Planning with Imperfect Information: Interceptor Assignment

by

Daniel B. McAllister

B.S., Operations Research and Mathematical Sciences
United States Air Force Academy, 2004

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Master of Science in Operations Research

at the   -

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2006

Author . . . .
Sloan School of Management
Interdepartmental Program in Operations Research
May 18, 2006

Approved by . . . .
Alexander Kahn
Member of the Technical Staff, The Charles Stark Draper Laboratory, Inc.
Technical Supervisor

Certified by . . . .
Leslie P. Kaelbling
Professor, Computer Science and Engineering
Research Director, MIT Artificial Intelligence Laboratory
Thesis Advisor

Certified by . . . .
Patrick Jaillet
Edmund K. Turner Professor and Head, Civil and Environmental Engineering
Thesis Supervisor

Accepted by . . . .
James B. Orlin
Edward Pennell Brooks Professor of Operations Research
Co-Director, Operations Research Center

Planning with Imperfect Information: Interceptor Assignment

by

Daniel B. McAllister

Abstract

We consider the problem of assigning a scarce number of interceptors to a wave of incoming atmospheric re-entry vehicles (RV). In this single wave, there is time to assign interceptors to a wave of incoming RVs. gain information on the intercept status, and then if necessary, assign interceptors once more. However, the status information of these RVs may not be reliable. This problem becomes challenging when considering the small inventory of interceptors, imperfect information from sensors, and the possibility of future waves of RVs.

This work formulates the problem as a partially observable Markov decision process (POMDP) in order to account for the uncertainty in information. We use a POMDP solution algorithm to find an optimal policy for assigning interceptors to RVs in a single wave. From there, three cases are compared in a simulation of a single wave. These cases are perfect information from sensors: imperfect information from sensors, but acting as it were perfect: and accounting for imperfect information from sensors using the POMDP formulation. Using a variety of parameter variation tests. we examine the performance of the POMDP formulation by comparing the probability of an incoming RV avoiding intercept and the interceptor inventory remaining. We vary the reliability of the sensors, as well as the number of interceptors in inventory. and the number of incoming RVs in the wave. The POMDP formulation consistently provides a policy that conserves more interceptors and approaches the probability of intercept of the other cases. However, situations do exist where the POMDP formulation produces a policy that performs less effectively than a strategy assuming perfect information.

Technical Supervisor: Alexander Kahn
Title: Member of the Technical Staff. The Charles Stark Draper Laboratory. Inc.

Thesis Advisor: Leslie P. Kaelbling
Title: Professor. Computer Science and Engineering

Thesis Supervisor: Patrick Jaillet
Title: Edmund K. Turner Professor and Head. Civil and Environmental Engineering

Acknowledgments

Assignment

Draper Laboratory Report Number T-1548

In consideration for the research opportunity and permission to prepare my thesis by and at The Charles Stark Draper Laboratory, Inc., I hereby assign my copyright of the thesis to The Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts.

Daniel B. McAllister, 2nd Lt, USAF

May 18, 2006

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

A Glossary of Acronyms and Terms

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Description

For decades, the United States has been vulnerable to ballistic missile attacks that could devastate the nation with nuclear, biological, or chemical weapons. In today's world, these attacks could come from not only a traditional foe such as North Korea or Iran, but also an accidental or unauthorized launch or, more likely, a stateless terrorist organization [15]. Many of these enemies view weapons of mass destruction as an asymmetric means to counter the conventional military might of the United States.

In recent years, ballistic missile technology has spread to more and more countries. Nations all over the world are developing missiles capable of reaching the United States [1]. On August 31, 1998, North Korea successfully launched the three-stage Taepo Dong 1 missile over Japan that almost reached Hawaii [5]. While it is not known whether this was a failed space launch or an intercontinental ballistic missile test, this initially undetected three-stage missile proved that North Korea had the capability to hit any point on earth with a several-hundred pound warhead [5]. Presently it is known that North Korea's Taepo Dong 2 missile could reach Alaska and Hawaii with a nuclear payload in a two-stage rocket configuration. If a third stage were added, this missile would likely be able to reach all of North America [15]. In addition to North Korea, China and Iran are also reported to be developing and

testing offensive ballistic missiles. These growing threats have led the U.S. to upgrade its current deterrence posture with a ballistic missile defense system. The goal of this system is to render missile attacks on the U.S. ineffective.

In 2004 the United States stood up its first defense against long-range missile attacks [15]. For the first time. the U.S. possesses the capability to intercept and destroy an incoming ballistic missile before it strikes its target [1]. While President Reagan envisioned a robust defense system capable of rendering missile attacks completely futile, the initial system is simpler and smaller. This Ground-based Midcourse Defense (GMD) system includes 10 silo-based interceptor missiles in central Alaska and southern California, which will be connected by an extensive command and control network to a mix of space- and land-based sensors [1]. Fort Greely, Alaska, currently has eight operational ground-based interceptor missiles and Vandenberg Air Force Base, California, has control of two more interceptors [3].

This ballistic missile defense system is designed to be the last line of defense if diplomacy and threats of retaliation fail. Employment of the ground-based interceptor missile is cued on satellite and radar data and then it uses its own sensors to identify targets launched from any site [5]. The interceptor correlates its observations with the information from the satellites and radar, and discriminates between decoys and actual warheads [5]. Interceptor missiles include a three-stage booster and are tipped with an Exoatmospheric Kill Vehicle (EKV) [15]. After the interceptor is approximately 140 miles in space. the kill vehicle detaches from the missile, locates an incoming missile, and destroys it with its sheer kinetic force.

As important as the interceptor missiles themselves is the sensor network used to detect an incoming attack. This network includes the Air Force's Defense Support Program (DSP) infrared early warning satellites. an upgraded early warning radar at Beale Air Force Base, California, an upgraded Cobra Dane surveillance radar on Shemya Island at the western end of the Aleutian islands. and three forward-deployed Navy Aegis destroyers equipped with Spy-1 radars [1]. These Aegis ships provide early target-track data [1]. All of these sensors and missile launch sites are connected to the heart of the system, the Command and Control, Battle Management and Communications network, based at Schriever Air Force

Base, Colorado.

The command and control aspect of this system ultimately relies on human operators to make decisions about how to defend against an incoming attack. In 2002, United States Northern Command (USNORTHCOM) was created and given the responsibility to defend the U.S. against any attack including a long-range missile attack [5]. In turn, the commander of USNORTHCOM holds that responsibility and would likely have the authority to make a decision as to how best to use the interceptor missiles to defend America against an attack. This commander will rely on United States Strategic Command (USSTRATCOM) to provide early warning from the previously described sensors and radars [5].

The following is a demonstration of how all components work together in an actual engagement [1]:

1. DSP satellites initially detect a threat missile's plume soon after it is launched.

2. This alerts the fire-control network which begins planning an intercept. Simultaneously, the other sensors such as Cobra Dane in Alaska, radars at Beale AFB, California, and Spy-1 radars on Aegis ships begin tracking the incoming missiles.

3. As operators receive higher quality data on the incoming attack, they launch their interceptor missiles.

4. As each EKV detaches from the missile and is deployed into space, radar continues to update it with track data.

5. Using these updates and its own sensors, the EKV locates the warhead of the incoming missile and collides with it.

6. Radar then assesses whether or not the incoming warhead was destroyed to determine if other interceptor missiles should be fired.

There are three phases of flight for an incoming ballistic missile: boost, midcourse, and terminal. The first phase, boost, usually lasts three to five minutes in which the missile is

19

powered by its engines [2]. During the midcourse phase, the missile travels above the atmosphere and releases its warheads becoming multiple objects [2]. When the warhead falls back into the atmosphere it enters the terminal phase [2]. Of these three stages, interceptors target and destroy incoming missiles in the midcourse phase-the longest duration of the three. During the 20 minute midcourse phase, a single engagement is assumed to be a "shoot-look-shoot" scenario, in which there are two opportunities to shoot interceptors at a wave of incoming missiles. We define a shot as a one-time assignment of multiple interceptors to multiple targets. The initial information regarding the number of incoming missiles is assumed to be completely accurate, and the decision maker has an opportunity to fire multiple interceptor missiles at this set of incoming missiles. Next, the decision maker has an opportunity to gain information on which incoming missiles were destroyed and which incoming missiles remain intact. Lastly, a final decision is made as to how many interceptors to fire at the believed remaining incoming missiles. We assume only enough time for two shots at a wave of incoming missiles. hence the term "shoot-look-shoot." The decision maker must weigh two important issues: saving some interceptor missiles for future waves of attacks, and stopping all incoming missiles from striking a target. This becomes a resource allocation problem under uncertainty with multiple objectives.

While this system aims to provide a very robust network of sensors to detect and track an incoming attack, there are several known limitations. The Cobra Dane radar's field of view can only detect a portion of North Korean missile launches [15]. The Beale AFB radar system has not completed all of its operational testing [15]. Overall, the entire system needs more extensive testing before America is assured to be safe from a ballistic missile attack.

The future holds a great deal of expansion for the ballistic missile defense system. As stated in the Missile Defense Agency's ballistic missile defense system overview, "The mission of the Missile Defense Agency is to develop an integrated, layered Ballistic Missile Defense System (BMDS) to defend the United States, its deployed forces, allies, and friends from ballistic missiles of all ranges and in all phases of flight" [1]. This means that in the future the defense system will include more than just the 10 ground-based interceptor missiles

designed to destroy missiles in the midcourse phase of flight. Eventually the BMDS will include Patriot Advanced Capability-3 missiles and Aegis Ballistic Missile Defense Standard Missile-3 missiles located on forward deployed ships used to destroy short- and medium-range ballistic missiles. The BMDS will have ground-based interceptors for intermediate-range and intercontinental ballistic missiles. An Airborne Laser will be added to the BMDS, employing a high-powered laser attached to an Air Force aircraft designed to destroy a missile in its boost phase. Lastly, the BMDS will have a terminal high altitude area defense element designed to destroy incoming missiles in their terminal phase [1]. In addition to adding more methods to shoot down incoming missiles, there will be improvements to current sensors and added sensors in other parts of the world to augment the current surveillance and detection component of the BMDS.

While all of these future components will likely prove to be important in the layered, integrated defense of the United States, this thesis will focus only on the GMD, as it is the newest and presently the only operational defense against a long-range missile attack.

1.2 Motivation

Because the single engagement problem is a "shoot-look-shoot" situation, there is information to be gained in between the first and second decisions. However, in order for this "shoot-look-shoot" technique to be successful, it requires accurate kill assessment after the first shot opportunity [17]. To the best of our knowledge, this is the first work that addresses imperfect kill assessment in this domain. Previous work has assumed that after the first shot, it is known with certainty whether each target has survived or not. This assumption, however, may not actually be valid. One of the main objectives of this thesis is to compare the performance of a system making this assumption and a system that tries to account for imperfect kill assessment. The focus of this thesis will be managing the uncertainty in kill assessment.

1.3 Overview of Thesis

This thesis describes the single engagement problem assuming imperfect kill assessment. We provide an overview of related research and previous approaches to this problem. We then introduce a partially observable Markov decision process (POMDP) formulation and assess the performance of this formulation compared to other methods of solving the problem. We measure the value of our formulation through a series of experiments and statistical analysis. The individual chapters are summarized as follows:

Chapter 2: Related Research

In this chapter we discuss the related research applicable to the single engagement problem. We begin with a discussion of dynamic programming and its characteristics, as well as guidelines for solving a dynamic programming problem. We continue with a description of Markov decision processes (MDP) as a class of problems typically solved by dynamic programming. We outline the components and decision cycle of an MDP. Next, we describe a variant of the MDP: the partially observable Markov decision process (POMDP). We discuss the differences between the MDP and POMDP and how they are handled. This chapter concludes with a discussion of the weapon-target assignment (WTA) problem as an approach to the single engagement problem. We explain how this approach fails to account for the imperfect information that is assumed by this thesis.

Chapter 3: Problem Formulation

This chapter outlines three cases of the single engagement problem that we will use to assess the impact of imperfect kill assessment: perfect information, imperfect information assumed perfect, and imperfect information taken into account. Case 1 acts as a best-case, and is the case assumed by previous approaches to this problem. Case 2 uses the same strategy as the first case, except that the assumptions of perfect information no longer exist. Case 3 accounts for this imperfect information and makes decisions based on this new assumption. We focus on the third case and formulate it as a POMDP.

Chapter 4: Implementation

We begin this chapter with a description of the solution process for POMDPs. We start with a description of the POMDP solver software and the solution algorithms it uses. Next we discuss how we simulate the single engagement using either the POMDP solver for Case 3 or the maximum marginal return (MMR) algorithm for Case 2 to generate a policy solution. This chapter continues with a description of the experimental design. We divide our experiments into three sets: initial experiments, a central composite design experiment, and a set of single-factor experiments. All experiments begin with a baseline setting for all factors and change factors from this scenario. First, we conduct initial experiments to examine the effect of three factors on the performance of the POMDP solver and MMR algorithm. These factors are left constant in the remaining experiments. Next, we use a central composite design (CCD) experiment testing the effects of five different factors on the difference in performance between the two cases with imperfect information. Lastly, we run a series of single-factor experiments that vary the same five factors individually. This provides a more detailed understanding of each factor's effect on the performance of each case.

Chapter 5: Results and Analysis

This chapter presents the results and analysis of the experiments described in Chapter 4. We begin with outcomes of the baseline scenario and the results from three initial experiments. We continue with statistical analysis on three quadratic models created from the CCD in Experiment 4. Lastly, we assess the impact of the factors in the final four one-factor experiments.

Chapter 6: Summary and Future Work

This chapter summarizes the single engagement problem and the POMDP formulation, along with experimental results and conclusions. It ends with a discussion of suggested future work for this problem.

1.4 Chapter Summary

The U.S. has begun to stand up its Ground-based Midcourse Defense–the first defense system designed to defend against long-range ballistic missile attacks. This system's 10 interceptor missiles are designed to locate and destroy incoming missiles in space based on information from a complex sensor network of satellites and radar. Due to the very limited number of interceptor missiles in inventory, each interceptor is a high-valued asset. While still being tested and upgraded, there is a great deal of uncertainty in this system. It is not known how effective the interceptors will be at destroying incoming missiles, and there may be problems detecting and tracking incoming missiles accurately with the current sensor network. The problem of assigning interceptors to incoming missiles in an attack becomes much more challenging due to the uncertainty in information from the sensor network. With only two shots at an incoming missile, it is very important to have accurate kill assessment; that is, to know which incoming missiles have been destroyed and which ones are still headed inbound. Finding a way to decide how many interceptors to use in an attack that accounts for this imperfect kill assessment could be very valuable. This task will be the focus of this thesis. We accomplish this by assessing the impact of a POMDP formulation that accounts for imperfect information, and comparing it to existing approaches that do not account for this uncertainty.

Chapter 2

Related Research

In this chapter we discuss the research related to this problem in order to formulate it mathematically and ultimately solve it. We begin with a discussion of dynamic programming and Markov decision processes. Then we discuss the partially observable Markov decision process, which will be used in our formulation. Finally, we discuss previous formulations of related problems and their applicability to other domains.

2.1   Dynamic Programming

The single engagement problem described in Chapter 1 is a sequential decision problem. One of the primary techniques used to solve a problem that optimizes an objective over several decisions is dynamic programming. Although dynamic programming problems do not have a specific formulation, they can be easily recognized by several characteristics [9]:

1. The problem can be partitioned into stages. At each stage a policy decision or action must be made.

2. Each stage has a number of states associated with that stage, which are the possible conditions that the system could be in at that stage. There may be a finite or infinite number of states.

3. The policy decision made in each stage will transform the current state into a state associated with the next stage.

4. A recursion can be created on the optimal cost/reward from the origin state to the destination state.

5. To solve the problem, an optimal policy over the entire problem must be found. This policy provides the optimal decision at each stage for each possible state.

6. An optimal policy for a future stage is only dependent on the current state and not the decisions made in previous stages. This property is the Markovian property and is the principle of optimality for dynamic programming.

7. The solution procedure begins by finding the optimal policy for the final stage.

8. There is a recursive relationship that provides the optimal policy for stage n given the optimal policy for stage $n + 1$.

9. The solution procedure uses the recursive relationship to start at the last stage and move backward iteratively finding the optimal policy at each stage. This is carried out until the optimal policy at the first stage is found.

In dynamic programming the time indices are called epochs. The 0-epoch begins at the end of the planning horizon at the final stage and the epochs increase until the first stage is reached. In other words an epoch is the number of stages left in which actions can be taken.

According to Bertsimas and Tsitsiklis, the following are guidelines for solving a dynamic programming problem [4]:

1. View the choice of a feasible solution as a sequence of decisions occurring in stages, and the total cost or reward as the sum of the costs of each decision.

2. Define the state as a summary of all relevant past decisions.

3. Let the cost/reward of the possible state transitions be the cost/reward of the corresponding decision.

2.2   Markov Decision Process

One variant of the typical dynamic programming problem in which state transitions are non-deterministic is a Markov decision process (MDP). An MDP is the specification of a sequential decision problem for a fully observable environment with a Markovian transition model and additive rewards [14]. An MDP is defined by four primary components:

1. A set of states: $s \in \mathcal{S}$

2. A set of actions for each state: $a \in \mathcal{A}$

3. A transition model: $T(s, a, s')$

4. A reward function for both intermediate and terminal rewards for each state: $R(s, a, s')$

The transition model specifies the probability of transitioning from one state, s, to another state, s', in one time step given an action, a. In an MDP there can be rewards for transitioning from one state to another in intermediate time steps as well as a terminal reward for being in a state at the final stage. An MDP may transition an infinite number of times (infinite horizon) or it may only transition a finite number of times (finite horizon). The goal of an MDP is to choose the optimal actions for the respective states when considering the expected rewards/costs of those actions. For infinite horizon problems, a discount factor, δ, is used to value current rewards over future rewards [14]. Again, the Markovian property, or "lack-of-memory property," applies because the transition probabilities are unaffected by the states in stages prior to the current stage [9].

The decision cycle of a Markov decision process is as follows:

1. Based on the current state, an optimal action or decision is chosen from a set of possible actions.

2. The selected action determines the probabilities of transitioning into a new state.

3. An immediate reward/cost is incurred.

4. The state of the system is determined after each transition.

5. The process is repeated.

A complete policy for the MDP is a specification of the optimal actions for each state. A solution maps a state to an action $(\mathcal{S} \rightarrow \mathcal{A})$ where $s \in \mathcal{S}$ and $a \in \mathcal{A}$. The objective is to find an optimal policy of actions considering both immediate and terminal rewards.

Markov decision processes are an important class of problems that are often solvable through dynamic programming. There are solution methods for MDPs that run in polynomial time in $|\mathcal{S}|$, $|\mathcal{A}|$, and finite horizon or infinite horizon with a discount of $\frac{1}{1-\delta}$. The concept of dynamic programming applied to MDPs forms the basis for the focus of this thesis: partially observable Markov decision processes.

2.3 Partially Observable Markov Decision Process

A Markov decision process as defined in Section 2.2 assumes that the environment is fully observable. This means that the state of the system is always known with certainty. However, in many real-world problems the environment is only partially observable, and the state of the system may not be known with certainty. As an example, this partial knowledge may occur if the observer is removed from the process in some way and must gain information over an imperfect communications channel [16]. In the world of ballistic missile defense, human operators are forced to rely on sensors and radar to determine the status of incoming ballistic missiles.

Using an MDP to model this type of partial observability falls short as step one of the decision cycle is not possible. In order to model systems with these characteristics. they are defined as partially observable Markov decision processes (POMDP). The POMDP. originally developed by Drake [8], but formalized by Sondik [16], is "the specification of a sequential decision problem for a partially observable environment with a Markovian transition model and additive rewards" [14]. A POMDP is an MDP that handles the case in which states can "look" the same or where the same state can "look" different each time it is visited. A

POMDP is defined by six primary components:

1. A set of states: $s \in \mathcal{S}$

2. A set of actions for each state: $a \in \mathcal{A}$

3. A transition model: $T(s, a, s')$

4. A set of observations: $o \in \mathcal{O}$

5. An observation model: $O(s, o, a, s')$

6. A reward function for both intermediate and terminal rewards for each state: $R(s, o, a, s')$

These elements are defined in more detail and in terms of the single engagement problem in Chapter 4.

A POMDP has the same elements as an MDP with the addition of the set of observations and the observation model. The observation model specifies the probability of perceiving observation o given that the system started in state s, ended in state s', and took action a to get there. In addition, the reward function may now also depend on observation o.

In MDPs the optimal action depends only on the current state, and a solution maps a state to an action. In POMDPs the current state is not known, so there is no way to map a state to an action. Without knowing the current state, the optimal action depends on the complete history of the system, including the initial information about the system, as well as all subsequent actions and observations. Sondik proved that a sufficient statistic for this complete history of the system is the belief state [16]. A belief state, $b \in \Pi(S)$, is the probability distribution over all possible states where $\Pi(S)$ is the set of all possible belief states [14]. Let $b(s)$ be the probability of being in the actual state s given the belief state b. In a POMDP, the optimal action depends only on the system's current belief state [14]. A solution maps the belief state to an action ($\Pi(S) \rightarrow \mathcal{A}$). A graphical depiction of a two state belief state is shown in Figure 2-1. In this two-state POMDP, the belief state can be represented by a single probability, p, of being in one state. The probability of being in the other state is simply $1 - p$. Therefore, the entire belief space can be represented as a line

29
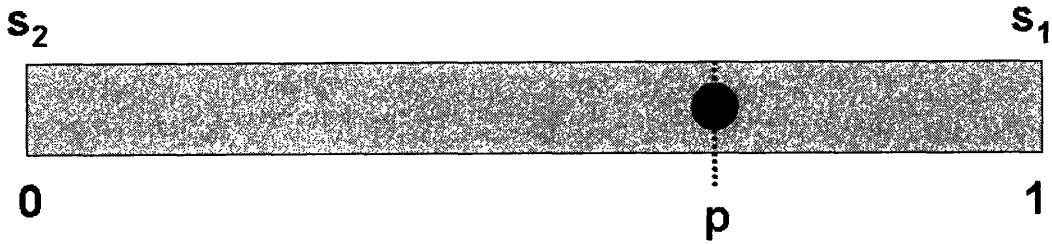
Figure 2-1: Two-state Belief State

segment. The point at 0 on the line segment indicates there is no way the system is in state s_1 and must be in state s_2. Likewise. the point 1 on the line segment indicates the system is in state s_1 with certainty, and there is no chance of being in state s_2. This means that $b = (p, 1 - p)$ where $b(s_1) = p$ and $b(s_2) = (1 - p)$.

While the Markovian property does not hold for the state of the system, it does hold for the belief state of the system. The optimal policy for any given stage is only dependent on the current belief state and not decisions made in previous stages.

The decision cycle in a POMDP formulation is now:

1. Based on the current belief state, an optimal action or decision is chosen from a set of possible actions.

2. The selected action determines the probabilities of transitioning into a new state.

3. An observation on the state of the system is made.

4. An immediate reward/cost is incurred.

5. The new belief state is calculated based on the action and observation after each transition.

6. The process is repeated.

The current belief state can be calculated as the conditional probability distribution over the actual states given the previous observations and actions so far. If b was the previous

belief state. action a was taken. and observation o was perceived. then the new belief state. b'. is calculated for each state. s'. by Equation 2.1.

$$b'(s') = \frac{O(s'.a.o) \sum_{s \in \mathcal{S}} T(s.a.s')b(s)}{\Pr(o|a.b)} \qquad (2.1)$$

The denominator normalizes the resulting belief state so that it sums to one. and can be computed by Equation 2.2.

$$\Pr(o|a.b) = \sum_{s' \in \mathcal{S}} \left[O(s'.a.o) \sum_{s \in \mathcal{S}} T(s.a.s')b(s) \right] \qquad (2.2)$$

As an example of updating the belief state. assume the system has two possible states (s_1 and s_2). two possible actions (a_1 and a_2). and two possible observations (o_1 and o_2). A graphical representation is shown in Figure 2-2. The larger black dot represents the
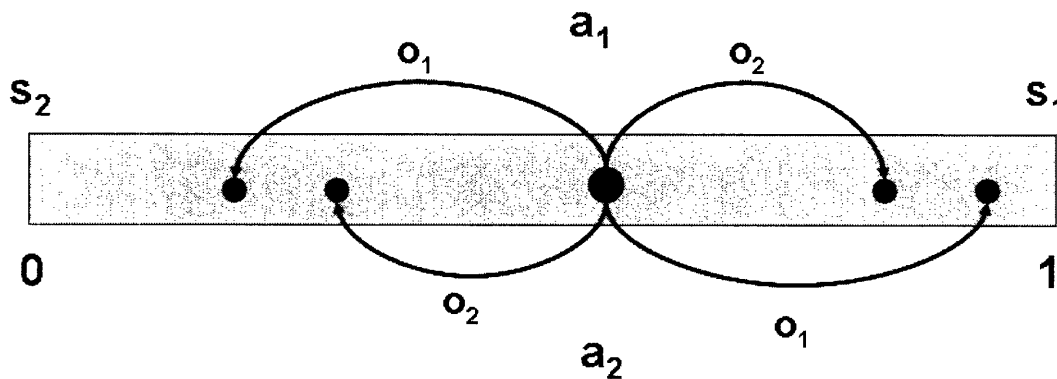
Figure 2-2: Updating the Belief State

starting belief state. and each of the smaller dots represent a possible resulting belief state given a certain action and observation. The arcs linking these belief states represent the transformation of belief states by Equation 2.1. In this example there are only four new possible belief states: one for each combination of actions and observations.

A complete policy for the POMDP is a specification of the optimal actions for each belief state. The objective is to find an optimal policy of actions considering both immediate

31

and terminal expected rewards. However, the challenge in finding an optimal policy for a POMDP is that, unlike the discrete state space in an MDP, the belief space for a POMDP is continuous. In contrast to MDPs, the belief state probabilities create a state space of infinite size. To deal with this, the belief space can be partitioned into regions where certain actions are optimal and the long-term value is a linear function of the belief state.

Assume now that the system has three possible actions. The belief space could be partitioned into three regions where each of these actions is optimal as shown in Figure 2-3. These lines in two dimensions, and hyperplanes in greater dimensions, are called alpha vec-
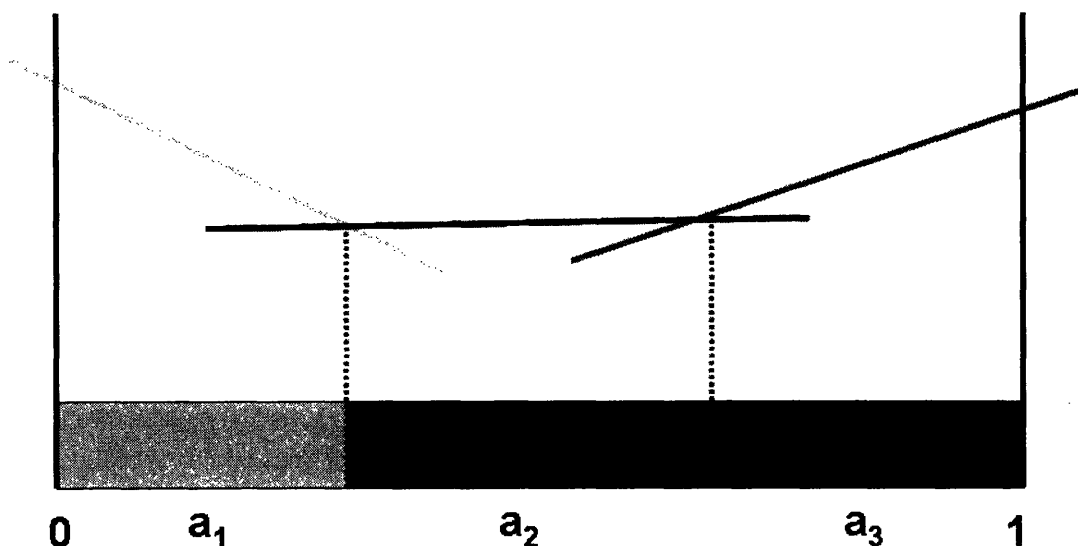
Figure 2-3: Belief State with Value Function

tors. They are simply vectors with a value for each state, and correspond to an action. An action is optimal where its alpha vector dominates other alpha vectors. Graphically, this means one alpha vector lies above another. The value function for a POMDP, $V(b)$, is simply the upper surface of the alpha vectors over the belief space–a piecewise linear combination of the alpha vectors. $V(b)$ is a mapping of the belief space to the expected total reward [7]. Because the value function is piecewise linear and convex, the belief space can be partitioned into regions where certain actions are optimal. Despite the simplicity of Figure 2-3, a belief

space may be partitioned into many more regions than actions, and therefore an action can be optimal in several different regions. The belief state can also be represented as a vector with probabilities for each state. Finding the optimal action for a given belief state requires calculating the dot product of the belief state vector and each alpha vector and finding which dot product has the greatest value.

2.4 Mathematical Approaches

One previous approach to problems similar to the single engagement problem is the weapon-target assignment (WTA) problem. In the static WTA, weapons are assigned to targets in order to minimize either the total expected number or the expected value of the remaining targets [10]. A value is assigned to each target, and each weapon-target pair has a kill probability associated with it. This is the probability that a certain weapon will destroy a certain target. The assignment of a weapon to a target is independent of all other weapons and targets.

A dynamic weapon-target assignment problem is a static weapon-target assignment problem that involves multiple stages. This means that the outcome of an assignment in one stage can affect the assignment in the next stage. Each stage consists of two steps:

1. Determine which targets have survived the assignment in the previous stage.

2. Assign a subset of the remaining weapons to the targets that survived based on the objective.

The missile defense single engagement problem can be defined as a dynamic weapon-target assignment problem. In this application, the weapons are interceptor missiles or kill vehicles, and the targets are incoming missiles. A certain portion of the inventory of interceptor missiles (the weapons) must be assigned to a number of incoming missiles (the targets). In a single engagement there are two stages, so that the outcome of the first "shot" in stage 1 may determine the assignment of interceptor missiles for the second "shot" in stage 2. The objective may be to minimize the probability an incoming missile leaks

through defenses. to minimize the damage done if incoming missiles are headed for different locations, or a variety of other potential objectives. The objective function may also be a weighted function maximizing not only the probability of no leakage, but also the remaining interceptor missiles left in inventory.

In this problem all incoming missiles (targets) are assumed to have the same value. In addition, because all interceptors are assumed to be identical, all kill probabilities are assumed equal. With these assumptions, Hosein, Walton, and Athans showed that in a dynamic problem with N targets and M weapons, it is optimal to spread the weapons evenly among all targets at each stage. In addition, given a two-stage problem in which $M > N$ with M_1 being the number of assigned weapons in stage one and M_2 being the number of assigned weapons in stage two, it was shown that the optimal assignment has the property that $M_1 \geq N$ [10].

These conclusions prove to be very useful in solving the single engagement problem. However, the addition of imperfect kill assessment after the first stage makes it more difficult to use the weapon-target assignment formulation. Under imperfect information, step one of each stage becomes very challenging: determine which targets have survived the last assignment. This information is no longer known with certainty, and this makes it much more difficult to accomplish step two: assign weapons to the targets which survived.

2.4.1 Applicability to Other Problems

By no means is this problem only applicable to ballistic missile defense. The work on this problem can easily be applied to a wide range of battle management problems. More specifically, the issues of a limited time window, limited resources, imperfect kill assessment, and severe consequences for every action are very relevant to many defense and non-defense related problems. As an example of a type of problem that could be formulated in this manner, we consider the use of unmanned aerial vehicles (UAVs) for reconnaissance and surveillance. A limited number of UAVs may be assigned to a number of different ground targets. Information on these targets may be required in a timely manner. Imagery from

the UAVs may not be complete or conclusive, but assigning more UAVs to a target may improve the information received. Assigning more UAVs to a target may come at some cost, such as losing information from other targets. This is one example of a problem to which the approach in this thesis may also apply.

2.5 Chapter Summary

This chapter progressed through a discussion of the mathematical tools used in this thesis beginning with the general technique called dynamic programming, a class of problems called Markov decision processes, and ending with a variant of MDPs, the partially observable Markov decision process. All of the nine characteristics of dynamic programming problems previously described are applicable to the POMDP when applied to the single engagement problem. In particular, POMDPs are solved backwards iteratively. The basis for the POMDP is the Markov decision process and its four primary elements. The POMDP is simply an MDP with only partial knowledge of the state. This complication adds two new elements to the MDP: the set of observations and the observation model. Instead of making decisions based on the current state, decisions must be made based on the belief state, a probability distribution over all states.

Previously, Hosein, Walton, and Athans formulated the single ballistic missile engagement as a weapon-target assignment problem. Using this formulation, several key results were proved about the optimal assignment of interceptors to incoming missiles. While this formulation provides valuable insight into this problem, it fails to account for the imperfect kill assessment in the GMD. Lastly, we discuss the applicability of our approach to other problems. The value of formulations accounting for imperfect information transcend ballistic missile defense. This approach could be applicable to any problem dealing with limited resources, uncertainty, and assignments.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Problem Formulation

This chapter discusses the three cases we use to assess the impact of imperfect information on the single engagement problem. These three cases describe different assumptions and realities in the single engagement problem: a system that has perfect kill assessment, a system that assumes perfect kill assessment incorrectly, and a system that makes decisions taking the imperfect kill assessment into account. We formulate the third case as a partially observable Markov decision process (POMDP). In order to assess the performance of this approach, we compare it to the other two cases.

3.1   Perfect Information

In the best case, Case 1, the information received after the first action would be completely accurate. In this "perfect information" case, the probability of observing a miss given a miss actually occurred, $\Pr(miss|miss)$, and the probability of observing a hit given a hit actually occurred, $\Pr(hit|hit)$, would both equal one. The assumption that an observation is completely accurate simplifies the problem. Case 1 describes the assumptions of previous work conducted on the single engagement problem.

3.2 Imperfect Information Assumed Perfect

Given that previous formulations of this problem have assumed perfect information. an important situation to analyze is one where $\Pr(miss|miss) \neq 1$ and $\Pr(hit|hit) \neq 1$. yet the policy is created assuming that $\Pr(miss|miss) = 1$ and $\Pr(hit|hit) = 1$. This means that the observations of which incoming targets were hit and missed are taken to be true. even though there is a chance those observations are incorrect. These assumptions could have disastrous consequences in an actual engagement. If an incoming target were falsely believed to be destroyed. and consequently no more interceptors were fired at it, it would be allowed to leak through defenses without being engaged. We refer to this situation as Case 2. In this case the decisions are made with the same assumptions as the first case. Reality. however, is different.

3.3 Imperfect Information: POMDP Formulation

In Case 3, the imperfect information from sensors after the first shot is known and the policy solution attempts to account for it. In order to do this. the single engagement problem is modeled as a partially observable Markov decision process (POMDP). This POMDP has a horizon of two stages to model the "shoot-look-shoot" aspect of the problem. Each decision or "shot" is the action of that stage.

States

We define a state, s, as the following:

$$s \equiv (\beta, \rho)$$

where β is the interceptor inventory remaining and ρ is the number of targets remaining. Given this state definition, an initial interceptor inventory β_0. and an initial wave of targets

ρ_0. the size of the state space is:

$$(\beta_0 + 1)(\rho_0 + 1)$$

Adding 1 to both β_0 and ρ_0 in this expression, accounts for the states in which $\beta = 0$ or $\rho = 0$.

Actions

We define an action, $a \in \mathcal{A}$, as the total number of interceptors assigned to all targets, given the current state. s. Many logical restrictions could be placed on the action. As an example of an action that could be restricted, consider the action of assigning fewer interceptors than targets even with enough interceptors in inventory. This action would allow a target to pass through defenses without being engaged. and appears not to be logical. However, in our formulation the only restriction placed on the allowable actions is $a \leq \beta$. By assigning a large negative value in the reward function. these impossible actions. in which $a > \beta$, are restricted. Although this is the only restriction placed on actions. in theory an optimal policy will not choose illogical actions given the proper reward function. Given the state. $s = (\beta, \rho)$, the number of allowable actions is equal to $\beta + 1$.

Transition Model

In our formulation the only uncertainty affecting the transition from one state to another is the single-shot probability of kill (SSPK), which is the probability a single interceptor hits a single target. The probability of transitioning from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ after taking action $a \in \mathcal{A}$ is denoted by $T(s, a, s')$. The transition model is the three-dimensional matrix of all of these values. Assuming that the interceptors are evenly distributed among all targets. either all targets will have the same number of interceptors assigned to them, or one group of g_1 targets will have n interceptors assigned to them and the remaining g_2 targets will have $n - 1$ interceptors assigned to them. As an example, if 7 interceptors were assigned to 3 targets. one target would have three interceptors assigned to it. and two targets would each have two interceptors assigned to them. This means that $g_1 = 1$ target, $g_2 = 2$ targets.

and $n = 3$ interceptors. Let PK_1 and PK_2 equal the overall probability of no leakage for one target given the number of interceptors assigned to each target in the groups containing g_1 and g_2 targets respectively. PK_1 and PK_2 can be calculated using Equation 3.1 and Equation 3.2.

$$PK_1 = 1 - (1 - SSPK)^n \tag{3.1}$$

$$PK_2 = 1 - (1 - SSPK)^{(n-1)} \tag{3.2}$$

Let h be the number of hits or number of targets destroyed. This value is calculated by Equation 3.3.

$$h = \rho_s - \rho_{s'} \tag{3.3}$$

Because there may be two groups of targets with PK_1 and PK_2 associated with them, there are many combinations of hits from each of the two groups of targets that result in the same number of overall hits and thus the same transition. To calculate the transition probabilities, $T(s, a, s')$, Equation 3.4 sums over all possible combinations that result in the same number of hits.

$$T(s, a, s') = \sum_{i=0}^{h} \left[\left[\binom{g_1}{i} PK_1{}^{g_1-i}(1 - PK_1)^{g_1} \right] \left[\binom{g_2}{h-i} PK_2{}^{g_2-h+i}(1 - PK_2)^{g_2} \right] \right], \forall_{s \in \mathcal{S}} \forall_{s' \in \mathcal{S}} \forall_{a \leq \beta} \tag{3.4}$$

It should also be noted that for any action, a, the transition probabilities sum to one over the ending states:

$$\sum_{s' \in \mathcal{S}} T(s, a, s') = 1$$

Observation Model

The observation model is the main component that differentiates a POMDP from an MDP. In this problem the observation probability, $O(s', a, o)$, is based on the probabilities $\Pr(miss|miss)$ and $\Pr(hit|hit)$. Table 3.1 depicts a confusion matrix of these probabilities. Ultimately, the observation model is a three-dimensional matrix dependent on the starting state, action, and resulting state.

40

		Observed	
		Hit	Miss
Actual	Hit	$\Pr(hit|hit)$	$\Pr(miss|hit)$
	Miss	$\Pr(hit|miss)$	$\Pr(miss|miss)$

Table 3.1: Confusion Matrix

In order to calculate the observation probabilities we define the following variables:

Variable	Definition
P_{mm}	$\Pr(miss|miss)$
P_{hm}	$\Pr(hit|miss)$
P_{hh}	$\Pr(hit|hit)$
P_{mh}	$\Pr(miss|hit)$
m_o	number of observed target misses
h_o	number of observed target hits
m_a	number of actual target misses
h_a	number of actual hits
lb	lower bound on number of actual misses
ub	upper bound on number of actual misses

Table 3.2: Variable Definition for Observation Probability Calculation

where $lb = max(0, (m_o - h_a))$ and $ub = min(m_o, m_a)$ in order to account for the correct combinations of possible observations. Equation 3.5 shows the equation to calculate each observation probability, where $O(s', a, o) = \Pr(o|s', a)$.

$$O(s', a, o) = \sum_{i=lb}^{ub} \left[\left[\binom{m_a}{i} P_{mm}{}^i (1 - P_{mm})^{m_a-i} \right] \left[\binom{h_a}{m_o-i} P_{mh}{}^{m_o-i} (1 - P_{mh})^{h_a-(m_o-i)} \right] \right] \qquad (3.5)$$

where

$$\sum_{o \in \mathcal{O}} O(s', a, o) = 1$$

Reward Model

The objective of the single engagement problem is to minimize the probability that any targets leak through defenses while maximizing the number of interceptors left in inventory after the engagement. To reconcile these two competing objectives, a weight, w_I, is used.

41

where $0 \leq w_I \leq 1.0$. Let P_{nl} equal the probability of no leakage for the entire single engagement. The reward function in Equation 3.6 balances the percentage of initial inventory of interceptors remaining and the probability that no targets leak through defenses after the transition. We scale the remaining inventory by the initial inventory in order for $0 \leq \frac{\beta_{s'}}{\beta_0} \leq 1$ just as $0 \leq P_{nl} \leq 1$. A w_I close to zero tells the POMDP solver to be much more conservative with its inventory of interceptors, while a w_I close to one tells the POMDP solver to value minimizing leakage much more than saving interceptors. This weight is varied in subsequent experiments to determine its impact on engagement success.

$$R(s, o, a, s') = (1 - w_I) \left(\frac{\beta_{s'}}{\beta_0} \right) + w_I P_{nl} \tag{3.6}$$

In the context of this problem, as with many POMDPs, the ending state is more important than the intermediate states. For example, targets remaining after the final shot have far more severe consequences than targets remaining when there is still one shot left at them. To account for this characteristic, terminal rewards can be specified for the POMDP. These rewards simply place a value on each of the possible final states. The terminal reward function in this problem took the form of Equation 3.7, where w_{T_1} is the weight given to the inventory remaining and w_{T_2} is the weight given to the targets remaining. In this equation, β_s is not scaled as it is in the intermediate reward function, because its competing metric is ρ_s, which is the number of targets remaining.

$$F(s) = w_{T_1} \beta_s - w_{T_2} \rho_s \tag{3.7}$$

3.4 Chapter Summary

This chapter discusses the three cases to be used for comparison to assess the effect of imperfect information on interceptor assignment. The first case assumes (correctly) perfect information from sensors. The second case assumes (incorrectly) perfect information from sensors in a world where information is not perfect. The third case attempts to account for

imperfect information in its decision making. The focus of this chapter is on the last case.
which is formulated as a partially observable Markov decision process.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Implementation

This chapter discusses the methodology used to solve and test each of the three cases outlined in Chapter 3. We begin with a description of the maximum marginal return (MMR) algorithm used to make assignments for Cases 1 and 2. Next, we discuss the POMDP solution algorithms used in Case 3. We then outline how these algorithms are used in the solution process. Finally, we discuss the experimental design utilized to compare the performance of the three cases.

4.1 MMR Algorithm

Although they deal with different information certainty, the first two cases described in Chapter 3 use the same algorithm to make interceptor assignments: the maximum marginal return (MMR) algorithm. The MMR algorithm variant used in this work assigns interceptors to targets in a single engagement. The objective of this algorithm is to minimize the number of interceptors used while meeting a probability of no leakage threshold. These two goals are in opposition to each other. In order to do this, the algorithm iteratively assigns interceptors to targets one-at-a-time until either the overall probability of no leakage reaches the threshold or no interceptors remain in inventory. The threshold for this MMR version is 0.99 to focus on maximizing P_{nl} without using all inventory. If the threshold was $P_{nl} = 1.0$, the algorithm

would consistently use all interceptor inventory. After each iteration, every target's marginal probability of leakage is calculated and the target with the highest probability is the next target to gain another interceptor assignment in the following iteration.

The MMR algorithm also assigns interceptors to one of the two time stages. For instance, the algorithm initially assigns some interceptors for stage one and some for stage two, but the assigned interceptors for stage two are not actually fired in stage one, but are planned to be fired. In this way the algorithm chooses the best two-stage strategy, with the knowledge that it will replan after kill assessment of the first stage. After the assignment is made in stage one, the algorithm is run again to make a new assignment for stage two based on the number of targets that still remain. During the first assignment, when determining which time stage to assign an interceptor, if none have been assigned to a target, the assignment is made to the first time stage. Otherwise, the assignment is made to the stage with fewer interceptors assigned, with the second stage gaining the assignment in the event of a tie. This second stage preference provides the same probability guarantee with fewer expected interceptors used. A description of this algorithm is shown in Algorithm 4.1.

Algorithm 4.1 Maximum Marginal Return Algorithm

$B \Leftarrow \beta$
$P_{nl} \Leftarrow 0$
while $P_{nl} < 0.99$ and $B > 0$ **do**
    for all ρ targets **do**
        Find target with highest probability of leakage
        Assign one interceptor to that target
        if In stage 1 **then**
            if First interceptor assigned to target **then**
                Interceptor assigned to first stage
            else if Each stage has equal interceptors assigned **then**
                Interceptor assigned to second stage
            else
                Interceptor assigned to stage with fewer interceptors assigned
    Recalculate P_{nl} based on new assignments
    $B \Leftarrow B - 1$

The MMR algorithm was tested on a variety of scenarios of varying interceptors and targets under the Case 1 assumptions. The algorithm provides a policy solution and from

that policy a probability of no leakage as well as an estimated inventory remaining are calculated for each scenario. These measures of performance. P_{nl} and β_2. are calculated using $SSPK$. The values shown in Table 4.1 and Table 4.2. were estimated using Monte Carlo simulation of 10.000 trials of the single engagement assuming $SSPK = 0.8$. These probabilities provide a good benchmark for the probabilities for the other cases. Likewise. the average remaining inventory for Case 1 provides a good benchmark for the other cases' remaining inventory. Hypothetically. P_{nl} is greater in this case. than in the case in which the information is imperfect.

Interceptors	Targets									
	1	2	3	4	5	6	7	8	9	10
1	0.7961	0	0	0	0	0	0	0	0	0
2	0.9593	0.6403	0	0	0	0	0	0	0	0
3	0.9922	0.8963	0.5076	0	0	0	0	0	0	0
4	0.9984	0.9708	0.8214	0.4059	0	0	0	0	0	0
5	0.9984	0.9888	0.9438	0.7445	0.3294	0	0	0	0	0
6	0.9982	0.9931	0.9737	0.9020	0.6626	0.2675	0	0	0	0
7	0.9982	0.9968	0.9872	0.9504	0.8541	0.5773	0.2060	0	0	0
8	0.9980	0.9973	0.9911	0.9775	0.9173	0.7915	0.5084	0.1671	0	0
9	0.9980	0.9985	0.9957	0.9812	0.9572	0.8794	0.7371	0.4406	0.1277	0
10	0.9987	0.9986	0.9963	0.9910	0.9684	0.9376	0.8428	0.6811	0.3731	0.1093
11	0.9979	0.9982	0.9975	0.9927	0.9879	0.9515	0.9047	0.8052	0.6194	0.3166
12	0.9985	0.9977	0.9988	0.9943	0.9871	0.9764	0.9325	0.8779	0.7610	0.5655
13	0.9988	0.9977	0.9990	0.9952	0.9915	0.9810	0.9664	0.9153	0.8439	0.7125
14	0.9987	0.9981	0.9995	0.9968	0.9943	0.9898	0.9746	0.9510	0.8915	0.8094
15	0.9982	0.9982	0.9991	0.9974	0.9951	0.9910	0.9843	0.9610	0.9326	0.8599
16	0.9984	0.9984	0.9996	0.9988	0.9957	0.9928	0.9880	0.9753	0.9526	0.9086

Table 4.1: Probability of No Leakage with Perfect Information (Case 1) using MMR Algorithm. $SSPK = 0.8$

4.2 POMDP Solver

To solve the POMDP used in Case 3, we use the software *pomdp-solve*, version 4.0. developed by Cassandra [6]. Using a basic dynamic programming approach working backwards in time. this software can use a variety of different algorithms to solve the POMDP. It is capable of solving both finite and infinite horizon problems and implements a number of

Interceptors	Targets									
	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0.8012	0	0	0	0	0	0	0	0	0
3	1.613	0.639	0	0	0	0	0	0	0	0
4	2.4	1.2772	0.5181	0	0	0	0	0	0	0
5	3.4081	1.9284	1.0178	0.4146	0	0	0	0	0	0
6	4.4102	2.8783	1.5303	0.8242	0.3323	0	0	0	0	0
7	5.4009	3.2964	2.4007	1.2546	0.6544	0.2624	0	0	0	0
8	6.385	4.2925	3.3375	2.0682	0.9771	0.5216	0.2056	0	0	0
9	7.4147	5.2842	4.2498	2.8728	1.727	0.7743	0.4052	0.1688	0	0
10	8.3841	6.2662	4.6554	3.6867	2.4393	1.4451	0.6363	0.3236	0.1334	0
11	9.3958	7.2503	5.1462	4.4978	3.1653	2.0832	1.2433	0.4938	0.2654	0.104
12	10.384	8.2628	5.6412	5.3329	3.9222	2.7741	1.7757	0.9779	0.3978	0.2114
13	11.402	9.2543	6.6572	5.907	4.6885	3.4174	2.3616	1.519	0.8415	0.3273
14	12.392	10.267	7.6374	6.4342	5.6035	4.091	2.9768	2.0154	1.2544	0.7169
15	13.408	11.266	8.6178	6.9909	6.517	4.943	3.5021	2.4982	1.6992	1.0675
16	14.39	12.252	9.6342	7.4991	7.1632	5.8672	4.3381	3.0246	2.1616	1.4079

Table 4.2: Average Inventory Remaining with Perfect Information (Case 1) using MMR Algorithm, $SSPK = 0.8$

algorithms including the enumeration. witness. and incremental pruning algorithms. The software requires an input file specifying the number of states, actions, and observations. as well as the complete transition model. observation model, and reward model. We wrote and used an input file writer to create such an input file. The input file writer begins with the basic settings: β_0, ρ_0. $SSPK$. P_{hh}. P_{mm}, and w_I. It then calculates the transition probabilities. observation probabilities. and reward matrix using the equations described in Chapter 3 and then writes them to a file.

4.2.1 POMDP Solution Algorithms

Ever since Sondik's formalization of the POMDP and his "One-Pass Algorithm," [16], solution algorithms for POMDPs have been proposed and researched. Because the ballistic missile defense single engagement problem is a "shoot-look-shoot" problem with two possible actions. it has a horizon of only two. Therefore, only finite-horizon algorithms are discussed in this section. All finite-horizon algorithms follow the same general structure as shown in Figure 4-1. First, the 0-epoch value function, $V_0(b)$, is constructed using the terminal values.

Initialize final stage with
0-epoch alpha vector set
to terminal values

Find next epoch
value function

No

Current epoch = Horizon ?

Yes

Return optimal policy
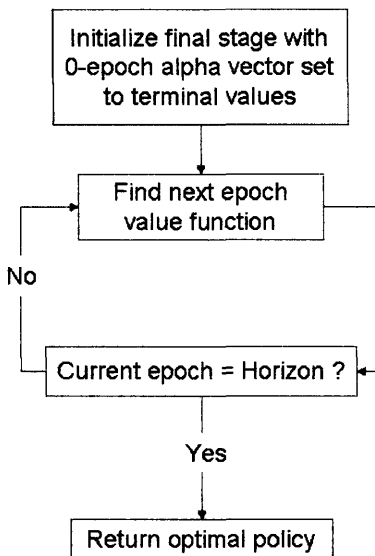
Figure 4-1: Finite-horizon POMDP Algorithm Structure

Terminal values place a reward or cost on each state for the final stage of the system. Next, the value function for the next epoch is computed. This dynamic programming update of the value function for each belief stage works backwards iteratively from the final stage in a recursive manner until the epoch equals the horizon of the problem. This process defines a new value function. $V'(b)$, from the current value function. $V(b)$ as shown in Equation 4.1 [7].

$$V'(b) = \max_{a \in \mathcal{A}} \left[\sum_{s \in \mathcal{S}} R(s, a)b(s) + \sum_{o \in \mathcal{O}} \Pr(o|a, b)V(b_o^a) \right] \quad (4.1)$$

This equation states that the value function for a belief state, b. is the value of the best action possible from b of the expected immediate reward for that action plus the expected value of the resulting belief state. b. This dynamic programming update is conducted until the horizon is reached. At that point. an optimal policy is produced [18]. This policy specifies the best action to take at that stage given the observation.

The main distinction between POMDP solution algorithms is the way they generate a finite set of points to build the alpha vectors for the value function. The process of finding dominant alpha vectors requires the use of linear programming. It should be noted that in some problems it is difficult to find regions where one alpha vector dominates others.

49

This may cause numerical instability in the linear programming problems. The work in this thesis investigated three primary algorithms to solve the POMDP: Monahan's enumerative algorithm, Littman's witness algorithm, and Zhang and Liu's incremental pruning algorithm.

Enumeration Algorithm: This type of algorithm, which was mentioned by Sondik in 1971, but formalized by Monahan in 1982, does not actually try to find a finite set of points to build the alpha vectors [6]. Instead it simply enumerates all alpha vectors [12]. From this superset of vectors, extraneous vectors are deleted if they are dominated by others. Ultimately the algorithm generates a set of dominant alpha vectors of minimal size. The problem with this algorithm is that the number of alpha vectors becomes very large as the horizon or number of epochs in the problem increase [6]. Even using the simple example in Figure 2-2 with two actions and two observations, the number of alpha vectors can become very large. This problem starts with only one alpha vector at the 0-epoch, which is the terminal value function. At each epoch the number of alpha vectors grows exponentially, so the total number of alpha vectors is doubly exponential in the horizon. It is clear that more complex problems with more possible actions and observations would require the generation of an excessively large number of alpha vectors. For this reason, enumerative algorithms are best suited to problems with small numbers of actions, observations, and a short horizon.

Witness Algorithm: This algorithm, developed by Littman, Cassandra, and Kaelbling, differs in the way it finds a set of alpha vectors of minimal size [11]. Instead of enumerating all possible alpha vectors and paring that set down, it builds up to that set one vector at a time. The witness algorithm defines regions for an alpha vector and looks for places where the vector is not dominant [11]. It starts with an arbitrary belief state, and finds the dominant alpha vector for this belief state. While it is known that the alpha vector is optimal for this point, it is not known where this vector is not dominant. The algorithm then defines a region of the belief space for this alpha vector and then searches for a point where it is not dominant. Unlike other algorithms, the witness algorithm defines a value function in this manner for each action separately. Then, it combines the value functions in the end to create the final value function. In addition to maximizing over actions in isolation,

50

the witness algorithm deals with one observation at a time. In choosing a vector for each observation. it chooses an action. The algorithm then searches one observation at a time for a choice that improves the overall value. If it finds an action and corresponding vector that improves the value function. then that serves as witness that the current value function is not the final value function [6].

Incremental Pruning Algorithm: This algorithm. originally proposed by Zhang and Liu [19] but developed by Cassandra. Littman. and Zhang [7]. is the latest and fastest algorithm for solving POMDPs. It combines elements of Monahan's enumeration algorithm and the witness algorithm [6]. Instead of finding the regions where alpha vectors dominate. this algorithm focuses on finding different combinations of future strategies. It begins by generating alpha vectors for a fixed action and observation. These vectors are compared and dominated vectors are removed. creating a dominant set of alpha vectors for only this action and observation. From there. the sets are combined for all the observations and dominated vectors are removed. creating a dominant set of alpha vectors for each action. Finally. the sets for each action are combined and dominated vectors are removed. creating the value function. $V(b)$.

4.3 Solution Process

The assumptions of this thesis make the single engagement a stochastic process based on several probabilities: $SSPK$, P_{hh}, and P_{mm}. In order to determine how the policies generated by the MMR algorithm and the POMDP solver perform. we used a Monte Carlo simulation of the single engagement, to calculate estimates for P_{nl} and β_2. This simulation can be run using either the MMR algorithm for Case 2 or the POMDP solver solution for Case 3. The Monte Carlo simulation for Case 1 is much simpler. as $P_{hh} = 1$ and $P_{mm} = 1$. Therefore, the only uncertainty comes from $SSPK$.

Figure 4-2 depicts the solution process for Case 2. In this case. the input file writer begins with the basic settings: interceptors. targets, $SSPK$. P_{hh}, P_{mm}, and w_I. It then calculates the transition probabilities and the observation probabilities. All of the initial settings. the

```
                                    Interceptors
                                    Targets
                                    SSPk
                                    Phh
                                    Pmm
                                    Weights

              ┌──────────────────────────────┐
              │       Input File Writer        │
              └──────────────────────────────┘

                                    Interceptors
                                    Targets
                                    SSPk
                                    Phh
                                    Pmm
                                    Weights
                                    Number of states
                                    Transition Model
                                    Observation Model

                Observation

   ┌──────────────────┐        ┌──────────────────┐
   │  MMR Algorithm    │◄═══════│    Simulation     │
   └──────────────────┘        └──────────────────┘
                Optimal Action          Optimal Policy
                                        Probability of No Leakage
                                        Average Inventory Remaining
```
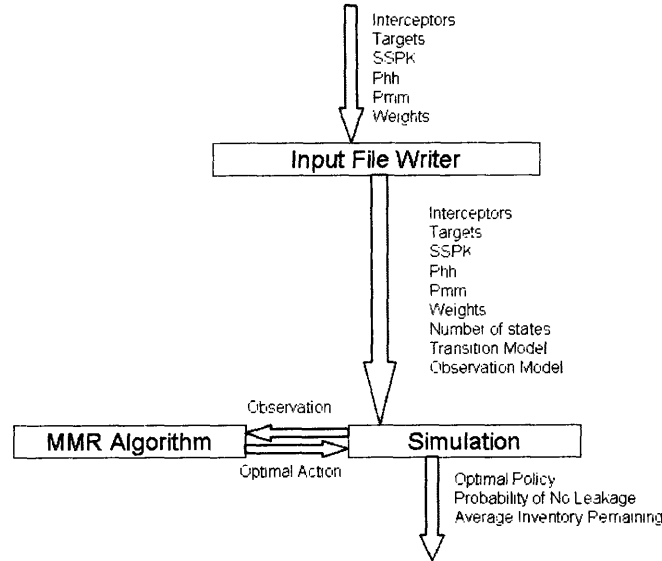
Figure 4-2: Solution Process Using MMR Algorithm

transition model, and the observation model are then used by the simulation. To determine the actions, the simulation calls the MMR algorithm, which takes the observation as the true number of targets remaining. The MMR algorithm then provides an optimal policy back to the simulation. The single engagement is simulated many times for the same settings in order to calculate average measures of performance.

When using the simulation with the POMDP solver of Case 3, the program flow is as depicted in Figure 4-3. As with Case 2, the input file writer begins with the basic settings and produces an input file for the simulation. In addition, it also produces two input files for the POMDP solver: one containing the transition model, observation model. and reward function and one containing the terminal rewards for each state. With these input files, the POMDP solver uses the selected POMDP solution algorithm, such as incremental pruning, and produces a solution file containing alpha vectors and their associated actions. After translating this file into a matrix for the alpha vectors and a vector for the actions corresponding to each alpha vector, the simulation uses them along with the initial settings from the input file writer. Again the simulation is run many times to estimate average probability of no leakage and average inventory remaining. The entire process depicted in
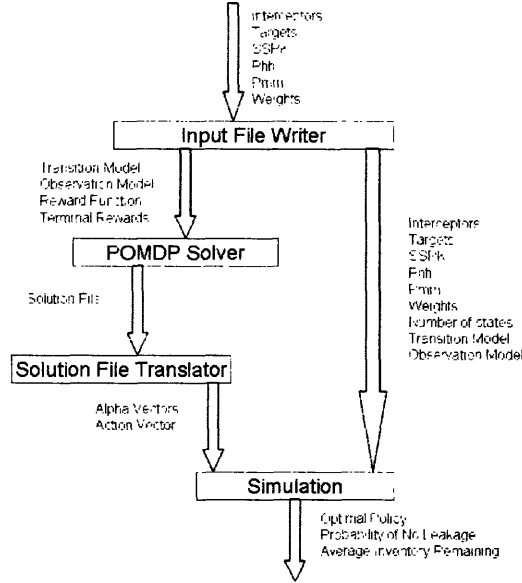
Figure 4-3: Solution Process Using POMDP Solver

Figure 4-2 for Case 2 and Figure 4-3 for Case 3 combine to form one run of each experiment to be described in detail in the following section.

The simulation was developed to run a simulated single engagement a large number of times to gain an accurate assessment of the strategy and settings chosen based on several response variables. It begins by using either the MMR algorithm or the POMDP solution policy to determine an initial action. If the simulation is using the POMDP strategy, the belief state is multiplied with each alpha vector to produce a value. The alpha vector resulting in the highest value corresponds to the best action to take. If the MMR strategy is used, the simulation simply invokes the MMR algorithm to determine the best action given the situation. The algorithm plans the assignment for two stages, and the simulation uses the first stage assignment as the first action. The simulation then determines how many targets are in g_1 and g_2 and how many interceptors are fired at each target n and $n-1$, respectively. With those values it calculates PK_1 and PK_2. It then generates a random number between zero and one for each target and compares it to PK_1 or PK_2. If the random number is less than PK_1 or PK_2, the target is hit. Then, a new random number is generated for each target. This number is compared to P_{hh} if the target was hit and P_{mm} if the target

53

was missed. If the random number is less than P_{hh} or P_{mm}, the observation is correct. Next, a second action is determined from either the POMDP solution policy or the MMR algorithm based on the observed number of targets remaining. For the POMDP case, this is accomplished by updating the belief state, and multiplying it with each alpha vector to find the highest value corresponding to the best action. For the MMR strategy, the algorithm replans its assignment for stage two based on the new observations, providing the second action. The process of generating a random number to compare to PK_1 and PK_2 is then repeated to determine the final state of the system. Running this simulation over many trials produces an average estimate of the response variables: inventory remaining, targets leaked, and probability of no leakage. A description of this simulation is depicted in Algorithm 4.2.

4.4 Experimental Design

The experimental design has three separate sets of experiments. First, three initial experiments were run varying factors that are later held constant. In this set of experiments we screen these variables to determine their effect on system performance, as well as establish an optimum set of values for the factors. Next, a central composite design of 87 runs was conducted varying five different factors at two levels each, in order to see how these factors and their interactions affected the results. Lastly, one-factor experiments were conducted on those five factors to determine how they affected the results when varied over a wide range of values. Whereas, in the second set of designed experiments, we used a CCD to determine if each factor had significant influence on the response variables, and if there were interactions between factors, this last set of experiments provides different information by showing the effect of the input variables over the full spectrum of their possible values.

Algorithm 4.2 Single Engagement Simulation

$I \Leftarrow 0.$ $L \Leftarrow 0$
for $i = 1$ to t **do**
  $h_a \Leftarrow 0.$ $m_a \Leftarrow 0.$ $h_o \Leftarrow 0.$ $m_o \Leftarrow 0$
  if Case 2 **then**
    Determine a from MMR Algorithm
  else {Case 3}
    Determine a from POMDP alpha vectors
  $\beta \Leftarrow \beta - a$
  Calculate $g_1.$ $g_2.$ $n.$ $PK_1.$ PK_2
  for all g_1 targets **do**
    Generate random number $0 \leq PK_{sim} \leq 1$
    if $PK_{sim} \leq PK_1$ **then**
      $h_a \Leftarrow h_a + 1$
    else
      $m_a \Leftarrow m_a + 1$
  for all g_2 targets **do**
    Generate random number $0 \leq PK_{sim} \leq 1$
    if $PK_{sim} \leq PK_2$ **then**
      $h_a \Leftarrow h_a + 1$
    else
      $m_a \Leftarrow m_a + 1$
  for all ρ targets **do**
    Generate random number $0 \leq P_{obs} \leq 1$
    if Target hit **then**
      if $P_{obs} \leq P_{hh}$ **then**
        $h_o \Leftarrow h_o + 1$
      else
        $m_o \Leftarrow m_o + 1$
    else {Target miss}
      if $P_{obs} \leq P_{mm}$ **then**
        $m_o \Leftarrow m_o + 1$
      else
        $h_o \Leftarrow h_o + 1$
  Repeat once
  if $m_a \geq 0$ **then**
    $L \Leftarrow L + 1$
  $I \Leftarrow I + \beta$
$P_{nl} \Leftarrow \frac{L}{t}$
$\beta_2 \Leftarrow \frac{I}{t}$

4.4.1 Initial Experiments

The initial experiments started with a baseline scenario that is a combination of input factors chosen as a likely real-world scenario. We then ran three different experiments varying the algorithm, terminal rewards, and the single-shot probability of kill (SSPK), leaving all other factors at the baseline level. The purpose of these experiments is to get a general idea of how these factors affected the POMDP solver software, before leaving them constant in the main experiments. The baseline scenario was first run for 10,000 trials of the simulation with the settings shown in Table 4.3.

Factor	Value
β_0	10
ρ_0	3
$SSPK$	0.8
P_{hh}	0.8
P_{mm}	0.8
w_I	0.7
w_{T_1}	1
w_{T_2}	100

Table 4.3: Baseline Scenario

Each of the three initial experiments began with the baseline scenario and varied one of the factors.

Experiment 1: This experiment varied the algorithm used in the POMDP solver software. The three algorithms examined are enumeration, witness, and incremental pruning. While Cassandra, Littman, and Zhang assert that the incremental pruning algorithm is the fastest algorithm to date, we conduct this experiment to test the algorithms on our problem. The response variables are the following: solving time, instability, policy solution, and number of alpha vectors. Each run of this experiment consisted of 10,000 simulation trials.

Experiment 2: This experiment varied the terminal reward function for the POMDP solver software. From Equation 3.7, w_{T_1} and w_{T_2} were varied. This experiment set w_{T_2} to

different orders of magnitude, and one experimental run also set both w_{T_1} and w_{T_2} to zero. One run also turned off the terminal reward setting for the solver. Each run is compared on the following response variables: solving time, instability, policy solution, and number of alpha vectors. Due to increased execution time of the simulation, each run in this experiment consisted of 1,000 simulation trials.

Experiment 3: This experiment varied $SSPK$ at different levels between 0.5 and 1 to examine its effect on the two cases assuming imperfect information. Due to the considerable time required to generate Table 4.1 and Table 4.2 for the perfect information case, further experiments were run using only one value: $SSPK = 0.8$. Runs were compared on the following response variables: policy solution, targets leaked, remaining inventory, and probability of no leakage. Each run in this experiment also consisted of 1,000 simulation trials.

Table 4.4 summarizes these three experiments.

Experiment	Varied Factor	Levels	Response Variables
1	Algorithms	Enumeration Witness Incremental Pruning	Solving Time Instability Policy Solution Alpha Vectors
2	Terminal Rewards, $F(s)$	None $w_{T_1} = 0,\ w_{T_2} = 0$ $w_{T_1} = 1,\ w_{T_2} = 1$ $w_{T_1} = 1,\ w_{T_2} = 10$ $w_{T_1} = 1,\ w_{T_2} = 100$ $w_{T_1} = 1,\ w_{T_2} = 1000$ $w_{T_1} = 1,\ w_{T_2} = 10000$	Solving Time Instability Policy Solution Alpha Vectors
3	$SSPK$	$0.5 \leq SSPK \leq 1$	Policy Solution Targets Leaked Remaining Inventory Prob of No Leakage

Table 4.4: Initial Experiments

4.4.2 Central Composite Design Experiment

The purpose of Experiment 4 is to understand how five different factors affect the outcome using the MMR algorithm and the POMDP solver. The five factors varied in this experiment were the number of interceptors (β_0), the number of targets (ρ_0), the observation probabilities $(P_{hh}$ and $P_{mm})$, and the intermediate weight (w_I). In order to truly know the effects of the five factors, including quadratic effects and interactions between factors, a central composite design (CCD) was used. The importance of this design is two-fold. First, it allows us to determine interaction effects of different factors. While "one factor at a time" experiments may show that the response increases as a factor increases, it may be true that the response actually decreases when that factor increases and another factor decreases. This implies that there is a significant effect on the response by an interaction between the two factors. If we only examine the effect of a factor as all other factors are held constant we really do not know how the response performs in other regions of the factor space. As a result, our conclusions are very dependent on the initial conditions and we may be led to a false conclusion. Secondly, the CCD allows the fitting of a second-order model [13]. This would imply that the effect of some factors on the response is not linear. Both of these occurrences seem likely with respect to our problem. First, it seems likely that factors such as the number of interceptors and targets would have significant interaction effects. Secondly, it seems likely that the effect of some factors on the response is nonlinear given that one response term is a probability.

The CCD begins with a 2^5 factorial design, which sets the five factors at a high and low level, creating 32 runs for all combinations of each of these levels. Then to check for curvature, axial runs and center points are added to the design. A center point simply sets all the factors to a level halfway between the high and low levels. Axial runs set all factors to the center level and one factor to a certain distance from the design center, α [13]. With five factors, this experiment had 10 axial points. A graphical depiction of a two-factor central composite design is shown in Figure 4-4. In order for the model to provide good predictions throughout the region of interest, the design must have rotatability, which means that the
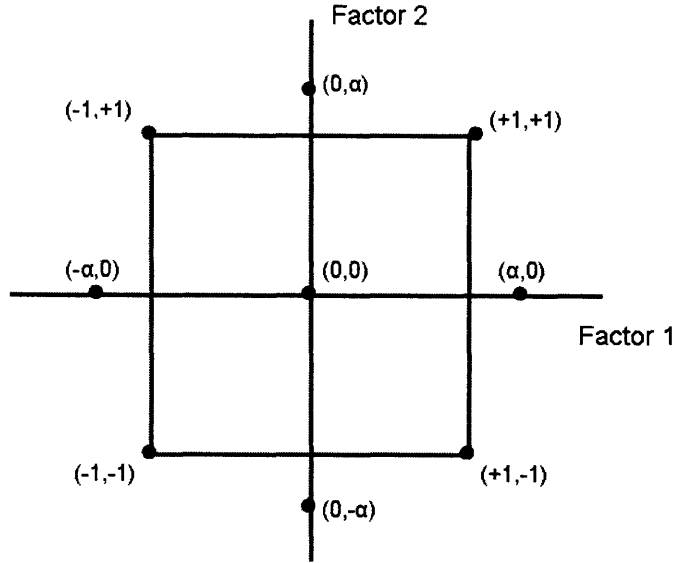
Figure 4-4: Two-factor Central Composite Design

variance of the predicted response should have equal variance at all design points that are equal distance from the design center [13]. This is attained by choosing the proper α. In general. setting $\alpha = n_f^{1/4}$ where n_f is the number of factorial runs leads to a rotatable design [13]. In this experiment, $n_f = 32$ so $\alpha = 32^{1/4} \approx 2.378$. Multiplying α by the distance from the factorial points to the design center provides the distance from the axial runs to the design center for each factor. This distance was rounded to the nearest integer for the factors β_0 and ρ_0.

The factorial runs. n_f, and the axial runs. n_a. were replicated twice. while the center point. n_c was replicated three times for a total of 87 runs. According to Montgomery, three to five center points provide reasonable stable variance of the response [13]. In experimental design, replication is used to obtain an estimate of experimental error and to obtain more precise estimates of the effects of the factors [13]. In this experiment, the same settings for the POMDP solver produce the same policy, and the output of the POMDP solver provides the input for the simulation. Therefore, the only variation in results comes from the stochastic simulation. If enough trials are used in the simulation, there should be very little difference in the response variables between replicates of the same factor settings.

Each run in this experiment consisted of 1,000 simulation trials. Table 4.5 shows the design matrix for this experiment without any of the replicates. Because the POMDP solver, MMR algorithm, and simulation have no memory. each run is completely independent. Because of this independence, randomization of trials is not necessary in this experiment. and runs were conducted in the order depicted in Table 4.5.

4.4.3 Single-Factor Experiments

Finally. one-factor experiments were conducted on each of the five factors varied in the CCD. In each of these experiments, all settings were set to the baseline level, except for the factor of interest. From there. that factor was varied over a wide range of relevant values. The goal of these experiments is to compare the performance of the three cases as each of the five factors changed over a wide range of values. They provide a more detailed depiction of what happens to the response variables as one factor changes. The important response variables examined in all of these experiments were: policy solution, targets leaked, inventory remaining, probability of no leakage, and a linear combination of inventory remaining and probability of no leakage based on the weight, w_I. Except for Experiment 5, each run in all experiments consisted of 1,000 simulation trials.

Experiment 5: This experiment varied the intermediate rewards weight. w_I, used by the POMDP solver software. We set w_I to values between zero and one at intervals of 0.1. Because the MMR algorithm does not depend on w_I to make decisions, Case 2 only required one experimental run. This decreased simulation time greatly, and each run consisted of 10,000 simulation trials.

Experiment 6: This experiment varied P_{hh} and P_{mm} simultaneously. While this is not truly a single-factor experiment, it was found that varying P_{hh} and P_{mm} separately produced the same results as varying them simultaneously. This experiment set P_{hh} and P_{mm} to values between 0.5 and 1. While it is possible to examine values between zero and one, the most relevant values were those in which $P_{hh} = P_{mm} \geq 0.5$. If sensors

Run	Type	β_0	ρ_0	P_{hh}	P_{mm}	w_I
1	n_f	7	3	0.6	0.6	0.3
2	n_f	11	3	0.6	0.6	0.3
3	n_f	7	5	0.6	0.6	0.3
4	n_f	11	5	0.6	0.6	0.3
5	n_f	7	3	0.8	0.6	0.3
6	n_f	11	3	0.8	0.6	0.3
7	n_f	7	5	0.8	0.6	0.3
8	n_f	11	5	0.8	0.6	0.3
9	n_f	7	3	0.6	0.8	0.3
10	n_f	11	3	0.6	0.8	0.3
11	n_f	7	5	0.6	0.8	0.3
12	n_f	11	5	0.6	0.8	0.3
13	n_f	7	3	0.8	0.8	0.3
14	n_f	11	3	0.8	0.8	0.3
15	n_f	7	5	0.8	0.8	0.3
16	n_f	11	5	0.8	0.8	0.3
17	n_f	7	3	0.6	0.6	0.7
18	n_f	11	3	0.6	0.6	0.7
19	n_f	7	5	0.6	0.6	0.7
20	n_f	11	5	0.6	0.6	0.7
21	n_f	7	3	0.8	0.6	0.7
22	n_f	11	3	0.8	0.6	0.7
23	n_f	7	5	0.8	0.6	0.7
24	n_f	11	5	0.8	0.6	0.7
25	n_f	7	3	0.6	0.8	0.7
26	n_f	11	3	0.6	0.8	0.7
27	n_f	7	5	0.6	0.8	0.7
28	n_f	11	5	0.6	0.8	0.7
29	n_f	7	3	0.8	0.8	0.7
30	n_f	11	3	0.8	0.8	0.7
31	n_f	7	5	0.8	0.8	0.7
32	n_f	11	5	0.8	0.8	0.7
33	n_a	5	4	0.7	0.7	0.5
34	n_a	13	4	0.7	0.7	0.5
35	n_a	9	2	0.7	0.7	0.5
36	n_a	9	6	0.7	0.7	0.5
37	n_a	9	4	0.46	0.7	0.5
38	n_a	9	4	0.94	0.7	0.5
39	n_a	9	4	0.7	0.46	0.5
40	n_a	9	4	0.7	0.94	0.5
41	n_a	9	4	0.7	0.7	0.02
42	n_a	9	4	0.7	0.7	0.98
43	n_c	9	4	0.7	0.7	0.5

Table 4.5: Central Composite Design Matrix

were so unreliable that they gave the wrong information most of the time, this entire exercise as well as the actual system would be ineffectual.

Experiment 7: This experiment varied the number of initial interceptors in the scenario between 3 and 16. The experiment did not include runs with the initial inventory less than 3, because the most relevant runs involved more interceptors than targets. With fewer or equal interceptors than targets, the best action is to assign all of the interceptors in inventory.

Experiment 8: The number of initial targets in this experiment was varied between 1 and 10. Again, as the number of targets approaches the number of interceptors, the resulting policy solution is less interesting, as all of the inventory will be assigned.

Table 4.6 summarizes these three experiments.

Experiment	Varied Factor	Levels	Response Variables
5	Intermediate Weight, w_I	$0 \leq w_I \leq 1$	Policy Solution Targets Leaked Remaining Inventory Prob of No Leakage
6	P_{mm} and P_{hh}	$0.5 \leq P_{mm} \leq 1$ $0.5 \leq P_{hh} \leq 1$ $P_{mm} = P_{hh}$	Policy Solution Targets Leaked Remaining Inventory Prob of No Leakage
7	Interceptors, β_0	3 to 16	Policy Solution Targets Leaked Remaining Inventory Prob of No Leakage
8	Targets, ρ_0	1 to 10	Policy Solution Targets Leaked Remaining Inventory Prob of No Leakage

Table 4.6: Single-Factor Experiments

62

4.5 Chapter Summary

This chapter describes the implementation of the problem formulation from Chapter 3. It begins with a discussion of how the MMR algorithm is used to provide a policy solution for the first two cases and how a POMDP solver is used to provide the policy solution for Case 3. It discusses the various POMDP solution algorithms and how they are differentiated by the method used to create a value function over the belief states. We then describe how the performance of the cases is estimated with a simulation of the single engagement using the policy solution created from the MMR algorithm or the POMDP solver. The chapter finishes with a discussion of the experimental design beginning with initial experiments. continuing with a central composite design, and ending with a series of single-factor experiments.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Results and Analysis

This chapter assesses the potential impact of imperfect information on the performance of interceptor assignment. and the possibility of accounting for this uncertainty with a POMDP approach. In order to do this we carry out a series of experiments that compare the decisions and performance of the three cases described in Chapter 3: perfect information, imperfect information assumed perfect. and imperfect information known to be imperfect. This chapter discusses the results of the experiments outlined in Chapter 4.

We begin with the results from the baseline scenario. This serves as a basis for comparison for all other results. We then compare the three POMDP solution algorithms in Experiment 1. Next, we examine the performance of the POMDP solver with various terminal reward functions in Experiment 2. With the last initial experiment, we assess the performance of each case with varying SSPKs.

Experiment 4 provides us with the data necessary to develop three statistical models. We conduct an analysis of variance (ANOVA) on each of these three quadratic models and then check for model adequacy. The response variables in each model are a difference in performance between Cases 2 and 3 using three different measures of performance. Each model includes five factors.

Our final four experiments assess the factors used in Experiment 4, by varying them individually. We assess the impact of w_I from the results of Experiment 5. In Experiment

6, we vary P_{hh} and P_{mm} simultaneously, and determine how they affect the performance of Cases 2 and 3. Finally, in Experiments 7 and 8, we assess the impact of the number of initial interceptors and initial targets respectively. We end this chapter with overall conclusions based on these experiments.

5.1 Initial Results

5.1.1 Baseline

As stated in Chapter 4, experimentation began with a baseline scenario. Chosen for its realistic settings, this baseline is the starting point for all following experiments. The results for the baseline scenario are shown in Table 5.1.

Response	Case 1	Case 2	Case 3
Execution Time	NA	NA	35.40 sec
Instability	NA	NA	243,388
Alpha Vectors	NA	NA	67
Policy Solution	4:0,3,6,6	4:0,3,6,6	3:1,2,2,3
Prob of No Leak	0.9963	0.9494	0.9367
Remaining Inventory	4.6554	3.468	5.2829
Leaked Targets	0.0037	0.0524	0.0697
Weighted Combination	0.83707	0.76862	0.814177

Table 5.1: Results from Baseline Scenario

In this table, "Case 1" corresponds to perfect information, "Case 2" corresponds to perfect information assumed perfect, and "Case 3" corresponds to imperfect information that is known to be imperfect. This terminology is used throughout the chapter. In Table 5.1, the first three results only apply to the POMDP solver (Case 3). "Execution Time" shows the time required for the POMDP solver to execute and solve the problem. This correlates to the size and complexity of the problem, as well as the speed of the algorithm used to solve it. "Instability" is the number of linear programming subproblems that had numerical instability during the execution of the POMDP solver. "Alpha Vectors" refers to the number of alpha vectors in the solution provided by the POMDP solver, and is highly correlated with

the instability. As the instability increases in a problem. it is more difficult to find which alpha vectors dominate over the belief space. Consequently, the number of alpha vectors in the solution increases. Much like instability, this also correlates to the complexity and size of the problem. These three results only apply to Case 3 and give a general baseline for the performance of the POMDP solver.

The next five results in Table 5.1 are used as a baseline for comparison between each of the three cases. The "Policy Solution" is depicted in the form

$$(a_1 : a_2^{o_1}, a_2^{o_2}, a_2^{o_3}, a_2^{o_4})$$

where a_1 is the first action and a_2^o is the second action based on an observation. Because this problem begins with three targets. there are only four possible observations. "Probability of no leak." P_{nl}. "Remaining Inventory." β_2. and "Leaked Targets" are direct ways to compare the performance of each case in this baseline scenario. Lastly. "Weighted Combination." W. is a method to assess each case based on the weight, w_I in the reward function. This value provides an overall metric of performance combining P_{nl} and β_2. W is calculated by Equation 5.1.

$$W = w_I P_{nl} + (1 - w_I) \left(\frac{\beta_2}{\beta_0} \right) \tag{5.1}$$

As can be seen in Table 5.1, Case 3 has a more conservative policy solution than the two other cases. and consequently has a greater remaining inventory. In spite of this conservatism. Case 3 almost matches the probability of no leakage of Case 2: 93.67% compared to 94.49%. Case 1 proves to have the highest probability of no leakage. allowing only 0.0037 targets leak through defenses on average. In comparison of W. Case 1 does the best. followed by Case 3. and Case 2. It is important to note that if the weight, w_I. is truly the importance of probability of no leakage compared to inventory remaining, then W is probably the best metric when comparing the three cases.

5.1.2 Experiment 1

Experiment 1 is simply a comparison of three algorithms used to solve the POMDP. The results of this experiment are shown in Table 5.2. The first important result from this exper-

Algorithm	Enum	Witness	Incprune
Execution Time	30 min 24.91 sec	23.81 sec	35.40 sec
Instability	892.667	256,785	243,388
Alpha Vectors	531	42	67
Policy Solution	3:1,2,2,3	3:1,2,2,3	3:1,2,2,3

Table 5.2: Results from Experiment 1

iment is that all three algorithms produce the same policy solution. So, aside from the fact that some algorithms may take longer than others, all three could be used interchangeably in further experiments. However, Table 5.2 clearly shows that the algorithm does matter when it comes to execution time. The enumeration algorithm takes over 30 minutes to run for this one scenario, while both the witness and incremental pruning algorithms require only around 30 seconds to run. In addition, the enumeration algorithm has far more alpha vectors and linear programming subproblems with instability than the other two algorithms. In this experiment the witness and incremental pruning algorithms are very similar in execution time and instability. Ultimately the incremental pruning algorithm was chosen for the remaining experiments not only because this experiment proved it to be fast and efficient, but also due to previous research by Littman, Cassandra, and Zhang that showed it to be the simplest and fastest algorithm to date [7].

5.1.3 Experiment 2

In the next experiment we examine various terminal reward functions, $F(s)$ and their effect on the POMDP solver. The results of this experiment are shown in Table 5.3 where the terminal reward function, $F(s)$, is described by (w_{T1}, w_{T2}). While the results of this experiment are not completely conclusive, they do provide some useful insights. First, it is clear that as w_{T2} increases by orders of magnitude, the execution time, instability, and number of

F(s)	None	(0,0)	(1,1)	(1,10)	(1,100)	(1,1000)	(1,10000)
Execution Time	0.24 sec	0.49 sec	1.76 sec	5.28 sec	35.39 sec	55.15 sec	49.72 sec
Instability	531	531	10.748	50.940	243.388	335.882	305.388
Alpha Vectors	24	24	46	44	67	74	45
Policy Solution	1:NA.NA.1.6	1:NA.NA.1.7	2:NA.2.1.1	3:1.2.1.2	3:1.2.2.3	6:1.2.2.2	6:2.2.2.2

Table 5.3: Results from Experiment 2

alpha vectors generally increase. We note that higher instability may not indicate an inferior solution. but more likely a larger or more complex problem.

Most importantly. this experiment shows that the policy solution highly depends on the terminal rewards. In this baseline scenario. no terminal rewards or even terminal rewards with small weights on targets remaining, w_{T2}. provide somewhat strange policy solutions. in which the first action is very small. When the first action is less than the number of targets. it is impossible to have an observation of zero or more targets depending on the difference between action and targets. This is depicted in Table 5.3 where a second action is listed as "NA." This indicates that an action is not applicable to that situation, because $a_1 < \rho_0$. However, even with these strange cases, as w_{T2} increases. the policy solution uses more interceptors. This result is logical. as increasing w_{T2} places more value on stopping targets compared to conserving interceptors.

Overall. this experiment shows that a logical and balanced policy solution results from $w_{T2} \approx 100$. With this setting. all actions were at least as great as the number of targets thought to be remaining. and more interceptors were used as more targets were observed. Subsequent experiments were conducted with the settings $w_{T1} = 1$ and $w_{T2} = 100$. In a sense this says that at the end of the engagement, we are 100 times more concerned about stopping targets from leaking through defenses than saving our inventory. To change the policy solution slightly based on the preferences of an actual decision maker, we could increase or decrease w_{T2} from a value of 100.

5.1.4 Experiment 3

Experiment 3 examines the effects of various single-shot probabilities of kill (SSPK) on the performance of each of the three cases. Table 5.4 shows the policy solutions for each of these cases. This table only lists values of $0.5 \leq SSPK \leq 0.98$. because those are the most

$SSPK$	Cases 1 and 2	Case 3
0.5	4:0,6,6,6	6:2.2.2.2
0.6	4:0,6,6,6	6:1,2.2,2
0.7	4:0,4,6,6	6:1.2.2.2
0.8	4:0,3,6,6	3:1,2.2.3
0.9	3:0,3,6,7	3:1.1.1,2
0.98	3:0,2.4,6	2:NA.1,1.1

Table 5.4: Policy Solutions for Experiment 3

relevant values. As previously mentioned, it makes little sense to use interceptors that have a higher probability of missing a target than hitting one. In addition, an interceptor with $SSPK = 1.0$, although operationally outstanding, provides little interesting insight into our work. In that scenario. imperfect kill assessment matters little when every target can be hit with certainty on the first shot.

In Table 5.4, we first note that Case 1 and Case 2 always have the same policy solution, as they both use the MMR algorithm to determine how many interceptors to assign to targets. The difference in the two cases is that in Case 1, an observation is always true, and in Case 2 it may not be true. This difference is not indicated in the policy solutions.

Table 5.4 also shows that in all of the cases, as $SSPK$ increases the policy solutions become more conservative with interceptor inventory. This result occurs because as kill probability of a single interceptor versus a single target increases. fewer interceptors should be required. The policy solutions in Cases 1 and 2 gradually become more conservative as $SSPK$ increases, while the policy solutions in Case 3 have a major decrease in number of interceptors assigned in action 1 from $SSPK = 0.7$ to $SSPK = 0.8$. This occurs because the POMDP solver generally assigns enough interceptors in action 1 so that each target is assigned the same number, while the MMR algorithm generally does not. For Cases 1 and

2. most policy solutions assign four interceptors to three targets in action 1. while Case 3 typically assigns either six or three interceptors to three targets in action 1.

While this trend of becoming more conservative as $SSPK$ increases exists in all cases. Case 3 begins much more conservatively in the first shot than the other cases, using six as opposed to four interceptors with $SSPK = 0.5$. Likewise. with this $SSPK$. Cases 1 and 2 use six interceptors when at least one target is observed. while Case 3 uses only two interceptors regardless of the observation. Although it seems illogical not to use as many interceptors as targets observed in Case 3, the POMDP solver knows that $Pmh > 0$. and that missing all three targets is unlikely. Therefore. while not necessarily the safest course of action. it does make sense to use only two interceptors even when three targets were observed.

Another major difference between Cases 1 and 2 and Case 3 is the number of interceptors they assign with an observation of no targets remaining. While the MMR algorithm never assigns any interceptors with an observation of no targets remaining. the POMDP solver always assigns at least one interceptor, as it accounts for imperfect kill assessment.

Finally. it is important to note that as $SSPK$ approaches values very close to one. the cases vary greatly. Cases 1 and 2 still assign one interceptor for each target in the first shot. and two interceptors for each target in the second shot. Case 3. however. continues to become more conservative. With $SSPK = 0.98$ the POMDP solver initially uses only two interceptors, and then uses only one more interceptor regardless of the observation. In essence it always uses three interceptors for three targets. when $SSPK \approx 1$.

In addition to comparing the policy solutions of each case, it is important to examine the performance of each case. We begin by comparing the probabilities of no leakage for various levels of $SSPK$ in Figure 5-1. This chart includes more experimental runs than depicted in Table 5.4. as we varied $SSPK$ at increments of 0.02 between $0.7 \leq SSPK \leq 1.0$. In this chart. the probability of no leakage generally increases as $SSPK$ increases for all three cases. Case 1 provides an upper bound on the probability of no leakage for the other cases. For these other two cases, Case 2 outperforms Case 3 with $SSPK = 0.5$ and $SSPK = 0.6$. After
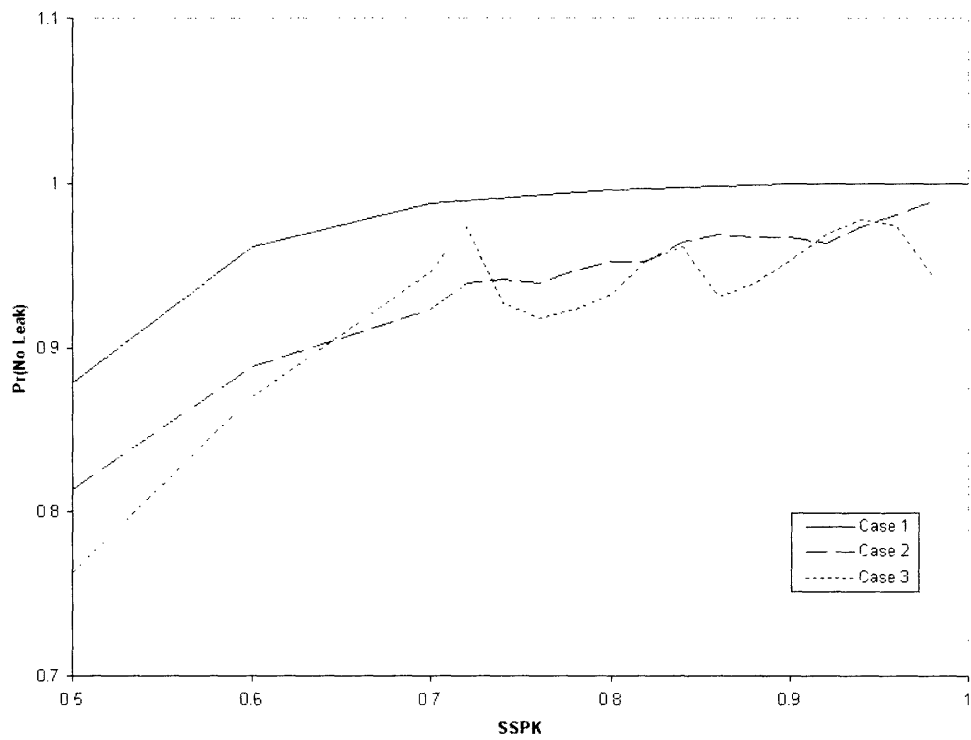
71

Figure 5-1: P_{nl} versus $SSPK$

that region. Case 3 generally matches the P_{nl} of Case 2. The P_{nl} in Case 2 increases much more gradually than that of Case 3, which can be attributed to the more gradual changes in policy solutions shown in Table 5.4. The chart shows that for Case 3. large decreases in the number of interceptors assigned in action 1 correspond to a decrease in P_{nl}.

In addition to comparing performance on probability of no leakage. we examine the inventory remaining for each of the three cases. Figure 5-2 depicts a chart of this metric as $SSPK$ is varied. This chart shows that for all cases. as $SSPK$ increases. the average
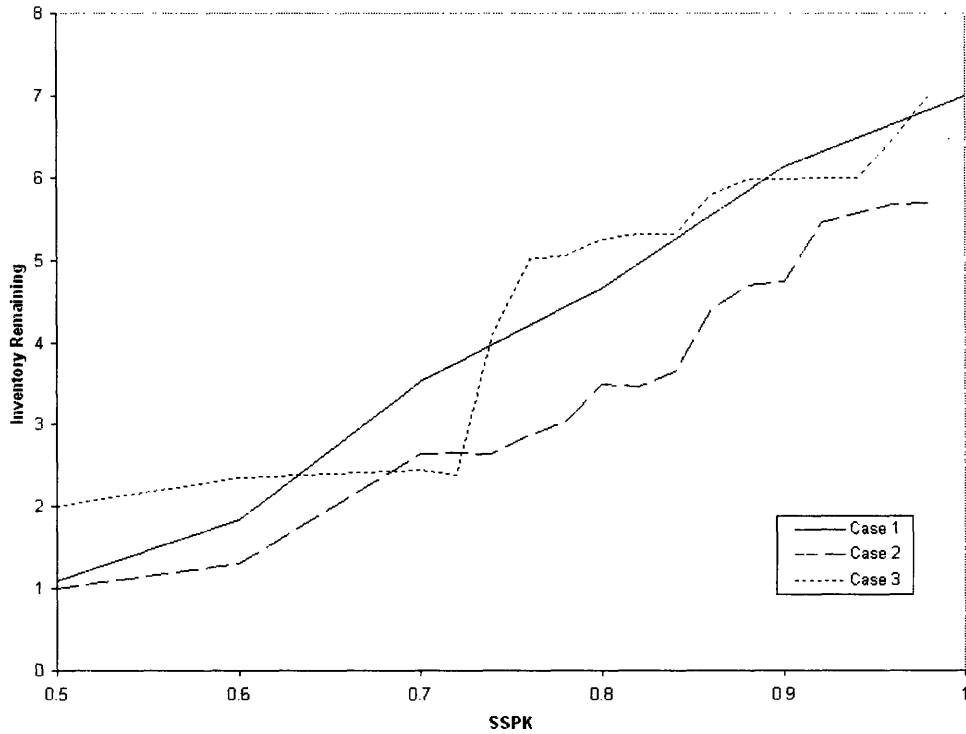

Figure 5-2: β_2 versus $SSPK$

inventory remaining also increases. This flows logically. as fewer interceptors should be used if each interceptor is more lethal. Except for $SSPK$ values around 0.7, the remaining inventory for Case 3 generally matches that of Case 1. with Case 2 typically the lowest of the three. Again. this makes sense, as the POMDP approach is the most conservative with inventory of the three cases. As with Figure 5-1, the inventory remaining for Case 3 increases less gradually due to its more drastic changes in policy solutions.

Lastly, we examine the effect of $SSPK$ on W. Figure 5-3 depicts a chart comparing each case. In a sense, this chart is a way to assess overall performance by combining the trends
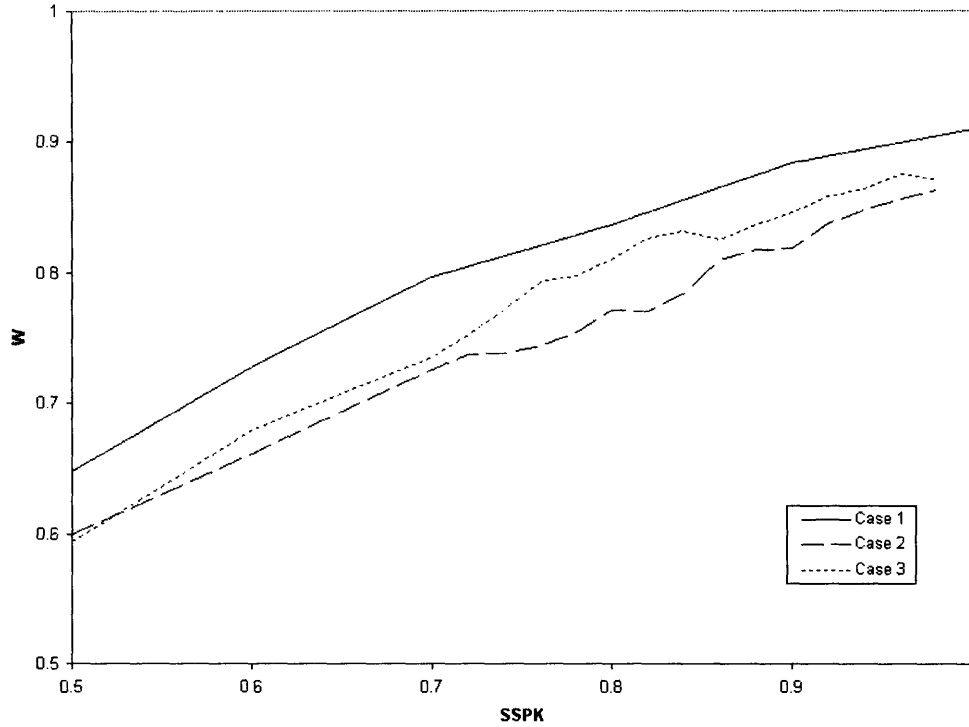

Figure 5-3: W versus $SSPK$

of the two previous charts using the weight, w_I. Figure 5-3 shows that as $SSPK$ increases, W also increases, as it is a linear combination of P_{nl} and inventory remaining, which were both shown to increase as well. Case 1 again proves to be an upper bound on the other two cases. In addition, Case 3 almost always outperforms Case 2. This indicates all other factors constant, as $SSPK$ is varied, accounting for imperfect kill assessment proves better than not accounting for it.

5.2 Central Composite Design Results

With a general idea of how the three cases perform from Experiments 1 through 3, we now conduct a full statistical analysis to better understand how Cases 2 and 3 compare. Our

74

focus in Experiment 4 is on the cases that have imperfect information. In order to estimate an appropriate statistical model we use the central composite design described in Chapter 4. which varies the factors β_0, ρ_0, P_{hh}, P_{mm}, and w_I simultaneously. In this section, we will refer to the effects of factors β_0, ρ_0, P_{hh}, P_{mm}, and w_I as A, B, C, D, and E respectively.

5.2.1 Model 1

Because we wish to compare the performance of Cases 2 and 3. we investigate three statistical models with different response variables: $\Delta_{P_{nl}}$, Δ_{β_2}, and Δ_W. These variables represent the difference in Case 2 and Case 3 probability of no leakage, remaining inventory, and W respectively. and are calculated by the following equations:

$$\Delta_{P_{nl}} = P_{nl}^3 - P_{nl}^2 \tag{5.2}$$

$$\Delta_{\beta_2} = \beta_2^3 - \beta_2^2 \tag{5.3}$$

$$\Delta_W = W^3 - W^2 \tag{5.4}$$

where the superscript indicates Case 2 or Case 3.

We begin with a quadratic model on $\Delta_{P_{nl}}$ that initially includes all five main effects, all two-way interactions, and all square terms. We pare down this model in a stepwise process to the significant factors at the $\alpha = 0.10$ significance level to produce the analysis of variance (ANOVA) results in Table 5.5. This table shows that the model is significant with a $p < 0.0001$. In addition, all main effects are significant. One quadratic effect, B^2. and three two-way interaction effects, AB, BD, CD, are also significant. The lack of fit significance indicates that this model may not fit, and that significant terms are omitted. However, this model has significant lack of fit regardless of the terms included. The model also has an $R^2 = 0.8415$ and $R^2_{Adj} = 0.8230$. This indicates that approximately 84% of the variability in the data is explained by this model [13]. R^2_{Adj} is an adjusted R^2 for the number of factors included in the model. R^2_{Adj} is useful, because in general, increasing the number of terms in a model alone increases R^2.

Source	Sum of Squares	DF	Mean Square	F Value	P-Value
Model	0.43	9	0.047	45.42	< 0.0001
A	0.029	1	0.029	27.62	< 0.0001
B	0.12	1	0.12	119.07	< 0.0001
C	0.020	1	0.020	18.81	< 0.0001
D	0.047	1	0.047	44.51	< 0.0001
E	4.577E-003	1	4.557E-003	4.38	0.0397
B^2	0.082	1	0.082	78.61	< 0.0001
AB	0.11	1	0.11	105.10	< 0.0001
BD	3.844E-003	1	3.844E-003	3.68	0.0589
CD	7.353E-003	1	7.35E-003	7.03	0.0097
Residual	0.081	77	1.046E-003		
Lack of Fit	0.073	33	2.217E-003	13.24	< 0.0001
Pure Error	7.365E-003	44	1.674E-004		
Cor Total	0.51	86			

Table 5.5: Analysis of Variance on $\Delta_{P_{nl}}$

Equation 5.5 shows the final quadratic model.

$$\hat{\Delta}_{P_{nl}} = -0.023 + 0.019A - 0.039B + 0.015C - 0.023D + 7.252 \times 10^{-3}E$$
$$- 0.039B^2 + 0.041AB + 7.750 \times 10^{-3}BD - 0.011CD \tag{5.5}$$

This equation indicates that although significant, E and BD have very little effect on the response. In other words, the weight, w_l and the interaction between targets, ρ_0, and P_{mm} do not greatly affect the difference in P_{nl} between Case 2 and 3. It is important to note that the effect of ρ_0 on this difference is quadratic. In addition, factors β_0 and ρ_0 and factors P_{hh} and P_{mm} both have strong interaction effects on this difference.

In order to test the adequacy of our model, we must make sure some assumptions hold true. If ϵ is the error between predicted values and actual values, we assume that ϵ is normally and independently distributed with a mean zero and constant variance [13]. We first examine the normality assumption with Figure 5-4. For the normality assumption to hold, the data points should fall along the line drawn through the chart. In this chart, we see that some points at the top right of the chart lie off of that line. This indicates slight departures from normality, but overall the majority of points lie close to the line. Therefore,
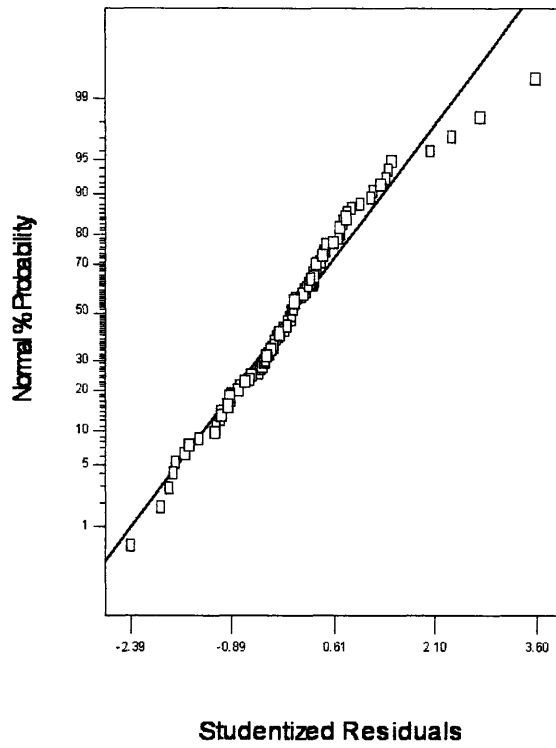
Figure 5-4: Normal Probability Plot of Residuals for Model 1

overall the normality assumption is valid.

Next, we examine the residuals for independence between runs. We already discussed that there should be no relationship between runs, as our simulation and POMDP solver have no memory. Therefore, we did not randomize our experiments. Regardless of this fact, we examine the independence of runs in Figure 5-5. This chart shows that there is no reason
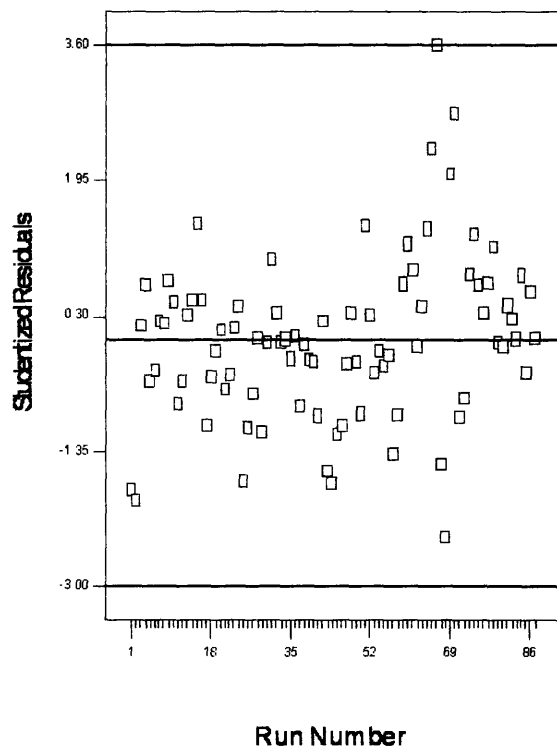

Figure 5-5: Residuals versus Runs in Model 1

to suspect any violation of the independence or constant variance assumption.

Lastly, we examine a plot of the residuals versus the predicted values from our model shown in Figure 5-6. If our assumptions hold true, the residuals should not be related to the predicted response variable. In this chart, no unusual structure or pattern is apparent. Overall, we have shown that our assumptions hold true and that our model is valid.
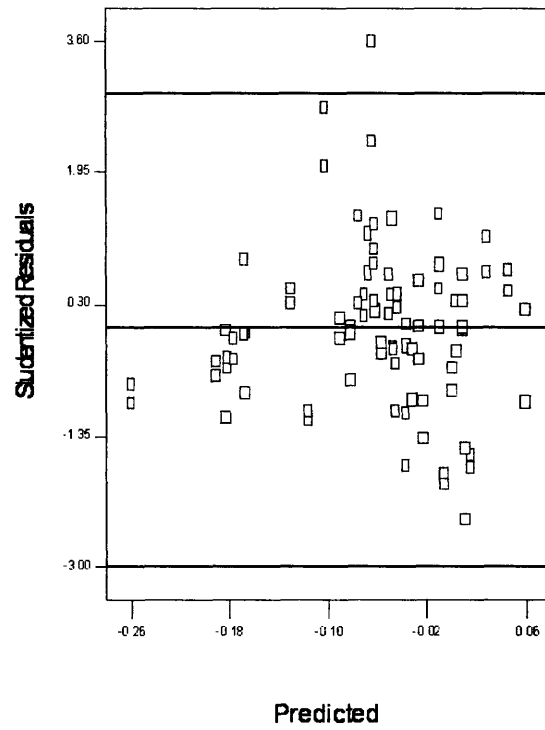
Figure 5-6: Residuals versus Predicted Values in Model 1

5.2.2 Model 2

Our second model fits a regression equation on the response variable, Δ_{β_2}. As with the first model, we begin with an all-inclusive quadratic model, and reduce the model in a stepwise manner to only significant terms. Table 5.6 shows the ANOVA results for this model. We

Source	Sum of Squares	DF	Mean Square	F Value	P-Value
Model	95.90	10	9.59	109.98	< 0.0001
A	59.73	1	59.73	685.00	< 0.0001
B	7.67	1	7.67	87.97	< 0.0001
C	12.58	1	12.58	144.25	< 0.0001
D	4.21	1	4.21	48.26	< 0.0001
E	0.80	1	0.80	9.14	0.0034
A^2	2.27	1	2.27	26.03	< 0.0001
B^2	2.39	1	2.39	27.36	< 0.0001
AB	4.10	1	4.10	47.01	< 0.0001
AC	2.40	1	2.40	27.49	< 0.0001
BC	0.46	1	0.46	5.29	0.0242
Residual	6.63	76	0.087		
Lack of Fit	6.46	32	0.20	52.94	< 0.0001
Pure Error	0.17	44	3.813E-003		
Cor Total	102.52	86			

Table 5.6: Analysis of Variance on Δ_{β_2}

see that with a $p < 0.0001$, Model 2 is significant. In addition to all main effects, two quadratic effects, A^2 and B^2, are significant. This indicates that the relationship between both interceptor inventory and number of targets to the difference in remaining inventory is non-linear. The model also includes three two-way interactions: AB, AC, and BC. This suggests that the interceptor inventory, number of targets, and P_{hh} have interacting effects on the remaining inventory. As with Model 1, this model has a significant lack of fit for likely the same reasons. However, with $R^2 = 0.9354$ and $R^2_{Adj} = 0.9269$, we know that about 93% of the variation in the response is explained by the model.

80

Our final quadratic model is shown in Equation 5.6.

$$\hat{\Delta}_{\beta_2} = 1.28 + 0.86A - 0.31B - 0.38C + 0.22D - 0.096E$$
$$+ 0.21A^2 + 0.21B^2 - 0.25AB - 0.19AC + 0.085BC$$

(5.6)

We again check the assumptions of our model through three separate charts. Figure 5-7 shows the normal probability plot for Model 2. This chart shows slight departures from
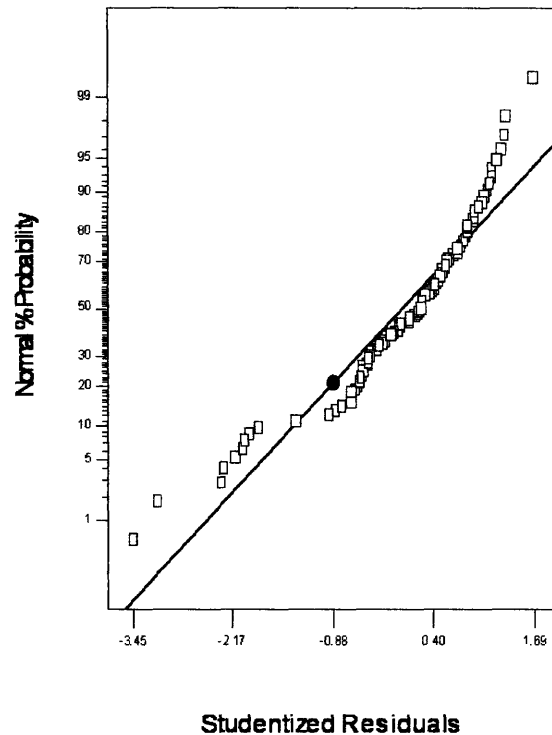

Studentized Residuals

Figure 5-7: Normal Probability Plot of Residuals for Model 2

normality, especially at the ends of the data points, with most points in the center falling along the line. According to Montgomery, slight deviations from normality such as these do not significantly impact the validity of the ANOVA results [13]. Therefore, we may proceed with our analysis of the model.

We next examine the independence between runs shown in Figure 5-8. This chart depicts no pattern between the runs and so there is no reason to suspect any violation of the independence or constant variance assumption.
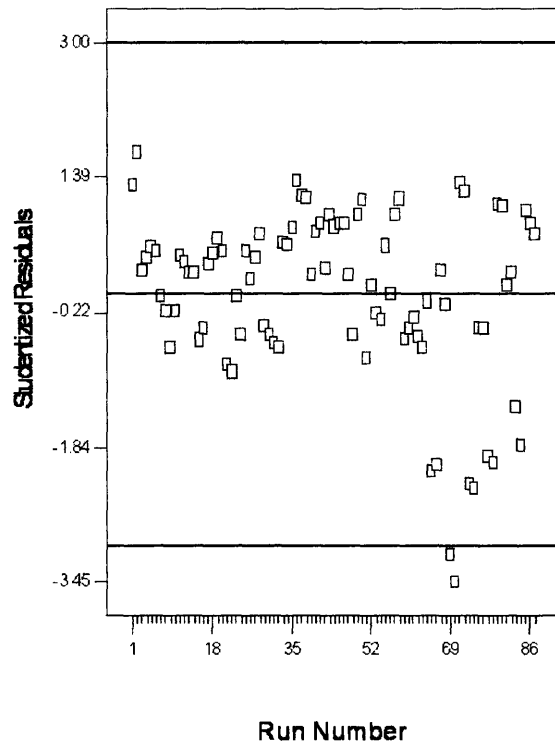
Figure 5-8: Residuals versus Runs in Model 2

Finally. Figure 5-9 shows a chart of the residuals versus the predicted values from our model. The model is valid if the error is not related to the predicted response variable. This
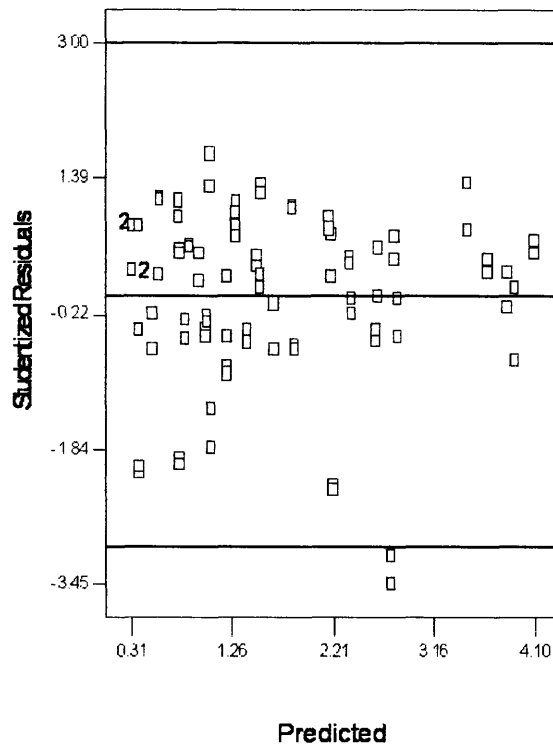

Figure 5-9: Residuals versus Predicted Values in Model 2

chart suggests no pattern or structure in the error. Our model proves to be valid as it does not violate any of the assumptions.

5.2.3 Model 3

In our third model. we fit a quadratic equation to the response variable Δ_W that includes terms based on their significance, determined in a stepwise manner. The results from the ANOVA are shown in Table 5.7.

Based on a $p < 0.0001$, Model 3 is significant. Although factor D is not statistically significant. it is included in the model. Despite a lack of statistical significance. we include P_{mm} because of its operational importance as a factor in a single engagement. In addition to main effects. this model has many other terms that are significant. Two square terms

Source	Sum of Squares	DF	Mean Square	F Value	P-Value
Model	0.45	13	0.035	100.79	< 0.0001
A	0.11	1	0.11	333.38	< 0.0001
B	0.10	1	0.10	304.24	< 0.0001
C	0.015	1	0.015	44.42	< 0.0001
D	3.920E-004	1	3.920E-004	1.14	0.2894
E	0.18	1	0.18	529.31	< 0.0001
B^2	3.614E-003	1	3.614E-003	10.50	0.0018
E^2	2.464E-003	1	2.464E-003	7.16	0.0092
AB	7.042E-003	1	7.042E-003	20.46	< 0.0001
AC	2.025E-003	1	2.025E-003	5.88	0.0178
AE	2.275E-003	1	2.275E-003	6.61	0.0122
BC	5.131E-003	1	5.131E-003	14.91	0.0002
CE	7.704E-003	1	7.704E-003	22.38	< 0.0001
Residual	0.025	73	3.442E-004		
Lack of Fit	0.023	29	7.811E-003	13.90	< 0.0001
Pure Error	2.473E-003	44	5.621E-005		
Cor Total	0.48	86			

Table 5.7: Analysis of Variance on Δ_W

are significant: B^2 and E^2. Therefore, we know that the number of targets and the weight have a quadratic effect on the difference in our weighted performance metric, W. Model 3 also includes six two-way interactions: AB, AC, AE, BC, and CE. It should be noted that these six effects are the three included in Model 2: AB, AC, BC. in addition to all of the statistically significant main effects interacting with factor E from Model 3. It is logical that factor E is significant in addition to its interactions as w_I has a direct impact on the calculation of Δ_W. Therefore, we know that w_I has a large impact on the overall performance of Case 3. Model 3 also has significant lack of fit regardless of the terms included. but its $R^2 = 0.9472$ and $R^2_{Adj} = 0.9378$.

Our final quadratic model is shown in Equation 5.7.

$$
\begin{aligned}
\hat{\Delta}_{\beta_2} = {} & 0.064 + 0.038A - 0.036B - 0.013C + 2.122 \times 10^{-3}D - 0.046E \\
& - 8.329 \times 10^{-3}B^2 + 4.847 \times 10^{-3}E^2 + 0.010AB - 5.624 \times 10^{-3}AC \\
& - 5.962 \times 10^{-3}AE + 8.954 \times 10^{-3}BC + 0.011CE - 5.356 \times 10^{-3}DE
\end{aligned} \tag{5.7}
$$

We check the validity of our model by verifying the assumptions about the error term. ϵ. First, we check the normality of the residuals with the chart in Figure 5-10. The points
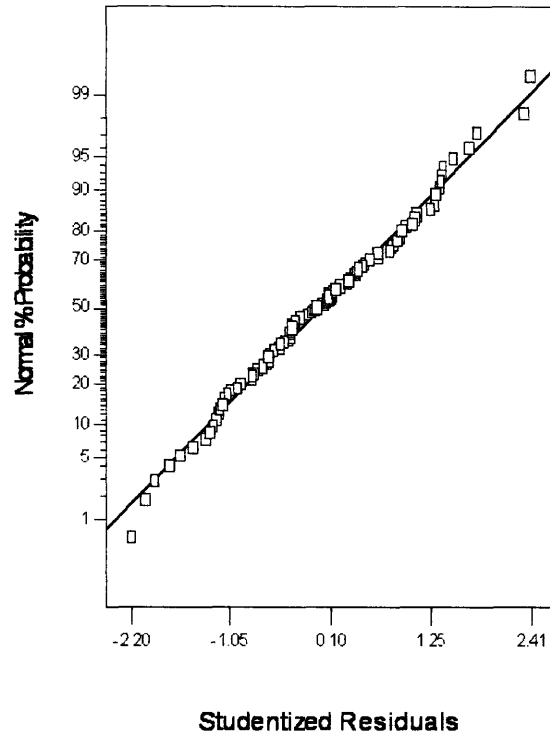

Figure 5-10: Normal Probability Plot of Residuals for Model 3

in this chart all lie very close to the normality line, so we can assume the error is normally distributed.

To determine if the runs are independent, we use the chart in Figure 5-11. The residuals in this chart appear completely random, which indicates independence and constant variance.

Third, we examine Figure 5-12 to check if the residuals are related to the predicted response. There is no pattern to suggest that the residuals are not independent of the response. Based on these three charts, we have checked all of the assumptions of Model 3.

From our three statistical models, we have seen that each of the five factors does not have a simple linear effect on the performance of Cases 2 and 3. In all three models, there were significant quadratic and two-way interaction terms. This implies that the response variables are many times determined by a complex interaction of factors. Knowing this
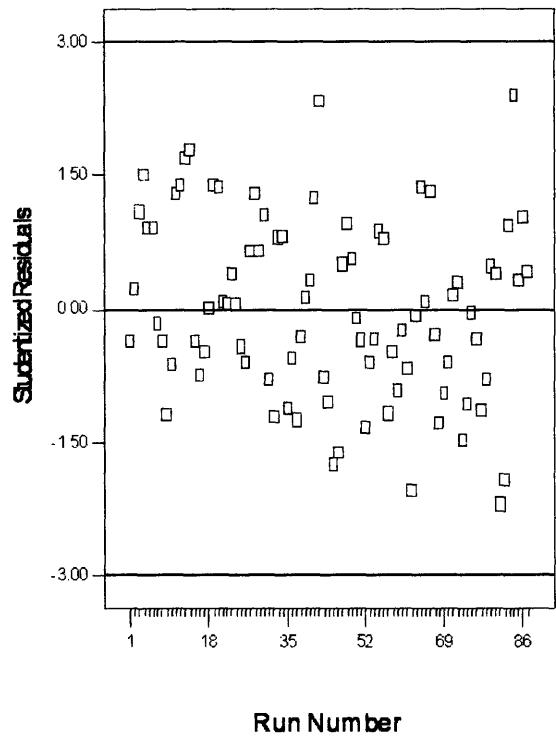
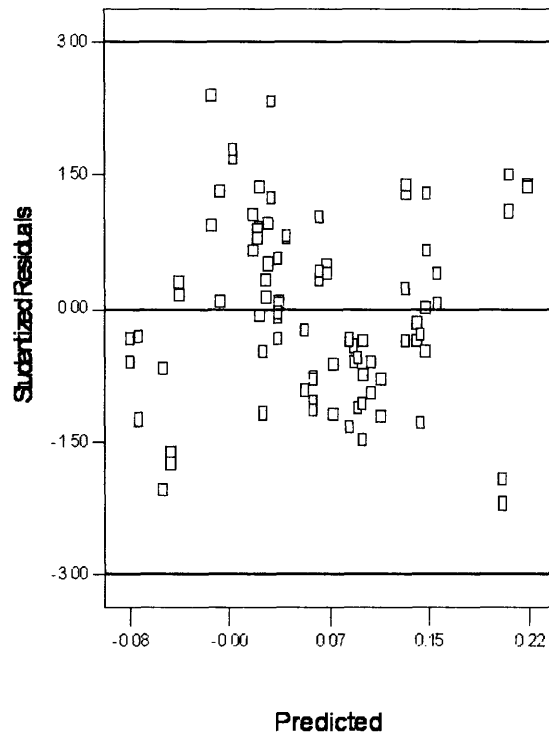Figure 5-11: Residuals versus Runs in Model 3

Figure 5-12: Residuals versus Predicted Values in Model 3

result. we proceed to our single-factor experiments.

5.3 Single-Factor Results

After gaining insight as to how five factors affect the performance of Cases 2 and 3, we further investigate the effects of these factors in single-factor experiments. Experiments 5 through 8 give us a more in-depth idea as to how sensitive the response variables are to changes in each factor. We note that based on the results of Experiment 4, we cannot expect that the results of these single-factor experiments to be completely typical of all scenarios. Due to interactions between factors, beginning these experiments with different baselines could prove to yield somewhat different results. Despite this fact, we still gain valuable insight from these one-factor experiments.

5.3.1 Experiment 5

We begin by varying the intermediate weight, w_I, in Experiment 5. In this experiment, we examine the performance of each case as w_I is varied at levels between zero and one. Table 5.8 shows the policy solutions for each case at various levels of w_I. In this table, Cases

w_I	Cases 1 and 2	Case 3
0.0	4:0.3,6,6	3:0,1,2,3
0.1	4:0,3,6,6	3:0,1,2,3
0.2	4:0.3,6,6	3:1,1,2,3
0.3	4:0.3,6,6	3:1,1,2,3
0.4	4:0,3,6,6	3:0,1,2,3
0.5	4:0,3,6,6	3:1,1,2,3
0.6	4:0.3.6.6	3:1,1,2,3
0.7	4:0,3,6,6	3:1,2,2,3
0.8	4:0.3.6.6	3:1,2,2,3
0.9	4:0,3.6.6	3:2,2,2,3
1.0	4:0,3,6.6	3:2.2.2,3

Table 5.8: Policy Solutions for Experiment 5

1 and 2 always have the same policy solution regardless of w_I. This occurs because the MMR

algorithm used in these cases does not take w_I into account when making decisions. The MMR algorithm relies only on β, ρ, and $SSPK$ to make decisions. In contrast, the POMDP solver used in Case 3 relies on the reward function calculated by w_I to determine a policy solution.

Table 5.8 also shows that in general as w_I increases, the policy solution uses more interceptors in the second action. The first action consistently remains at $a_1 = 3$ regardless of the w_I. This decrease in conservatism occurs due to the nature of the reward function in Equation 3.6. Higher levels of w_I correspond to a $R(s, o, a, s')$ that values P_{nl}, and likewise lower levels of w_I correspond to a $R(s, o, a, s')$ that values $\frac{\beta_{s'}}{\beta_0}$. We should note that while the policy solutions for Case 3 generally add more interceptors to action 2 as w_I increases, this trend does not occur for $w_I = 0.4$. In this scenario the POMDP solver chooses to use one less interceptor for a_2^0 than the policy solutions for $w_I = 0.3$ and $w_I = 0.5$. This aberration may result from some instability in the POMDP solver solution.

We further investigate the impact of w_I on the performance of each case by examining the probabilities of no leakage as w_I changes. A chart of w_I versus P_{nl} is shown in Figure 5-13. In this chart, Cases 1 and 2 are denoted by a single line. This occurs because the MMR algorithm in both of these cases has the same policy solution regardless of w_I. Figure 5-13 shows that for Case 3 as w_I increases, P_{nl} generally increases as well. P_{nl} begins at approximately 84% and continually increases until it approaches the P_{nl} for Cases 1 and 2 at approximately 95%. Once $w_I > 0.7$, we see that P_{nl} remains greater than 93%, almost if not equaling the performance of Cases 1 and 2. We also notice the effect of the aberation in policy solution at $w_I = 0.4$ on P_{nl}, as it decreases momentarily against the general trend.

Next, we consider a plot of the inventory remaining against varying levels of w_I shown in Figure 5-14. This plot shows that Cases 1 and 2 have an average remaining inventory of approximately 3.5 interceptors. The policy solutions of Case 3 gradually become less conservative as w_I increases and leave less interceptors in inventory. For this case, a $w_I \approx 0$ corresponds to $\beta_2 \approx 6$, and a $w_I \approx 1$ corresponds to $\beta_2 \approx 5$. Again, we notice the same aberration at $w_I = 0.4$ as the only point on the line where the slope is positive.
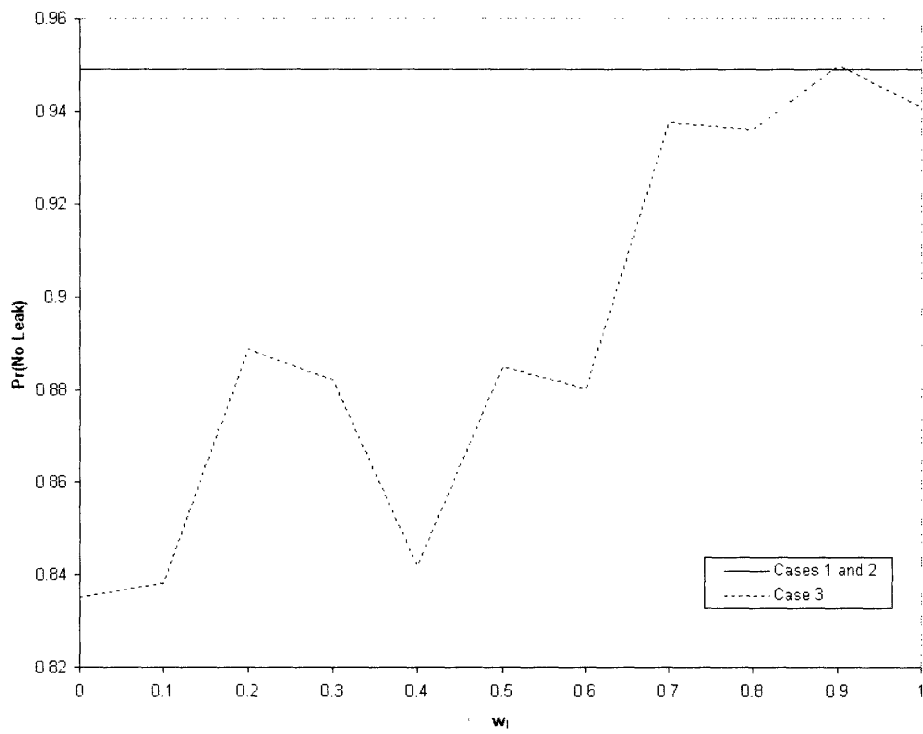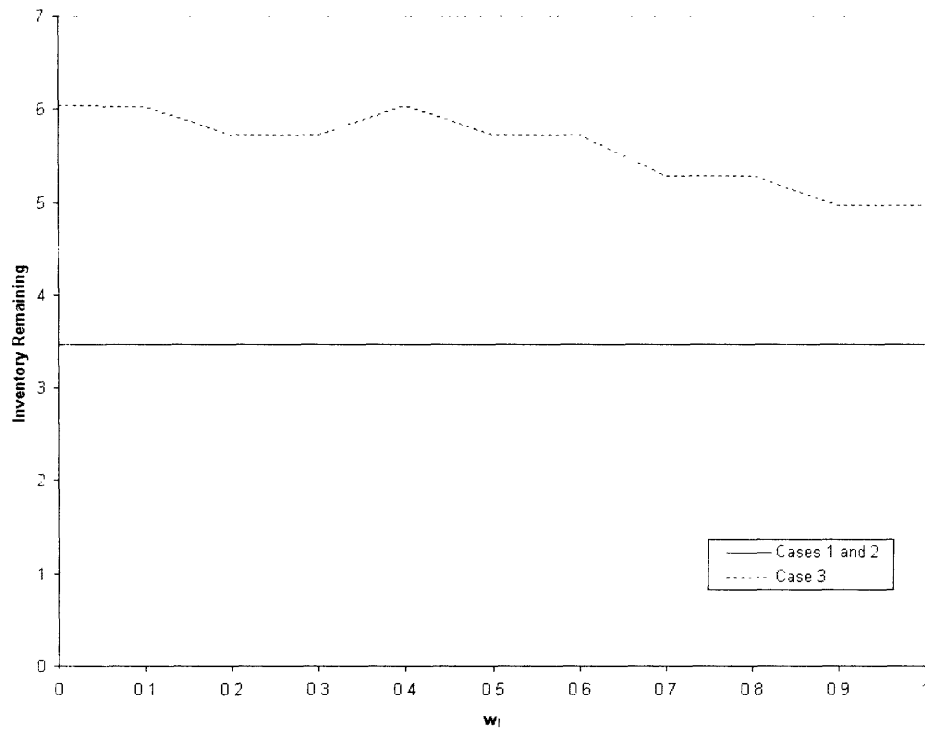
Figure 5-13: P_{nl} versus w_I

Figure 5-14: β_2 versus w_I

Lastly, we examine the effect of w_I on W. A chart of this data is shown in Figure 5-15. Based on the W measure of performance, Case 3 always outperforms the other cases, except
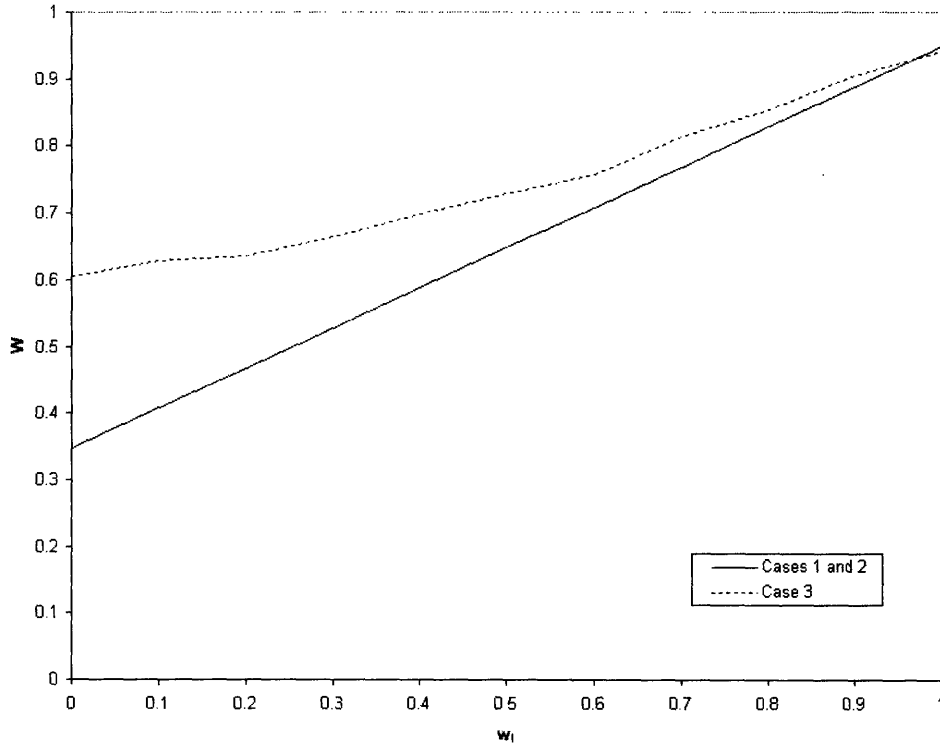

Figure 5-15: W versus w_I

for $w_I = 1.0$ where $\Delta_W \approx 0$, which could be explained by random error in the simulation. The W of Case 3 gradually increases as w_I increases, but it does not increase as much as W for Cases 1 and 2. This difference in W is greatest when $w_I = 0$ and decreases gradually until there is no significant difference at $w_I = 1.0$. Based on W, as we increase w_I, the advantage of Case 3 over Cases 1 and 2 decreases.

5.3.2 Experiment 6

In Experiment 6 we vary P_{hh} and P_{mm} simultaneously to examine the effect on the performance of Cases 2 and 3. It should be noted that in Case 1 $P_{hh} = P_{mm} = 1$, so there is only one data point for comparison. Table 5.9 shows a table of policy solutions for Cases 2 and 3 as P_{hh} and P_{mm} are varied between values of 0.5 and 1.0. In this experiment the policy

$P_{hh} = P_{mm}$	Case 2	Case 3
0.5	4:0.3.6.6	3:2.2.2.2
0.6	4:0.3.6.6	3:1.2.2.2
0.7	4:0.3.6.6	3:1.2.2.3
0.8	4:0.3.6.6	3:1.2.2.3
0.9	4:0.3.6.6	3:1.2.3.3
1.0	4:0.3.6.6	3:0.2.3.3

Table 5.9: Policy Solutions for Experiment 6

solution for Case 2 is always constant, as it does not account for varying P_{hh} or P_{mm}. In other words Case 2 always assumes that $P_{hh} = P_{mm} = 1$, and consequently takes the same actions. In contrast, Case 3 does account for this imperfect kill assessment.

The values of P_{hh} and P_{mm} affect the policy solutions in two different and independent ways. P_{hh} affects the number of interceptors used with none or few targets observed. P_{mm} affects the number of interceptors used when a higher number of targets are observed. With lower values of P_{hh}, the POMDP solver uses fewer interceptors in action 2 with observations of many targets remaining. In a sense, it does not trust these observations and does not use as many interceptors as seems appropriate. This occurs due to low P_{hh}, which implies a larger P_{mh}. This means that we think we missed more targets than we actually missed. Therefore, when many targets are observed remaining, there is a good chance some of those have been hit. Table 5.9 shows that when $P_{hh} = P_{mm} = 0.5$ and three targets are observed, the POMDP solver only uses two interceptors. Given this P_{hh} and $SSPK = 0.8$, it is unlikely that all targets were missed even if they were all observed missed. As P_{hh} improves, the POMDP solver gradually uses more interceptors with observations of two or three targets. In a sense, it can trust the observations more.

While P_{hh} affected the policy solution with larger observations, P_{mm} affects the policy solution when fewer targets are observed. With lower values of P_{mm}, the POMDP solver uses more interceptors in action 2 when observing zero targets remaining. This occurs due to a low P_{mm}, which implies a higher P_{hm}. This means that we think we hit more targets than we actually did. When zero targets are observed remaining, there is a good chance some still remain. In the scenario $P_{hh} = P_{mm} = 0.5$, the POMDP solver in Case 3 uses two interceptors

when it observes zero targets remaining. As P_{mm} improves, the POMDP solver can trust its observations more and gradually uses fewer interceptors with observations of zero targets. Finally, when $P_{mm} = 1$, the POMDP solver uses zero interceptors for an observation of zero. In summary, the two effects of P_{hh} and P_{mm} produce the following result: as P_{hh} and P_{mm} increase, a_2^0 decreases and a_2^2 and a_2^3 increase.

In addition to the two independent effects of P_{hh} and Γ_{mm}, we observe that as P_{hh} and P_{mm} both increase, action 2 goes from being completely independent of the observation to being very dependent on the observation. The POMDP solver cannot trust the observations when $P_{hh} = P_{mm} = 0.5$, so it always assigns two interceptors. However, when $P_{hh} = P_{mm} = 1.0$, the POMDP solver assigns very differently depending on the observation.

While the policy solutions provide an idea of how the decisions are made, we also need to examine how P_{hh} and P_{mm} actually affect the performance of Cases 2 and 3. Figure 5-16 shows a plot of probability of no leakage versus P_{hh} and P_{mm}. This chart shows that most
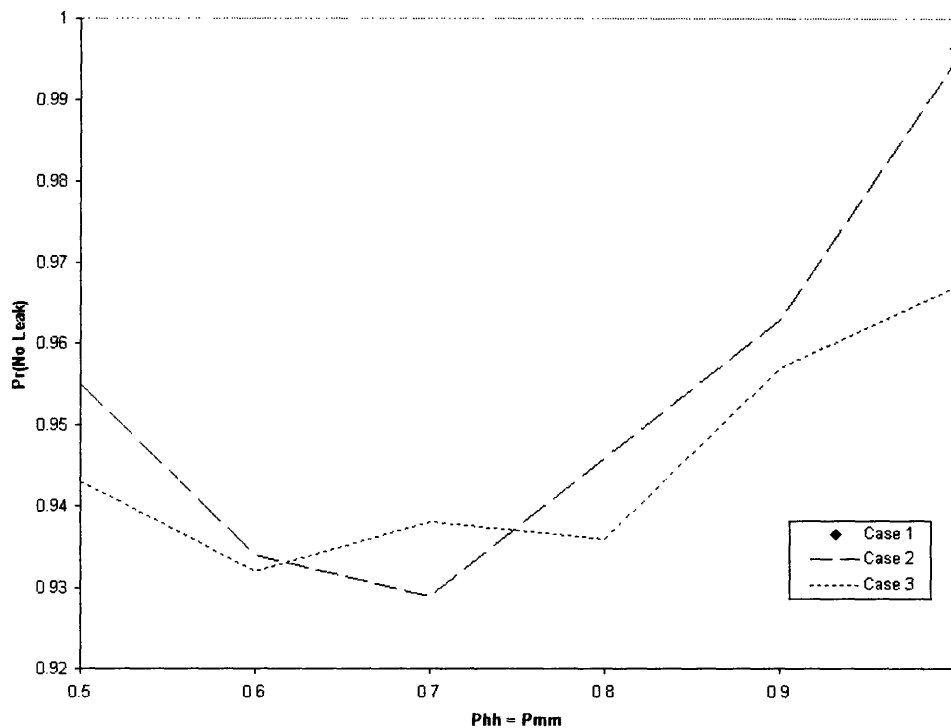

Figure 5-16: P_{nl} versus P_{hh} and P_{mm}

of the time Case 2 outperforms Case 3 in terms of P_{nl}. Case 1 provides one point. which is an upper bound. that is only matched by Case 2 when $P_{hh} = P_{mm} = 1$. This makes sense. because when $P_{hh} = P_{mm} = 1$. Case 1 and Case 2 are essentially the same. While Case 2 always outperforms Case 3 except when $P_{hh} = P_{mm} = 0.7$. it is important to note that Case 3 always almost matches the P_{nl} of Case 2. There is never a difference in P_{nl} greater than 3%. and Case 3 values for P_{nl} never fall below 93%.

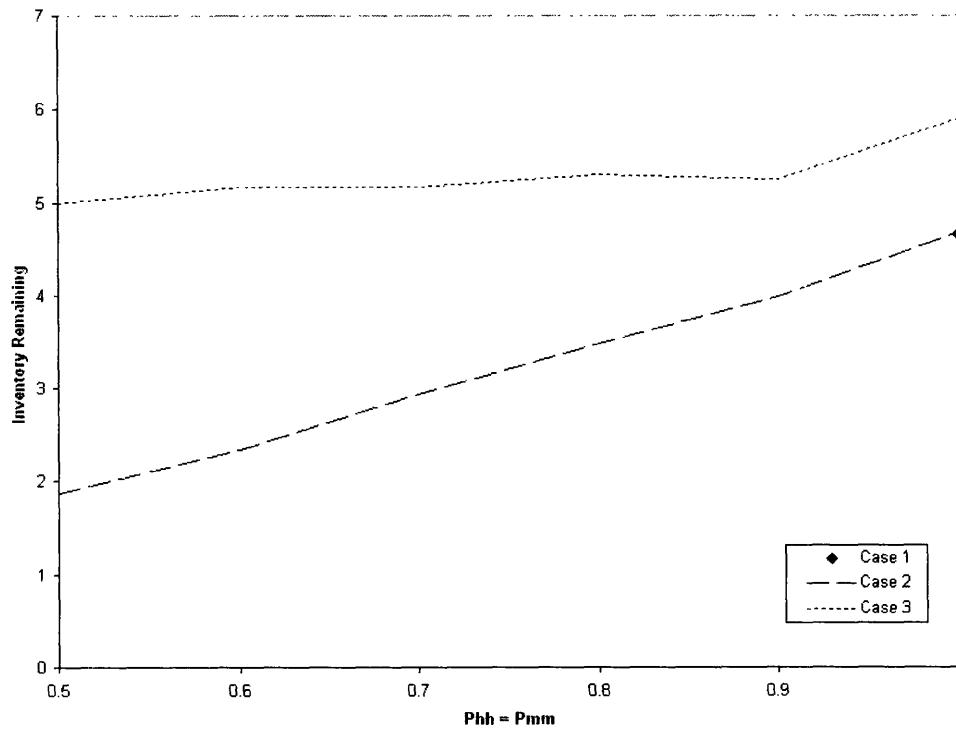We also examine a plot of P_{hh} and P_{mm} versus inventory remaining in Figure 5-17. This

Figure 5-17: β_2 versus P_{hh} and P_{mm}

figure shows that Case 3 always has a higher average inventory remaining than Cases 1 or 2. The difference between the remaining inventories of these cases does decrease as P_{hh} and P_{mm} increase. but it never falls below one interceptor. When $P_{hh} = P_{mm} = 0.5$, $\Delta_{\beta_2} > 3$. This is an important result, because the policy solutions of Case 3 provide P_{nl} that almost match those of Case 2 while saving between one and three extra interceptors.

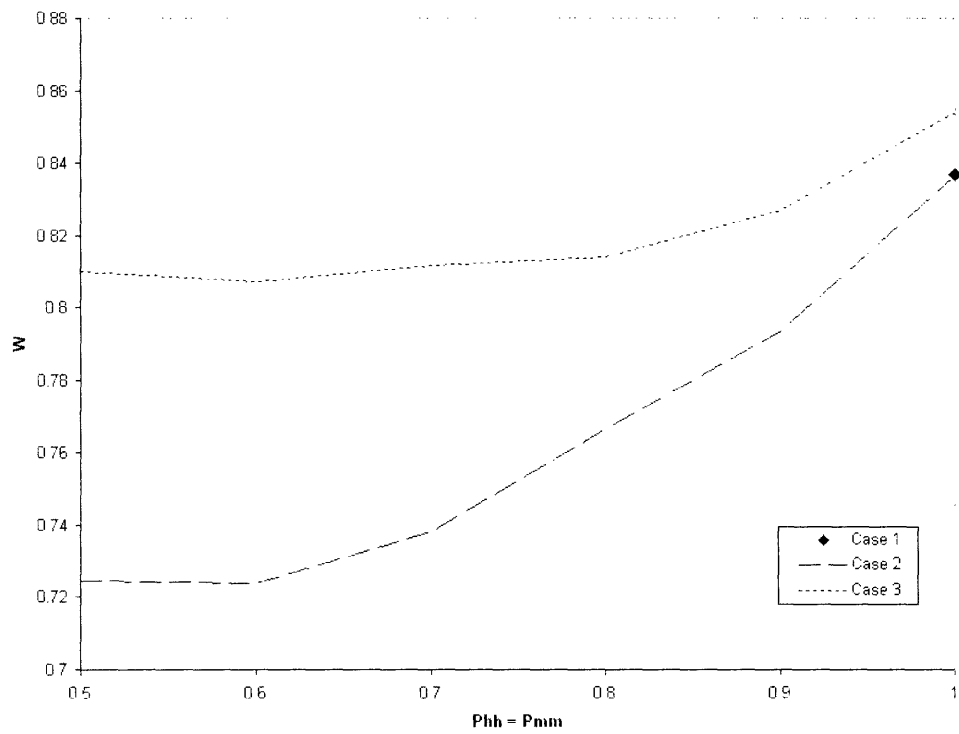Lastly. we analyze a plot of W versus P_{hh} and P_{mm} shown in Figure 5-18. This chart

Figure 5-18: W versus P_{hh} and P_{mm}

proves to be very similar to Figure 5-17. Case 3 always has a higher $W^\cdot$ than Cases 1 and 2, and the difference between them, Δ_W, decreases as P_{hh} and P_{mm} decrease.

Overall, Experiment 6 showed that over various levels for P_{hh} and P_{mm}, Case 3 almost matches P_{nl} for Case 2 and always outperforms Cases 1 and 2 with respect to inventory remaining and $W^\cdot$.

5.3.3 Experiment 7

Experiment 7 varies the number of initial interceptors in order to compare the performance of all three cases. Table 5.10 shows the policy solutions for the three cases in this experiment. Again, in this experiment, Cases 1 and 2 have the same policy solution based on the MMR

Interceptors	Cases 1 and 2	Case 3
4	3:0,1,1,1	2:NA,1,1,1
5	3:0,2,2,2	3:1,1,1,1
6	3:0,3,3,3	3:1,1,1,1
7	3:0,3,4,4	3:1,1,2,2
8	3:0,3,5,5	3:1,2,2,3
9	3:0,3,6,6	3:1,2,2,3
10	4:0,3,6,6	3:1,2,2,3
11	5:0,3,6,6	3:1,2,2,3
12	6:0,3,6,6	3:1,2,2,3
13	6:0,3,7,7	3:1,2,2,3
14	6:0,3,8,8	3:1,2,2,3
15	6:0,3,8,9	3:1,2,2,3
16	6:0,3,8,10	3:1,2,2,3

Table 5.10: Policy Solutions for Experiment 7

algorithm. The policy solutions for all cases gradually use more interceptors as the inventory increases. However, there are many differences between the policy solutions. The first major difference between Cases 1 and 2 and Case 3 is that the POMDP solver in Case 3 always assigns one interceptor for action 2 when the observation is zero targets. In contrast, the MMR algorithm never assigns an interceptor when no targets are observed. Another difference between the cases is how each case uses its inventory. In Cases 1 and 2 the algorithm

97

takes advantage of its inventory, assigning more interceptors as its inventory increases. The POMDP solver for Case 3 is far more conservative. Regardless of its inventory, it never assigns more than six interceptors in a single engagement. When $\beta_0 \geq 8$ the policy solution is always (3:1,2,2,3). In addition, for a small initial inventory of interceptors, Case 3 does not always use all of its inventory, while Cases 1 and 2 always use their full inventory. In fact with four initial interceptors, Case 3 only uses three interceptors, assigning $a_1 = 2$ and $a_2 = 1$ regardless of the observation.

While the policy solutions provide some insight as to the sensitivity of each case to changes in initial inventory, we also examine the sensitivity of the probability of no leakage, remaining inventory, and W. Figure 5-19 depicts a chart of P_{nl} versus interceptors. This
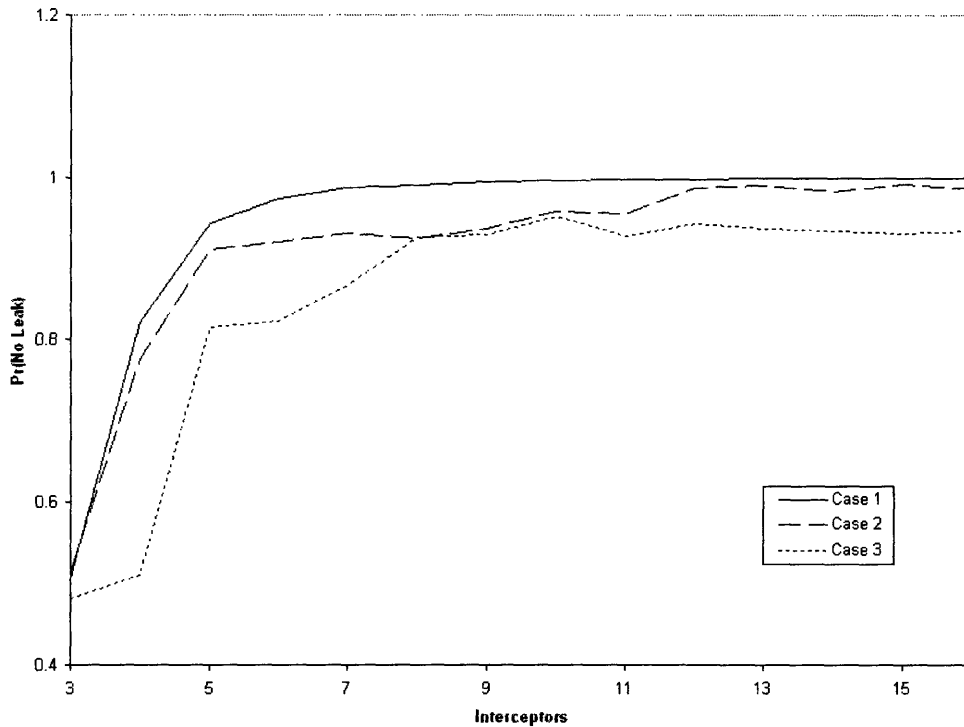

Figure 5-19: P_{nl} versus β_0

table shows that Case 1 is an upper bound on P_{nl} that is almost reached by Case 2 for high values of β_0. In general, Case 2 also does better than Case 3. Except for $\beta_0 = 4$ and $\beta_0 = 5$, the difference between Cases 2 and 3 is not too considerable. When $\beta_0 \geq 8$, the difference

in P_{nl} is never greater than 6%.

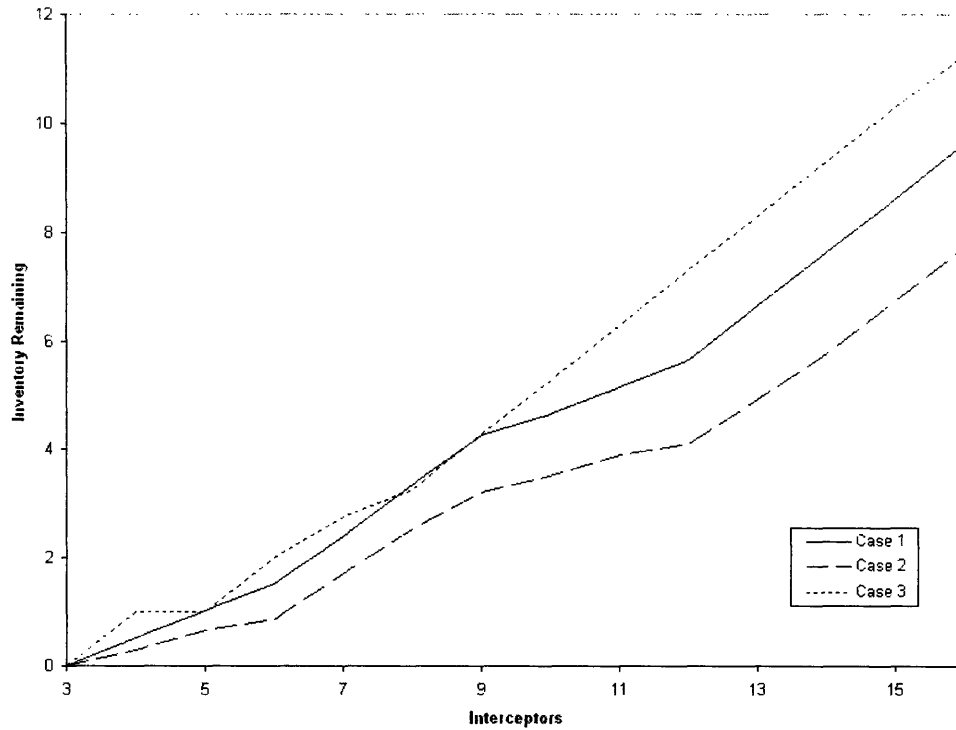Next we consider a plot of interceptors versus inventory remaining in Figure 5-20. This

Figure 5-20: β_2 versus β_0

chart shows that Case 3 almost always has more remaining inventory than Cases 1 and 2, and Case 1 performs better than Case 2. This occurs because Case 3 typically has a much more conservative policy solution. Although they have the same policy solutions, Case 1 has more remaining inventory than Case 2 because it sees observations of three and two targets much more rarely, as in Case 1, $P_{hh} = 1$.

We further explore the relationship of initial inventory to each case's performance with a plot of interceptors versus W in Figure 5-21. In this figure, we observe that as in Figure 5-19, Case 1 serves as an upper bound on W for the other two cases. The difference from that plot is that Case 3 has a higher W than Case 2 after $\beta_0 \geq 7$. Even when $4 < \beta_0 < 7$, the difference in W between Cases 2 and 3 never exceeds 0.05.

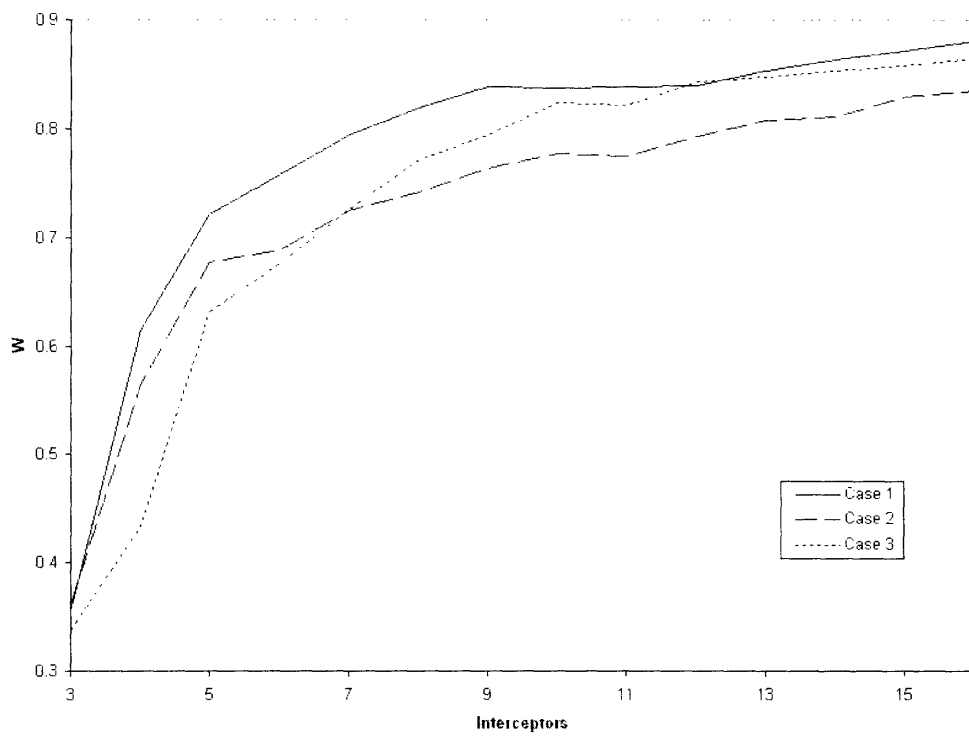Overall, this experiment showed that the performance of Case 3 is somewhat sensitive to

99

Figure 5-21: W versus β_0

changes in initial inventory, particularly with respect to P_{nl}. While Case 3 outperforms the other cases in terms of inventory remaining, it does not have P_{nl} levels as high as Cases 1 or 2 for lower initial inventories of interceptors. In addition, W for Case 3 does not match that of Case 2 for lower numbers of interceptors.

5.3.4   Experiment 8

In our final experiment, we vary the number of initial targets, ρ_0, between one and ten, leaving all other factors at the baseline level. The policy solutions for this experiment are shown in Table 5.11. The policy solutions in this table are not extremely different between

Targets	Cases 1 and 2	Case 3
1	1:0,3	2:0,2
2	3:0,3,7	2:1,2,2
3	4:0,3,6,6	3:1,2,2,3
4	4:0,3,6,6,6	4:1,2,2,3,4
5	5:0,3,5,5,5,5	5:1,2,3,3,3,3
6	6:0,3,4,4,4,4,4	6:1,2,2,2,2,2,2
7	7:0,3,3,3,3,3,3,3	7:1,1,1,1,1,1,1,2
8	8:0,2,2,2,2,2,2,2,2	8:1,1,1,1,1,1,1,1,1
9	9:0,1,1,1,1,1,1,1,1,1	8:NA,1,1,1,1,1,1,1,1,1
10	10:0,0,0,0,0,0,0,0,0,0,0	9:NA,1,1,1,1,0,0,0,0,0,0

Table 5.11: Policy Solutions for Experiment 8

Cases 1 and 2 and Case 3. Except when $\rho_0 = 1$, the second action given an observation of zero targets for Case 3 is always $a_1 = 1$. In general, Case 3 is much more conservative in terms of its second actions. Particularly when $\rho_0 > 5$, the POMDP solver in Case 3 does not always use as many interceptors in action 2 as targets observed. This occurs because the reward function values remaining inventory, and it is still fairly unlikely to miss half of the targets given the baseline $SSPK$.

The number of targets versus probability of no leakage is plotted in Figure 5-22. In this figure, Case 1 always performs the best with respect to P_{nl}, and Case 2 performs better than Case 3. Case 3, however, almost matches the performance of Case 2 for $\rho_0 \leq 5$.
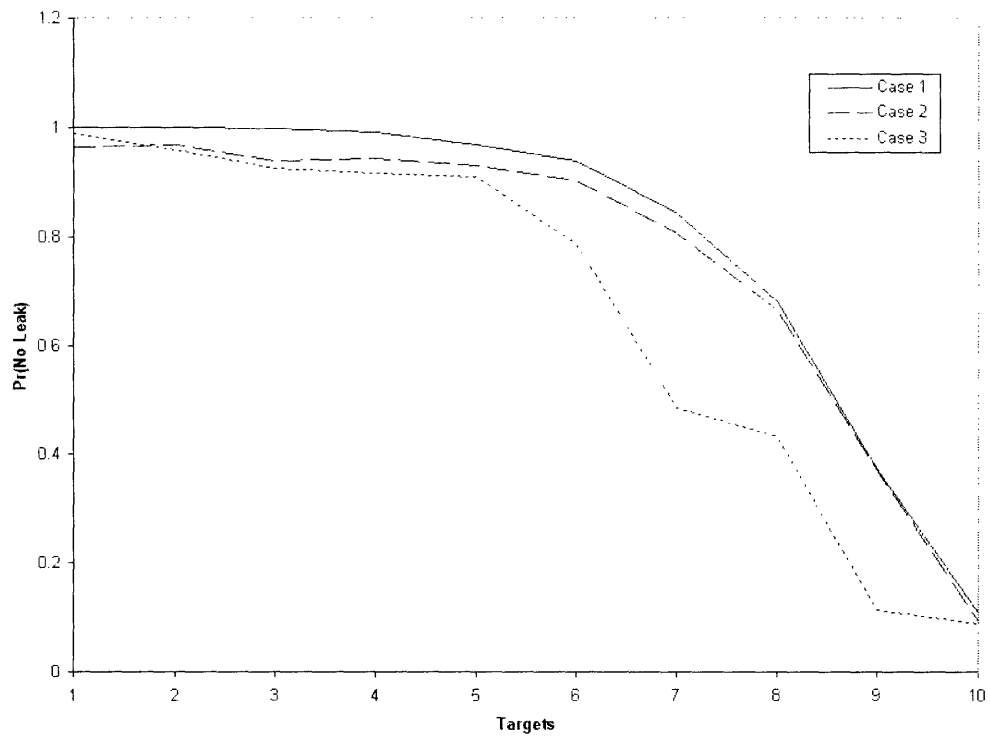
Figure 5-22: P_{nl} versus ρ_0

Again. we assess the effect of varying the number of targets with a chart of inventory remaining in Figure 5-23. In this chart, as with most of the other experiments, Case 3
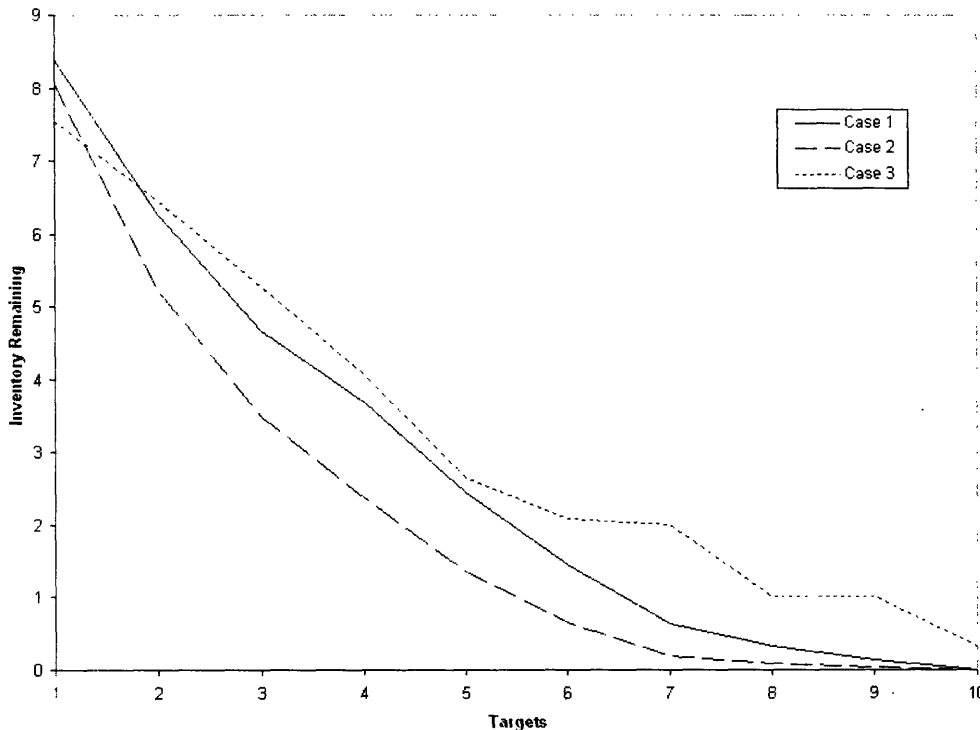

Figure 5-23: β_2 versus ρ_0

generally has the greatest remaining inventory, with Case 2 having the least remaining inventory. Also. similar to previous experiments. the difference between Cases 2 and 3 becomes greater as the number of targets increases.

Finally, we compare the three cases with the weighted combination of P_{nl} and β_2 in Figure 5-24. This chart appears much like that of Figure 5-22. in which Case 1 performs the best. However, with respect to W, Case 3 does better for $\rho_0 \leq 5$ and Case 2 does better for $\rho_0 > 5$. This chart, along with the other two from this experiment show that Case 3 is very sensitive to changes in the number of initial targets. Specifically, as ρ_0 approaches β_0, Case 3 becomes much less effective, and the MMR algorithm of Case 1 and 2 proves superior with respect to P_{nl}. β_2, and W.
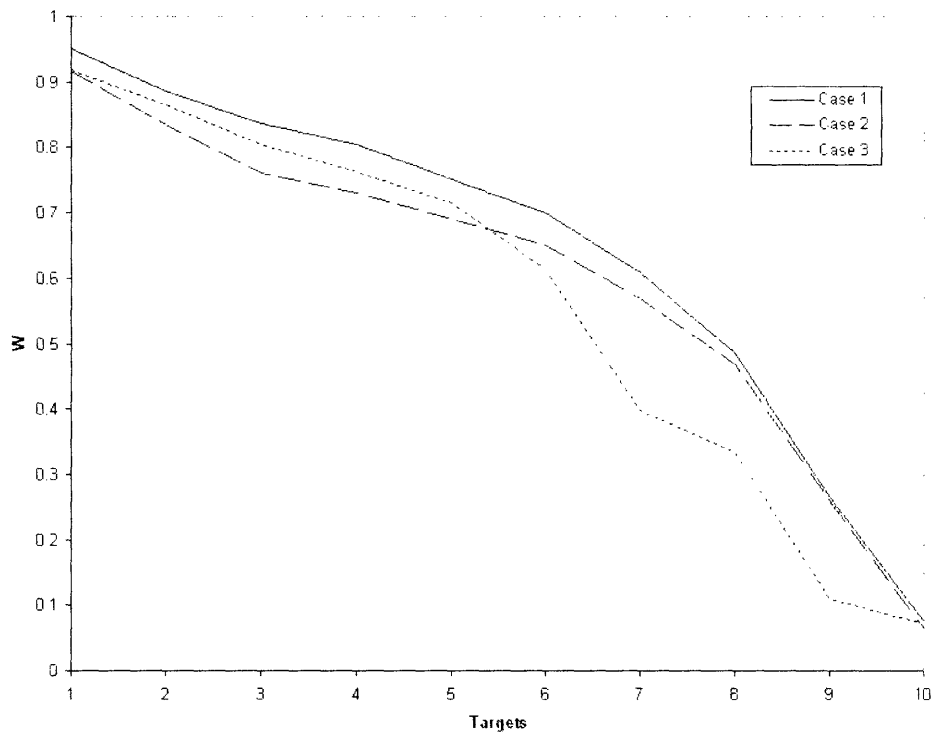
Figure 5-24: W versus ρ_0

5.4 Chapter Summary

This chapter contains the results and analysis for all eight experiments. It begins with a discussion of the baseline case. chosen as a possible real-world scenario. This baseline is the starting point for all other experiments. The first three experiments varied factors that would later be held constant: algorithm. terminal reward function. and $SSPK$. We found that all algorithms provide the same policy solution. but incremental pruning generally provides the fastest and most stable algorithm. In Experiment 2 we found that the POMDP solver is very sensitive to the terminal reward function, $F(s)$. and we determined a setting for this function that produces reasonable results. Experiment 3 showed that the performance of each case is very dependent on $SSPK$, however, Case 3 generally performs as well if not better than the other cases regardless of $SSPK$.

After conducting the initial experiments, we examined the way five factors affect both Case 2 and Case 3. We ran Experiment 4 in a central composite design in order to test for quadratic terms and factor interactions. We set up three quadratic models on the differences of P_{nl}. β_2. and W between Case 2 and Case 3. We found that all three models proved significant. and that all three had significant quadratic terms and two-way interactions. This tells us that there are complex relationships among the factors that affect the performance difference between Cases 2 and 3.

Lastly, we ran four single-factor experiments to further analyze the effect on the performance for each of the five factors in Experiment 4: w_I, P_{hh}. P_{mm}, β_0, and ρ_0. The overall conclusions from these four experiments were generally the same. We found that Case 3 typically has lower P_{nl} than the other two cases, but for most scenarios, this difference is very small. Many times Case 3 is within 3% to 6% of Case 2 in terms of P_{nl}. At the same time. Case 3 typically conserves many more interceptors. This can be attributed to Case 3 always assigning an interceptor with an observation of no targets. These two facts lead to a Case 3 W that is generally better than Case 2 and sometimes better than Case 1. Overall. we found that using the POMDP solver in Case 3 provides policy solutions that achieve almost equal P_{nl} as in Case 2. but consistently have a greater inventory of interceptors re-

maining. Using W as an overall metric, Case 3 generally does better than Case 2. These experiments also showed that while Case 3 was somewhat sensitive to all factors, it proved most sensitive to β_0 and ρ_0. The performance of this case was especially questionable as the scenario approached $\beta_0 \approx \rho_0$.

.

Chapter 6

Summary and Future Work

This chapter serves as a summary of the thesis and some final conclusions. It also provides a description of possible future work expanding on this research or applying it to other areas.

6.1 Thesis Summary

The goal of this thesis is to address the issue of imperfect information received from sensors in a ballistic missile single engagement and to investigate a method for making decisions in light of this uncertainty. To our knowledge, this is the first work that addresses the issue of imperfect kill assessment in the single engagement problem (consisting of a wave of incoming targets and a set of interceptors). We deal with the imperfect information by formulating the problem as a partially observable Markov decision process (POMDP). We assess the performance of this formulation by comparing it to two other cases in a series of experiments.

In **Chapter 1** we outlined the motivating problem for this work: a Ground-based Midcourse Defense (GMD) system. As this system grows and improves, uncertainty in sensor reliability may be an issue. The single engagement problem is assumed to be a "shoot-look-shoot" scenario. After an initial shot of a set of interceptors at a set of targets, sensors observe which targets were hit and which targets were missed before the second shot is taken.

Imperfect information from these sensors could have serious affects on the decision of how many interceptors to use in the second shot.

Chapter 2 focuses on the basis for our formulation. We discuss the use of dynamic programming to optimize an objective over multiple decisions. We describe the class of problems known as Markov decision processes (MDP), which are the basis for POMDPs. We outline the components of MDPs and the decision cycle. Next, we expand on this set of problems to describe the POMDP as an MDP in which the state of the system is not known with certainty. We explain the use of the belief state as a sufficient statistic for the state. Chapter 2 concludes with a description of the weapon-target assignment (WTA) problem. While the WTA approach does not account for the imperfect kill assessment addressed in this thesis, it does provide some useful mathematical ideas about methods for interceptor assignment.

In **Chapter 3** we present three cases for comparison: perfect information from our sensors (Case 1), imperfect information from sensors assumed perfect (Case 2), and imperfect information from sensors that decisions account for (Case 3). We formulate the third case as a POMDP.

Chapter 4 provides a description of the process used to solve and test the performance of each of the three cases. We begin with a description of the maximum marginal return (MMR) algorithm used to make interceptor assignments in Cases 1 and 2. From there we discuss the POMDP solver and its solution algorithms used to make interceptor assignments in Case 3. Next, we explain the solution process for a single experimental run. In this process, a simulation for the single engagement uses either the MMR algorithm or the POMDP solver to make interceptor assignments. This simulation over many trials estimates the performance of each case. Chapter 4 continues with a description of the experimental design. We begin with experiments to assess the effect of factors to be held constant in later experiments. Next, we conduct statistical analysis on three models to determine how five different factors impact the performance for Cases 2 and 3. Lastly, we conduct single-factor experiments on these five factors to gain a more detailed understanding of how they affect performance.

In **Chapter 5** we present the results from the experiments described in Chapter 4. For the initial experiments we find that the incremental pruning algorithm for solving POMDPs is the fastest and most stable algorithm. We also find that the algorithm is very sensitive to the terminal reward function, and we find a setting that provides reasonable results. Lastly, we show that all three cases are very sensitive to single-shot probability of kill (SSPK), but Case 3 generally performs better when varying this factor. In Experiment 4 we find that all three statistical models are significant. We find that not only all five factors, but at times their quadratic effects and interactions, highly impact the response variables. For the single-factor experiments we find that for most of the experiments, the POMDP approach of Case 3 conserves more interceptors and still approaches the probability of no leakage of Case 2. Based on the overall performance metric, W, we show that Case 3 typically outperforms Case 2.

In conclusion, the purpose of this thesis was to investigate the impact of imperfect kill assessment. We showed that assuming perfect information in a world where it is imperfect may significantly decrease the performance of the system, leading to a much lower probability of no leakage and wasted inventory. Our POMDP approach consistently conserved far more interceptors and generally performed well in terms of probability of no leakage. At the very least, this approach showed that using a single interceptor when no targets are observed can improve the overall probability of no leakage significantly. This approach, however, was very sensitive to the scenario, in particular the initial interceptors and initial targets. The policy solutions produced by the POMDP solver were not always reasonable. It is unlikely that a decision maker would use fewer interceptors than targets observed, unless that observation were extremely unlikely. Overall, the POMDP approach proved a valuable tool for making decisions under uncertainty in the single engagement problem.

6.2   Future Work

The work in this thesis on imperfect kill assessment could easily be expanded and continued to handle a broader array of missile defense scenarios. We suggest the following areas of

further research in the missile defense field:

- Our work assumed uniform reachability between targets and interceptors. In reality, incoming missiles have varying degrees of reachability depending on target destination, from where they are launched. and the location of the interceptors in relation to the flight path. In particular. there are currently two different interceptor locations.

- Our work also assumed identical single-shot probabilities of kill, SSPK, for each target. Targets may actually have different SSPKs based on each target-interceptor assignment; that is, some targets may be more difficult to destroy than others.

- This work did not address the existence of decoy targets. In reality, it is possible that some of the initial or observed targets are decoys and not actually warheads. This discrimination between decoys and actual targets adds a new element of uncertainty to the problem that was not formulated in this thesis.

- We also assumed that each target had an equal value. It is very reasonable that not every incoming target has the same value, especially if they are headed towards different locations. Certain cities or military installations have greater strategic value than others based on their population or on other factors. Thus, the value of any individual incoming missile might vary depending on its destination.

- This work also assumed that the initial state of the system is completely observable; that is, the initial wave of targets is known with certainty. It is quite possible that this may not be true. Future work could formulate a POMDP with a different initial belief state.

- Our work only considered one wave of incoming targets. Considering multiple waves of targets and modeling the state uncertainty in multiple waves would be a logical extension of this research.

In addition to expanding this research in the context of the missile defense problem, the work in this thesis could easily apply to a variety of other problems. The POMDP

110

formulation as well as the techniques used to solve the POMDP may be applied to other battle management problems. Specifically, problems involving allocating limited resources in a limited time-frame under uncertainty with consequences for every action may closely resemble the single engagement problem. These problems could be defense or non-defense related.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Glossary of Acronyms and Terms

action : decision made at each stage in a POMDP

alpha vector : vector with a value for each state corresponding to an action

ANOVA : Analysis of Variance

belief space : set of all possible belief states

belief state : probability distribution over all possible states

BMDS : Ballistic Missile Defense System

boost phase : first phase of missile flight in which it is powered by engines

case : set of assumptions and realities for the single engagement problem

CCD : Central Composite Design

DSP : Defense Support Program

EKV : Exoatmospheric Kill Vehicle

epoch : number of stages left in which actions can be taken

experiment : test in which changes are made to input variables of a process in order to observe the reasons for changes in the output variables [13]

factor : input variable that affects the outcome of the experiment

GMD : Ground-based Midcourse Defense

interceptor : defensive missile designed to destroy incoming offensive missiles

kill assessment : the conclusion by a sensor network of whether an incoming target was destroyed

leakage : allowing a target to pass through defenses and strike its destination

MDP : Markov Decision Process

midcourse phase : second phase of missile flight in which it travels above the atmosphere and releases warheads

MMR : Maximum Marginal Return

observation : perceived state of the system

policy solution : provides the optimal action at each stage for each possible state

POMDP : Partially Observable Markov Decision Process

response : output variable from an experiment

reward : consequence of an action

RV : Re-entry Vehicle

shot : one-time assignment of multiple interceptors to multiple targets

single engagement : shoot-look-shoot opportunity against one wave of incoming targets

SSPK : Single-shot Probability of Kill

stage : partition of a dynamic programming problem in which action must be made

state : condition of the system

target : incoming offensive missile

terminal phase : third phase of missile flight in which warhead falls back into atmosphere

transition : system change from one state to another

UAV : unmanned aerial vehicle

USNORTHCOM : United States Northern Command

USSTRATCOM : United States Strategic Command

value function : piecewise linear combination of alpha vectors over a belief space

WTA : Weapon-Target Assignment

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix B

Notation

B.1 POMDP Formulation

$s \in \mathcal{S}$: state

$a \in \mathcal{A}$: action

$o \in \mathcal{O}$: observation

$T(s, a, s')$: transition model

$O(s, o, a, s')$: observation model

$R(s, o, a, s')$: intermediate reward model

$F(s)$: terminal reward model

$b \in \pi(s)$: belief state

$b(s) = p$: probability of being in state s

$V(b)$: POMDP value function

δ : discount factor for finite horizon POMDPs

B.2 Problem Implementation

β_0 : number of initial interceptors

ρ_0 : number of initial targets

β : number of interceptors remaining in inventory

ρ : number of targets remaining

$SSPK$: single-shot probability of kill

g_1 : number of targets with most interceptors assigned to them

g_2 : number of targets with fewer interceptors assigned to them

n : number of interceptors assigned to each of the g_1 targets

PK_1 : overall probability of no leakage for each of the g_1 targets

PK_2 : overall probability of no leakage for each of the g_2 targets

PK_{sim} : randomly generated number to compare to PK_1 or PK_2 in simulation

h : number of hits from an assignment

P_{mm} : probability of observing a miss given a miss actually occurred

P_{hm} : probability of observing a hit given a miss actually occurred

P_{hh} : probability of observing a hit given a hit actually occurred

P_{mh} : probability of observing a miss given a hit actually occurred

P_{obs} : randomly generated number to compare to P_{hh} or P_{mm} in simulation

m_o : number of observed misses

h_o : number of observed hits

m_a : number of actual misses

h_a : number of actual hits

lb : lower bound on observations

ub : upper bound on observations

w_I : intermediate reward weight

w_{T_1} : terminal reward weight on inventory remaining

w_{T_2} : terminal reward weight on targets remaining

B : number of interceptors remaining in inventory during MMR assignment planning

I : total number of interceptors remaining in inventory for all simulation trials

L : total number of simulation trials that allowed target to leak through defenses

t : number of simulation trials

B.3   Experimental Design

α : distance from center points for axial runs

n_f : number of factorial runs

n_a : number of axial runs

n_c : number of center points

B.4   Results

P_{nl} : probability of no leakage

β_2 : interceptors remaining in inventory after second shot

W : weighted combination of probability of no leakage and inventory remaining

$\Delta_{P_{nl}}$: difference in probabilities of no leakage between Case 2 and 3

Δ_{β_2} : difference in inventory remaining between Case 2 and 3

Δ_W : difference in W between Case 2 and 3

ϵ : residual or error between predicted and actual response

Bibliography

[1] A Historic Beginning: Ballistic Missile Defense System. World Wide Web. Missile Defense Agency. http://www.mda.mil/mdalink/pdf/bmdsbook.pdf.

[2] MDA Glossary. World Wide Web. Missile Defense Agency. http://www.mda.mil/mdalink/pdf/glossary.pdf.

[3] Tenth Interceptor Emplaced for the Ballistic Missile Defense System. page 1. December 2005.

[4] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization.* volume 6 of *Athena Scientific Optimization and Computation Series.* Athena Scientific. Belmont, Massachusetts, 1997.

[5] Gregory H. Canavan. Missile Defense for the 21st Century. *Ballistic Missile Defense Technical Studies Series*, page 165. 2003.

[6] Anthony Cassandra. Tony's POMDP Page. World Wide Web. Computer Science Department. Brown University, http://www.cs.brown.edu/research/ai/pomdp/index.html. 1999.

[7] Anthony Cassandra, Michael L. Littman. and Nevin L. Zhang. Incremental Pruning: A Simple, Fast, Exact Method for Partially Observable Markov Decision Processes. *Uncertainty in Artificial Intelligence*. 1997.

[8] Alvin William Drake. *Observation of a Markov Process Through a Noisy Channel.* PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering, June 1962.

[9] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research.* McGraw-Hill, New York, New York, seventh edition, 2001.

[10] Patrick A. Hosein, James T. Walton, and Michael Athans. Dynamic Weapon-Target Assignment Problems with Vulnerable C^2 Nodes. Technical report, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems, Cambridge, Massachusetts, June 1988.

[11] Michael L. Littman. The Witness Algorithm: Solving Partially Observable Markov Decision Processes. Technical Report CS-94-40, Brown University, Department of Computer Science, Providence, Rhode Island, December 1994.

[12] George E. Monahan. *On Optimal Stopping in a Partially Observable Markov Chain with Costly Information.* PhD dissertation, Northwestern University, 1977.

[13] Douglas C. Montgomery. *Design and Analysis of Experiments.* Wiley, New York, New York, fifth edition, 2001.

[14] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach,* chapter 17, pages 618–652. Prentice Hall, Upper Saddle River, New Jersey, second edition, 2003.

[15] Michael C. Sirak. Year of the Missile. *Air Force Magazine,* 87(1):8, January 2004.

[16] Edward Jay Sondik. *The Optimal Control of Partially Observable Markov Processes.* PhD dissertation, Stanford University, Department of Electrical Engineering, May 1971.

[17] Dean A. Wilkening. A Simple Model for Calculating Ballistic Missile Defense Effectiveness. *Science and Global Security,* 8(2):183–215, 1999.

[18] Eric J. Zarybnisky. Allocation of Air Resources Against an Intelligent Adversary. Master's thesis. Massachusetts Institute of Technology, Sloan School of Management, May 2003.

[19] Nevin L. Zhang and Wenju Liu. Planning in Stochastic Domains: Problem Characteristics and Approximations. Technical Report HKUST-CS96-31, The Hong Kong University of Science and Technology, Department of Computer Science, Clear Water Bay, Kowloon, Hong Kong, 1996.