosphere profiles for the month of October provided by MIT Lincoln Laboratory. For each half kilometer interval from 1 km to 25 km, the profiles contain wind speed, wind direction, and air density. Above this height, the wind speed is assumed to be zero and air density is assumed to have a continuous exponential decay with increasing altitude.

The debris catalog separates the debris into 11 categories based on weight, ballistic coefficient, and size. This model includes the expected number of pieces of debris for each category. Figure 2.2a shows the trajectories of all eleven types of debris for an anomaly at 100 seconds after launch, and Fig. 2.2b shows the trajectories of one type of debris for various times of anomaly.

The debris information, launch vehicle information, time of anomaly, and atmosphere profiles are input to the Range Safety Assessment Tool (RSAT) to generate
an estimated trajectory for the debris [28]. The initial position of the debris matches the position of the launch vehicle when the anomaly occurs. The impulse velocity of the debris is uniformly selected from 0 m/s to 100 m/s. Because many of the pieces of debris have similar trajectories, only a representative set are modeled [9]. The various MDP formulations use the following sets of debris profiles:

- For the MDP formulations that consider all the debris at once, 50 debris profiles are used in total: 25 debris profiles are used when solving the MDP, five debris profiles are used for parameter tuning as discussed in Section 4.2.4, and 20 profiles are used when calculating the simulation results. A logarithmic histogram of the debris locations for the 25 debris profiles used when solving the MDP at 35,000 ft is shown in Fig. 2.3.
• For the MDP formulation that look at a single piece of debris at a time, only three debris profiles are used: one debris profile is used when solving the MDP, one is used for parameter tuning as discussed in Section 4.2.4, and one is used when calculating the simulation results.

![Logarithmic histogram of potential debris locations](image)

Figure 2.3: Logarithmic histogram of potential debris locations

### 2.1.2 Debris Avoidance Methodologies

The scenario considers aircraft flying at 35,000 ft. In order to resolve an interaction between an aircraft and debris, the aircraft trajectory can be adjusted to avoid the debris in either space or time. The heading action MDP was formulated to produce optimal actions that should adjust the trajectory in space to avoid debris as shown in Fig. 2.4 where the blue circles represent debris, the black path represents the nominal trajectory, and the teal path represents the rerouted trajectory. A separate metering MDP was formulated to produce optimal actions that provide speed commands to adjust the trajectory in time to avoid debris as shown in Fig. 2.5 where the blue rectangles represent debris, the black path represents the nominal trajectory, and the teal path represents the rerouted trajectory.
2.2 Heading Actions

The initial approach of rerouting aircraft during a space launch was modeled after severe weather rerouting procedures [30]. During severe weather, air traffic control (ATC) instructs pilots to reroute their aircraft around the active weather system. This can be done by providing a change of heading or direction of deviation. During space launches, the same process should apply except instead of ATC providing reroute commands around weather, they would provide reroute commands around potential debris locations. The goal of this thesis is to provide the methodology to build a decision support tool for ATC during space launches. This section describes how to model the problem as an MDP as a step in developing a decision support tool that provides heading commands.

2.2.1 State Space

A state $s \in S$ has five components: aircraft east position ($e \in E$), aircraft north position ($n \in N$), aircraft heading ($\psi \in \Psi$), time of anomaly ($t_{anom} \in T_{anom}$), and
CHAPTER 2. MDP FORMULATION

Time since launch ($\delta t \in \Delta T_{\text{launch}}$). The state space is discretized into a five-dimensional grid.

The $e$ and $n$ coordinates are defined in an east, north, up reference frame centered at Kennedy Space Center. The coordinates are found experimentally as the extent of the airspace that would result in rerouted flights. The details of this procedure are discussed in Section 3.4.2. There are four experimented discretizations for $E$ and $N$. The first set of discretizations are based on potential state space size limitation of 50, 75, and 100 million states. The final discretization is selected to create a comparison policy for an innovative solution technique and uses a matching state space discretization.

To match a standard rate and half standard rate turn, $\Psi$ is discretized at $15^\circ$ intervals (the MDP is solved in 10 second increments). The state variable $t_{\text{anom}}$ denotes when an anomaly has occurred. In this formulation, $t_{\text{anom}} < 0$ indicates no anomaly. The variable $t$ is the time since the launch. The modeled range of $t$ is sufficient to capture debris that would cross 35,000 ft within 10 min of anomaly and the modeled range of $t_{\text{anom}}$ is sufficient to create debris that would cross 35,000 ft within

Figure 2.5: Example of metering actions avoiding debris in time
Table 2.1: Fixed discretization aircraft state space variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Minimum</th>
<th>Steps</th>
<th>Maximum</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$</td>
<td>-83,474.5</td>
<td>11, 43, 53, 61</td>
<td>138,866.2</td>
<td>m</td>
</tr>
<tr>
<td>$N$</td>
<td>-99,476.5</td>
<td>11, 43, 53, 61</td>
<td>119,351.2</td>
<td>m</td>
</tr>
<tr>
<td>$\Psi$</td>
<td>-180</td>
<td>25</td>
<td>180</td>
<td>°</td>
</tr>
<tr>
<td>$T_{\text{anom}}$</td>
<td>-10</td>
<td>13</td>
<td>110</td>
<td>s</td>
</tr>
<tr>
<td>$\Delta T_{\text{launch}}$</td>
<td>0</td>
<td>82</td>
<td>810</td>
<td>s</td>
</tr>
</tbody>
</table>

10 min of anomaly. A summary of the variables and their modeled discretization are presented in Table 2.1.

### 2.2.2 Action Space

An action $a \in A$ is the heading advisory to be provided to the pilots. The model considers both standard rate ($3^\circ/s$) and half standard rate ($1.5^\circ/s$) turns. The resultant action space contains the following commands: strong left ($-30^\circ$), weak left ($-15^\circ$), maintain ($0^\circ$), no advisory (NIL), weak right ($15^\circ$), and strong right ($30^\circ$).

### 2.2.3 Reward Model

There is a (negative) reward of $r_{\text{safe}}$ when an aircraft is either a distance $d$ less than $\delta$ from a launch vehicle or an RSAT modeled piece of debris. This formulation uses $\delta = 10 \times d_{\text{NMAC}}$ where $d_{\text{NMAC}} = 500$ ft so that $\delta$ matches an inflated version of aircraft regulations that define a near midair collision (NMAC) between two aircraft [31]. To account for uncertainty, the safety region for a piece of debris is active for its time of anomaly $\pm 10$ s and from the time it intersect 35,000 ft $\pm 20$ s. Path deviations result in a (negative) reward of $r_{\text{eff}}$. Since the state space does not require a flight plan, $r_{\text{eff}}$ are modeled by the commanded action $a$. The total reward is given by $r_{\text{safe}} + \lambda r_{\text{eff}}$, where $\lambda$ controls the tradeoff between safety and efficiency.

Table 2.2 summarizes the reward parameters where $d_{lv}$ and $d_{deb}$ are the distance from the aircraft to the launch vehicle or closest piece of debris, respectively.
Table 2.2: Reward parameters

<table>
<thead>
<tr>
<th>Safety</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(d_{lv} \leq \delta)</td>
<td>-1</td>
</tr>
<tr>
<td>(d_{lv} &gt; \delta)</td>
<td>0</td>
</tr>
<tr>
<td>(d_{deb} \leq \delta)</td>
<td>-1</td>
</tr>
<tr>
<td>(d_{deb} &gt; \delta)</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Efficiency</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(a = \text{NIL})</td>
<td>0</td>
</tr>
<tr>
<td>(a = 0^\circ)</td>
<td>-0.01</td>
</tr>
<tr>
<td>(a = \pm 15^\circ)</td>
<td>-0.5</td>
</tr>
<tr>
<td>(a = \pm 30^\circ)</td>
<td>-1</td>
</tr>
</tbody>
</table>

### 2.2.4 Transition Model

This model captures uncertainty in the launch vehicle trajectory, aircraft trajectory, potential debris, data turnaround time, and pilot response. The data turnaround time encompasses the time it takes for a launch provider to determine an anomaly has occurred, provide that information to the air traffic control, and the air traffic controllers to relay the information to the pilot. The transition model depends on the advisory command \(a\), which accounts for the data turnaround time and pilot response rate. If the current state heading is \(\psi\) and the next state heading is \(\psi'\), then \(P(\psi' \mid \psi, a)\) is defined as

\[
P(\psi' \mid \psi, a) = \begin{cases} 
1 & \text{if } a = 0^\circ \\
0.4 & \text{if } a = \text{NIL} \text{ and } \psi' = \psi \\
0.5 & \text{if } a = \text{NIL} \text{ and } |\psi' - \psi| = 15^\circ \\
0.1 & \text{if } a = \text{NIL} \text{ and } |\psi' - \psi| = 30^\circ \\
0.5 & \text{if } a \neq \text{NIL}, a \neq 0^\circ \text{, and } \psi' = \psi + a \\
0.5 & \text{if } a \neq \text{NIL}, a \neq 0^\circ \text{, and } \psi' = \psi \\
0 & \text{otherwise}
\end{cases}
\]  

(2.1)
When $a = 0$, the aircraft always continues straight. When $a = \text{NIL}$, the aircraft follows its current heading 40% of the time, turns weak left 25% of the time, turns weak right 25% of the time, turns strong left 5% of the time, and turns strong right 5% of the time. The NIL action distribution models the uncertainty in aircraft path. When $a \neq \text{NIL}$ the aircraft follows a model similar to [32] based on the pilot response rate of TCAS and has the pilot respond to the commanded action 50% of the time, resulting in an average data turnaround time and pilot response rate of 20s. The east and north position updates are deterministic with their current position, $\psi'$, and cruise velocity, $s_{ac}$, is modeled as Mach 0.84. The time of anomaly only updates if $t_{\text{anom}} < 0$. For the scenario, when $t_{\text{anom}}$ is updated, it updates to $t$ at the next time step with a probability 0.052, otherwise it remains as no anomaly, modeled as less than 0. This structure ensures that an anomaly can only occur once during a launch and results in a 50% chance of anomaly during the first stage of the launch. The probability $p$ is the solution of the cumulative distribution function for a geometric distribution

$$P = 1 - (1 - p)^k$$  \hspace{1cm} (2.2)

where $P$ is the overall probability of failure and $k$ is the number of potential failure timesteps in the first stage of launch. The time variable increments by $t_{\text{step}} = 10$ s at each step. The problem terminates when $t = 810$ s.

### 2.3 Metering Actions

The formulation in Section 2.2 requires aircraft to maintain communication with air traffic controllers. During oceanic operations there are regularly long spans of time when the aircraft do not communicate with ATC. Air traffic controllers relay waypoints and arrival times to aircraft rather than specific flight headings [33]. To address oceanic operations, this section describes how to model the problem to develop a decision support tool that meters when aircraft reach predefined waypoints.
2.3.1 State Space

A state \( s \in S \) has five components: current waypoint \((w \in W)\), aircraft speed \((s_{ac} \in S_{ac})\), time into flight \((t_{flight} \in T_{flight})\), time of launch \((t_{launch} \in T_{launch})\), and time of anomaly \((t_{anom} \in T_{anom})\). The state space is discretized into a five-dimensional grid.

The oceanic air traffic is considered to follow a series of waypoints and is only given air traffic control (ATC) commands at waypoints, which means commands can be given over an hour apart. The ATC commands meter when aircraft arrive at waypoints. Waypoints are published in the North Atlantic Route Planning Chart [34]. Figure 2.6 shows waypoints near Florida. To initialize the state space, a path of waypoints \( w_1, \ldots, w_n \) is specified. Each path is selected such that it will interact with potential pieces of debris.

The variable \( s_{ac} \) ranges from 283.0 m/s to 293.0 m/s in 2.5 m/s intervals. The variable \( t_{flight} \) is the time that an aircraft can be at a waypoint and is discretized in 5 s intervals. Time of launch \( t_{launch} \) is the time a launch occurs and is discretized in 5 s intervals. The state variable \( t_{anom} \) denotes when an anomaly has occurred and is discretized in 10 s intervals. When \( t_{anom} \) is less than 0, no anomaly has occurred.

For each path, the values of \( t_{anom} \) are computed and indicate values that produce
Table 2.3: Reward parameters

<table>
<thead>
<tr>
<th>Safety</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$d \leq \delta$</td>
<td>-1.0</td>
</tr>
<tr>
<td>otherwise</td>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Efficiency</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$a = 0^\circ$</td>
<td>0.0</td>
</tr>
<tr>
<td>$a = \pm 2.5 \text{ m/s}$</td>
<td>-0.5</td>
</tr>
<tr>
<td>$a = \pm 5.0 \text{ m/s}$</td>
<td>-1.0</td>
</tr>
</tbody>
</table>

debris that comes within the safety threshold of the aircraft path. This set is denoted $T_{\text{anom}}$. For each $t_{\text{anom}}$ the minimum and maximum time during a flight path where an aircraft could interact with debris is determined and denoted $t_{\text{min}}$ and $t_{\text{max}}$. To increase robustness, a 20 s buffer is subtracted from $t_{\text{min}}$ and added to $t_{\text{max}}$. The set of values of $t_{\text{launch}}$ that causes the aircraft path to intersect the debris safety threshold, denoted $T_{\text{launch}}$, spans from $t_{\text{anom}} + t_{\text{min}}$ to $t_{\text{anom}} + t_{\text{max}}$.

2.3.2 Action Space

An action $a \in A$ is the speed advisory provided to the pilots that meters when the aircraft should arrive at the next waypoint. The model considers both large and small speed advisories. The resultant action space contains the following commands: large speed up ($+5 \text{ m/s}$), small speed up ($+2.5 \text{ m/s}$), maintain speed ($+0 \text{ m/s}$), small slow down ($-2.5 \text{ m/s}$), and large slow down ($-5 \text{ m/s}$).

2.3.3 Reward Model

The metering MDP formulation uses the (negative) reward model presented in Section 2.2.3 with the parameters summarized in Table 2.3.
2.3.4 Transition Model

This model captures uncertainty in the launch vehicle trajectory, aircraft trajectory, potential debris, data turnaround time, and pilot response. The transition model is defined for each state component. The current waypoint $w_i$ always increments to the next waypoint in the path $w_{i+1}$. The $s_{ac}$ update depends on the advisory command $a$, which accounts for the data turnaround time and pilot response rate. If the current state speed is $s_{ac}$ and the next state speed is $s'_{ac}$, then $P(s'_{ac} | s_{ac}, a)$ is defined as

$$P(s'_{ac} | s_{ac}, a) = \begin{cases} 1 & \text{if } a = 0 \text{ and } s'_{ac} = s_{ac} \\ 0.5 & \text{if } a \neq 0 \text{ and } s'_{ac} = s_{ac} + a \\ 0.5 & \text{if } a \neq 0 \text{ and } s'_{ac} = s_{ac} \\ 0 & \text{otherwise} \end{cases}$$  

(2.3)

The probabilities above account for whether or not the pilot will respond to the given action. The time at the next state $t'_{flight}$ is updated by adding $t_{flight}$ to the leg distance divided by $s'_{ac}$. While the straight line distance between two waypoints is known and fixed, it is likely that the aircraft will deviate slightly from the straight line path. If the straight line distance between the two waypoints in meters is denoted $d_s$ and the leg distance used to updated $t_{flight}$ is denoted $d_u$ then $P(d_u | d_s)$ is defined as

$$P(d_u | d_s) = \begin{cases} 0.4 & d_s \times 1.0 \\ 0.3 & d_w \times 1.01 \\ 0.2 & d_w \times 1.015 \\ 0.1 & d_w \times 1.02 \\ 0 & \text{otherwise} \end{cases}$$  

(2.4)

This formulation considers a single launch so $t_{launch}$ stays constant. A separate MDP is run for each $t_{anom}$ and an anomaly is always considered to occur, resulting in conservative actions when an anomaly has not yet occurred or never occurs.

The reward function is dependent on the transition function to determine whether
the aircraft will intersect the debris safety threshold. The new $s_{ac}$ and $t_{anom}$ are used to query whether the aircraft is within the safety threshold of debris between waypoints. To do this, the aircraft is modeled as flying at $s'_{ac}$ on the straight line path between waypoints. The aircraft position is updated and checked every 5 s to determine whether it intersects the debris safety threshold.

2.4 Discussion

This chapter presented modeling the problem of rerouting aircraft during a space launch with both a heading and metering MDP. The model selection and design were driven by two main design principles. First, the model strives to accurately represent the scenario to have more efficient airspace allocation during a launch without an adverse effect on safety. Second, the model is designed to have a solution that can be the foundation for an easily adaptable decision support tool. This section discusses how the models were inspired by the main design principles.

This chapter defined the problem, including everything from launch location, which affects weather and how debris will travel to the at-risk aircraft and its location and speed. Once the scenario was defined, the exact model was selected. Based on the sequential and stochastic nature of the problem MDPs were selected.

The MDP models make a few assumptions. The states are discretized while the actual problem is continuous. It is assumed that there is perfect state information, so the relative position of an aircraft and pieces of debris is known. Current space launch operations do not provide perfect state information but there is a push to update the FAA’s technology and information sharing during a launch to improve the available information [35]. Finally, based on the state and action space limitations associated with MDPs, discussed further in Chapter 3, the heading action formulation is constrained to initial rerouting commands associated with a launch rather than commands for the entire time after an anomaly would have occurred.

In terms of the desired solution, the MDPs were designed to have actions that are similar to what air traffic controllers are familiar with. This was done to be consistent with what is already done and encourage easy integration and adoption
into air traffic management. The heading action formulation matches some of the procedures used during weather events in the national airspace. The metering action formulation matches some of the procedures used with oceanic flights. Further, the MDP model was selected for its solution outputs and techniques, which are described in the next chapter.
Chapter 3

Solving MDPs

Once we have a model in place, the MDP is solved for a policy, which provides the preferred action to take at each state. This chapter begins by introducing the principals of rational decision making that drive solving for the optimal utility function or optimal policy. The chapter introduces solution techniques, including adaptive spatial discretization.

3.1 Rational Decision Making

Using rational preferences, a utility function assigns a real-valued utility or preference for every state. The utility function \( U'(s) \) is a mapping of state \( s \) to value. Rational decisions are made by following the principle of maximum expected utility [36], which assigns the optimal action \( a \) in the set of possible actions \( A \) to

\[
\arg\max_{a \in A} \sum_{s' \in S} T(s' \mid a, s) U(s'),
\]

where \( s \) is the current state, \( s' \) is the next state, \( S \) is the set of possible states, \( T(s' \mid a, s) \) is the transition probability of moving from \( s \) to \( s' \) by taking \( a \). This can be extended to a series of decisions for sequential problems such as an MDP. An optimal policy provides an \( a \) for each \( s \) that maximizes the expected sum of utilities. For an MDP, the optimal utility at each state denoted \( U^*(s) \) is defined by the Bellman
equation [37] which states that
\[
U^*(s) = \max_{a \in A} \left[ R(s, a) + \sum_{s' \in S} T(s' \mid s, a) U^*(s') \right] 
\] (3.2)

where \(R(s, a)\) is the immediate reward for taking \(a\) in \(s\). The optimal policy denoted \(\pi^*(s)\) is defined as
\[
\pi^*(s) = \arg\max_{a \in A} \left[ R(s, a) + \sum_{s' \in S} T(s' \mid s, a) U^*(s') \right] 
\] (3.3)

There are a variety of solution techniques that solve for \(U^*(s)\) or \(\pi^*(s)\).

3.2 Solution Techniques

There are exact and approximate offline as well as online solution techniques for solving MDPs. This section provides an overview of a few solution techniques.

3.2.1 Exact Offline Solution Techniques

Dynamic programming [37] can be used to solve for \(U^*(s)\) exactly offline via value iteration. Value iteration starts by initializing the utility function and then iteratively solves for the utility for each state until convergence. Similarly, policy iteration can be used to solve for \(\pi^*(s)\). Policy iteration starts by initializing a policy and then iterates between policy evaluation and policy improvement until the policy converges [38]. Value and policy iteration learn how future states affect current states through their iterative learning and therefore are considered closed-loop planning. Further, the results from value and policy iteration can be stored and quickly queried in real time.

Value iteration is used in this thesis because it is more straightforward to implement than policy iteration. The value iteration algorithm is detailed at length in Section 3.3. Further, there are many variations of value iteration. Among others, asynchronous value iteration only updates some of the states during each iteration.
Gauss-Siedel value iteration updates the utility function in place for states in a specified order, and backward induction value iteration (BIVI) can be used when there is a time state variable and solves the problem backward in time, iteration only once. BIVI is explained further in Section 3.3.1 [39].

3.2.2 Approximate Offline Solution Techniques

Both value iteration and policy iteration require a discrete state and action space. Since some problems have a continuous state space the solutions are approximated by discretizing their state spaces and using interpolation.

Further, as the MDP problem size increases, it can be computationally intractable to solve for the utility function or policy. This issue is referred to as the curse of dimensionality [40]. One method to address this issue is to use local approximation value iteration with a more coarsely discretized state space. There are also variable resolution techniques for fixed state space limits [41], [42]. Section 3.4 introduces adaptive spatial discretization that dynamically focuses the state space discretization on the region of interest.

While local approximation uses a set of representative states to approximate the utility at arbitrary states, global approximation uses a set of parameters to approximate the utility at all states. Global approximation techniques include linear regression and deep learning. Both of these techniques work in the continuous domain and do not require a discretized state space. Linear regression approximates the value function using a linear combination of a fixed set of parameters and basis functions [18]. Similarly, deep reinforcement learning can be used to solve for a utility function or policy [43], [44]. A discussion on how deep learning was applied to rerouting aircraft during launches and the resulting limitations is presented in Section 5.2.

3.2.3 Online Solution Techniques

For the offline methods, the utility function or policy is solved for the entire state space before any decisions are actually made. Online techniques only reason about states reachable from the current state. These methods can handle a larger state
space, require less storage, but involves more computation during decision making.

Forward search and branch and bound are examples of online solution techniques. Forward search is conducted from an initial state by exploring the potential state-action combinations for a predetermined number of states in the future. From this exploration, the best action is selected. This method is optimal through the predetermined number of states and may lack knowledge about states further in the future.

To speed up forward search, branch and bound prunes portions of the state-action search tree based on precomputed lower and upper bounds [45]. Other methods include sampling methods such as sparse sampling [46] and Monte Carlo tree search and are not guaranteed to provide the optimal action [47].

### 3.2.4 Offline vs. Online Tradeoff

There are advantages and disadvantages to both offline and online techniques. The main advantage of offline techniques is that they solve the problem ahead of time for the entire state space, so when the solution is needed it can be easily queried. The main disadvantage with offline techniques is scalability.

The main advantage of online techniques is that they can accommodate a larger state space because they only explore the state space reachable from the current state. The main disadvantages of online techniques are that they need to be calculated in real time, which can be slow, and that the runtime of the calculations are based on the reachable state space and search depth; so, more informed solutions often take more time.

Moving forward, the selected solution method needs to meet the demands of a decision support tool that could be used in real time by air traffic controllers during a space launch. This means that the solution technique needs to provide fast and safe rerouting commands for a constantly evolving system. Offline approximate techniques were selected to create solutions with all future knowledge (such as relevant debris information) and to address the operational speed concern. In the future, algorithm and computational advancements could make online solutions feasible in real time. Further, approximately optimal solutions are often sufficient to achieve the desired
safety and efficiency.

3.3 Value Iteration

This section explains how value iteration and its variations were implemented to solve the MDP models presented in this thesis. The state-action utility function $Q^*(s,a)$ was used instead of the state utility function $U^*(s)$ because it provides the value of each state action pair. All of the $Q^*(s,a)$ utilities are initialized to 0 and the state-action version of the Bellman equation is defined as

$$Q^*(s,a) = R(s,a) + \sum_{s' \in S} T(s' \mid s, a) \max_{a' \in A} Q^*(s', a')$$  (3.4)

where the state-action utility function $Q^*(s,a)$ is the expected sum of rewards when $a$ is taken from $s$ and then an optimal policy is followed. Similar to the state utility function, the state-action utility function can be iteratively solved for each state until convergence. Once $Q^*(s,a)$ is computed for, an optimal policy can be extracted

$$\pi^*(s) = \arg\max_{a \in A} Q^*(s,a)$$  (3.5)

Since $Q^*(s,a)$ involves iterating over discrete state and action spaces, multilinear interpolation is used to estimate $Q^*(s,a)$ at states that are not part of the original discretization.

3.3.1 Backward Induction Value Iteration

To reduce the number of iterations required until the state-action utility function converges, BIVI is used. The goal of rerouting aircraft during a space launch is to avoid debris if an anomaly occurs. All the information needed to make the best rerouting decisions comes from the information about whether an anomaly occurred and, if so, the location of the pieces of debris. The problem has a finite horizon and time in the state space. Using BIVI, the states are ordered from the final timestep (after the final piece of debris has fallen through the altitude threshold) to the initial
timestep (interacting with various pieces of debris along the way) and $Q^*(s, a)$ can be solved in exactly one iteration. This optimization is possible because the ordering allows the solution to use information about future states to compute the state-action utility function at the current state.

The algorithm for BIVI is outlined in Algorithm 1. First, $Q^*(s, a) = 0$ is initialized for all states that occur at the final time step $t_h$. The subset of states that may occur at a specific time $t$ is denoted $S(t)$ and the time variable increments by $t_{step}$. Then, $Q^*(s, a)$ for all the states that occur at $t_h - t_{step}$ can be computed exactly using the Bellman equation. This is continued until $t = 0$.

<table>
<thead>
<tr>
<th>Algorithm 1 BIVI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: function \textsc{BackwardInduction}($S, A, R, T, t_h, t_{step}$)</td>
</tr>
<tr>
<td>2: \hspace{1em} $t \leftarrow t_h$</td>
</tr>
<tr>
<td>3: \hspace{1em} for $a \in A$, $s \in S(t)$</td>
</tr>
<tr>
<td>4: \hspace{1em} $Q^*(s, a) \leftarrow 0$</td>
</tr>
<tr>
<td>5: \hspace{1em} while $t &gt; 0$</td>
</tr>
<tr>
<td>6: \hspace{1em} \hspace{1em} $t \leftarrow t - t_{step}$</td>
</tr>
<tr>
<td>7: \hspace{1em} \hspace{1em} for $a \in A$, $s \in S(t)$</td>
</tr>
<tr>
<td>8: \hspace{1em} \hspace{1em} \hspace{1em} $Q^*(s, a) \leftarrow R(s, a) + \sum_{s' \in S} T(s'</td>
</tr>
<tr>
<td>9: \hspace{1em} return $Q^*$</td>
</tr>
</tbody>
</table>

BIVI is the basis for the solution techniques used to solve aircraft rerouting during a space launch throughout this thesis. A simplified example of how BIVI is used focuses on a subset of the state space with a single piece of debris that falls through the altitude threshold at $t = 100$ s. Figure 3.1a shows 35 cells representing 35 states that exist in a subset of the possible aircraft east and north states. The star represents the state where the aircraft is within the safety threshold of the debris. At $t = 100$ s, only the cell containing the debris has a large cost due to the safety threshold violation, as represented by the red in Fig. 3.1a.

We can solve for the utilities of the states one time step earlier, $t = 90$ s. For the example, we will first do this for an aircraft located in the grid cell immediately to the left of the debris, flying towards the debris ($\psi = 0^\circ$), with the limited commands: right, left, and straight. Figure 3.1b shows the available actions and arrows from
(a) Subset of state space at $t = 100\text{ s}$

(b) Subset of state space at $t = 90\text{ s}$ showing the Transition Function

(c) Example policy at $t = 90\text{ s}$

(d) Example policy at $t = 80\text{ s}$

Figure 3.1: Example of solving for an optimal policy for a single piece of debris
those actions with width proportional to the probability of moving to the specific next state. The colors of the actions relate to the sum of their immediate costs and expected sum of future costs. The straight command is red because it has a large cost for landing in the grid cell of the debris. The right and left turn commands are orange because they are not as costly as going straight into debris, but due to the transition function they might still encounter the debris, and turning has a cost. For this scenario, left and right would have the same outcome, so we select left without loss of generality. This process is done for all of the states at this time step. The policy for each state in the grid with $\psi = 0^\circ$ is shown in Fig. 3.1c.

Using this information, the process is repeated for one time step earlier, $t = 80$ s, and is shown in Fig. 3.1d. At this time step, some grid cells are yellow to represent the cost of the action and the sum of future rewards, which is less than those of the orange grid cells. This process is repeated until $t = 0$ s. While this is a simple example, the same process is extended to solve for the whole state space and all of the debris.

### 3.4 Adaptive Spatial Discretization

As previously mentioned, using value iteration is subject to the curse of dimensionality. Backward induction uses the structure of the problem to optimize value iteration. To further optimize the solution and break the curse of dimensionality, instead of using a fixed discretization for all of the debris at each time step, the problem is decomposed by pieces of debris and solved using adaptive spatial discretization for backward induction value iteration (ASD). This section will outline the ASD algorithm and how the problem can be solved using decomposition.

#### 3.4.1 Adaptive Spatial Discretization Algorithm

ASD uses the information about future states to dynamically focus the state space discretization on the region of interest. In order to dynamically focus on the smallest subset of the state space at a time, the problem is decomposed by piece of debris,
solved individually, and then recombined. The ASD algorithm for a single piece is outlined in Algorithm 2. Initially, the discretization is assigned a uniform \( x_{\text{steps}} \) by \( x_{\text{steps}} \) grid. The grid is centered on \((e_{\text{deb}}, n_{\text{deb}})\), which corresponds to the location the debris or where the launch vehicle passes through the altitude threshold. The east and north extents of the grid are \([e_{\text{deb}} - D_{\text{buffer}}, e_{\text{deb}} + D_{\text{buffer}}]\) and \([n_{\text{deb}} - D_{\text{buffer}}, n_{\text{deb}} + D_{\text{buffer}}]\), where \( D_{\text{buffer}} = \delta + s_{\text{ac}} \times t_{\text{step}} \). The \( E \) and \( N \) state space is defined in Algorithm 2 using the \texttt{linspace} function, which creates an array of \( x_{\text{steps}} \) linearly spaced elements from the east and north grid extents. These limits capture the region where an aircraft could potentially require rerouting.

\begin{algorithm}
\caption{ASD}
\begin{algorithmic}[1]
\Function{ASD}{$S, A, R, T, t_h, t_{\text{step}}, e_{\text{deb}}, n_{\text{deb}}, D_{\text{buffer}}$}
\State \( t \leftarrow t_h \)
\State \( e_{\text{init}}, e_{\text{max}} \leftarrow e_{\text{deb}} - D_{\text{buffer}}, e_{\text{deb}} + D_{\text{buffer}} \)
\State \( n_{\text{init}}, n_{\text{max}} \leftarrow n_{\text{deb}} - D_{\text{buffer}}, n_{\text{deb}} + D_{\text{buffer}} \)
\State \( E \leftarrow \text{linspace}(e_{\text{init}}, e_{\text{max}}, x_{\text{steps}}) \)
\State \( N \leftarrow \text{linspace}(n_{\text{init}}, n_{\text{max}}, x_{\text{steps}}) \)
\State \( S(t) \leftarrow E \times N \times \Psi \times T_{\text{anom}} \)
\For{$a \in A, s \in S(t)$}
\State \( Q^*(s, a) \leftarrow 0 \)
\EndFor
\While{$t > 0$}
\State \( t \leftarrow t - t_{\text{step}} \)
\For{$a \in A, s \in S(t)$}
\State \( Q^*(s, a) \leftarrow R(s, a) + \sum_{s' \in S(t+t_{\text{step}})} T(s' | s, a) \max_{a' \in A} Q^*(s', a') \)
\EndFor
\If{$\min_{s \in S(t), a \in A} Q^*(s, a) = 0$}
\State \( e_{\text{min}}, e_{\text{max}}, n_{\text{min}}, n_{\text{max}} \leftarrow e_{\text{init}}, e_{\text{init}}, n_{\text{init}}, n_{\text{init}} \)
\Else
\State \( e_{\text{min}} \leftarrow \min_{s \in S(t), a \in A} \{ s.e \mid Q^*(s, a) < 0 \} \)
\State \( e_{\text{max}} \leftarrow \max_{s \in S(t), a \in A} \{ s.e \mid Q^*(s, a) < 0 \} \)
\State \( n_{\text{min}} \leftarrow \min_{s \in S(t), a \in A} \{ s.n \mid Q^*(s, a) < 0 \} \)
\State \( n_{\text{max}} \leftarrow \max_{s \in S(t), a \in A} \{ s.n \mid Q^*(s, a) < 0 \} \)
\EndIf
\State \( E \leftarrow \text{linspace}(e_{\text{min}} - D_{\text{buffer}}, e_{\text{max}} + D_{\text{buffer}}, x_{\text{steps}}) \)
\State \( N \leftarrow \text{linspace}(n_{\text{min}} - D_{\text{buffer}}, n_{\text{max}} + D_{\text{buffer}}, x_{\text{steps}}) \)
\State \( S(t - t_{\text{step}}) \leftarrow E \times N \times \Psi \times T_{\text{anom}} \)
\EndWhile
\EndFunction
\end{algorithmic}
\end{algorithm}
As with regular BIVI, \( Q^*(s, a) = 0 \) for each state at \( t_h \) and then \( U^*(s) \) is solved for every state at \( t_h - t_{\text{step}} \). The algorithm then identifies the rectangular spatial region that encloses the states where the utility function is negative, indicating that there is a nonzero probability of a safety violation. The east and north components of each \( s \) are denoted \( s.e \) and \( s.n \), respectively. If \( Q^*(s, a) = 0 \) for all \( s \in S(t_h - t_{\text{step}}) \), indicating the optimal action is no advisory (NIL) everywhere, the limits are set to the initial bounds. Again, at \( t_h - 2 \times t_{\text{step}} \), ASD defines the state space as the set of states that can transition into the region where an aircraft should be rerouted. ASD continues this process until \( t = 0 \).

An example of this process is shown in Fig. 3.2. The first image in Fig. 3.2a shows the where the example debris passes through the altitude threshold in yellow and the extents of the safety threshold in pink. The next image shows the extents of \( D_{\text{buffer}} \) from the intersection location in blue. The blue region is used to determine the \( E \) and \( N \) state space and discretization for the previous time step, \( t = t_h - t_{\text{step}} \). The \( E \) and \( N \) state space is represented as the grid. In the next image, the states that have a rerouting optimal action at \( t = t_h - t_{\text{step}} \) are shown in green. The final image in Fig. 3.2a shows the \( D_{\text{buffer}} \) region around the rerouting region in blue. The first image and updated grid in Fig. 3.2b shows how the blue region is used to determine the \( E \) and \( N \) state space and discretization for the previous timestep, \( t = t_h - 2 \times t_{\text{step}} \). The second image shows the states that have a rerouting optimal action at \( t = t_h - 2 \times t_{\text{step}} \). The process of updating the state space continues with the buffer expansion in the final image of Fig. 3.2b and the updated state space in Fig. 3.2c.

The example shows not only how the process works but also why they process is useful. In the example, in just three time steps, the extents of the state space more than doubled. Without ASD, the extents would have to be set to this larger value from the start, limiting information gathering in safety critical regions.

### 3.4.2 Utility Decomposition

The multiple threat problem involves accounting for all the pieces of debris. One can take an approach similar to that of Chryssanthacopoulos and Kochenderfer [48]. The
CHAPTER 3. SOLVING MDPS

(a) $E$ and $N$ state space for $t = t_h - t_{step}$

(b) $E$ and $N$ state space for $t = t_h - 2 \times t_{step}$

(c) $E$ and $N$ state space for $t = t_h - 3 \times t_{step}$

- debris location
- safety threshold
- states with reroute action
- $D_{buffer}$ region

Figure 3.2: Example evolution of $E$ and $N$ state space with ASD
state-action utility functions for different pieces of debris are combined to form

$$Q^*_{\text{multi}}(s,a) = \min_i Q^{(i)}_{\text{single}}(s,a)$$  \hspace{1cm} (3.6)$$

where $Q^{(i)}_{\text{single}}$ is the state-action utility function associated with the $i$th piece of debris. This approximation of the multiple threat state-action utility function tends to work well because it is driven by the worst-case piece of debris.

Unfortunately, this multiple threat approach requires querying the state-action utility function for each of the thousands of pieces of debris, which may be too computationally expensive for real time operation. In addition, storing approximately 4000 $Q^*_{\text{single}}(s,a)$ tables is memory intensive and impractical. Instead, immediately after computing $Q^*_{\text{single}}(s,a)$ for a particular piece of debris, we incorporate it into $Q^*_{\text{multi}}(s,a)$. To retain the added information from the dynamic discretization of ASD, $x_{\text{step}}$ for $Q^*_{\text{multi}}$ is increased to 200.

At each $Q^*_{\text{single}}(s,a)$ incorporation, $Q^*_{\text{multi}}(s,a)$ is updated as presented in Algorithm 3. The algorithm starts by comparing $E,N$ for the single and multiple state spaces, $S_{\text{single}}.E,S_{\text{single}}.N$ and $S_{\text{multi}}.E,S_{\text{multi}}.N$, respectively. The superset is calculated and used moving forward. To show how this works a smaller example is shown in Fig. 3.3. This example starts in Fig. 3.3a with an initial 10×10 $E_{\text{multi}},N_{\text{multi}}$ grid. In Fig. 3.3b, the initial grid is shown in black and an example 4×4 $E_{\text{single}},N_{\text{single}}$ grid is shown in red. The superset of the $E$ and $N$ state spaces are shown in Fig. 3.3c in blue. The superset is divided into the updated 10×10 $E_{\text{multi}},N_{\text{multi}}$ grid. In order to ensure that the worst scenario for each action is stored, $Q^*(s,a)$ is calculated for each point in the new state space by taking the minimum utility of $Q^*_{\text{multi}}(s,a)$ and $Q^*_{\text{single}}(s,a)$. The resulting $Q^*(s,a)$ becomes the new $Q^*_{\text{multi}}(s,a)$. 
Algorithm 3 Multiple Threat Combination

1: function \text{Combine}(Q_{\text{single}}^{*}, Q_{\text{multi}}^{*}, S_{\text{single}}, S_{\text{multi}})
2: \hspace{1em} for \( t \in T_{\text{launch}} \)
3: \hspace{2em} e_{\text{min}} \leftarrow \min(S_{\text{single}}(t).E \cup S_{\text{multi}}(t).E)
4: \hspace{2em} e_{\text{max}} \leftarrow \max(S_{\text{single}}(t).E \cup S_{\text{multi}}(t).E)
5: \hspace{2em} n_{\text{min}} \leftarrow \min(S_{\text{single}}(t).N \cup S_{\text{multi}}(t).N)
6: \hspace{2em} n_{\text{max}} \leftarrow \max(S_{\text{single}}(t).N \cup S_{\text{multi}}(t).N)
7: \hspace{1em} E = \text{linspace}(e_{\text{min}}, e_{\text{max}}, x_{\text{steps}})
8: \hspace{1em} N = \text{linspace}(n_{\text{min}}, n_{\text{max}}, x_{\text{steps}})
9: \hspace{1em} S(t) \leftarrow E \times N \times \Psi \times T_{\text{anom}}
10: \hspace{1em} for \( a \in A, s \in S(t) \)
11: \hspace{2em} Q_{\text{multi}}^{*}(s, a) \leftarrow \min\{Q_{\text{single}}^{*}(s, a), Q_{\text{multi}}^{*}(s, a)\}
12: \hspace{1em} \text{return } Q_{\text{multi}}^{*}

The goal of decomposing the problem and solving with ASD is to dynamically focus the state space. The superset of all state spaces for all timesteps is the state space that BIVI solution models.

3.5 Discussion

This chapter introduced utility functions and policies as MDP solutions. While various solution techniques were described to solve for the optimal utility function or policy, they all used rational decision making and variations of the principle of maximum expected utility. Different techniques work better for different problems.

Based on the sequential nature of when debris information is learned, and the fact that rerouting aircraft during a space launch can be decomposed into subproblems, the solutions in this thesis are found with BIVI and ASD. Both solution techniques take advantage of the problem by having time as a state variable, being finite horizon, and the value of information of future states on the solution at the current state. These techniques order states from those at the final timestep to the first timestep and therefore only require one sweep through the utility function, speeding up the
Figure 3.3: Example updating $E_{\text{multi}}, N_{\text{multi}}$ grid with $E_{\text{single}}, N_{\text{single}}$
solution process by removing the need to iterate.

There are limitations to the size of the state space that can be solved with BIVI. To focus on the safety critical regions, ASD focuses on the region of interest. BIVI is solved with all of the debris at once. The problem is decomposed for ASD, which solves a policy for each piece of debris and then uses multiple threat combination. This method allows for finer discretization in a more safety critical region at the cost of computational compute time.

The next chapter presents the utility functions and policies solved using BIVI and ASD. It also presents a simulation framework and simulation results to understand how the solutions would preform in the NAS.
Chapter 4

Simulation Framework and Results

The heading action MDP was solved using both BIVI and ASD. The metering action MDP defined in Section 2.3 was solved using BIVI. This section will start by discussing the different utility functions and their resulting policies. These policies were used in simulation to evaluate their effect on safety and efficiency of the airspace. This section continues by presenting the simulation framework and simulation results. This section concludes with a discussion on useful additional studies and preliminary results.

4.1 Utility and Policy Results

This section presents the utility function and policy results for the heading and metering action MDPs.

4.1.1 Heading Action Solutions

The heading action solutions can be broken into three parts: the BIVI solution, the ASD solution for a single piece of debris, and the multiple threat ASD solution.

Backward Induction Value Iteration Solution

As mentioned in Section 2.2.1, multiple state space discretizations for $E$ and $N$ were used for the BIVI solution. Table 4.1 shows the four different utility and policy
Table 4.1: Experimental $E$ and $N$ discretizations

<table>
<thead>
<tr>
<th>Utility Function</th>
<th>Policy</th>
<th>$E$ and $N$ Divisions</th>
<th>State Space Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_{\text{head, BIVI ASD}}$</td>
<td>$\pi_{\text{head, BIVI ASD}}$</td>
<td>11</td>
<td>3,185,325</td>
</tr>
<tr>
<td>$u_{\text{head, BIVI 50M}}$</td>
<td>$\pi_{\text{head, BIVI 50M}}$</td>
<td>43</td>
<td>48,674,925</td>
</tr>
<tr>
<td>$u_{\text{head, BIVI 75M}}$</td>
<td>$\pi_{\text{head, BIVI 75M}}$</td>
<td>53</td>
<td>73,946,925</td>
</tr>
<tr>
<td>$u_{\text{head, BIVI 100M}}$</td>
<td>$\pi_{\text{head, BIVI 100M}}$</td>
<td>61</td>
<td>97,955,325</td>
</tr>
</tbody>
</table>

names, the number of $E$ and $N$ divisions, and the corresponding size of the overall state space.

A subset of the utility function and policy are shown in Fig. 4.1 and Fig. 4.2, respectively, where the red region represents an elliptical approximation of the TFR presented in Fig. 1.2. These plots present the utility function and policy over time for the discretizations presented in Table 4.1. The subset considers an aircraft anywhere in the $E$ and $N$ grid traveling at $225^\circ$ (southwest) when an anomaly occurs 80 s into launch.

For the utility plots, each column represents a different time after launch and each row represents a different discretization. It is important to note that the utility scale is fixed for all of the utility plots and sometimes there are regions with very small negative utility that might not appear to the naked eye. When the launch first occurs, the first column, an anomaly has not occurred. At this time step, there is minimal risk of debris yet, but it is known that the launch vehicle will pass through the altitude threshold at 50 s. In the second column, no anomaly has occurred but with the probability of anomaly modeled, there is a growing risk of potential debris and the utility plots start to show more regions with negative utility. The regions of negative utility are regions where an aircraft might be at risk of intersecting the safety threshold around a piece of debris if there is no heading command. In the third and fourth columns, an anomaly has occurred and debris is actively falling through the altitude threshold. With active debris, there are more regions of larger negative utility. Since the aircraft is traveling southwest, to help the aircraft maneuver around the debris, hopefully with weaker actions, the utility function has more regions of negative utility spreading from the debris locations to the northwest.
Figure 4.1: Utility for different discretizations over time
Figure 4.2: Policy for different discretizations over time
The \( E \) and \( N \) discretization increases from the top to the bottom row. Due to the coarse discretization of Fig. 4.1a, compared to Fig. 4.1d, the utility values are spread over a coarse discretization for the different pieces of debris and create less efficient rerouting commands. Finer discretization is not always better because the state space discretization is set using a uniform grid that is not always aligned with where most pieces of debris or the launch vehicle are located. An example of this is in Fig. 4.1b. The second column highlights a region of potential concern that other discretizations miss.

For the policy plots, the key at the bottom indicates the commanded action and, similar to the utility functions, each column represents a different time after launch and each row represents a different discretization. In general, the policy plots reflect the same trends of the utility functions but in the form of actions. In the first column, the aircraft is given commands to reroute around the location at which the launch vehicle will pass through the altitude threshold. In the second column, an anomaly has not occurred but because there is a probability that an anomaly will occur, preventative actions are commanded to reroute the aircraft around regions where debris could fall through the airspace if there were an anomaly in the future. In the third and fourth column, an anomaly has already occurred and reactive commands reroute aircraft around debris.

Moving down the rows, the size of the rerouting region decreases as the discretization becomes finer. This happens because the finer discretization allows more precise multilinear interpolation. In general, the states further from the potential debris have maneuvers that are less disruptive. Similar to in the utility plots, the maneuver region expands to the northeast, the opposite direction of the aircraft’s motion. Since the model does not know the aircraft’s path and the aircraft is not required to maintain its heading, the policies show an expansion of less disruptive maneuvers perpendicular to the aircraft motion to ensure the aircraft’s path does not change course towards the debris.
Single Adaptive Spatial Discretization for BIVI Solution

The use of ASD allows the MDP to dynamically focus on the region of interest. A subset of the utility function and policy for a single piece of debris are shown in Fig. 4.3 and Fig. 4.4, respectively. These plots present the utility function and policy over time for an aircraft anywhere in the $E$ and $N$ grid traveling at $225^\circ$ (southwest) when an anomaly occurs 80 s into launch.

![Utility plots](image1)

**Figure 4.3:** Utility for a single piece of debris over time

![Policy plots](image2)

**Figure 4.4:** Policy for a single piece of debris over time

The utility plot presents the utility on a logarithmic intensity scale, where the darker the plot, the lower the utility. As the time to debris intersection decreases from right to left, utility decreases. This makes sense because the shorter the time until debris intersections, the more at risk the aircraft is of intersecting the debris safety threshold and the stronger the rerouting actions need to be to fly clear of the debris. The debris intersection plot in Fig. 4.3 continues to have low utility when
the debris passes the threshold because of the uncertainty model. To account for any assumptions made in debris modeling and the uncertainty of failure modes in general, the MDP models the each piece of debris as falling through the airspace for multiple time steps. For all of the utility plots, the low utility regions expand in the opposite direction of the aircraft motion because as the aircraft travels towards the intersection location, it becomes less safe.

For the policy plot, the key indicates the commanded actions. The policy plots show trends similar to those of the utility function. When there is more time to maneuver the maneuvers to avoid the debris are less intrusive (maintain) and when there is less time, the aircraft is commanded more aggressive maneuvers (strong left or strong right). Because the reward function is structured such that less disruptive maneuvers are less expensive, they are commanded more often.

Although the plots are presented chronologically in both Figs. 4.3 and 4.4, they are solved for from right to left. The plots solved for first reflect when the debris intersects the altitude slice and have the smallest and therefore most focused state space. For the far right plots, the $E$ and $N$ states span just under 18,000 m each. The plots solved for later have an expanded state space, allowing the aircraft to maneuver around the debris earlier and safer. For comparison, for the far left plots, the $E$ and $N$ states span just over 130,000 m each. This is an order of magnitude change in state space size, which allows earlier timesteps to have a much finer discretization, less interpolation, and more accurate values.

To further see the results of ASD on the state space size and its ability to focus on the rerouting region, Fig. 4.5 shows the same policies as Fig. 4.4 but has fixed bounds equal to the bounds of the largest, focused state space (the state space from 3 min prior to intersection). The red region represents an elliptical approximation of the TFR presented in Fig. 1.2.

**Multi-Threat Adaptive Spatial Discretization for BIVI Solution**

The utility and policy over time of the multiple threat solution is presented in Fig. 4.6 and Fig. 4.7, respectively, where the red region represents an elliptical approximation of the TFR presented in Fig. 1.2. These plots display the results for an aircraft with
CHAPTER 4. SIMULATION FRAMEWORK AND RESULTS

Figure 4.5: Policy for a single piece of debris over time with the least restrictive state space

an initial heading of 225° (southwest) when an anomaly occurs 80 s after the launch. These results show how the model adapts to information over time, recognizes the danger of debris and the launch vehicle, and reacts to all of the individual threats.

Figure 4.6: ASD utility over time (s after launch)

Figure 4.7: ASD policy over time (s after launch)
The utility plot presents the utility on a logarithmic intensity scale, where the darker the plot, the lower the utility. At 0 s after launch, the utility plot shows a low utility for where the launch vehicle will pass through the airspace and the associated region where the aircraft should be rerouted to avoid the launch vehicle. At 50 s after launch, the anomaly has not yet occurred, and the launch vehicle is actively passing through the airspace, so the aircraft is no longer rerouted around it. At 250 s and 500 s after launch, an anomaly has occurred and debris is actually falling through the airspace. The utility trend for these time steps is similar to that in Fig. 4.3, except Fig. 4.6 fuses results from many pieces of debris and therefore there are multiple regions with very low utility. The regions of low utility continue to expand in the direction opposite to the aircraft motion because as the aircraft travels towards the intersection location, it is less safe. Further, regions perpendicular to the aircraft motion have some regions of low utility because the aircraft trajectory is unknown and actions are provided to ensure the aircraft does not turn towards debris.

For the policy plot, the key indicates the commanded actions. At 0 s after launch, the policy plot maneuvers the aircraft to avoid the launch vehicle. At 50 s after launch, there has not yet been an anomaly and the probability of anomaly is low enough, and the time it takes for debris to fall through the altitude threshold is long enough, that no actions are commanded for this subset of the state space. At 250 s and 500 s after launch, the policy is a combination of the policies for the various pieces of debris falling through the airspace.

Compared to the BIVI policy solutions presented in Fig. 4.2 some of the ASD decision boundaries are less smooth. This can be attributed to the added details of solving for each piece of debris separately, thereby increasing the model fidelity and limiting interpolation. Further, the lack of smoothness can be a result of how the individual policies are combined to create the multiple threat solution.

4.1.2 Metering Action Solution

Based on the metering state space, only the policy is visualized. The policy is visualized in the $E$ and $N$ plane, where the circles represent waypoint locations, the filled
circles are the reachable waypoints at the denoted $t_{\text{flight}}$, and the color indicates the commanded action. Fig. 4.8 shows the policy for a path over time when the initial speed is 293 m/s, a launch occurs 53.3 min into the flight, and an anomaly occurs 2.8 min into the launch.

The policy accounts for the pilot response, periods between waypoints, and probability of anomaly. For the metering results, each path is run for each of the potential times of anomaly assuming that an anomaly will occur for that potential time of anomaly. Assuming that an anomaly will occur for each potential time of anomaly creates conservative policies.

When the policy is queried, if it is before $t_{\text{anom}}$, the utility results for all $t_{\text{anom}}$ are combined to select the optimal action. Similar to the procedure described in Section 3.4.2, the $Q^*(s,a)$ is calculated for each $t_{\text{anom}}$ and minimum utility for each action makes up $Q^*_{\text{multi}}(s,a)$. The optimal action is still the action that maximizes $Q^*_{\text{multi}}(s,a)$.

The conservative nature of the policy results in many preventative actions such as the slowest command given at 0 s into flight, well before an anomaly occurs. The anomaly occurs between the first and second waypoint so the action at 6 min is reactive. By the time the aircraft reaches the third and fourth waypoints, at 20.5 min and 47.6 min respectively, there is more clearance between the aircraft and debris so there is a reduction in action strength. After the fourth waypoint, the debris is cleared and the commanded speed is to remain constant.

4.2 Simulation Framework

Experiments were run to evaluate the effectiveness of the resulting policies in improving safety and efficiency compared to historic or heuristic rerouting techniques. The experiments are divided into domestic flight paths, used with the BIVI and ASD models, and oceanic flight paths, used with the metering model. This section describes the two sets of simulated trajectories, procedures during the simulations, and comparison policies.
Figure 4.8: Policy for a single path over time
4.2.1 Domestic Flight Paths

To explore domestic rerouting, 101 recorded flight paths in the Kennedy Space Center area, shown in Fig. 4.9, were used. All of the flight paths were gathered as a set of waypoints gathered from NASA’s FACET tool [49] and represent a portion of the daily activity in the historic rerouting region. The historic rerouting region is represented by the red region in Fig. 4.9, which is an elliptical approximation of the TFR presented in Fig. 1.2. The simulations were run with a simulation set of debris that was generated as described in Section 2.1.1 and separate from the MDP solution generating and parameter selection debris sets.

![Flight trajectories gathered from FACET](image)

**Figure 4.9:** Flight trajectories gathered from FACET

To understand how each of these flight trajectories might be impacted, each flight was simulated with 100 different start times and various times of anomaly resulting in 121,200 total simulations. The start time variation affects the position and heading of the aircraft relative to the launch vehicle. The various times of anomaly match the range of time that produces debris at the 35,000 ft altitude slice within the duration of the launch plus 10 min. Cycling over each anomaly time allows each flight to experience all of the potential debris trajectories. During simulation, the pilots are assumed to immediately respond to commands.
4.2.2 Oceanic Flight Trajectories

To explore oceanic rerouting, 33 simulation paths were created, used to find the MDP solution, and used for simulation. Each path consists of randomly generated sets of waypoints from the North Atlantic Route Planning Chart [34] that come within a distance $\delta$ of potential debris locations. There are a total of 406 waypoints, and each path has at most 10 waypoints. The paths are generated as follows:

1. An initial waypoint $w_1$ is randomly selected and added to the path.

2. A random waypoint $w_2$ is selected from the previous waypoint’s $nn$ nearest neighbors and added to the path. For this thesis, $nn = 75$.

3. A random waypoint $w_i$ is selected from the previous waypoint’s $nn$ nearest neighbors. If the angle from $w_{i-1}$ to $w_i$ is within $45^\circ$ of the angle from $w_{i-2}$ to $w_{i-1}$ the waypoint is added to the path; if not, another random waypoint is selected from the neighbors. If none of the $nn$ neighbors achieve this criterion, the current path is returned.

4. Repeat 3 until the path has been returned or has 10 waypoints.

After the path is generated, it is checked with potential debris locations to make sure it would intersect with the debris safety threshold. Further, we determine the $T_{\text{anom}}$ and $T_{\text{launch}}$ that create these interactions. We only keep flights that require all positive $T_{\text{launch}}$ because a negative $t_{\text{launch}}$ means that metering should occur before the aircraft takes off and our current study is for metering in the air. Figure 4.10 shows the modeled 33 paths laid over a histogram of debris locations. The starting location of each path is shown in light blue.

To test robustness, the experimental simulations are run on path variants. The path variants require the flights to pass over the waypoints but add noise between waypoints by sampling a k-dimensional normal distribution parameterized by mean $\mu$ and covariance matrix $\Sigma$.

A distribution is defined between each set of waypoints within a path. We set $k = \lfloor d_s/10^5 \rfloor + 1$ where $d_s$ is the distance between two waypoints in meters. The values
for \( \mu \) is the concatenated set of east and north coordinates for the \( k \) evenly spaced, linearly interpolated values between the two waypoints, \([e_1, e_2, \ldots, e_k, n_1, n_2, \ldots, n_k] \). The covariance is a \((2k \times 2k)\) matrix. We designed the covariance matrix such that there is only correlation between two subsequent east values and two subsequent north values. The covariance decreases as the east and north points get further away from one another. For example, if \( k = 4 \), then

\[
\Sigma = \begin{bmatrix}
c & \frac{c}{2} & \frac{c}{4} & 0 & 0 & 0 & 0 \\
\frac{c}{2} & c & \frac{c}{2} & \frac{c}{4} & 0 & 0 & 0 \\
\frac{c}{4} & \frac{c}{2} & c & \frac{c}{2} & 0 & 0 & 0 \\
0 & \frac{c}{4} & \frac{c}{2} & c & 0 & 0 & 0 \\
0 & 0 & 0 & c & \frac{c}{2} & \frac{c}{4} & 0 \\
0 & 0 & 0 & \frac{c}{2} & c & \frac{c}{2} & \frac{c}{4} \\
0 & 0 & 0 & \frac{c}{4} & \frac{c}{2} & c & \frac{c}{2} \\
0 & 0 & 0 & 0 & \frac{c}{4} & \frac{c}{2} & c
\end{bmatrix} \tag{4.1}
\]

where \( c = 100 \times d_s \). The set of waypoints and intermediate values represent a simulation trajectory. Figure 4.11 shows 33 trajectories between two waypoints in light blue for a single path shown in royal blue.
After the paths are created, they are used for simulation. For each path, the number of simulations run is proportional to the size of that path’s state space. Each path has a $n_c$ potential $s_{ac}$, $t_{anom}$, and $t_{launch}$ combinations. To simulate a broad range of combinations and path variations, each path is simulated $2 \times n_c$ times. For 33 paths, this results in 101,840 simulations.

The simulation framework runs each trajectory with a randomly selected $t_{launch}$ and $s_{ac}$. First, the trajectory is run with no debris to obtain a baseline trajectory flight time. Next, the trajectory is run with the MDP solution. The trajectory is run following the MDP policy for a randomly selected $t_{anom}$ as well as no anomaly. When following the MDP, the policy is queried at each waypoint, an advisory is issued, and the pilot always immediately responds. Finally, the trajectory is run with the same $t_{anom}$, following the simple heuristic.
4.2.3 Comparison Policies

The first comparison policy, which is used for both domestic and oceanic flights, is the nominal policy denoted $\pi_{\text{nominal}}$. This policy has the aircraft follow the nominal path at the nominal speed. This gives a baseline for how many aircraft are nominally unsafe in the experimental simulations.

For domestic flights, the second comparison policy is the historic policy denoted $\pi_{\text{historic}}$. This policy reroutes the aircraft around the historically restricted region by having the aircraft follow its nominal path until it reaches the restricted region and chooses the shortest path around the perimeter of the restricted region to where the nominal path would exit the restricted region. Actual rerouting procedures are somewhat more complicated; for example a reroute may begin some time before the aircraft touches the restricted region and not follow a perfectly curved path.

For oceanic flights, the second comparison policy is a simple heuristic denoted $\pi_{\text{heuristic}}$. Currently, rerouting during a space launch is not done through metering, so there is no historic method to compare against. For this reason, the metering trajectories are compared against a simple heuristic. The heuristic has the aircraft fly at a constant speed until an anomaly occurs. Once an anomaly occurs, the heuristic uses information about the future debris locations to make speed changes at upcoming waypoints. If there is debris between waypoints and metering can avoid the debris, the heuristic will command the most efficient action commands to meter around the debris. This is repeated at every waypoint.

4.2.4 Safety vs. Efficiency Tradeoff Analysis

For all of the MDP models, the reward function is defined as

$$r_{\text{safe}} + \lambda r_{\text{eff}}$$

where $\lambda$ controls the tradeoff between safety and efficiency. For each solution method and discretization, a solution is found with five different $\lambda$ values: $0$, $5 \times 10^{-6}$, $5 \times 10^{-5}$, $5 \times 10^{-4}$, and $5 \times 10^{-3}$. Following the procedures outlined in Section 1.2.1,
the results for each $\lambda$ are run on a full set of simulations with a validation set of debris that was generated as described in Section 2.1.1 and visualized.

The $\lambda$ sweep for the domestic flight results is presented in Fig. 4.12. The key at the bottom of the figure is an example key of how the transparency relates to $\lambda$ value. This figure includes the five different BIVI discretization results, the ASD results, and the nominal results. In general, the results are monotonic, the smaller the $\lambda$, the safer the results, at the expense of more rerouted flights. The non-monotonic results could be from overfitting or the specific locations of state space cut points. For the BIVI test results $\lambda = 5 \times 10^{-6}$. For the ASD test results $\lambda = 5 \times 10^{-5}$.

The $\lambda$ sweep for the oceanic flight results is presented in Fig. 4.13. Some of the results are very close and sometimes overlapping. For the metering test results $\lambda = 5 \times 10^{-4}$.

### 4.3 Simulation Results

Using the simulation framework, simulations were run with a test set of debris, and their aggregate results are presented in this section. The aggregate results are presented as likelihood-weighted results. The simulation results reflect a 50% chance of anomaly. For the potential modeled times of anomaly there is a uniform distribution over what time the anomaly could occur. The weighted results are the probability of the cumulative distribution function for a geometric distribution presented in Section 2.2.4.

#### 4.3.1 Domestic Simulation Results

The likelihood-weighted simulation results for the domestic flight paths are presented in Table 4.2 and shown in Fig. 4.14. Two nominal results are presented, $\pi_{\text{nominal, BIVI}}$ and $\pi_{\text{nominal, ASD}}$ for the BIVI and ASD simulations, respectively, to account for the different simulation debris.

For the BIVI results, in general the finer the discretization, the safer the solution, which results in $\pi_{\text{head, BIVI, 100M}}$ being the safest policy. Compared to $\pi_{\text{nominal, BIVI}}$,
Figure 4.12: Heading safety vs. efficiency lambda exploration
Figure 4.13: Metering safety vs. efficiency lambda exploration

Table 4.2: Domestic simulation results

<table>
<thead>
<tr>
<th>Policy</th>
<th>p(Rerouted)</th>
<th>Avg Added Dist (m)</th>
<th>p(Pass $\delta$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_{\text{historic}}$</td>
<td>1.0</td>
<td>6992</td>
<td>0.0</td>
</tr>
<tr>
<td>$\pi_{\text{nominal, BIVI}}$</td>
<td>0.0</td>
<td>0</td>
<td>$2.80 \times 10^{-2}$</td>
</tr>
<tr>
<td>$\pi_{\text{head, BIVI, ASD}}$</td>
<td>$8.0 \times 10^{-2}$</td>
<td>3140</td>
<td>$1.95 \times 10^{-2}$</td>
</tr>
<tr>
<td>$\pi_{\text{head, BIVI, 50M}}$</td>
<td>$1.6 \times 10^{-1}$</td>
<td>579</td>
<td>$1.50 \times 10^{-2}$</td>
</tr>
<tr>
<td>$\pi_{\text{head, BIVI, 75M}}$</td>
<td>$5.9 \times 10^{-2}$</td>
<td>296</td>
<td>$1.83 \times 10^{-2}$</td>
</tr>
<tr>
<td>$\pi_{\text{head, BIVI, 100M}}$</td>
<td>$6.1 \times 10^{-2}$</td>
<td>532</td>
<td>$1.10 \times 10^{-2}$</td>
</tr>
<tr>
<td>$\pi_{\text{nominal, ASD}}$</td>
<td>0.0</td>
<td>0</td>
<td>$1.69 \times 10^{-2}$</td>
</tr>
<tr>
<td>$\pi_{\text{head, ASD}}$</td>
<td>$7.8 \times 10^{-2}$</td>
<td>583</td>
<td>$1.28 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

$\pi_{\text{head, BIVI, 100M}}$ is 60.71% safer and 26.67% safer than the second-best BIVI result, $\pi_{\text{head, BIVI, 50M}}$. While it was expected that $\pi_{\text{head, BIVI, 75M}}$ would be safer than $\pi_{\text{head, BIVI, 50M}}$ the opposite was true. This could be because the state space discretization for $\pi_{\text{head, BIVI, 50M}}$ being able to better generalize, avoiding overfitting.

It is important to note that comparing the BIVI and ASD results is not a direct comparison because the simulation debris profiles are different. However, if the $\pi_{\text{head, BIVI, 100M}}$ and $\pi_{\text{head, ASD}}$ simulation results are compared, $\pi_{\text{head, ASD}}$ is only 14.06% less safe and was solved for with a much coarser discretization. Compared to $\pi_{\text{head, BIVI, ASD}}$, which has a matching size state space, $\pi_{\text{head, ASD}}$ is 34.36% safer.
Further, $\pi_{\text{head, ASD}}$ is 24.26% safer than $\pi_{\text{nominal, ASD}}$.

In terms of efficiency, all of the proposed solutions significantly reduce the number of rerouted aircraft and average added flight distance per rerouted flight compared to the historic method. Note that there is not a strong correlation between discretization and efficiency.

![Graph showing P(Metered Flight) vs. P(Unsafe Flight) for different formulations.]

**Figure 4.14:** Heading safety vs. efficiency results

There are a few ways in which safety could be improved. In terms of the ASD formulation, $\pi_{\text{ASD}}$, was solved for with a coarse discretization for each threat. Based on the positive trend of increased safety with increased discretization, solving each threat could produce safer results. Further, solving with BIVI with a finer discretization could also potentially increase safety results.

For all of the domestic flight frameworks, some of the flights might be unsolvable. All of the paths are checked to make sure in the baseline case that they start outside of the historic restricted region but each path is run with different start times, and therefore, it is not guaranteed that every trajectory is in a region from which an aircraft can be safely rerouted with the current system. Further, to ensure safety, it might be necessary to increase the time discretization, which would increase debris
model accuracy and would allow more frequent reroute commands. In turn, the pilot response rate would need to be revisited and it would be important to adjust it based on how long it takes information from launch providers to reach air traffic control and then for the pilots to respond.

Another thing that could improve safety, is using more realistic safety thresholds. Currently, $\delta = 10 \times d_{\text{NMAC}}$. With the correct state space discretization, a solution for $\delta = d_{\text{NMAC}}$ would be more efficient and potentially safer. Further, this smaller $\delta$ could reduce the number of inherently unsolvable simulations. As a preview into how the results would change with a smaller $\delta$, Table 4.3 shows the likelihood-weighted simulation safety results when the policies solved with $\delta = 10 \times d_{\text{NMAC}}$ are simulated with $\delta = 10 \times d_{\text{NMAC}}$ and $\delta = 1 \times d_{\text{NMAC}}$, resulting in an order of magnitude increase in safety.

Table 4.3: Domestic safety results for different thresholds

<table>
<thead>
<tr>
<th>Policy</th>
<th>p(Pass $\delta$)</th>
<th>p(Pass $d_{\text{NMAC}}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_{\text{historic}}$</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>$\pi_{\text{nominal, BIVI}}$</td>
<td>$2.80 \times 10^{-2}$</td>
<td>$6.47 \times 10^{-3}$</td>
</tr>
<tr>
<td>$\pi_{\text{head, BIVI, ASD}}$</td>
<td>$1.95 \times 10^{-2}$</td>
<td>$6.17 \times 10^{-3}$</td>
</tr>
<tr>
<td>$\pi_{\text{head, BIVI, 50M}}$</td>
<td>$1.50 \times 10^{-2}$</td>
<td>$1.01 \times 10^{-3}$</td>
</tr>
<tr>
<td>$\pi_{\text{head, BIVI, 75M}}$</td>
<td>$1.83 \times 10^{-2}$</td>
<td>$4.04 \times 10^{-3}$</td>
</tr>
<tr>
<td>$\pi_{\text{head, BIVI, 100M}}$</td>
<td>$1.10 \times 10^{-2}$</td>
<td>$3.67 \times 10^{-3}$</td>
</tr>
<tr>
<td>$\pi_{\text{nominal, ASD}}$</td>
<td>$1.69 \times 10^{-2}$</td>
<td>$4.2 \times 10^{-3}$</td>
</tr>
<tr>
<td>$\pi_{\text{head, ASD}}$</td>
<td>$1.28 \times 10^{-2}$</td>
<td>$3.8 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

4.3.2 Oceanic Simulation Results

The likelihood-weighted simulation results for the oceanic flight paths are presented in Table 4.4. The Anomaly Status column describes whether the results include the no anomaly or anomaly cases. The no anomaly case has no anomaly 100% of the time whereas the anomaly case has an anomaly 100% of the time. The no anomaly and anomaly case has an anomaly occur 50% of the time. In terms of safety, $\pi_{\text{meter}}$ was 12.32% safer than $\pi_{\text{heuristic}}$ and 58.63% safer than $\pi_{\text{nominal}}$. Due to the likelihood
weighting, the efficiency results for the no anomaly and anomaly cases for the MDP solution are different. Because rerouting commands are only provided at waypoints to encourage more successful rerouting, the MDP was modeled considering an anomaly to always occur, resulting in a more conservative policy with considerable preventative metering. This is why the no anomaly case has a high metering percentage. Further, the heuristic efficiency metric only counts if the metering creates safety between two waypoints, which lowers the metric.

There are a few ways in which safety could be improved. To start, Fig. 4.15 shows how many trajectories hit debris between sets of waypoints. Trajectories can hit more than one piece of debris, so the total count is greater than the number of trajectories. According to the histogram, five trajectories intersect debris between the first and the second waypoint, 15 trajectories between the first and third waypoint, and 25 trajectories between the first and fourth waypoint. To solve these encounters, it might be necessary for commands to happen earlier, which would require rerouting before the plane reaches the first waypoint in the region or ground holding. Another way to have a quicker response is with stronger commands. Currently the speed changes modeled are small, and over the long distances between waypoints it might be preferable to have larger speed changes. Since the time between waypoints could be over an hour and the distance traveled between waypoints could be over 1,000,000 m, to ensure desired safety metrics, it might be necessary to provide commands more often than once every waypoint.
4.4 Discussion

This chapter begins by presenting the MDP solutions as utility functions and policies. The BIVI utility function shows regions of negative utility leading up to where pieces of debris pass through the altitude threshold. This is correlated with the risk of hitting a potential piece of debris and the magnitude of the commanded action in those regions. The more-at-risk locations have stronger commands and there is a region that expands from the debris intersection location with weaker actions to encourage safer flights with less disruption. The BIVI policy shows the same trends as the utility function but also provides the actual commands. The maintain action is the cheapest and is used most often compared to the strong right or left that are used in more critical situations.

The ASD utility function and policy follow similar trends as the BIVI solution, with the additional focus on how the ASD algorithm is able to focus the state space on the region of interest. To show how the state space changes and focuses on safety critical regions, the policy for a single piece of debris with ASD is shown with the ASD limits as well as the BIVI limits. This comparison shows how ASD is able to support a finer state space resolution without changing the size of the state space. The
combined ASD utility function and policy for all of the pieces of debris were presented. The combined utility function and policies both showed how ASD was able to localize solutions around smaller groups of debris compared to the BIVI solutions.

While BIVI and ASD were used for the heading formulation, only BIVI was used for the metering formulation because the metering state space uses waypoint rather than $E$ and $N$. The chapter continues by presenting an example metering action policy. The policy shows how actions are used to separate aircraft and debris in time and add space between the two. The example provided showed how preventative maneuvers are used to initiate separation and reactive maneuvers provide additional separation.

Next, the simulation framework was presented. Domestic flight data was available, so real trajectories were used for the applicable simulations. Oceanic flight data was not available, so generated trajectories were used for the applicable simulations. The generated trajectories were made by following a set of rules to create realistic flight paths of available oceanic waypoints. Once the trajectories were generated, paths were created with noise added between each set of waypoints by sampling a multidimensional normal distribution. The simulation framework uses the various flight paths for reward function parameter tuning and to collect aggregate metrics.

Simulations are run to create Pareto frontiers to explore the reward function trade-off between safety and efficiency. The tradeoff is done using the weighting parameter, $\lambda$, and a parameter sweep is run for the various solution methods to select the $\lambda$ value that would be used in practice. The selected $\lambda$ values are used for the simulations that gather the aggregate metrics.

The aggregate metrics use likelihood-weighted results to evaluate the safety and efficiency performance of the proposed solutions. For BIVI, the safest policy, $\pi_{\text{head, BIVI, 100M}}$ is 60.71% safer than $\pi_{\text{nominal, BIVI}}$. For ASD, $\pi_{\text{head, ASD}}$ is 24.26% safer than $\pi_{\text{nominal, ASD}}$, and, while not a direct comparison, $\pi_{\text{head, ASD}}$ is 34.36% safer than $\pi_{\text{head, BIVI, ASD}}$ which are solved with matching discretization. For metering, $\pi_{\text{meter}}$ is 12.32% safer than $\pi_{\text{heuristic}}$ and 58.63% safer than $\pi_{\text{nominal}}$. These results show that there is confidence in both the heading and metering formulations but that both also have room for improvement. While this chapter touched upon some of the areas for
improvement, the following chapter will describe future work that could improve both methods.
Chapter 5

Summary and Future Work

To mitigate the costs of increased space launches to airlines, this thesis presents methods for rerouting aircraft during space launches that result in fewer rerouted aircraft and when an aircraft is rerouted, smaller flight path deviations. This chapter summarizes the methods and thesis contributions, presents how they could be incorporated into today’s air traffic management, and concludes with a discussion of future work.

5.1 Contributions

To increase efficiency, instead of rerouting aircraft around a large, static restricted region that is potentially active for hours, aircraft can be rerouted in real time based on the current or potential hazards resulting in smaller, dynamic restrictions. This thesis has four contributions. The first two contributions of this thesis are modeling the problem as a Markov decision process to solve for both heading and metering reroute commands. The third contribution is a solution technique, adaptive spatial discretization for backward induction value iteration. The final contribution is a simulation framework to evaluate the robustness of the solutions.
CHAPTER 5. SUMMARY AND FUTURE WORK

5.1.1 MDP Formulations

The problem of rerouting aircraft during a space launch can be modeled as a Markov decision process (MDP). The MDP is solved for optimal rerouting actions. The first two contributions of this thesis presented in Section 2.2 and Section 2.3 are modeling the problem as an MDP to solve for heading reroutes and metering reroutes, respectively.

The first model follows how aircraft are rerouted during severe weather, and solves for heading reroutes. This MDP models the aircraft location in the launch region, the time along the launch trajectory, launch vehicle locations, launch vehicle probability of anomaly, potential debris locations, and pilot response rate. The solution is an optimal heading command for every state space location.

The second model takes into account that oceanic communications are unreliable, air traffic control rarely communicates with aircraft, and reroutes are often done by changing the time that an aircraft should reach a specific waypoint, known as metering. By adjusting the waypoint arrival time, the aircraft is implicitly given a speed command. This MDP is solved for metering reroutes by modeling the aircraft location along its trajectory, trajectory variability, aircraft speed, the time along the launch trajectory, launch vehicle locations, launch vehicle probability of anomaly, potential debris locations, and pilot response rate. The solution is an optimal speed command at every waypoint.

5.1.2 Implemented Solution Techniques

Both MDP models are solved using backward induction value iteration. This method generally requires a coarse discretization due to the dimensionality of the problem. In the heading action formulation, the aircraft location is modeled as discretized east and north coordinates. The discretization of these state variables can alter how the optimal action is selected to avoid the launch vehicle and pieces of debris. To address this, Section 3.4 presents decomposing the heading action formulation by piece of debris, solving the MDP for each piece of debris or the launch vehicle with a novel approach, adaptive spatial discretization for backward induction value iteration.
(ASD), and recombining the MDP solutions to model the multiple threat problem. The ASD algorithm is the third thesis contribution.

For each threat, launch vehicle or piece of debris, ASD focuses the state space around the point of impact and as the solution moves backward in time, expands the state space to encompass the maneuver producing region. This method allows a finer state space discretization in maneuver critical regions (areas and times closest to a potential threat) and a coarser state space discretization in less maneuver critical regions (areas and times further from a potential threat).

5.1.3 Simulation Framework

The final contribution of this thesis is the oceanic simulation framework, specifically the flight path generation used to evaluate the MDP solutions presented in Section 4.2.2. Unlike the domestic simulations, which had available flight data and comparable historic procedures, the oceanic simulations used generated trajectories and a heuristic. Oceanic trajectories were generated by randomly selecting oceanic waypoints, ensuring the produced path was realistic, and then adding noise to the trajectories between the waypoints. This allowed the simulations to model a potential airspace and test the robustness of the solution. All of the MDP solutions were run through the appropriate simulations and aggregate results were presented in Section 4.3.

5.2 Future Work

This thesis showed how using MDP solutions to reroute aircraft during space launches can create a more efficient airspace with limited safety degradation. In the future, this work can be expanded to create safer MDP solutions by leveraging more accurate models that produce better solutions.

The first way to increase and better interpret safety is to parse resolvable and unresolvable scenarios. A resolvable scenario is one where an aircraft can feasibly escape a hazardous region using the available actions while considering the aircraft’s
capabilities and limitations. In contrast, an unresolvable scenario is one where an aircraft’s capabilities and limitations prevent it from feasibly escaping a hazardous region using the available actions. This task defines the policy operating procedures and limitations. The operating procedures and limitations could adjust how the MDPs are formulated in the future.

While improving the simulations can inspire MDP formulation changes, there are already some changes that future work should address. The MDP can be adjusted to use a more realistic safety threshold around debris. In general, this would require a finer state space discretization. This thesis presented a method to implement a finer east and north discretization, and future work can continue to address this as well as address a finer time of anomaly, time of launch, and time into flight discretizations. Improving the time discretizations can allow more accurate debris modeling and more frequent reroute commands.

With the current time discretizations and assumptions, the current metering formulation only has aircraft receiving reroute commands at waypoints, which could be over an hour apart from one another. The safety of the metering formulation solutions would improve with more frequent reroute commands and potentially from a larger metering actions space. Growing the action space would allow for stronger speed commands that could potentially meter the aircraft quicker and could introduce ground holding. To gain additional separation between aircraft and debris, ground holding would delay the start of the aircraft path making up for limited rerouting commands or aircraft capabilities. Another safety factor with the metering formulation is the data turnaround time and pilot response rate. Currently, the modeled data turnaround time and pilot response rate is borrowed from the response rate of pilots given frequent heading commands from an TCAS and should be investigated for the sparse metering commands.

In terms of modeling considerations, throughout this thesis, the problem is modeled as an MDP and considers no state uncertainty. While the current advances with ADS-B provides precise aircraft locations, we do not always know the exact location of a piece of debris [50]. A partially observable Markov decision process that contains
state uncertainty would be a more accurate model but would have inherent computational tractability concerns. Future work could explore the benefits from adding the state uncertainty and how it effects the formulation due to the added computational constraints.

Another way to enhance model accuracy is to move from a discretized to continuous domain. One method of doing this is by using deep learning. The universal approximation theorem [51] suggests that a deep neural network could represent \( Q^*(s,a) \) well. In conjunction with the work presented in this thesis, work was done to model the problem with deep learning two ways. The first method takes into account that a discrete dynamic programming solution is a good approximation of the continuous solution and takes the following two steps:

1. The network is trained to regress a solved discrete dynamic programming solution using the asymmetric \( \ell^2 \) loss function proposed by Julian et al. [52] between the neural network solution denoted \( Q_\theta(s,a) \) and the discretized solution denoted \( \tilde{Q}^*(s,a) \). The network was able to represent \( \tilde{Q}^*(s,a) \) well.

2. The second step is to use deep reinforcement learning to train the network further to learn the continuous state-action utility function using the same asymmetric \( \ell^2 \) loss function but between \( Q_\theta(s,a) \) and \( Q^*(s,a) \). It is important to note that \( Q(s,a) \) does use the current \( Q_\theta(s,a) \) utilities when considering future states.

The second method only does the second half of the previous method, uses deep learning to learn \( Q^*(s,a) \). It was assumed that air traffic control would still provide decisions at discrete time steps. Therefore, learning \( Q_\theta(s,a) \) could be done using backward induction where the future utilities are learned first. In practice, when the loss function no longer or never considered \( \tilde{Q}^*(s,a) \), \( Q_\theta(s,a) \) would learn to do nothing. Future work could explore deep reinforcement learning further.

In the bigger picture, future work should expand the scope of this method by relaxing assumptions about number of aircraft, launch location, altitude, and velocity. This expansion is necessary for adoption. When considering adoption, it is possible that the presented method could be improved by combining the heading and metering
techniques. For instance, an oceanic aircraft can be rerouted using metering until its in the domestic airspace and if safety would increase, the aircraft could transition to heading reroutes. Combining the heading and metering techniques would produce a comprehensive design for domestic and oceanic flights.

It is also possible that the presented offline method could be improved if it was combined with an online technique. This thesis suggests a method of solving all of the policies offline and querying them in real time but future work should consider the alternative of providing preventative actions from an offline solution and reactive actions from an online solution. The offline solution would model the problem with the predicted weather, launch trajectory, failure modes, etc. and reroute aircraft when no anomaly has occurred to ensure the aircraft is in a resolvable location if an anomaly was to occur. If an anomaly does occur, then an online solution would be generated with the actual weather, launch trajectory, failure modes, etc.

\section{5.3 Potential Implementation}

Recently there have been numerous media articles highlighting the added airspace congestion \cite{53} and airline cost \cite{4} with the increase of space launches. Further, there was a Senate hearing to encourage the Federal Aviation Administration (FAA) be mandated to modernize and update their launch procedures to increase airspace efficiency \cite{6}. This thesis suggests doing this by switching from overly conservative, static, restrictive airspace closures to smaller, dynamic safety regions. To limit the impact on air traffic controllers, the policy results presented in this thesis should be transformed into a decision support tool that can seamlessly integrate space launch operations into air traffic management systems.

The decision support tool would take launch vehicle inputs directly from the launch provider and aircraft inputs directly from preexisting air traffic management systems. The decision support tool would use the inputs to query policies, like the ones presented in this thesis, and provide heading and metering commands that air traffic controllers can relay to affected aircraft. This method would limit the impact
on the air traffic controller by providing the reroute procedure rather than requiring the air traffic controller to generate the reroute procedure. Further, since this system integrates into the air traffic management system it is familiar to air traffic controllers increasing adoption and limiting additional training. Additionally, heading commands could be integrated into data link technology removing the reliance and burden on air traffic controllers [54]. In the future, this technology could be integrated into aircraft autopilot to reroute the aircraft during a space launch.

Simulations show that this method results in a more efficient airspace and with limited degradation in safety that future work could address. The presented models and solutions are parameterized for a specific launch, location, weather, and region of air traffic. If the scope of scenarios is expanded, this research could lead to an operable decision support tool.
Bibliography


[7] Testimony of Eric Stallmer President, Commercial Spaceflight Federation, Before the Committee on Commerce, Science and Transportation, United States


