2006-09

Efficient multiple hypothesis track processing of boost-phase ballistic missiles using IMPULSEÂ©-generated threat models

Rakdham, Bert

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

EFFICIENT MULTIPLE HYPOTHESIS TRACK PROCESSING OF BOOST-PHASE BALLISTIC MISSILES USING IMPULSE©-GENERATED THREAT MODELS

by

Bert Rakdham

September 2006

Co–Advisors: Murali Tummala
Phillip E. Pace

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2006	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Efficient Multiple Hypothesis Track Processing of Boost-Phase Ballistic Missiles Using IMPULSE©-Generated Threat Models			**5. FUNDING NUMBERS**
6. AUTHOR(S) Bert Rakdham			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Center for Joint Services Electronic Warfare Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) Missile Defense Agency			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT

In this thesis, a multiple hypotheses tracking (MHT) algorithm is developed to successfully track multiple ballistic missiles within the boost phase. The success of previous work on the MHT algorithm and its application in other scientific fields enables this study to realize an efficient form of the algorithm and examine its feasibility in tracking multiple crossing ballistic missiles even though various accelerations due to staging are present. A framework is developed for the MHT, which includes a linear assignment problem approach used to search the measurement-to-contact association matrix for the set of exact N-best feasible hypotheses. To test the new MHT, an event in which multiple ballistic missiles have been launched and threaten the North American continent is considered. To aid in the interception and destruction of the threat far from their intended targets, the research focuses on the boost-phase portion of the missile flight. The near-simultaneous attacks are detected by a network of radar sensors positioned near the missile launch sites. Each sensor provides position reports or track files for the MHT routine to process. To quantify the performance of the algorithm, data from the National Air and Space Intelligence Center's IMPULSE ICBM model is used and demonstrates the feasibility of this approach. This is especially significant to the U.S. Missile Defense Agency since the IMPULSE model represents the cognizant analyst's accurate representation of the ballistic threats in a realistic environment. The results show that this new algorithm works exceptionally well in a realistic environment where complex interactions of missile staging, non-linear thrust profiles and sensor noise can significantly degrade the track algorithm performance especially in multiple target scenarios.

14. SUBJECT TERMS KEYWORDS: Multiple Hypotheses Tracking, Boost-phase, Ballistic Missile, Impulse Modeling, IMPULSE, Kalman Filtering, RF Sensors, Association, Simulation, Linear Assignment Problem, LAP	**15. NUMBER OF PAGES** 109
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

EFFICIENT MULTIPLE HYPOTHESIS TRACK PROCESSING OF BOOST-PHASE BALLISTIC MISSILES USING IMPULSE©-GENERATED THREAT MODELS

Bert Rakdham
Captain, United States Marine Corps
B.S., University of Illinois at Chicago, 1997

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2006

Author: Bert Rakdham

Approved by: Murali Tummala
Co-Advisor

Phillip E. Pace
Co-Advisor

Jeffrey B. Knorr
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In this thesis, a multiple hypotheses tracking (MHT) algorithm is developed to successfully track multiple ballistic missiles within the boost phase. The success of previous work on the MHT algorithm and its application in other scientific fields enables this study to realize an efficient form of the algorithm and examine its feasibility in tracking multiple crossing ballistic missiles even though various accelerations due to staging are present. A framework is developed for the MHT, which includes a linear assignment problem approach used to search the measurement-to-contact association matrix for the set of exact *N*-best feasible hypotheses. To test the new MHT, an event in which multiple ballistic missiles have been launched and threaten the North American continent is considered. To aid in the interception and destruction of the threat far from their intended targets, the research focuses on the boost-phase portion of the missile flight. The near-simultaneous attacks are detected by a network of radar sensors positioned near the missile launch sites. Each sensor provides position reports or track files for the MHT routine to process. To quantify the performance of the algorithm, data from the National Air and Space Intelligence Center's IMPULSE ICBM model is used and demonstrates the feasibility of this approach. This is especially significant to the U.S. Missile Defense Agency since the IMPULSE model represents the cognizant analyst's accurate representation of the ballistic threats in a realistic environment. The results show that this new algorithm works exceptionally well in a realistic environment where complex interactions of missile staging, non-linear thrust profiles and sensor noise can significantly degrade the track algorithm performance especially in multiple target scenarios.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my wife, Kendra for her unwavering support and patience throughout my studies at the Naval Postgraduate School.

I would like to thank my thesis advisors, Professor Murali Tummala and Professor Phillip E. Pace, for their insight and guidance in conducting graduate-level research. I would also like to thank Zachary Pace of Duke University for his help in scripting many of the MATLAB® coordinate conversion subroutines; his help was both selfless and invaluable.

I would like to that the National Air & Space Intelligence Center for the use of the IMPULSE© software.

My deepest gratitude is extended to my parents, Chintana and Vaivuthi Rakdham, for the positive roles they have exemplified as well as for their active participation in my early education and extra curricular activities.

Without the above individuals and organization, this work would not have been possible.

This work was supported by the Missile Defense Agency

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The U.S. Defense Department is currently examining many aspects of a defense system for the purposes of providing early warning information and protective measures in the event of an intercontinental ballistic missile attack. In defining the problem we consider a situation where multiple long-range missiles are launched on a ballistic trajectory toward the North American continent. Surface-based sensors are pre-positioned such that they are in an optimal location for detecting and providing position update during the missiles' boost-phase of flight.

This thesis investigates the use of the multiple hypotheses tracking (MHT) algorithm to process track files from a sensor of a surface-based sensor network. In particular, a framework is developed for an efficient form of the MHT to expedite the tracking process. This study applies the algorithm to simulated multiple ballistic missile launches and examines the feasibility and appropriateness of the modified algorithm for this specific application.

This study also makes use of the IMPULSE© simulation tool for generating ballistic missile flight profiles. This software was developed by the National Air & Space Intelligence Center (NASIC). This organization is recognized as the cognizant analysts' representative for threat platforms. IMPULSE© is currently used by engineers and researchers to conduct missile guidance testing and targeting studies. Furthermore, the software enables users to study ballistic missiles at a classified level, i.e., enemy missile data are available so users at the appropriate security levels may study the flight profiles and performances of these particular missiles. Essentially, this study uses a realistic flight model to generate missile trajectories more sophisticated than simple, constant-rate, parabolic motion models.

The IMPULSE© generated flight profiles are then considered from the viewpoint of a sensor network. In particular, the multiple flight trajectories generated through the simulation tool are 'fed' into a sensor model. The study examines the missile's motion as it moves throughout the sensor's field of view and its effect on the sensor's measurement

accuracy. Of interest is the sensor's signal-to-noise ratio and its role in generating measurement error when reporting the missile's location over time. The IMPULSE© missile data, as reported by the sensor (corrupted by noise and precision error), will be used as input to the implementation of the multiple hypotheses tracking algorithm.

A critical drawback of the MHT is its growth in computational requirements. As the number of targets in a scanning region increases, the number of measurement-to-known-targets hypotheses increases exponentially between scans; thus, the standard approach to the MHT is not practical. A modification on the MHT, namely the inclusion of the *linear assignment problem* (LAP), provides an efficient means of identifying the likely measurement-to-target associations. The method for determining the association likelihood probability as outlined by Danchick and Newnam is used in this work and serves as an efficient means to successfully identify correct target-to-next-measurement pairings.

The intent of this thesis research is to exploit the computational efficiency of the modified MHT and demonstrate its suitability for ballistic missile tracking. The simulated sensors used to provide missile position information and the modified MHT are implemented using MATLAB® software. The results herein show that this particular strategy, namely the LAP, used in the implementation of the multiple hypotheses tracking algorithm is successful in correctly tracking numerous—and near-simultaneous—ballistic missile launches in an environment where complex interactions of missile staging, non-linear thrust profiles and sensor noise can significantly degrade the track algorithm performance, especially in multiple target scenarios.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. NATIONAL POSTURE ON MISSILE DEFENSE

In April of 2006, the director of the Missile Defense Agency announced plans to dedicate the missile defense facilities at Vandenberg Air Force Base, Calif. as the "Ronald W. Reagan Missile Defense Site." The ceremony was to honor the 40th President for his vision and commitment in advancing the development of missile defense technologies to protect the United States and its allies from ballistic missile attack.

President Reagan addressed the nation in the spring of 1983 and expressed his concern for the nation's deficiency in effective missile defense measures. He then challenged the American scientific and technical community to explore the methods needed to successfully detect, intercept and destroy ballistic missiles before they could harm American interests. In his speech he said, "I know this is a formidable, technical task…yet, current technology has attained a level of sophistication where it's reasonable for us to begin this effort. It will take years, probably decades of efforts on many fronts. There will be failures and setbacks, just as there will be successes and breakthroughs [1]."

Since that landmark speech, the nation's finest scientists, engineers and missile technology experts have developed and fielded the initial elements of the first missile defense system capable of providing limited defense of all 50 states against long-range ballistic missile attack. The events of September 11, 2001 reminded the world of the resolve exhibited by those who wish to do harm to America by exploiting any means available. An increased proliferation of ballistic missile technology, combined with efforts by extremists to develop warheads capable of inflicting mass casualty and damage, justifies the need to address this threat with the necessary resources to protect the nation [2].

Currently, there are many platforms in the U.S. missile defense program available to address the threat of ballistic missiles. These systems include many variations of intercept-to-kill vehicles designed solely for the purpose of intercepting and neutralizing a ballistic missile. More importantly, there are many subsystems in service, which enable

physical counter measures to successfully intercept ballistic missiles. These systems include powerful land and sea-based radars, early warning infrared satellites, and an integrated command, control, battle management, and communication elements. These elements serve to detect a ballistic missile launch, compute fly-out trajectories, and provide navigational guidance to interceptors in an effort to destroy the threat missile far from friendly territory.

B. MOTIVATION FOR BOOST PHASE DEFENSE

The flight profile of a ballistic missile is comprised of three distinct phases: boost, midcourse, and terminal. Defending against an attack in either of these phases poses several advantages and disadvantages. A midcourse and terminal-phase-based defense is appealing as the requirement for forward deployed logistics, personnel and missile countermeasures, with the exception of early warning sensors, is minimal. Conversely, a boost phase defense system seeks to detect and intercept the ballistic missile in the initial minutes of flight far from friendly territory. These phases of flight are illustrated in Figure 1. The latter approach would mitigate the threat of falling debris or warhead remains from continuing their ballistic trajectory to the intended target—conceivable in a terminal or midcourse interception strategy. This study, thus, seeks to contribute to the boost-phase portion of missile defense protection measures.

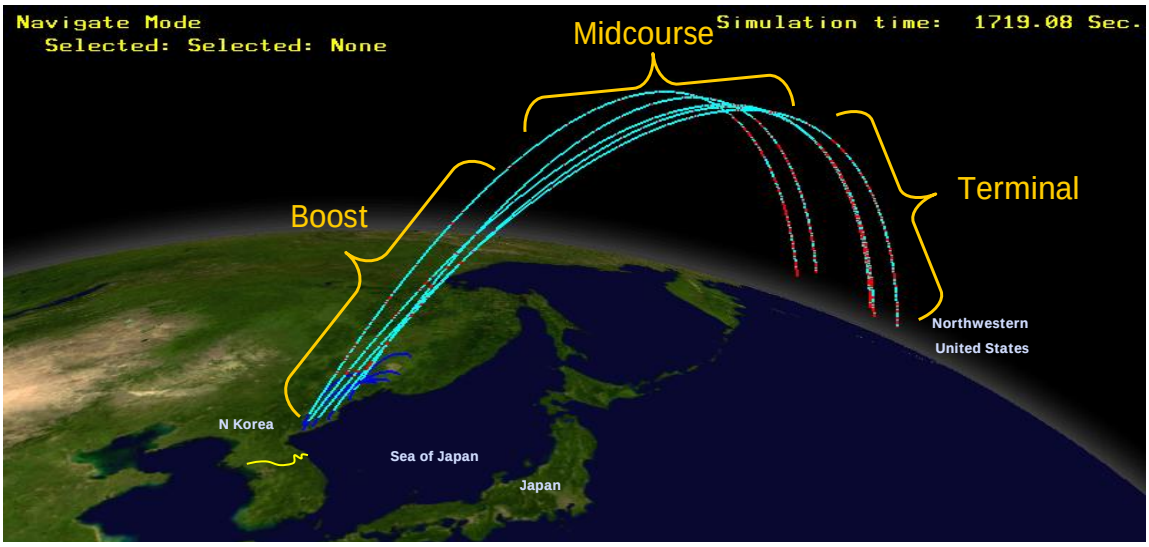


Figure 1. Phases of a ballistic missile flight.

C. PRINCIPAL CONTRIBUTION

This research investigates the application of the multiple hypotheses tracking algorithm in a realistic ballistic missile launch setting. In particular, a strategy for efficiently determining the most likely measurement-to-target association hypotheses, within many possible associations, is studied. The approach used in this work involved the use of computer simulation tools. Specifically, the MATLAB® programming language is used to model the various areas of investigation, including missile track file generation, sensor modeling and the multiple hypotheses tracking algorithm.

The ballistic missile behavior and flight profile—motion—are simulated using a proprietary add-on MATLAB module, specifically, IMPULSE©. Developed by the National Air & Space Intelligence Center (NASIC)—the cognizant analyst threat representative—this software is currently used by engineers and researchers to conduct missile guidance testing and targeting studies. The software provides a model that reflects behaviors commensurate of real-world counterparts [3]. Many physical parameters, such as missile staging, acceleration discontinuities; fuel-flow rates effects on velocity, propellant and missile mass; and atmospheric drag are all dynamically modeled by this software. Essentially, this study uses a flight model that is more sophisticated than the profiles represented in earlier studies. IMPULSE© provides a missile motion-model more complicated than a simple, constant-rate, parabolic profile. The missile flights are generated in the geodetic latitude, longitude and altitude coordinate system. Finally, the software also allows for the possible inclusion of classified missile profiles for future research.

Another area of study in this work is the simulation of the radar sensor network that is used to detect the missile launch and provide measurement updates. An X-Band, low pulse frequency, monostatic radar is considered and is used to generate observations of the missile fly-out. The signal-to-noise ratio characteristics of the sensor are used to introduce precision errors into the sensor-to-missile observations. The computations (and addition) of the sensor measurement-inaccuracies are accomplished in sensor coordinates, i.e., polar coordinates, namely, azimuth, elevation and slant range. The measurements are then converted to the three-dimensional, Earth Centered, Earth Fixed (ECEF) Cartesian

coordinate system for use in the multiple hypothesis tracking algorithm. Both the sensor locations and the missile trajectory are passed to tracking routine in the ECEF coordinate system.

In previous studies, the multiple hypotheses tracking algorithm has been noted as computationally prohibitive as well as difficult to implement. This is due to the measurement-to-target association algorithm being combinatory in nature—leading to an explosive growth of potential association hypothesis [4]. Furthermore, it has been argued that other variations of its implementation try to make feasible associations from infeasible hypothesis from previous scans—making the MHT further inappropriate for such an application. The method for determining the association likelihood probability as outlined in [4] is investigated in this study and serves as an efficient means to expediently identify correct target-to-next-measurement pairings. As a result, this method enables the MHT algorithm to successfully track multiple missiles and their staging events.

The linear assignment problem approach is taken from [4] provides an efficient method to locating *N*-best measurement-to-target association pairs. As a result, the best overall association hypothesis (per scan frame) may be found. An advantage, according to the authors, is that the approach is non-combinatoric and avoids carrying unlikely association hypotheses forward for later scan computations. Components of an extended Kalman filter are used to help in the computation of the measurement-to-target likelihood as well as to provide track smoothing when displaying the individual missile tracks. Again, the management of track files, computation of associations and hypotheses, and generating the estimated missile tracks are also accomplished in MATLAB. Furthermore, the tracking algorithm is implemented with respect to an individual sensor's position. Computation of innovation matrices are performed in sensor polar coordinates, i.e., azimuth, elevation and range. The position of the missile, upon successful tracking of the algorithm, is reported in a Local Center Local Vertical (LCLV) (also referred to as north, east and up) coordinate system centered about the radar platform. Tracked missiles are then converted to geodetic latitude, longitude and altitude coordinates for display.

The key contribution of this study is the success the multiple hypotheses algorithm has displayed while tracking multiple ballistic missiles—in the initial minutes

of flight—as they are detected by one or more sensors. Moreover, the algorithm allows for initialization of new contacts in the scanning region, e.g., upon the missile shedding lower-stage components, the algorithm is successful in tracking objects that were not present in previous scans. Overall, the algorithm provides one method of managing ballistic missile track files within a sensor network.

D. THESIS ORGANIZATION

Chapter II introduces the IMPULSE© software and discusses the generation of the ballistic missile flight profiles. This software enables the user to load either predefined generic or classified missile models, simulate its flight, and display the profile in both a text and graphical format. The most important advantage to using IMPULSE© is the high-fidelity flight information obtained with the aid of the software. IMPULSE© recognizes many factors when performing flight calculations; it considers both the model's parameters as well as its interaction with the physical world. The physics of ballistic missile flight have been exhaustively modeled in this software; thus, it will serve to support the study.

Chapter III discusses the sensor platform used in this study, namely, surface-based radar. The mathematical relationships determining the performance of this particular sensor are presented. In particular, the radar system's signal-to-noise and its affect on measurement precision are investigated. The results of this chapter will create "sensor-reported" sensed observations, which will vary slightly from true object positions as generated by the missile modeling software. Once the reported positions are obtained on each missile fly-out, they are used to generate a large "radar database", which is further used as the input file to the multiple target tracking module.

Chapter IV presents the main area of effort. Several terms are introduced such as *scans*, *measurements*, *associations* and *hypotheses* to aid in the discussion of multiple-target tracking problem. A comparison of single and multiple-target tracking is also made and the problems associated with the latter are outlined. A (concatenated) key expression for each inter-scan association hypotheses probability is presented and the shortfalls brought about by the direct implementation of this equation and approach are also discussed. The linear assignment problem solution is then introduced and is shown to form the correct pairings of targets to measurements within each scan. The overall

multiple hypotheses tracking algorithm, developed in this study, is applied to the measurement data generated in Chapters II and III. Performance plots are shown for the algorithm, e.g., the algorithm's precision in reporting the missile's position throughout flight. The chapter concludes by discussing the computational complexity of the algorithm.

Chapter V summarizes the contributions of this work and provides remarks on the appropriateness of the algorithm for sensor networks. Suggestions for follow-on research are recommended. Appendix A describes a step-by-step use of the IMPULSE© GUI (as used in this study). A flow chart is provided in Appendix B for all MATLAB source code used to perform the simulation. Finally, in Appendix C, a review of the extended Kalman filter equations used in this study are presented.

II. GENERATING BALLISTIC MISSILE PROFILES

This chapter details the generation of the ballistic missile flight profiles which will serve as input data to the sensor models and tracking routine, the subject of Chapters III and IV, respectively. The modeling of the missile is achieved through using IMPULSE©, a specialized add-on toolkit that operates in conjunction with MATLAB©. The purpose for using this modeling software is to simulate multistage, intercontinental ballistic missiles capable of reaching the North American continent from East Asia. Most importantly, the software will provide the study with missile flight profiles that exhibit behavioral characteristics comparable to real-world platforms—subject to physical, gravitational and atmospheric effects. Figure 2 below shows the major functional blocks throughout this document. The highlighted block denotes the main subject of this chapter. Inclusion of the IMPULSE-generated missile flight profiles is beneficial, as it serves to authenticate the practical nature of the data used in the research and validate the applicability of the multiple-target tracking routine—the focus of Chapter IV.

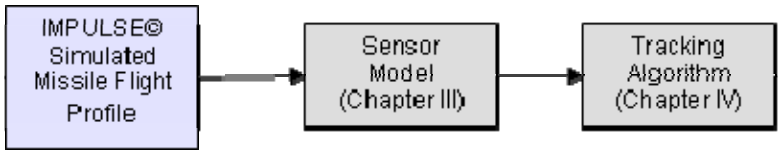


Figure 2. Major functional blocks to be examined in this study.

A. MISSILE MODELING CONSIDERATIONS

Today's ballistic missiles are complex machines and many physical parameters must be correctly modeled in order to realistically simulate a high-fidelity flight profile. Unlike fixed-wing or rotary aircraft, to counter the force of gravity, these machines develop upward thrust by rapid expulsion of solid or liquid fuel in the form of hot gases. Moreover, a missile differs from other vehicles in that it carries both its fuel and an oxidizer internally. This enables the missile's propulsion system to generate thrust within the Earth's atmosphere as well as in the vacuum of space.

The major components of a modern missile assembly are a rocket motor or engine, propellant consisting of fuel and an oxidizer, a guidance system, a payload such

as a warhead, and a frame to hold the components. Accurately modeling the flight dynamics of a missile requires the consideration of changes in all of these physical components throughout its flight. For example, consider a multistage missile's flight profile. As propellant is consumed from liftoff to initial stage burnout, the overall mass of the missile changes accordingly; in turn, this affects the performance of the missile during boost phase. Furthermore, simulating a constant thrust and acceleration throughout the missile trajectory are also insufficient to precisely predict the missile's flight path. In fact, it will be shown that the thrust generated by the motors is indeed a non-constant factor, further complicating the task of modeling of these machines. Still yet, a realistic flight profile must reflect changes in missile physical structure, e.g., shedding of coupling components as well as spent stages and payload—warhead—release.

There are also external factors that govern the performance of a ballistic missile. Drag induced by the atmosphere on the missile is another complicated parameter requiring consideration on the performance of flight. During boost phase, the missile experiences the effects of the dense air at sea-level. Modern missile thrust control systems, in fact, generate less thrust on lift off, allowing the missile to climb into the thinner atmospheric layers before maximum thrust is applied. This leads to greater fuel efficiency while giving the missile a longer range projection [5]. Once again, this is a reminder that neither fuel consumption nor the application of thrust is a linearly changing variable. Another parameter that must be considered in the missile model is the vehicle's performance upon entering the vacuum of space. Suborbital flight is also a complicated parameter to accurately model. Gravitational forces acting on the missile must also be modeled into the flight profile. The IMPULSE© toolkit, as presented in the next section, is used in this study to aid in the simulation of realistic missile flight profiles.

B. MODELING THE MISSILE FLIGHT DATA WITH IMPULSE©

The actual design of the ballistic missile as well as accounting for all environmental factors is beyond the scope of this study. Undoubtedly, modeling flight performance to reflect accurate missile physical parameters, interaction with the atmosphere, and modeling correct gravitational effects as well as suborbital flight parameters is a daunting task. This study relies upon defining the missile threat profiles with the aid of IMPULSE© to model the ballistic missile threats and generate each

respective—high-fidelity—flight profile [3]. The software is an add-on module for MATLAB and, at the time of this work, IMPULSE Version 1.0 was available for research use. Lastly, this particular release of the software was only compatible with MATLAB R14, Service Pack 1.

1. Scenario Parameters

This study is concerned with tracking multiple missile launches; thus, six fictitious two-stage missile flight profiles are generated with the IMPULSE software, namely, *Ballistic Missile 1*, *Ballistic Missile 2*,…, and *Ballistic Missile 6*. The acronyms *BM1*, *BM2*,…, *BM6* will be used for brevity. In the IMPULSE© *Boost Analysis* graphical user interface (GUI) the *Missile Type* and *Loaded Missile* parameters are set to "SampleUnclassModels" and "(U) BOOST Unclassified Sample," respectively.

A detailed guide to the operation of the IMPULSE software and its GUIs are covered in Appendix A, Section 1. The user simply sets specific missile launch-scenario parameters or selects predefined models from the IMPULSE© data base. Classified models are also available for use in enhancing potential future studies. In this research, generic missile models are selected to keep the nature of the research unclassified. On the other hand, a potential real-world launch scenario is examined.

There are four suspected North Korean long-range ballistic missile launch sites represented in the simulation: BM1 and BM2 are launched from the *No-Dong* missile facility located at N40°51'17" E129°39'58," BM3 originates from *Toksong-gun* base at N40°25'00" E128°10'00," BM5 starts from *Yongo-Dong* at N42°11'47" E130°11'48, and BM5 and BM6 are launched from the *Mayang* facility at N40°00'14" E128°11'04" (Unclassified). The coordinates of these facilities, given in the World Geodetic System 1984 (WGS84) coordinate system, are used to initialize the *Launch Latitude*, *Longitude* and *Launch Altitude* parameters within the IMPULSE © "Config Scenario" block of the analysis.

The remainder of the simulation scenario parameters is set such that the ballistic missiles are launched from the given coordinates and assume flight trajectories that threaten the western United States. The *Kick Angle* commands the missile to the required angle-of-attack immediately after launch to assume the ballistic profile. Furthermore, the

aiming azimuth defines the missile heading after to achieve the desired impact point. Launch azimuths of 31°, 33°, 35°, 29°, 31° and 35° are used for BM1 through BM6, respectively, to impact the western U.S. [6]. The *Final Stage Burn Time* of the missile is left at the default value for the selected model—65.01 seconds.

Environment and physical parameters are adjusted as well. The *Rotation* parameter enables the program to simulate the earth's rotation while the missile is in flight, which can influence the desired impact point. Still yet, the Earth is not an ideal sphere and can be accurately modeled as an oblate shape in the simulation. *Atmosphere* and *Gravity* parameters are left as default values [7]. Table 1 reflects the above discussion and summarizes the scenario settings for *BM1*—the missile launched from the No-Dong missile facility. A comprehensive list of tables for *BM2* through *BM6* parameters can be found in Appendix A, Section 2.

Parameter	**Value**
Missile Type	"SampleUnclassModels"
Loaded Missile	"(U) BOOST Unclassified Sample"
Launch Facility	*No-Dong*
Launch Latitude,	N 40°51'17"
Launch Longitude	E 128°11'04"
Launch Altitude	20 m
Launch Azimuth	33°
Kick Angle	11.5°
Final Stage Burn Time	65.01 (default)
Rotation [Earth]	Rotating
Shape [of Earth]	Oblate/ Spherical
Gravity	WGS84

Table 1. Ballistic Missile 1 scenario parameters used in the IMPULSE© Boost-Phase Analysis GUI.

Once the parameters are entered into the GUI, the user selects the "Simulate" command in the interface window. The software computes the fly-out telemetry and populates the MATLAB workspace with the missile flight data. The generated data are examined and the missile position is extracted and plotted for visualization. The flight

profile for *BM1* can be seen in Figure 3. This graphical representation served as a test "flight" of the vehicle and helped to verify that the chosen model and launch parameters fulfill the desired scenario. Each missile successfully assumed a north-easterly trajectory and, in approximately 26 minutes, covered the distance between East Asia and the northwestern region of the United States; impacts were recorded in desired locations. Each missile also jettisons its spent lower stages upon burnout. Flight data for the falling booster debris were made available and will also be included in the study; thus, at a given time, there may be as many as 12 objects of interest.

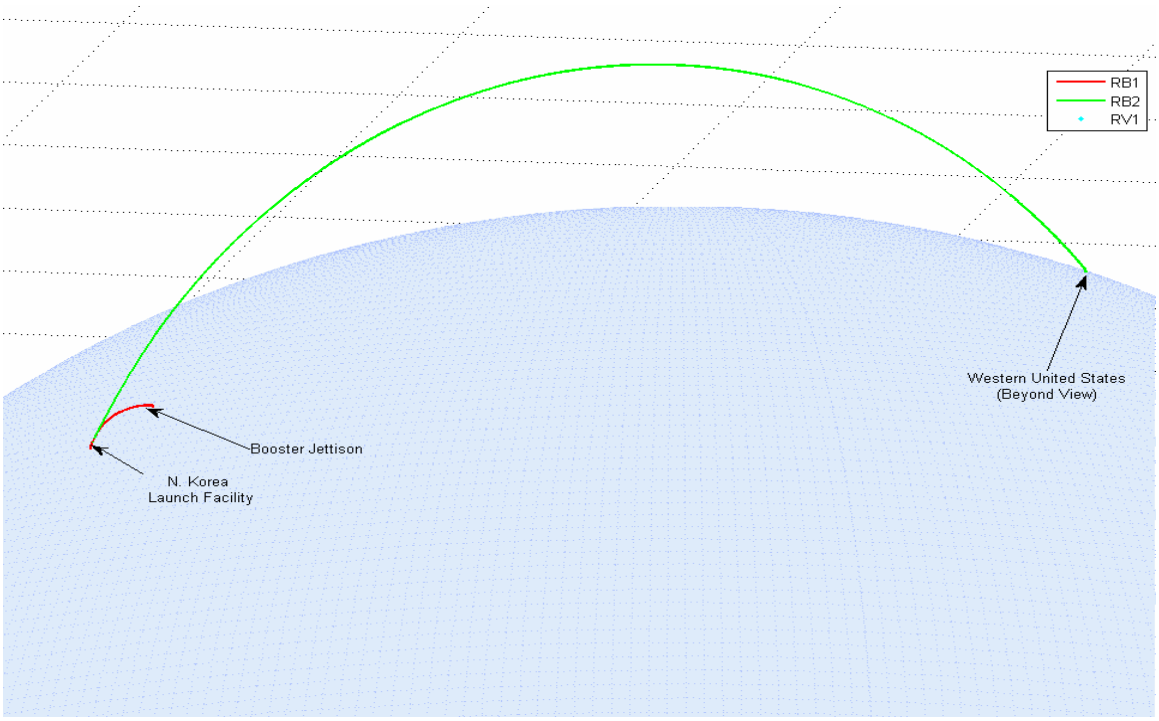


Figure 3. Missile "test" flight for BM1 to validate the scenario parameters. The missile is launched from a location in North Korea and performs a transcontinental flight where it impacts the Northwestern United States.

Lastly, a sample of the ballistic missile flight information, as generated by IMPULSE and outputted to the MATLAB command window, is provided in Appendix A for the reader to review. Additionally, the text files recording the missile fly out positions are included in the same appendix.

2. Missile Flight Data

This section examines several of the complex missile flight parameters simulated by the IMPULSE software. The first parameter to be discussed is the missile's velocity over the duration of the flight. Figure 4 shows the velocity plot of the upper and lower stage of Ballistic Missile 1 for the duration of its flight. The information reveals that the missile's velocity is not constant. A steady increase in speed can be seen during the boost phase of flight. The lapses in velocity are observed at Time = 65 and 130 seconds—at stage jettison and upper stage burnout, respectively. Maximum values are reached just prior to exhaustion of the propellant in both the booster and upper stage and again just before impact. The m*idcourse* segment is where the missile is in the outer atmospheric phase of its flight. The a*poapsis* is the peak of the trajectory while the *terminal* segment denotes the missile descent back into the atmosphere to impact.

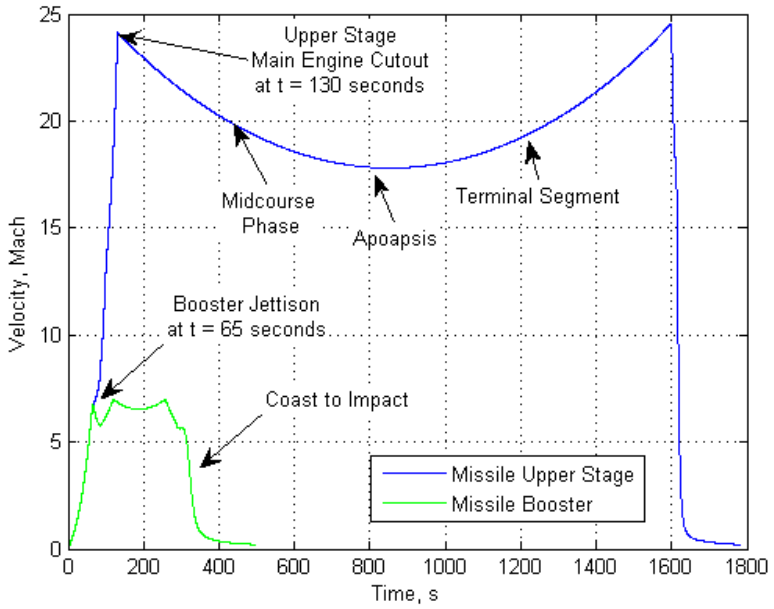


Figure 4. Ballistic Missile 1 velocity during the boost-phase portion of flight for an unclassified two-stage ballistic missile.

Figure 5 captures the induced atmospheric drag as the vehicle ascends during lift off and, once again, during the terminal phase of flight. The plot indicates that the missile is experiencing the greatest atmospheric drag while in the more dense levels of the

atmosphere—as expected. The drag drops to near-zero values as the missile enters the vacuum of space. The second green spike, at approximately 320 seconds, indicates that the booster has re-entered—and is again experiencing interface friction—the atmosphere. The upper stage and warhead continue their ascent in the low-drag environment until they also re-enter and impact the target.

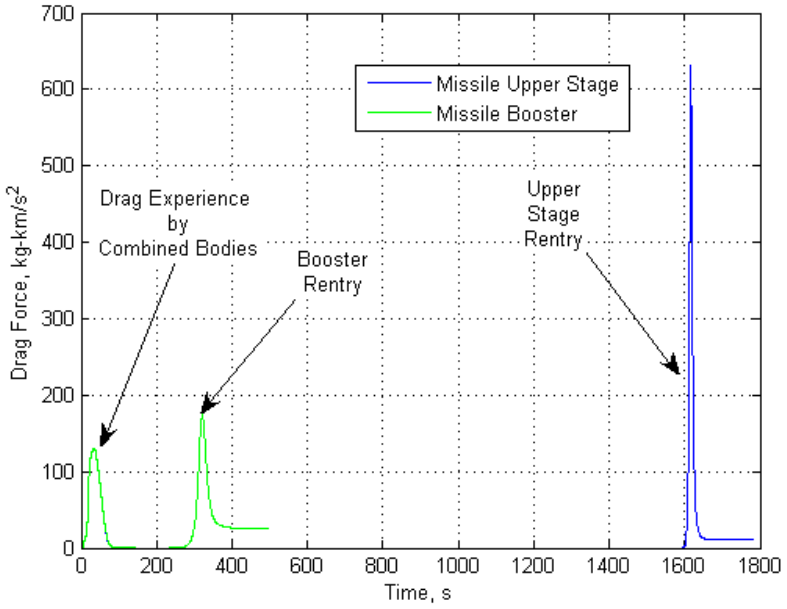


Figure 5. BM1 induced drag force due to vehicle's interaction with the atmosphere.

Another variable IMPULSE© includes in its model is the change in the missile's mass as fuel is consumed and as changes in the missile's physical composition occur—staging. Figure 6 presents the decrease in the missile's mass over the flight duration. During the initial minute-and-four-seconds of flight, IMPULSE models the mass of the missile as a single element that includes both the lower and upper stage masses. The missile data then reflects the expulsion of the child components, i.e., the booster at 65 seconds when the staging event takes place. The introduction of the blue line at 65 seconds indicates that the mass of the upper stage now exists within the simulation and is an independent element to the discarded booster can. It is also observed that the mass of the booster abruptly drops as the mass value no longer includes the mass of the upper

stage component. Next, the mass of the upper stage is seen slowly decaying between 65 and 130 seconds as the fuel is consumed. The final change in mass occurs at 130 seconds and implies the release of the missile warhead.

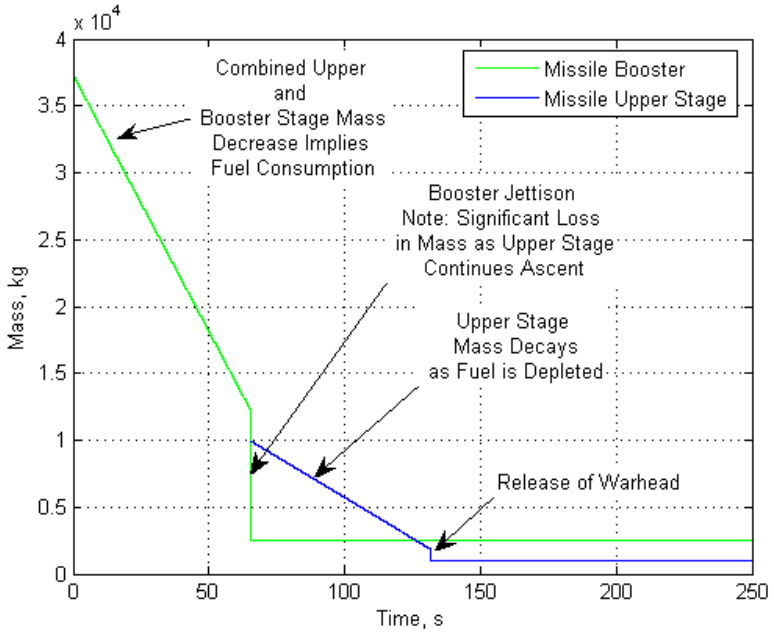


Figure 6. Measurement of BM1's, an unclassified two-stage missile, mass during the boost-phase portion of flight.

IMPULSE© further includes in its simulation the rate at which its propellant is consumed and appropriately computes the remaining fuel level within the missile. Clearly, these parameters interact and obviously contribute to a majority of the total mass of the missile as discussed above. The fuel level remaining is a parameter that is dependent upon the engine propellant flow-rate. The plot in Figure 7 shows that for a large flow rate, as seen in the boost stage—represented by the green line—the remaining propellant in the upper stage decays rather rapidly; hence, the steep slope at a constant flow rate of approximately 385 kg/s. The fuel is exhausted at 65 seconds in flight, which coincides with the jettison event.

On the other hand, the upper stage data—indicated by the blue line—denotes a slower fuel-flow rate of approximately 120 kg/s and, thus, this stage has a longer burn time. The fuel is completely exhausted at 130 seconds of flight, which is consistent with

the data given in earlier figures. Furthermore, the differences in fuel flow-rate also offer insight into the stages of flight each represent. Recall that the boost stage must lift the vehicle off the launch pad starting from zero inertia value as well as through the more dense parts of the atmosphere; as a result, a higher flow rate is required. Conversely, the upper stage is nearly in the outer atmosphere and seconds from entering suborbital flight. The interaction of missile fuel, propulsion flow and mass are satisfactorily modeled by the IMPULSE© software.

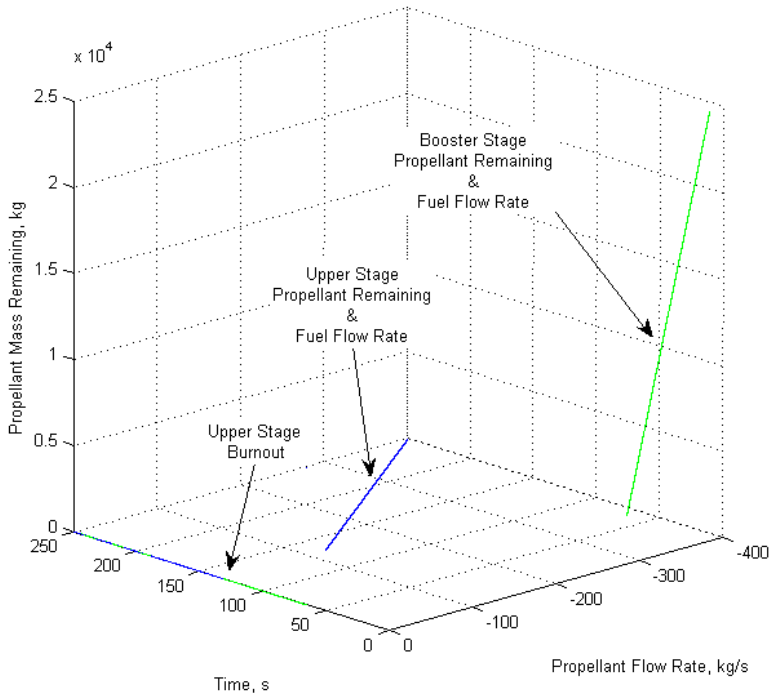


Figure 7. Propellant remaining and fuel flow-rate measurements during the boost-phase portion of flight for an unclassified two-stage ballistic missile.

Yet another important parameter IMPULSE© is capable of modeling is the missile's thrust magnitude during flight. This parameter is recorded and presented in Figure 8. As previously mentioned, the missile's boost stage can be observed producing less thrust during the initial ascent period. This characteristic is commanded by the guidance and engine control system to increase fuel efficiency while the rocket body

experiences the greatest aerodynamic pressure in the lower atmosphere. Conserving fuel during this phase extends the missiles range [5], thus enabling the missile to reach targets at greater distances.

A parameter used to classify a missile's engine performance is its impulse. The impulse, sometimes called total impulse, is the product of thrust, in Newtons, and the effective firing duration. From Figure 6, the impulse of *BM1* may be estimated by taking 9.5 N and multiplying it by the duration of flight—70 seconds. This results in a total impulse calculation of 665 N-s. In comparison, the upper stage only exhibits a total impulse of approximately 300 N-s [5].

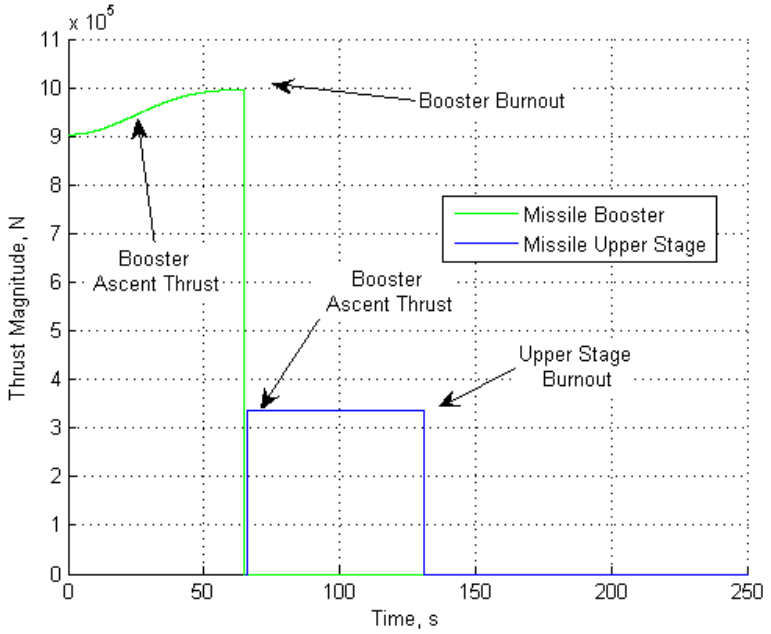


Figure 8. Missile thrust over time measurements during the boost-phase portion of flight for an unclassified two-stage ballistic missile.

Impulse further simulates the missile-body acceleration. The plot for the acceleration parameter is reserved for Chapter IV (see Figure 26). It is presented later as it is used to support the discussion on the tracking algorithm. In short, the acceleration parameter caused complications in the implementation of the multiple hypotheses tracking algorithm during the testing phase of this study. It will be shown how the

booster-jettison staging event introduces unexpected acceleration 'jerk' and can complicate the measurement-to-target association process. These terms will also be clarified further in Chapter IV.

Clearly, the IMPULSE© tool recognizes numerous parameters that define a ballistic missile's flight. The information provided above supports and validates this software as an excellent source of high-fidelity flight data. The study at hand includes a modeling method that produced far more realistic data than in previous studies [8][9]. As seen throughout out this chapter, there are many missile physical parameters that must be modeled to accurately represent the missile's flight profile. Accurate simulation of the interaction between these parameters ensures the generation of a high-fidelity flight trajectory.

3. IMPULSE Output

The files listed in Table 2 contain the IMPULSE© output of each missile's flight as exported by the General Write Utility (GWU). Information regarding the use of this tool can be found in Appendix A. These files will serve as the input data to the sensor models and tracking algorithm—the subject of Chapters III and IV.

Missile Profile	**File**
Ballistic Missile 1 Upper Stage data	*Flt1UStage.txt*
Ballistic Missile 1 Lower Stage data	*Flt1LStage.txt*
Ballistic Missile 2 Upper Stage data	*Flt2UStage.txt*
Ballistic Missile 2 Lower Stage data	*Flt2LStage.txt*
Ballistic Missile 3 Upper Stage data	*Flt3UStage.txt*
Ballistic Missile 3 Lower Stage data	*Flt3LStage.txt*
Ballistic Missile 4 Upper Stage data	*Flt4UStage.txt*
Ballistic Missile 4 Lower Stage data	*Flt4LStage.txt*
Ballistic Missile 5 Upper Stage data	*Flt5UStage.txt*
Ballistic Missile 5 Lower Stage data	*Flt5LStage.txt*
Ballistic Missile 6 Upper Stage data	*Flt6UStage.txt*
Ballistic Missile 6 Lower Stage data	*Flt6LStage.txt*

Table 2. Output files from IMPULSE General Write Utility

Table 3 details the file format for each text file listed in Table 2. The simulation time is given in the first column. The latitude and longitude are represented in the second and third columns. The forth and fifth columns give the altitude in meters and the thrust magnitude in Newtons, respectively.

Time, s	Geodetic Latitude, rad	Geodetic Longitude, rad	Altitude, km	Thurst Magnitude, N
0	0.70969	2.2372	0.02	2.03E+05
1	0.70969	2.2372	0.027182	2.03E+05
2	0.70969	2.2372	0.048894	2.03E+05
3	0.70969	2.2372	0.085382	2.03E+05
4	0.70970	2.2372	0.13561	2.03E+05
5	0.70970	2.2372	0.20043	2.04E+05
6	0.70970	2.2372	0.27988	2.04E+05
7	0.70971	2.2372	0.37395	2.04E+05
8	0.70971	2.2372	0.48265	2.04E+05
9	0.70972	2.2372	0.60599	2.04E+05
10	0.70973	2.2372	0.74394	2.05E+05

Table 3. Column format for each Ballistic Missile file listed in Table 2 as the output from the General Write Utility GUI.

The ballistic missile positions in these files are given in the WGS84 geodetic coordinate system. Furthermore, the latitude and longitude values in these files are in units of radians with respect to the center of the Earth. The data can be easily converted to North/South, Degrees/Minutes/Seconds (N/S DD/MM/SS) and East/West, Degrees/Minutes/Seconds (E/W DD/MM/SS) format by using the *rad2deg.m* function in the MATLAB mapping toolbox.

In summary, this chapter described the generation of the ballistic missile fly out profiles for analysis in follow-on chapters. IMPULSE© enables the user to load either predefined generic or classified missile parameters, simulate its flight and display the profile in both a text and a graphical format. The high-fidelity flight profile along with the position information obtained through the simulation tool is the primary advantage presented when using the IMPULSE© simulation software. IMPULSE© considers many factors when performing flight calculations, including the model's physical performance parameters as well as its compliance with the laws of physics when interacting with the

real-world. As a result, the missiles in the simulation display flight performances that one would expect a real-world counterpart to exhibit. In this work, the physics of ballistic missile flight have been modeled using this software. Six missiles were generated with potential real-world launch consideration and will be studied in the remainder of this work. Next, Chapter III presents the sensor models used to detect the launch and provide constant position updates of ballistic missiles.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SENSOR MODELS

In order to detect the threat of a ballistic missile attack, there are a multitude of sensors available to provide early warning information if such an event were to occur. Two particular types of sensors are appropriate for the boost-phase missile tracking problem. In this chapter, we investigate surface-based radio-frequency (RF) sensors. Furthermore, the mathematical relationships, which determine the performance of each sensor, will be examined. The results will enable us to take the IMPULSE© missile flight information and introduce appropriate error as a means to model the sensed position information as reported by each sensor.

A. RADAR

Radar systems use modulated waveforms and directional antennas to transmit the electromagnetic energy into a region of interest to scan for objects. Targets that are in the specified search volume will reflect a small portion of this energy back to the transmitting sensor. Information, such as range, bearing and velocity, can be extracted from the returned pulses of energy. In this study, the signal-to-noise ratio (SNR) of the radar as a function of target-to-sensor range is the parameter of interest. This information will be used to simulate the sensor's error in angular and range measurements when reporting the missile's position. Figure 9 depicts a simple schematic showing the input of the IMPULSE©-generated missile flight profile into a sensor model block. Noise is generated as a function of the signal-to-noise ratio and added to the flight data. The resulting 'noisy' trajectory will later be used for the tracking experiment in Chapter IV.

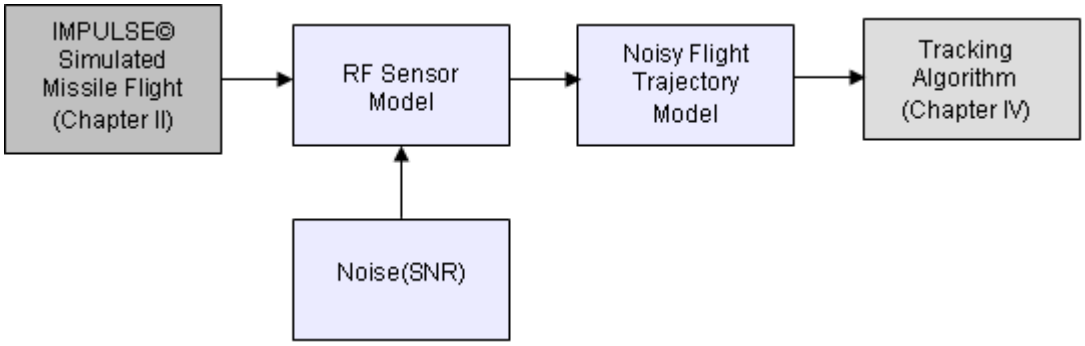


Figure 9. Modeling of the RF sensor by adding noise to the "true" missile flight trajectory as generated by IMPULSE©.

This research considers the use of the following parameters: the radar is a low pulse repetition frequency (LPRF) sensor that operates in the X-band. The high angular and range resolution of these particular sensors provide a capability for detecting and tracking ballistic missiles during boost phase [10]. For the simulated ballistic missile scenario, the positions for the two sensors, RF1 and RF2 are as follows: RF1 is located at N 40°50 E 131°46 while RF2 is positioned at N 41°50 E 132°46. Both sensors lie within the Sea of Japan and are in an optimal position for viewing a ballistic missile launch [6]. The simulation, depicted in Figure 10, is run for a single ballistic missile that is launched from North Korea towards the west coast of the United States.

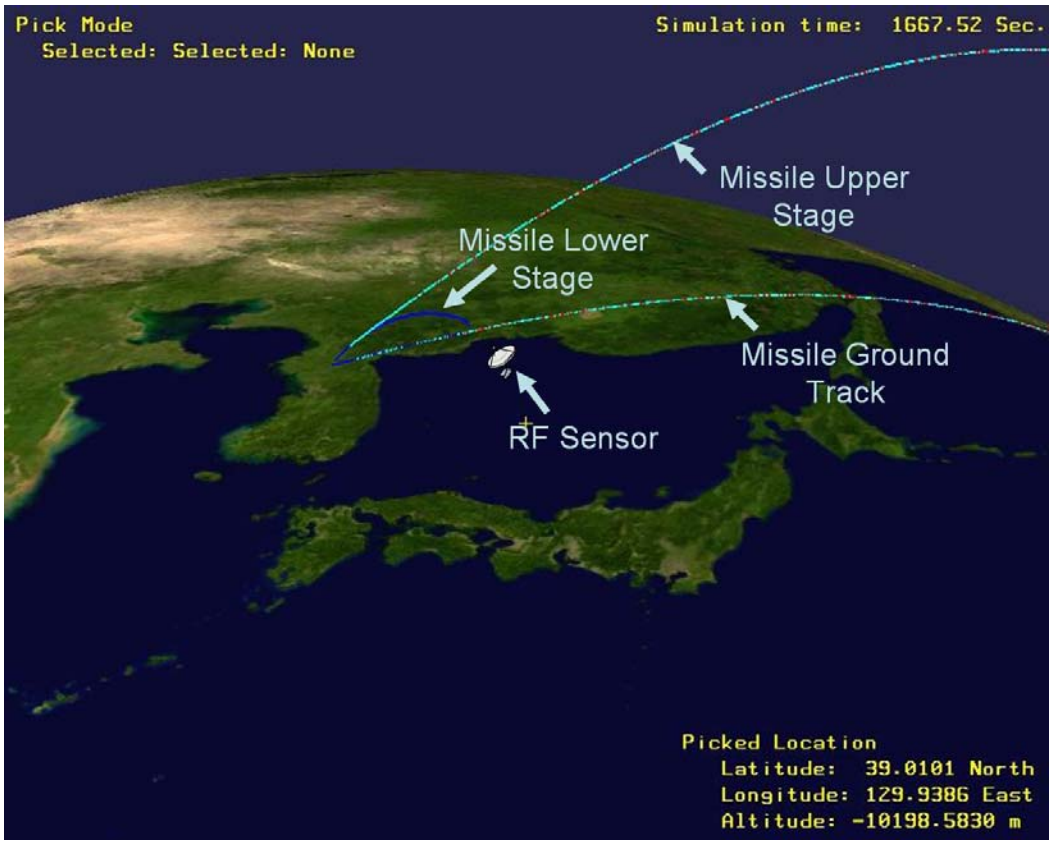


Figure 10. RF sensor placement in relationship to the launch of a test ballistic missile. [Map generated using the IMPULSE© Blue Marble Viewer GUI]

Two tracks of the missile are clearly seen in the figure: the lower, jettisoned, booster stage and the upper stage as it continues its ascent toward its intended target. The missile ground track provides clarity on the missile position with respect to the ground

when testing the simulation. The missile was generated in a similar fashion to the six flight profiles discussed in Chapter II; however, the launch point for this particular fly-out is arbitrary as it is still within the northern portion of the Korean peninsula. This missile served as a test case for the sensor under study.

1. Sensor Signal-to-Noise Ratio

The sensor equations are now developed. The next few paragraphs cover the radar equation followed by two relationships used in the simulation of the sensor to introduce measurement noise. The RF sensor under consideration uses the same antenna in transmitting and receiving its signal, i.e., monostatic operation. Furthermore, the number of pulses integrated is assumed to be singular; thus, a monopulse-radar is implied. The SNR (S/N) for a single-pulse radar is given by [11][12]

$$S/N = \frac{P_t G_t^2 \tau \lambda^2 \sigma}{(4\pi)^3 k T_0 F R^4} \tag{3.1}$$

where P_t denotes the peak transmitted power, G is the antenna gain, σ is the radar cross section of the object , τ is the compresses pulse width, λ is the wavelength, F is the noise factor, k is Boltzmann's constant, T_0 is the system temperature, widely accepted as 290° K, and R is the range to the target. Generally, the radar losses are also considered but are assumed to be zero for this study. The sensor-to-target slant range, R, is obtained in the simulation by using the MATLAB *elevation* function. This returns the actual target-to-sensor range.

Figure 11 shows the SNR computed at radar sensor 1 (RF1) as it observes the upper and lower stages of the ballistic missile in our simulation. An increasing SNR is observed as the missile passes near and relatively over the sensor during the boost phase of flight. This is observed in the first 100 seconds of the missile's flight. Staging occurs at 65 seconds where a second SNR value—the green line—is initiated.

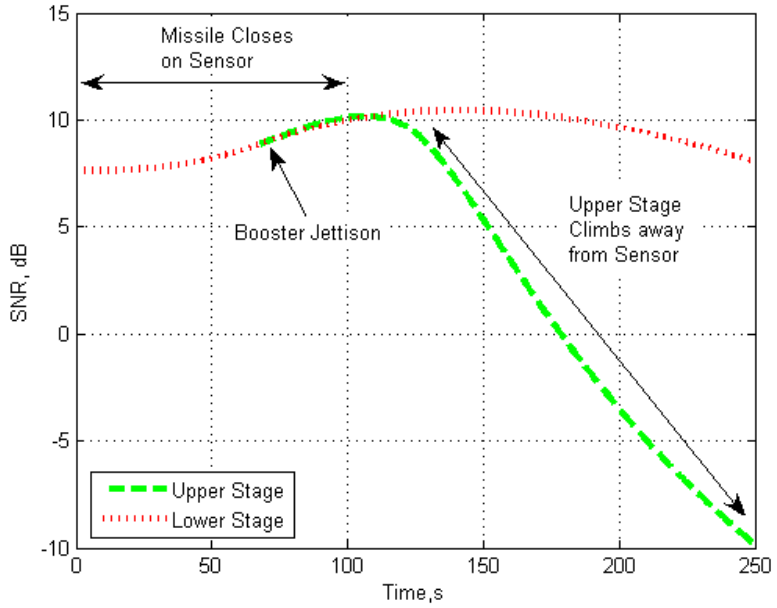


Figure 11. Radar Sensor 1 SNR versus time while observing BM1.

The SNR value observed on the booster—the red plot—remains at a nominal value of 8 dB as it never leaves the sensor's immediate field of view. Conversely, the upper stage of the missile continues its flight away from the sensor. The SNR is inversely proportional to the forth power of the range; as a result, the SNR decays rather rapidly beyond 130 seconds of the simulation. As the SNR changes throughout the course of the missile trajectory, the values will be used in the computation of the error in the reported angular and range measurements.

2. Sensor Measurement Precision

There are several sources of measurement error for an RF sensor. One type, *bias errors*, is caused by simple mis-calibration of the radar prior to operation. Mechanical errors, such as *servo-induced-error*, are another type and occur when the voltages required for the movement of the antenna pedestal have small inaccuracies. Refraction of the signal as it passes through the atmosphere—*propagation medium errors*—may cause range and elevation angle errors is another type of error [12]. One particular cause of measurement error is due to noise which is attributed to the finite signal-to-noise ratio; this study examines the latter problem.

The *time-of-arrival* error is one such noise-induced measurement. It is caused by interference contaminating the returning echo pulse. As a result, the center of the signal occurs at a time other than that of the actual center of the pulse. Figure 12 depicts the relationship between the signal and the contaminating noise.

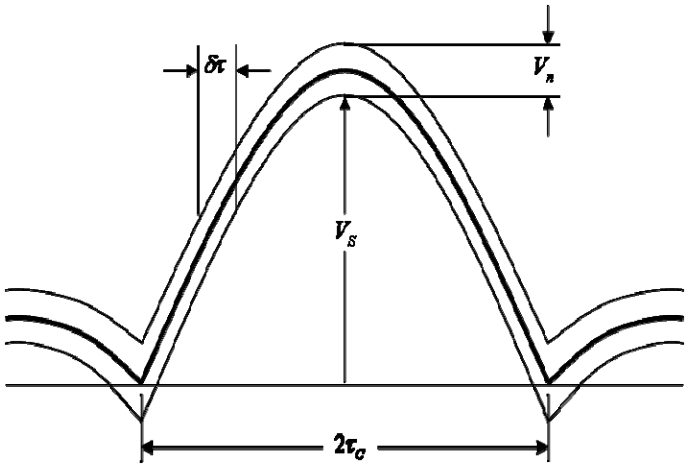


Figure 12. Range noise error due to interference contaminating the return pulse (After [12]).

The RMS time-of-arrival error, $\delta\tau$, is taken as the average slope between zero and the signal voltage peak value, V_S. The signal, noise and time error relationship is given by the proportion [12]

$$\frac{\delta\tau}{V_n} = \frac{2\tau_C}{V_S}$$

where V_n is the RMS noise voltage, and $2\tau_C$ is the compressed pulse width (provided compression is used); otherwise, it is simply the pulse width. The above proportion is rearranged for the time of arrival error, $\delta\tau$, and the power $(S/N)_1$ is substituted for the ratio of voltages such that

$$(S/N)_1 = 1/2(V_S/V_n)^2$$

thereby yielding the expression for noise error

$$\delta\tau = \frac{2\tau_C}{\sqrt{2(S/N)_1}}$$

The above is a simplistic approach since range measurements are never made with a single pulse. For a multiple-pulse approach, the time-of-arrival error is given by

$$\delta\tau = \frac{2\tau_C}{\sqrt{2\, f_{PRF}\, T_d \left(S/N\right)_1}}$$

where f_{PRF} is the pulse repetition frequency of the pulsed radar and T_d is the dwell time or time over which data is gathered. The number of coherently integrated pulses, M_i, may be substituted for the product of the PRF and T_d; thus

$$\delta\tau = \frac{2\tau_C}{\sqrt{2\left(S/N\right)_1 M_i}}$$

The range error is found by substitution of the following relationship. By letting $\sigma_R = v_P\, \delta\tau/2$, where v_P is the propagation velocity. By letting $v_P = c$, the speed of light, the following is obtained [13]

$$\sigma_R = \frac{c}{2B}\frac{1}{K\sqrt{\left(2\left(S/N\right)_1 M_i\right)}}$$

where $B = 1/\tau_C$ and $1 \le K \le 2$ is an error-slope coefficient. Finally, letting $S/N = \left(S/N\right)_1 M_i$. We have

$$\sigma_R = \frac{c}{2B}\frac{1}{K\sqrt{\left(2\left(S/N\right)\right)}} \tag{3.2}$$

A similar approach is used for computing angular (azimuth and elevation) measurement errors. The next set of relationships is presented in Figure 13. The beam shape is approximated as a triangle such that the error slope is taken at the midpoint from the zero point and the signal peak (approximately the 3-dB beamwidth); thus, the proportion may be expressed as

$$\frac{\delta\theta}{V_n} = \frac{2\theta_3}{V_S}$$

Where $\delta\theta$ is the RMS angle error and θ_3 is the 3-dB beamwidth. The remaining terms are the same as above. The RMS angle error is solved to give

$$\delta\theta = \frac{2\theta_3}{\sqrt{2\left(S/N\right)_1}}$$

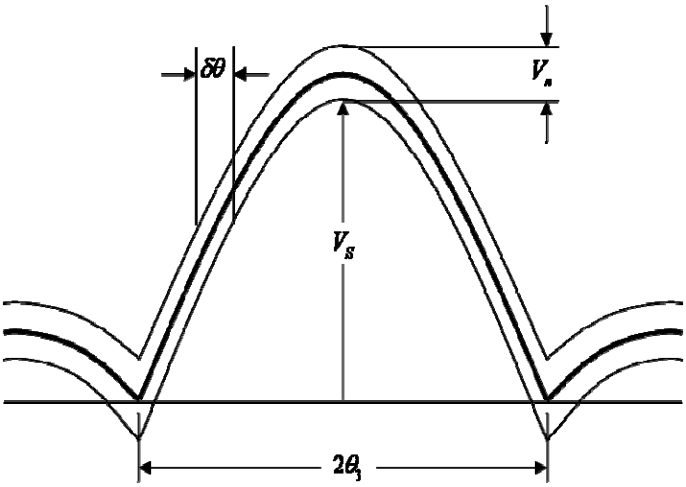


Figure 13. Angular noise error due to interference contaminating the return pulse (After [12]).

The final form of the angular error as given in [13] is

$$\sigma_\theta = \frac{\theta_{3\text{dB}}}{K\sqrt{\left(2\left(S/N\right)_1\right)M_i}} \tag{3.3}$$

where θ_{3dB} is the half-power beamwidth and. For this study, *K* will assume the value 1.7 for a monopulse radar as given in [6]. By letting $\left(S/N\right)_1 M_i = S/N$, the quantity

calculated in Equation 3.1, it is now used in the computation of Equations 3.2 and 3.3. In the MATLAB simulation, the *awgn* function, in addition to the computed values for σ_R and σ_θ, was used to add measurement error to the true sensor-to-target azimuth, elevation and slant range that was obtained earlier from the *elevation* function.

The error in angular measurement and range measurement for the booster and upper-stage of the missile are shown in Figures 14 and 15, respectively. A complete picture depicting how the error affects the measurement can be seen in Figure 16. For clarity, Figure 16 only shows the sensor measurement for the lower booster-stage. The true missile flight path is represented by the red, dashed line while the measurement or position, as reported by the sensor, is represented by the single blue points along the flight-path curve. Each of the three figures illustrates that there is a significant difference between the actual missile location and the report measurements in the initial minutes of the flight. The range to the sensor is the main contributing factor to the error. As the

missile reaches a point in its flight where it is nearest to the sensor, the measurement error is at a minimum quantity. This is seen at 150 seconds as show in Figure 14 and 15. Furthermore, large measurement errors are recorded as the upper stage continues its ascent and leaves the sensor's view beyond 180 seconds. To summarize the information provided by the figures above, the mean angular error is approximately 1.28 degrees and 1.16 degrees for the lower and upper stages, respectively. The mean range measurement error is approximately 0.39 km and 0.33 km for the booster and upper stage, respectively.

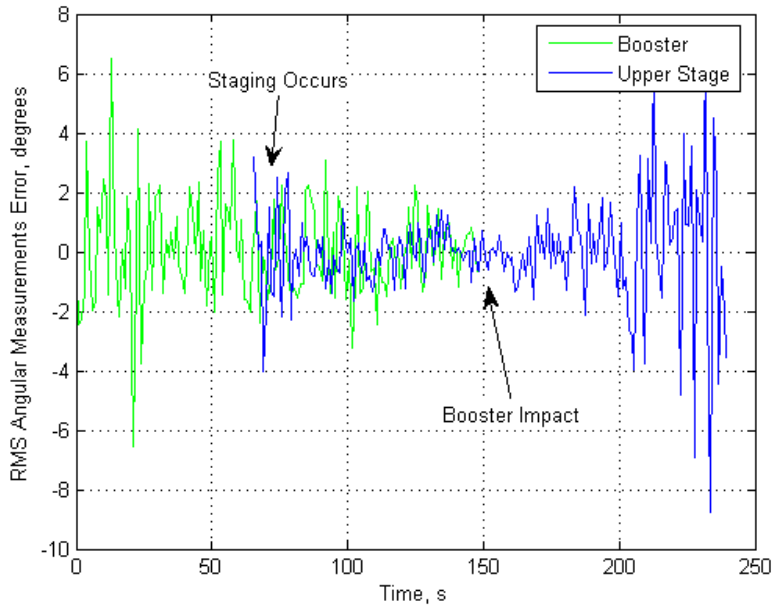


Figure 14. An examination of the angular azimuth and elevation measurement error, due to noise in the RF sensor model, compared to the known true angular measurement (centered about zero) during the initial 240 seconds of flight.

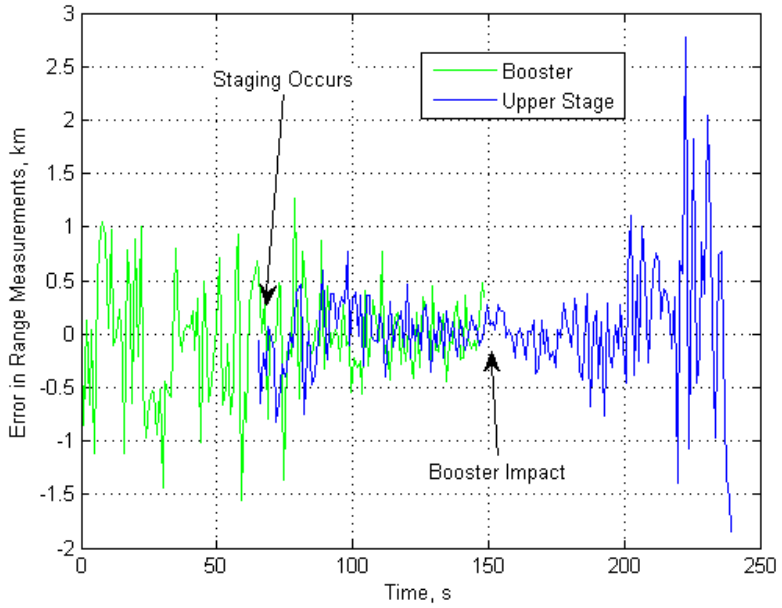


Figure 15. An examination of the error in range as reported by the RF sensor compared to the true (centered about zero) missile-to-sensor range during the initial 240 seconds of flight.

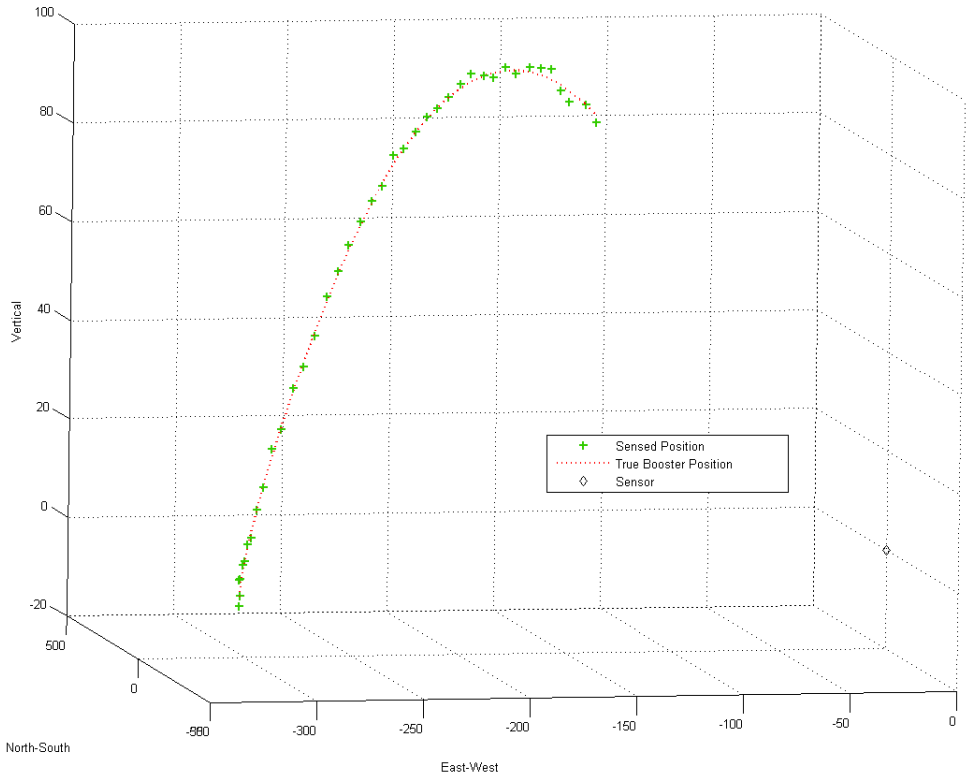


Figure 16. Angular and range uncertainty due to sensor noise. This plot shows the booster stage only.

In [6], the author conducted experiments by varying several parameters of the radar equation of an X-band RF sensor to include: the peak power, P_t, antenna half-power beamwidth, θ_3, the pulsewidth, τ, and the number of pulses integrated, N_i. The author exhaustively investigated the influence each parameter had on the sensor's accuracy in reporting the target's position as well as determining values that optimize the sensor's measurement accuracy. Here, the parameter values found in [6] are used in this study. These parameters for sensors *RF1* and *RF2* are summarized in Table 4.

Parameter	**Value**
Frequency, f	10 GHz
Peak Power, P_t	1.0 MW
Antenna Gain, G	42 dB
Half Power Beam Width, θ_B	0.5-1.0°
Pulse Width, τ_c	50 μs
Noise Figure, F	3 dB
Pulses Integrated, M_i	20

Table 4. RF1 and RF2 sensor parameters.

The simulation of the sensor behavior and the computation of range and angular errors were completed in MATLAB. By using the radar parameters given in Table 4 and evaluating the radar equation for SNR, given in Equation 3.1, and applying the quantity to Equations 3.2 and 3.3, the error in the sensor's precision—in reporting azimuth, elevation and range to the missile—can be simulated. The MATLAB script *RFobserve*, written for this study, uses the preceding equations to generate the RF sensor's observed measurements and simulated inaccuracies. The input to the *RFobserve* function is the data from the IMPULSE General Write Utility as given in Table 2 with the format of Table 3. The output of the script is the missile position—with error—in an Earth Centered Earth Fixed (ECEF) coordinate system.

In summary, this chapter established the relationships that govern the performance of the radar sensor. Measurement errors due to interference in the return echo were discussed. The radar the signal-to-noise ratio was examined. The important

result that was presented was the measurement error obtained as a function of the SNR. The results were used to model the sensor's limitation in precision when detecting and reporting ballistic missile position. Next, in Chapter IV, we discuss the primary method of tracking simultaneous missile launches.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. EFFICIENT MULTIPLE BALLISTIC MISSILE TRACKING

In this chapter, a strategy for tracking simultaneous missile launches is examined. The ballistic missile data, generated in Chapter II and the respective observations reported by the given sensor model in Chapter III, are used as the input to the tracking algorithm. More concisely, the missiles created with the simulation tool and as reported by the sensor will be passed to this tracking routine. The output will be the coherent tracks of each missile. The overall concept of the multiple target tracking approach is depicted in Figure 17.

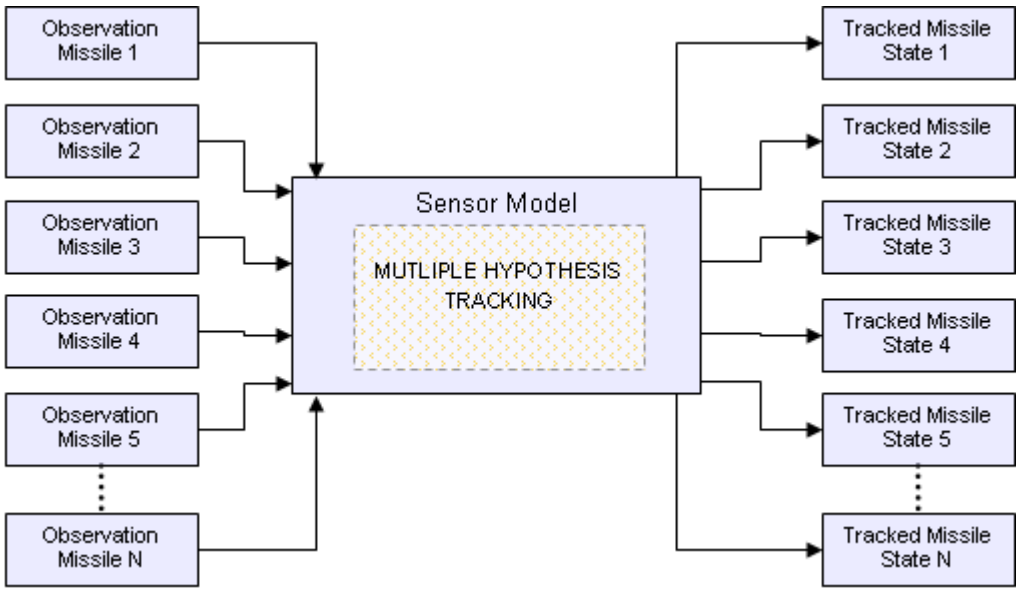


Figure 17. The MHT in the missile tracking process.

This chapter is organized as follows: first, concepts relevant to multiple target tracking problem are introduced. Next, a strategy to help realize an efficient form of multiple hypotheses tracking (MHT) is examined. A flow diagram is presented to show the MHT implementation and how the algorithm is applied to the ballistic missile data based on the discussion in Chapter II. The chapter concludes with the presentation of the results and a discussion on the effectiveness of the algorithm in tracking a missile's boost phase trajectory. Runtime statistics are made available for the reader to review.

A. MULTIPLE HYPOTHESES TRACKING

This section introduces terms that will aid in the discussion of the tracking algorithm. Secondly, it is important to discuss the principal difficulty encountered in the multiple-target tracking process. In Section 2, a comparison is made between single and multi-target tracking. The problems associated with the latter are revealed and a strategy for addressing the issue will be presented later in Section B.

1. Contacts, Targets, Scans and Associations

Several terms must be established to provide clarity for the remainder of the discussion in this chapter. Firstly, the definition of a *contact* is introduced. This is an observation that consists of a detection—by a sensor—and its corresponding *measurement*. Commonly, a detection occurs when the signal-to-noise ratio of a sensor meets or exceeds a predefined threshold. In the sensor model simulation, as discussed in Chapter III, the RF signal return from the missile was such that it enabled the sensor to detect the object. Furthermore, the *measurement* corresponding to this detection consists of the position of the object. In limiting the sensor responses to contacts, a restriction is imposed such that a signal return is of interest only when the object meets a predetermined criterion. This allows the contact to be "flagged" for further examination. For the remainder of this study, the term *measurement* will also be used interchangeably to mean contact. These terms, as well as the next several terms given, are illustrated in Figure 18 for a single target tracking problem.

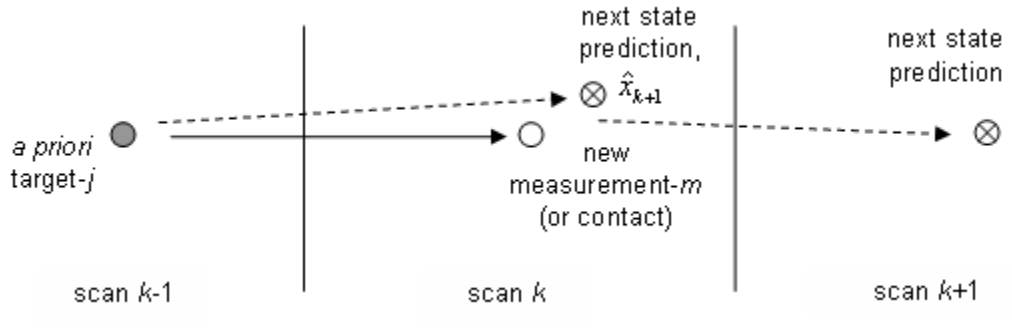


Figure 18. Single target tracking (in 2-D) illustrating measurement-to-target state prediction pairing, and next-state prediction correction.

A *target* is a *measurement* from a previous sample time, which has been flagged and declared an object of interest; hence, the term, *a priori target*, is used. This object's

information, or track file, may consist of position, velocity and acceleration parameters. Certain characteristics of a measurement are "screened" and, if they fall within predetermined values, the *measurement* is declared a *target* and stored in a database to await further information update. A contact's velocity information, for instance, may be used to mark the measurement for further observation.

The term *scan* will be used to refer to successive time periods. Allowable measurements and targets will now be further restricted to specific time scans. Finally, the term *association* is used to describe the pairing of a measurement, in a present scan, with that of a target in a previous time scan. It is important to note that there must be no ambiguity in a target-to-measurement pairing from one time period to the next time period.

2. Single and Multiple-Target Tracking Environment Comparison

There are two essential assumptions made in a single target tracking problem. First, there is one and only one target present in the observation region. Secondly, all sensor observations in successive scans are generated by the same target. Given the *a priori* target, *j*, as in Figure 18, the problem is approached as follows: based on prior information about the target, a *next-state* prediction is generated. This is given by [Bar]

$$\hat{x}_{k+1} = F_k x_k + \omega_k$$

where x_{k+1} is target next-state vector (prediction), F_k is the known state transition matrix, ω_k is the plant-noise associated with the target and k is the time index. Next, a measurement is taken in the ensuing scan. Since the measurement is likely to originate from the target that caused the contact in the previous scan, a direct comparison may be made between the new observed *contact* and the expected *next-state* prediction. This difference is also referred to as the *innovation* and an appropriate expression will be presented later.

The algorithm which computes the next-state prediction is updated with the new observed information. Adjustments are made to improve follow-on estimations. The algorithm eventually stabilizes such that each next-state prediction approaches on the actual measurement. Obviously, tracking is trivial when there is no uncertainty about the origin of successive measurements.

On the other hand, in a multiple-target environment, for any given scan, the sensor responses—measurements—may be due to any target. Thus, each possible movement of established targets must be hypothesized in the observation space. The primary difficulty in multiple target tracking is correctly assigning successive contacts to a corresponding established target's next state prediction as shown in Figure 19. This process of making possible assignments is referred to as *measurement-to-target* association.

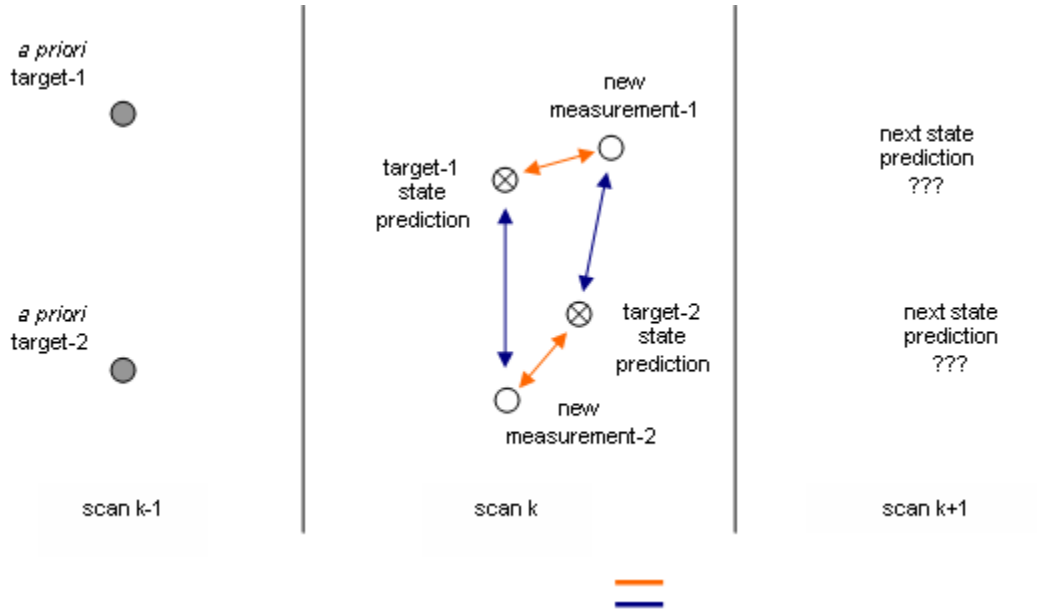


Figure 19. Multiple-target tracking (2-D) scenario. An illustration of the difficulty in correctly pairing new measurements with their state prediction so subsequent predictions converge on the true track.

This is an ideal place to introduce the term association hypothesis in which each hypothesis attempts to provide a likely explanation as to the source of each new measurement. Hence, we have the algorithm's namesake—MHT.

An association hypothesis maps each measurement to a possible target's next-state prediction. Furthermore, if the association of contacts were always obvious, the problem would simplify to independent single target problems—the recursive process of measurement, state prediction, observation, and, finally, update of each target. However, due to the unpredictable nature of each target in a multiple-target space, the associations are ambiguous.

B. MHT IMPLEMENTATION

The following paragraphs cover the implementation of the MHT using the terms and concepts that have been established in Section A. Furthermore, the strategy for solving the association problem is presented in detail.

1. Generation of Association Hypotheses

The first step in the measurement-to-target association process is to form multiple feasible hypotheses. The assumptions made in this study are as follows:

- Each target causes, at most, one measurement to be generated by the sensor. Clustering as presented in [14] is an example whereby multiple contacts with common measurements can be represented by a single contact.
- Each measurement corresponds to only one known target. The possibility for sensor false alarms is not addressed in this study.
- Sensor measurements are updated every second.
- Targets, in this case ballistic missiles, do not deploy chaff or use electronic countermeasures.

The following discussion is based upon [14] and provides the framework for the implementation of the MHT. Let $Z^k = \left\{ z_{k,m},\, m = 1, 2, ..., m_k \right\}$ denote the set of measurements *m* in scan time *k*. In the simulation, a single measurement $z_{k,m}$ can be further decomposed such that $z_{k,m} = \left\{ \xi_{k,m},\, \gamma_{k,m},\, \zeta_{k,m} \right\}$ where ξ, γ, and ζ denote the position of the measurement of interest in a predefined coordinate system. In the study of ballistic missiles, a local vertical coordinate system—north, east and up—centered about the sensor's location will be used. Let $Z^k = \left\{ Z^1,\, Z^2,\, Z^3,\, ...,\, Z^k \right\}$ represent the cumulative set of measurements up to scan time *k*. Let $\Omega^k = \left\{ \Omega_i^k,\, i = 1,\, 2,\, ...,\, I \right\}$ represent the set of all measurement-to-target hypotheses at scan *k*. The index, i, denotes the hypothesis number. This is the association of measurement in a current scan with *a priori* targets established in preceding scans. Furthermore, the hypothesis Ω_i^k of the *k*th scan is taken as the joint hypothesis formed from all prior hypotheses Ω_i^{k-1} and the association hypothesis for the current data scan. Using Figure 19 as an illustration, each hypothesis comprises two possible association pairings.

The probability of each feasible hypothesis may now be derived. Let P_i^k be the probability of hypothesis, Ω_i^k, given measurements up through time *k*. To further clarify, allow Ω_1^k to represent *hypothesis*-1 where *measurement*-1 is associated to *target*-1 (implied by pairing with state *estimate*-1*)* and, within the same hypothesis, *measurement*-2 is associated to *target*-2 (see Figure 19). The probability of this hypothesis being the correct pairing is represented by P_1^k. Alternatively, *hypothesis*-2, Ω_2^k, with probability P_2^k corresponds to the alternate hypothesis where *measurement*-1 is associated with *target*-2 while *measurement*-2 originates from *target*-1. Also, let P_g^{k-1} represent the probability Ω_g^{k-1}; the past relationships. To successfully perform multiple hypothesis tracking where computational time must remain minimal, a recursive relationship must be established between P_i^k and P_g^{k-1} to avoid reevaluating all past data whenever a present scan set, Z^k, is received.

The association probability, P_i^k, is a conditional probability of Ω_g^k given the set of measurements, Z^k, and the hypothesis, Ω^k. In [15], Nagarajan, Chidambara, and Sharma (NCS) give the relationship

$$P_i^k = P\left(\Omega_g^k \mid Z(k), \Omega^k\right) = P\left(\{\Omega_g^{k-1}, \psi_h\} \mid Z^k, \Omega^k\right)$$

where $\psi_h = \left\{\psi_{mj_h},\ m = 1, 2, 3, ..., m_k\right\}$ and ψ_{mj_h} represents the event that the *m*th measurement corresponds (originated from) to the *j*th target as per association hypothesis ψ_h. Furthermore, *g* denotes the global index while *h* denotes the hypothesis index. By using Bayes' rule, we can express the above equation as

$$P_i^k = \frac{1}{P\left(\Omega^k\right)} P\left(\Omega_g^{k-1}, \psi_h \mid Z^k\right)$$

By setting $P\left(\Omega^k\right) = C$, a constant, the expression becomes

$$P_i^k = \frac{1}{C} P\left(\Omega_g^{k-1}, \psi_h \mid Z^k\right)$$

which can be further written as

$$P_i^k = \frac{1}{C} P\left(\Omega_g^{k-1}\right) P\left(\psi_h \mid \Omega_g^{k-1},\, Z^k\right)$$

where we have again applied the Bayes' rule. Since the events ψ_{mj_h} that comprise ψ_h are independent, we have

$$P_i^k = \frac{1}{C} P\left(\Omega_g^{k-1}\right) \prod_{m=1}^{m_k} P\left(\psi_{mj_h} \mid \Omega_g^{k-1},\, Z^k\right)$$

The right-most term is made possible by observing that the current scan association is affected only by Z^k and all past data have been included through the term Ω_g^{k-1}. From [15], by defining $\beta\left(m, j_h, \Omega_g^{k-1}\right) = P\left(\psi_{mj_h} \mid \Omega_g^{k-1},\ Z^k\right)$, we have

$$P_i^k = \frac{P\left(\Omega_g^{k-1}\right) \prod_{m=1}^{m_k} \beta\left(m, j_h, \Omega_g^{k-1}\right)}{C} \tag{4.1}$$

The term $\beta\left(m, j_h, \Omega_g^{k-1}\right)$ represents the probability that measurement *m* corresponds to target j_h as per the present scan hypothesis ψ_h and the past scan hypothesis Ω_g^{k-1}.

To solve for the above probability, let *J* number of known *a prior* targets $j_{k-1,1}$, $j_{k-1,2}$, ..., $j_{k-1,J}$ have predicted next-state measurement vectors $\hat{x}_{k,j1}$, $\hat{x}_{k,j2}$, ..., $\hat{x}_{k,jN}$ and corresponding innovation covariance matrices $\Sigma_{k,1}$, $\Sigma_{k,2}$, ..., $\Sigma_{k,N}$, respectively, as per Ω_g^{k-1} retained from previous scan *k*-1. When the new measurement $z_{k,m}$ corresponds to a confirmed target j_N whose existence is implied by the prior hypothesis Ω_g^{k-1}, then $\beta\left(m, j_h, \Omega_g^{k-1}\right)$ is characterized by the probability density function [18][19]

$$N\left(\tilde{z}_{k,m,j}, \Sigma\right) = \frac{1}{\left(2\pi\right)^{n/2} \left|\Sigma_{k,m,j}\right|^{1/2}} \exp\left[-\frac{1}{2} \tilde{z}^T_{k,m,j}\, \Sigma^{-1}_{k,m,j}\, \tilde{z}_{k,m,j}\right] \tag{4.2}$$

where $\tilde{z}_{m,j}$ is the measurement-to-target innovation, $\left|\Sigma_{k,m,j}\right|$ is the determinant of the innovation covariance, and *n* denotes the degrees of freedom of the measurement (in this study, *n* = 3 as the sensor's measurements consist of range, azimuth and elevation).

Furthermore, the innovation is given by [16] [17]

$$\tilde{z}_{k,m,j} \equiv z_{k,m} - H_j \hat{x}_{j,k|k-l} \tag{4.3}$$

and represents the difference between the observed measurement, $z_{k,m}$, and the predicted—or expected—measurement $\hat{x}_{j,k|k-1}$. The observation matrix, H_j, is the used to select the elements of $\hat{x}_{j,k|k-1}$ for comparison. The innovation (or residual) covariance given by

$$\sum_{k,m,j} \equiv H_{\Gamma,k,j} P_{j,k|k-1} H_{\Gamma,k,j}^T + R_k \tag{4.4}$$

The quantities $\tilde{z}_{k,m,j}$ and $\sum_{k,m,j}$ above are obtained from the output of the extended Kalman (EKF) filter as outlined in Appendix B, with the update estimate covariance, $P_{j,k|k-1}$,. The noise covariance, R_k, and the observation matrix $H_{\Gamma,k,j}$ are further explained in Equations B.2 and B.3 of the same appendix. The appendix further provides a comprehensive review of the EKF equations and the motion model used to predict the next state position $\hat{x}_j$ of the target (the missile).

The *j*-to-*m* probability given in Equation 4.1 is computed for each possible pairing for all hypotheses. Table 5 illustrates an example of the possible measurement-to-target association and each of their respective likelihoods. Note that there are five targets, $J = 5$, and seven measurements, $M = 7$, in this example.

Measurements, M

Targets, J	m_1	m_2	m_3	m_4	m_5	m_6	m_7
j_1	0.40	0.00	0.36	0.00	0.39	0.00	0.42
j_2	0.00	0.41	0.06	0.38	0.00	0.43	0.05
j_3	0.15	0.27	0.17	0.11	0.29	0.09	0.21
j_4	0.28	0.13	0.26	0.18	0.10	0.20	0.22
j_5	0.14	0.16	0.12	0.30	0.19	0.25	0.07

Table 5. Association probability matrix $\beta(\bullet)$ The measurement, m, to a priori target, j, association likelihoods as for each possible assignment in scan k (After [4]).

To solve the multiple hypotheses tracking problem, the "best" parings must be found. The most likely set of measurement-to-target associations in the list further implies that the hypothesis corresponding Ω_g^k is the most probable explanation as to the source of each measurement. The next section will introduce an alternate—more computationally efficient—approach to solving the multiple target motion tracking.

2. Efficient Determination of the Most Likely Associations

The strategy used in [15] is a concatenated form of the classical MHT developed in by Reid [4]. The NCS approach sought to improve Reid's original algorithm by minimizing the exponential growth of hypotheses as the number of tracks and observations increased. The undesirable growth in association hypotheses was a weakness in the classical approach, making its implementation impractical. However, Danchick and Newnam [4] argue that the NCS algorithm carries forward non-feasible hypothesis, which are then used to generate feasible solutions. The growth of the feasible solutions from non-feasible hypotheses carried forward is observed to be combinatorial, thus undesirable.

Using the method outlined in [4] the problem of determining the optimal hypothesis can be solved in a computationally efficient manner. The proposed method solves for the *N*-best feasible hypotheses using a linear assignment problem (LAP) solution applied directly to the $\beta(\bullet)$ matrix in Table 5. An advantage, as claimed by the authors, is that their method only deals with feasible solutions, thus enabling the MHT to be practically implemented in an operational setting. This technique is exploited in implementing the MHT in this thesis research. Its effectiveness in tracking ballistic missiles will be presented later.

An outline of the method as described in [4] is presented in the following paragraphs. Firstly, the individual association probabilities are computed as

$$P_{m,j} = \beta\left(m, j_h, \Omega_g^{k-1}\right) \tag{4.5}$$

which yields an association matrix, P_A, of size $J \times M$. Furthermore, the entries in the P_A matrix are sequentially numbered along the rows from top to bottom. These numbers will serve as the "pointers" to the individual cells. Table 6 illustrates the resulting probability of association matrix. From these probabilities, the LAP will search the matrix to locate *N*-best feasible associations is selected based on the highest values where *N* is defined by the user (in this study, $N = 10$ best solutions are found). From Equation 1.4, the probability of this hypothesis is computed as

$$P_H^{(q)} = P_i^k = \frac{1}{C'}\prod_{l=1}^{J} P_{m,j}\left(\alpha_q^l\right), \qquad q = 1, 2, ..., N \tag{4.6}$$

where $\left\{\alpha_q^l\right\}$ is the set of selected cell pointers and C' is a constant defined as

$$C' = \frac{P\left(\Omega_g^{k-1}\right)}{C}$$

In the MATLAB implementation of the algorithm, the computation is simplified by setting $C' = 1$.

		Measurements						
P_A		m_1	m_2	m_3	m_4	m_5	m_6	m_7
Targets	j_1	**1** 0.40	**2** 0.00	**3** 0.36	**4** 0.00	**5** 0.39	**6** 0.00	**7** **0.42**
	j_2	**8** 0.00	**9** 0.41	**10** 0.06	**11** 0.38	**12** 0.00	**13** **0.43**	**14** 0.05
	j_3	**15** 0.15	**16** 0.27	**17** 0.17	**18** 0.11	**19** **0.29**	**20** 0.09	**21** 0.21
	j_4	**22** **0.28**	**23** 0.13	**24** 0.26	**25** 0.18	**26** 0.10	**27** 0.20	**28** 0.22
	j_5	**29** 0.14	**30** 0.16	**31** 0.12	**32** **0.30**	**33** 0.19	**34** 0.25	**35** 0.07

Table 6. Pointers of probability matrix P_A: the cells in the measurement-to-target pairing are each given a pointer (bold) value. The Linear Assignment Problem approach uses these pointers to search the matrix for the N-best set of most probable associations. The most likely pairings are highlighted in red after the first sweep (After [4]).

To locate the set of *N*-best hypothesis and each corresponding set of associations, the algorithm uses a two-step strategy: an *initialization* step followed by a series of iterative *sweeps*. The parameter *N* is set to ten, that is, the 10-best hypothesis of measurement-to-targets are of interest for a particular scan. The first pass over the matrix, P_A, locates the *unconstrained-solution* where each matrix cell is considered. This yields the first hypothesis. The corresponding probability of this particular hypothesis, $P_H^{(0)}$, is the product of all cell values given by α_0 (highlighted in red in Table 6) and computed using Equation 1.9. The following is obtained for the initialization step (sweep 0) [4]

$$\alpha_0 = \{7\ 13\ 19\ 22\ 32\}$$

$$P_H^{(0)} = 0.42 \times 0.43 \times 0.29 \times 0.28 \times 0.30 = 0.00440$$

The remaining $(N-1)$-best feasible associations are obtained by continuing the sweep process. On subsequent sweeps, a cell member from the first hypothesis list is omitted from consideration. For example, in the P_A table for the immediate sweep, '7' would be eliminated. Thus, the pointer must select cell number '1'—the next highest *m*-to-*j* paring of 0.40 of the same row—to establishing the next list and its respective probability of hypothesis. For this sweep, the following is obtained:

SWEEP 1-A

α_1 = {1 13 19 24 32} with {7} eliminated

$P_H^{(1)}$ = 0.00389

The reader should observe that the forth pointer, '22', in α_0 was also removed and replaced by '24' to yield α_1. This occurred as a result of pointer '1' already assigning the first measurement to the first target, i.e., m_1 to j_1. This ensures that a measurement may only correspond to one target—the second constraint listed in Section B.1. The next possible assignment for j_4 is m_3.

The process continues with 'SWEEP 1-B', in which '7' is restored to the list and '13' is removed from consideration to generate another hypothesis and probability, which then yields

SWEEP 1-B

α_1 = {7 9 19 24 32} with {13} eliminated

$P_H^{(1)}$ = 0.00419

A master list of the probabilities $P_H^{(q)}$ corresponding to the set of pointers $\{\alpha_q\}$ is formed. Each time a row in P_A is processed, a new probability is computed, it is compared to the previous probability and the list is maintained in descending order for each ensuing sweep.

On 'SWEEP 2-A', one pointer is removed from the first row of P_A while another is removed from the second row. The process is repeated in a similar fashion. As the algorithm continues, *n*-tuples of pointers are removed on each subsequent sweep.

Reference [4] is recommended to the reader for a more comprehensive discussion and illustration of further sweeps. The results, after 20 sweeps, are provided in Table 7.

Number, q	Cell Pointers $\{\alpha_q\}$					Probability $P_H^{(q)}$
1	7	13	19	22	32	0.00440
2	7	9	19	22	32	0.00419
3	7	13	16	22	32	0.00410
4	7	13	19	24	32	0.00409
5	1	9	19	22	32	0.00400
6	1	13	16	22	32	0.00390
7	7	9	19	24	32	0.00390
8	1	13	19	24	32	0.00389
9	5	13	16	22	32	0.00380
10	7	13	16	24	32	0.00380

Table 7. *N*-best hypotheses in descending order (N = 10). The first row is the most probable assignment group. The cell list $\{\alpha_q\}$ contains pointers to the P_A matrix to make m-to-j associations and its respective probability $P_H^{(q)}$.

is generated during the ballistic missile tracking routing and can be found in the *Main_List* variable within the MATLAB workspace.

In Table 7, the first row, hypothesis '1' or Ω_1^k, contains pointers to the P_A matrix in Table 6 and makes the best *m*-to-*j* pairings. The assignment's respective probability $P_H^{(q)}$ is listed in the right-most column. An examination of Table 7 for this particular hypothesis yields the following measurement-to-target associations: *m7-j1*, *m6-j2*, *m5-j3*, *m1-j4*, *m4-j5*, *m2-j3*, *m3*-spurios measurement or possible new target. The pointers in this table are used to generate the *measurement-oriented* list in Table 8

Number	*m*1	*m2*	*m3*	*m4*	*M5*	*m6*	*m7*
1	*j*4	0	0	*j*5	*j*3	*j*2	*j*1
2	*j*4	*j*2	0	*j*5	*j*3	0	*j*1
3	*j*4	*j*3	0	*j*5	0	*j*2	*j*1
4	0	0	*j*4	*j*5	*j*3	*j*2	*j*1
5	*j*4	*j*2	0	*j*5	*j*3	0	0
6	*j*4	*j*3	0	*j*5	0	*j*2	0
7	0	*j*2	*j*4	*j*5	*j*3	0	*j*1
8	*j*1	*0*	4	*j*5	*j*3	*j*2	0
9	*j*4	*j*3	*0*	*j*5	*j*1	*j*2	0
10	*0*	*j*3	*j*4	*j*5	0	*j*2	*j*1

Table 8. Most likely association (measurement-oriented per hypothesis) list. For each scan k, a table as such is generated to give the *N*-best association hypothesis.

In the simulation, the MATLAB script *LAP* performs the *linear assignment problem* computation and returns the *N*-best set of measurement oriented hypotheses; however, only the first set of associations is reported, i.e., $N = 1$ is chosen within each scan when making the measurement associations and target next-state prediction. It can be seen that lower-ranked, as well as non-feasible associations, are pruned and prevented from being carried forward. Furthermore, the script returns the above table in the *TRK_ORNT_List* matrix within the MATLAB workspace. The follow-on section will give an overview of the mechanics of the entire routine of scanning, establishing targets, taking new measurements, association and, finally, track updating for output to display.

3. The Algorithm

This section gives a general overview of how the methods and techniques discussed in Sections B are implemented to address the multiple-missile tracking problem. The process-flow diagram is given in Figure 20 illustrates a process flow-diagram of the MHT implementation. The first scan iteration begins by receiving the 'radar feed' or, in the case of simulation, the position of each missile $z_{k,m}$ within the set Z^k, the set from the sensor at time *k*.

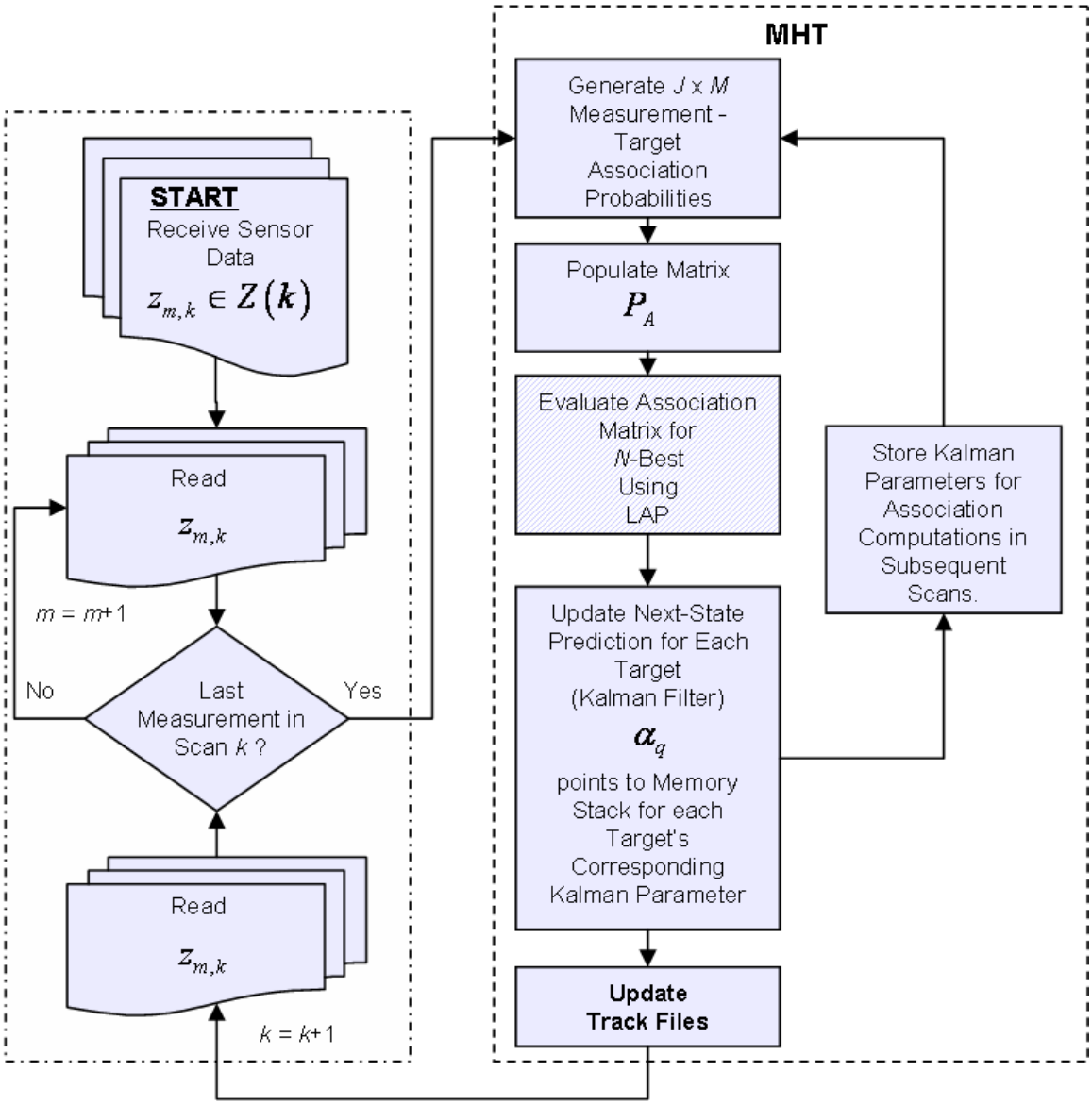


Figure 20. Sensor- level multiple-target track processing.

It is assumed that the measurements are first screened for characteristics typically exhibited by a ballistic missile profile. The screening criteria are not discussed here and assumed to be part of the sensor pre-processing; however, velocity and vertical acceleration are examples of screening parameters that may used to discriminate the missile from an air-breathing target, such as an aircraft. On the initial time-scan, each contact is assigned a corresponding tentative target, j_m. Next, a state prediction $\hat{x}_k$ is determined for each target, and the algorithm waits for the next set of measurement in the (k + 1)th scan. The MHT now generates multiple hypotheses of the measurements-to-

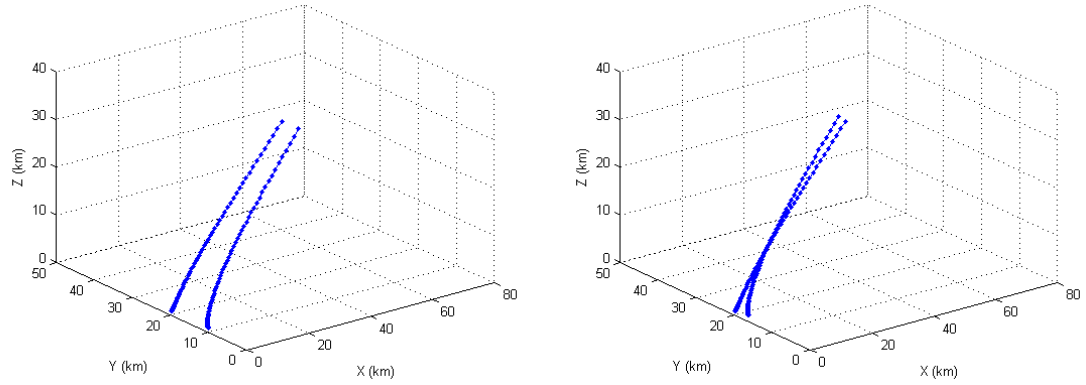
established targets—implied by their next-state update. The LAP makes the most likely assignments while the Extended Kalman filter is used to update each target's next state estimate, covariance, innovation covariance, and Kalman-gain for each target. Correct pairings of all of the above parameters to their respective targets are done within memory "stacks" in the simulation. When correct associations are made, the Kalman filter parameters that correspond to a particular target are retrieved from the appropriate position in the storage array. This was implemented in the computer simulation by storing the filter parameters in such an order that the pointers in the set α_q could also be used as an index to the desired positions in the memory stacks. The process then repeats for all subsequent scans and sets Z^k.

C. APPLICATION OF THE MHT TO THE MISSILE FLIGHT PROFILES

This section discusses initial testing of the developed MHT on test data generated in [9] as well as data original to this work—IMPULSE track files. The section is organized as such: firstly, an experiment of the multiple hypotheses tracking algorithm on the data from San Jose is performed. Plots are presented to show the algorithm's success in tracking multiple missiles, each following different launch profiles. Next, a study of the performance of the algorithm on the data generated by IMPULSE is conducted. The section further discusses the complications encountered in the application of the MHT and presents a working solution. The chapter concludes by presenting an analysis of the number of association hypotheses and computational times required per scan when the algorithm is successfully run on a six-missile launch scenario where each missile jettisons its booster can.

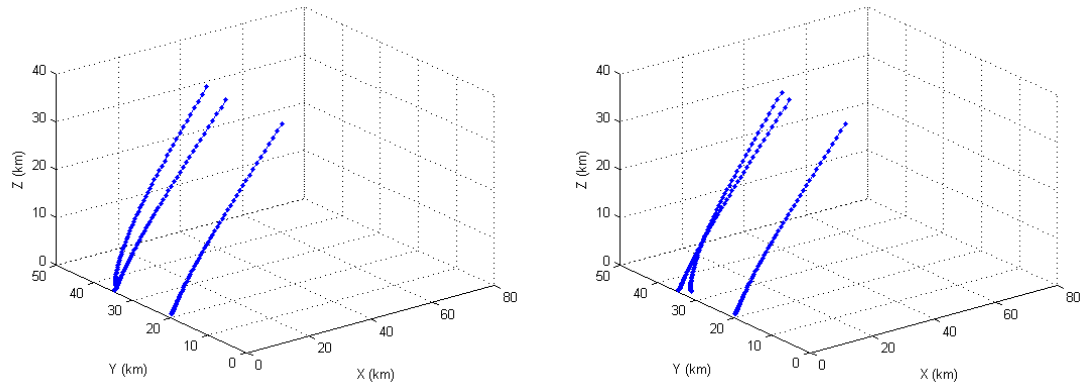
1. Examination of the Algorithm on Test Missile Data

The ballistic missile profile represented in this section was collected from the launch of a Titan-II missile as studied in [9]. The missile launch-data were replicated for the same time period to simulate multiple missile launches. The MHT algorithm is applied to single and multiple launch scenarios and can be seen tracking the missiles' position throughout the flight in Figures 21 and 22.



(a) (b)

Figure 21. Efficient MHT algorithm testing on data provided from [9] Titan II missile data. (a) Two missiles closely launched and converging, (b) Two missiles launch while one crosses in front of the other. Sensor location at x = 80, y = 20, z = 0 for both cases.



(a) (b)

Figure 22. Efficient MHT algorithm testing on data provided from [9] Titan II missile data. (a) Three missiles closely launched, (b) Three missiles launch while one crosses in front of the other. Sensor location at x = 80, y = 20, z = 0 for both cases.

This data served to provide initial validation of the MHT routine. The solid lines represent the missile's true position throughout its flight. The MHT, as developed in this study, accurately tracked the flight path of each missile in each multiple-missile case. The algorithm's report on each missile's location is represented by the blue-dotted points over

the flight path as reported by the sensor; thus, the algorithm also works in the presence of measurement noise. Furthermore, the coherent plots also imply that within each scan, the correct measurement-to-target association is made.

The second phase of the testing involved applying the algorithm to the IMPULSE©-generated data. Recall that IMPULSE© provides realistic motion modeling of missile flight profiles due to the many physical parameters integrated into the IMPULSE© software. Furthermore, IMPULSE© allows the study to be conducted on missiles which exhibit staging—jettisoning of spent components. This phenomenon can present problems for the MHT algorithm due to acceleration gradients.

Initially, the MHT was applied to two simultaneous missile launches; however, only the flight profiles of each lower stage was considered. In Figure 23, the tracking simulation stops at 65 seconds into the flight. Although the fly-out data appears similar to [9], the IMPULSE© model has included acceleration 'jerk' (nonlinearities) upon staging. As the missile's lower stage propulsion system is exhausted, there is a sudden deceleration in the missile flight.

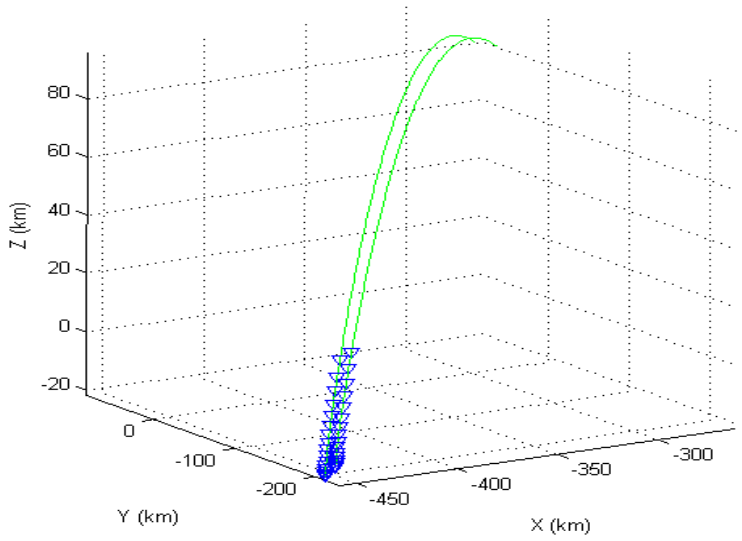


Figure 23. MHT testing on simultaneous launches of two missiles generated with IMPULSE (lower stage only). The algorithm fails at 65 seconds upon missile staging due to acceleration gradients.

The algorithm was tested again on a single missile profile and, because the routine only needed to make one association within each successive scan, the MHT was successful in tracking the ballistic missile. Figure 24 shows the second test on a single flight profile.

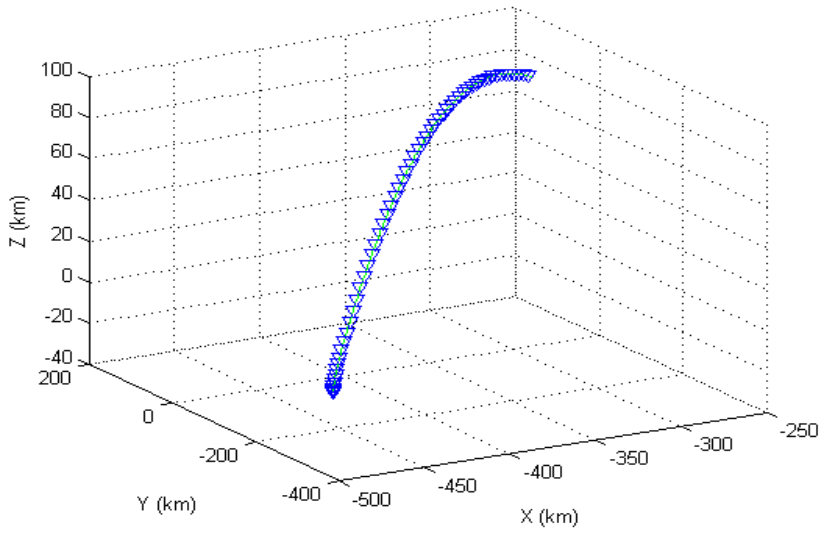


Figure 24. Second test of the MHT on a single boost stage to collect innovation data. The algorithm required adjustment to the routine to allow successful tracking of two-or-more missiles.

Inspection of the IMPULSE data, as seen in Figure 25, on *missile body acceleration* reveals that there is a lapse in acceleration.

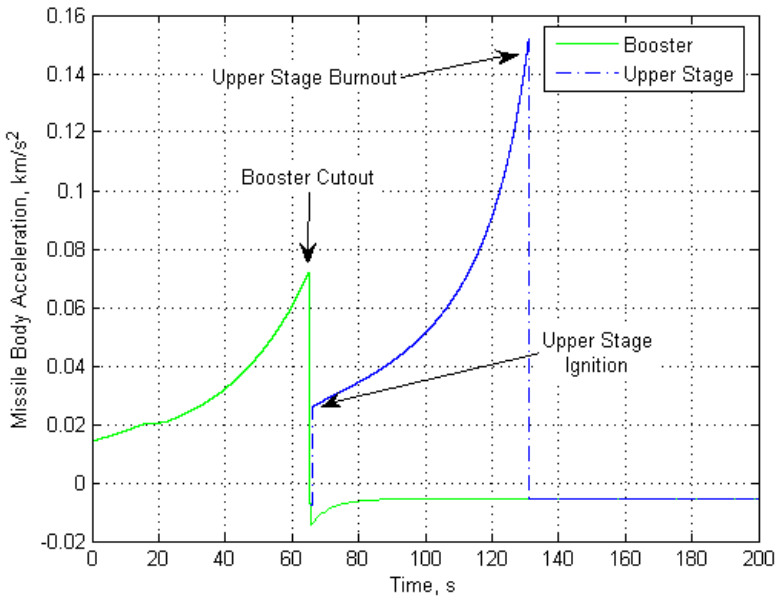


Figure 25. Missile body acceleration over time. Peaks are indicative of staging.

The abrupt change in acceleration causes the computation of Equation 1.6 to be very large. Essentially, the Kalman filter, within the MHT, expects to see the next measurement, z_m, relatively close to the state prediction $\hat{x}_j$. Figure 26 shows the value of the components comprising $\tilde{z}_{m,j}$ during the testing of the MHT on a single booster body. The timescale in this figure is the same as in Figure 25 for easy viewing of the errors that occur during booster jettison and upper-stage burnout. The error in the angular measurements, *azimuth* and *elevation*, are relatively small compared to the *range* component of the innovation. As $\tilde{z}_{m,j}$ grows large, the argument to the exponential term in Equation 1.5 becomes large and negative. Hence, computation of the measurement-to-target probability, $P_{m,j}$, results in an association probability of zero—even for the correct pairing. Moreover, all other incorrect pairings will have near-zero association-probabilities—as they should. In this situation, the *LAP* is unable to process this information if all confidences are zero.

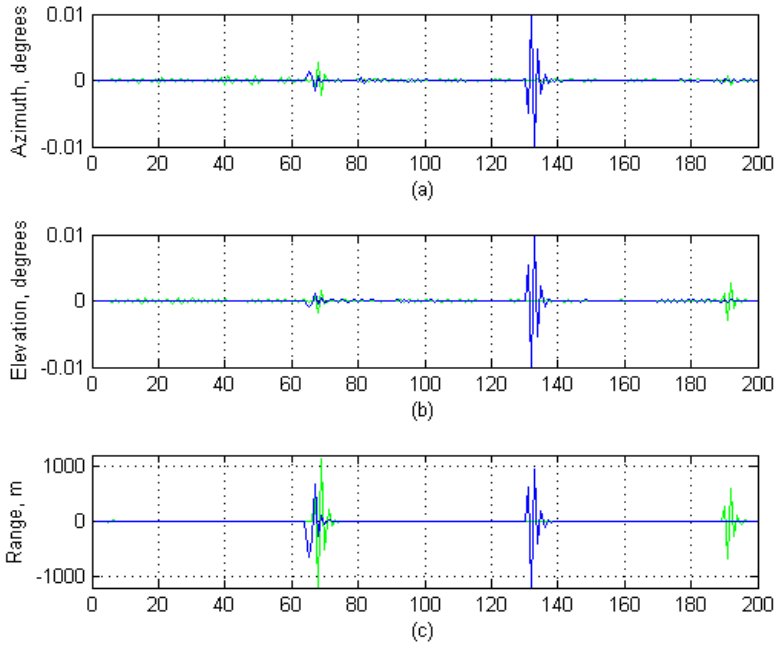


Figure 26. Innovation, $\tilde{z}_{m,j}$, plot as reported by the MHT. Peaks coincide with staging.

Table 9 shows the measurement-to-target assignment matrix at scan k = 65 of the IMPULSE-generated two booster-body flight. A solution to the problem is presented next.

P_A	m_1	m_2
j_1	**1** 0.00	**2** 0.00
j_2	**3** 0.00	**4** 0.00

Table 9. The pairing matrix at scan = 65 while running the MHT on the two-booster scenario. All measurement-to-target pairings are zero. Cell pointers used by the LAP are in bold while association values appear in gray.

Recall that Equation 1.8 provides a scoring of each possible measurement-to-target assignment based on a likelihood function. Maximizing the function implies minimizing the distance between the expected value and the measured value. The distance can be found using the weight of the innovation covariance such that

$$D = \tilde{z}_{m,j}^{T} \sum_{k,m,j}^{-1} \tilde{z}_{k,m,j} \leq G$$

where $\tilde{z}_{k,m,j}$ and $\sum_{k,m,j}^{-1}$ are given in Equations 1.6 and 1.7, respectively and G is a threshold value. Further, a constraint is placed on the absolute value of this distance such that when a threshold G is exceeded, the MHT simply chooses the *m-to-j* pairing with the minimum distance. A value of $G = 2$ was chosen as the threshold as all correct pairings were observed to have $|D| << 2$ over several trial simulations.

The MHT was next tested on a two-missile, two-stage flight scenario. In Figure 27, two missiles are launched. The algorithm successfully distinguishes each missile throughout each respective flight.

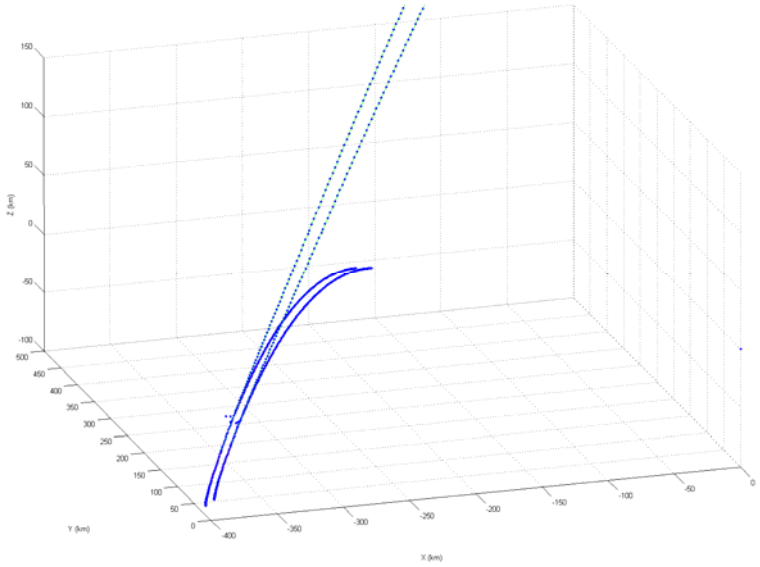


Figure 27. The MHT algorithm successfully tracks two simultaneous missile launches. Slight position report error is observed at the moment staging occurs.

However, the anomaly seen at the instant staging occurs is caused by the sudden introduction of two new contacts, i.e., at $k = 64,\ j = 2$ and at $k = 65,\ m = 4$. The algorithm initializes two additional measurements and makes some incorrect associations for six scans following the staging event. Upon the seventy-fifth scan, the MHT converges on each missile's true track. Further testing on three or more simultaneous missile launches revealed that the problem compounded with the introduction of more missile tracks. If the simulation is executed such that all missiles are launched simultaneously, each missile jettison occurs at exactly the same moment—an artificiality of the simulation world. Nevertheless, the MHT was slower to converge and upon running the algorithm for only four simultaneous launches, the algorithm is unable to make the correct measurement-to-target associations. This problem was solved by relaxing a constraint and making the assumption that launches and staging events do not occur simultaneously. In the computer analysis, this was accomplished by padding the *radar_data.mat* file in the workspace with zeros to simulate a delay in each missile's launch by 2 to 10 seconds. Figure 28 shows that the MHT algorithm is successful in tracking six missiles launched with a two to ten second delay. Staging occurs between 65 and 75 seconds into the simulation.

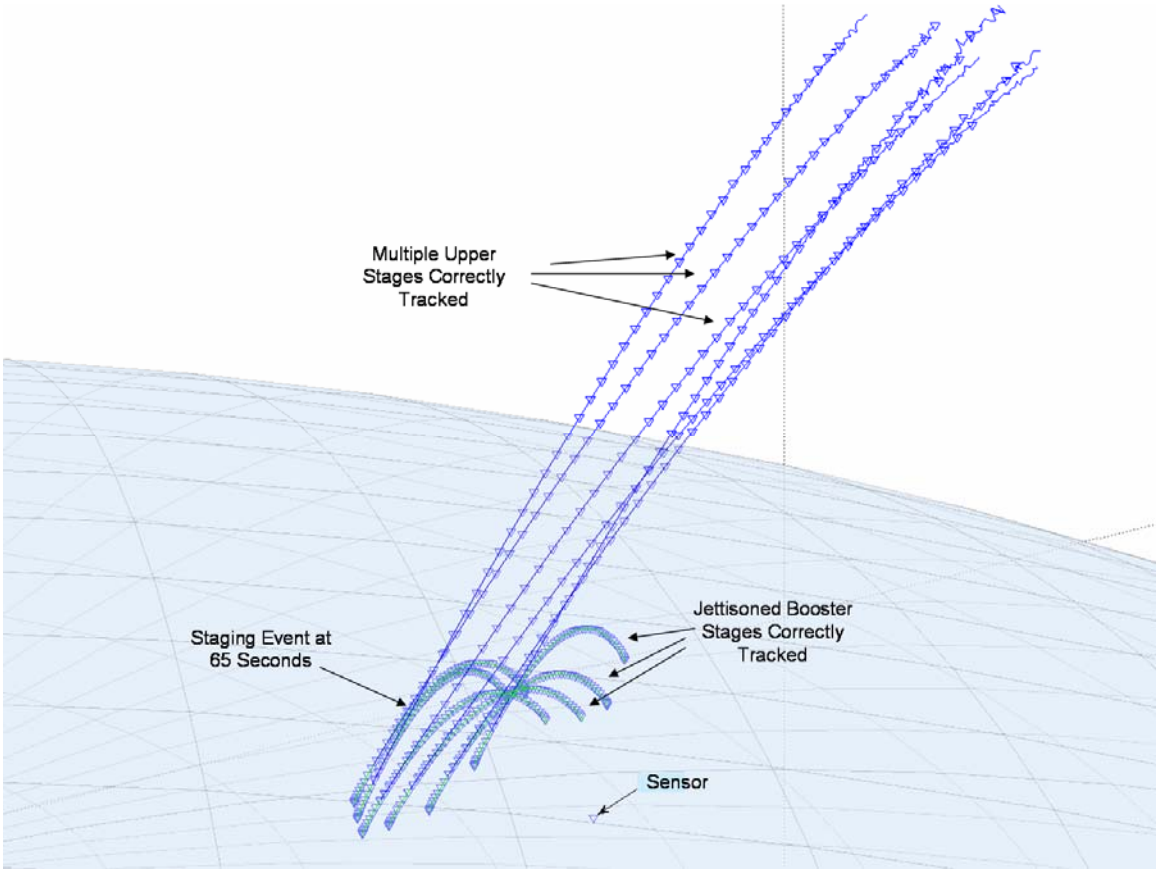


Figure 28. Application of the MHT to six near-simultaneous launches of ballistic missiles (viewed looking north). The delay ranges from four to ten second intervals. Missile staging occurs at 65 to 85 seconds, simulation time. The coordinate system here is local vertical (north-east-up) with respect to the sensor.

The algorithm is also successful in tracking missile flight trajectories that cross paths. In Figure 29, a West-to-East view of the same fly-out in Figure 28 is shown and it is clear the MHT coherently tracks the missile flight trajectory even as the missiles cross in the view of the sensor.

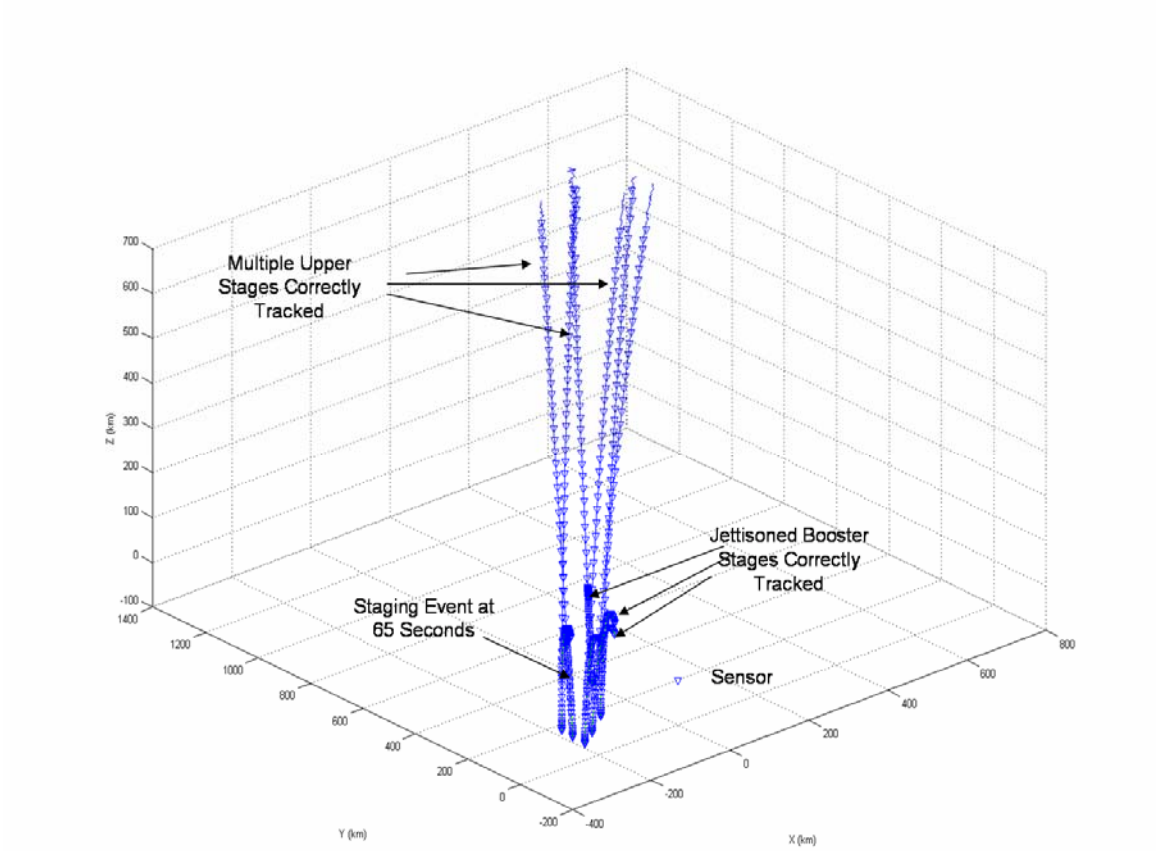


Figure 29. Same profile as in Figure 29: Six, near-simultaneous, ballistic missile launches (looking East-to-West). This view is used to emphasize the success of the algorithm despite missile tracks crossing flight paths.

D. COMPUTATIONAL COMPLEXITY OF MHT WITH LAP APPROACH

Several values were recorded to quantify the complexity of the algorithm in its application to the six-missile tracking scenario. Within each scan the following parameters were noted: time, the number of measurements observed, the number of potential measurement-to-target hypotheses, the time required to populate the P_A matrix, and finally, the time required by the LAP to find is set of *N*-Best solutions (where N = 10).

The results have been truncated for brevity and are listed in Table 10. An '~' indicates the time required for the simulation to complete the task was less than 0.00001 second. In each sweep of the LAP, $J \times M$ possible measurement-to-target assignments are still generated and tested for a likely pairing until the *N*-best associations are found.

An apparent advantage to the MHT with the LAP approach is that only a limited number of sweeps must be conducted to find the *N*-best solution. The number of sweeps is relatively small compared to the number of potential hypothesis. On the other hand, a notable argument is that the method is still quite demanding on the amount of memory needed to successfully perform the tasks within each scan. The basis for this statement is that within each scan, and within each *m*-to-*j* association, many prediction and update matrices for the extended Kalman filter must be maintained. For example, the innovation covariance in the simulation given by Equation 1.7, is an 81 element array. Numerous associations can generate many such arrays and require large memory space to store.

Time Scan	Number of Measurements	Number of Potential Hypothesis	Time Required to Formulate Hypotheses*	Time Required for LAP to find *N*- Best solutions
1.00	1	1	~	0.015625
2.00	1	1	~	0.0132
3.00	1	1	~	0.0132
11.00	2	2	0.03125	0.015625
14.00	2	2	~	0.0132
15.00	2	2	~	0.0132
16.00	3	6	~	0.0132
20.00	3	6	~	0.0132
21.00	4	24	~	0.0132
24.00	4	24	~	0.0132
25.00	4	24	~	0.0132
26.00	5	120	~	0.0132
29.00	5	120	~	0.0132
64.00	6	720	0.015625	0.015625
65.00	6	720	0.015625	0.015625
74.00	7	5040	0.015625	0.0132
75.00	7	5040	0.015625	0.015625
76.00	8	40320	0.015625	0.0132
77.00	8	40320	0.015625	0.015625
80.00	8	40320	0.015625	0.015625
81.00	9	362880	0.015625	0.0132
85.00	9	362880	0.015625	0.0132
86.00	10	3628800	0.03125	0.015625
90.00	10	3628800	0.03125	0.015625
91.00	11	39917000	0.03125	0.015625
94.00	11	39917000	0.03125	0.015625
95.00	11	39917000	0.03125	0.015625
359.00	12	479000000	0.046875	0.03125
360.00	12	479000000	0.046875	0.03125

Table 10. Runtime statistics of the MHT on a six-missile fly-out tracking problem. * A '~' indicates the time required to complete the task was less than 0.00001 of a second.

In summary, this chapter has demonstrated how the proposed linear assignment in the multiple hypotheses tracking may be effectively applied to the ballistic missile tracking problem. Previously, the multiple hypotheses tracking algorithm has been noted as infeasible and difficult to implement. The measurement-to-target association algorithm is combinatory in nature which can lead to an explosive growth of potential association hypotheses. Furthermore, it has been argued that the NCS routine in attempts to make feasible associations from infeasible hypothesis from previous scans [4]. These two principal disadvantages can prove detrimental to the time-sensitive, time-critical event of tracking ballistic missiles—making the MHT further prohibitive for such an application. The method for determining the association likelihood probability as outlined by Danchick and Newnam was exploited in this study and served as an efficient means to expediently identify correct target-to-next-measurement pairings within each scan. The linear assignment approach, studied here, avoids enumerating unnecessary and infeasible hypotheses. As a result, this enabled the MHT algorithm to successfully track multiple missiles and their staging events even with the unpredictable changes in acceleration during stage jettison.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION

This thesis investigated an approach for using RF sensors to detect and track near-simultaneously launched, ballistic missiles in their boost-phase of flight. The multiple hypotheses tracking algorithm, combined with a linear assignment approach to solve the *k*th-scan measurement-to-target pairing matrix for the *N*-best set of hypotheses, was implemented and resulted in the correct tracking of each missile's trajectory. To test the performance of the algorithm under realistic conditions, data generated by the National Air and Space Intelligence Center's IMPULSE© ICBM model were used.

A. SUMMARY OF FINDINGS

The likeness of the missile data as compared to real world launch-vehicles helped to demonstrate the feasibility and appropriateness of the MHT to the application of ballistic missile tracking. The inclusion of the IMPULSE© model is especially significant to the U.S. Missile Defense Agency as it represents the cognizant analyst's simulation of ballistic threats in a realistic physical environment. The results show that a technique called the linear assignment problem (LAP) used in the implementation of the algorithm is successful in an environment where complex interactions of missile staging, non-linear thrust profiles and sensor noise can significantly degrade the algorithm's performance, especially in multiple target scenarios. Determining the *N*-best associations and each respective probability as outlined in [4] was beneficial as it offered a means to expediently identify correct target-to-next-measurement pairings within each scan. The linear assignment approach avoided propagating infeasible hypotheses to later scans and reduced the computational complexity.

The tracking problem in this study examines at most 12 targets throughout the entire simulation, thus, for this number of targets, there are 479 million potential hypotheses (*M!)* per time scan. However, the linear assignment problem is still computationally effective and efficient. By generating $J \times M$ associations when populating the P_A assignment array, and using a series of iterative sweeps, the most probable measurement-to-target assignments can be found. The number of required sweeps to find the likely parings is relatively small compared to the number of hypothesis

that must be searched in the NCS approach. Still yet, the algorithm may not be suitable for "small-sensor" wireless networks. The MHT has extensive memory and computational requirements. Each target must have a corresponding set of state prediction and update matrices for the filtering algorithm used. The central processing unit (CPU) must also store a state transition matrix, covariance matrix, innovation covariance matrix, and measurement innovation matrix for each potential measurement-to-target assignment. The ability for a small sensor node with limited computational power to adequately handle a similar tracking scenario is, thus, brought into question.

B. SUGGESTIONS FOR FUTURE STUDIES

The algorithm performed well under the constraint that the ballistic missiles do not use radar evading tactics, such as chaff or electronic counter measures (ECM). The inclusion of chaff would require the radar database to be manually modified after the *RFsimulation* and *Radar_Data* scripts have been executed. In order to include spurious measurements—to represent chaff—individual ξ, ψ and ζ coordinates would need to be specified at random scans throughout the database. Additionally, the IMPULSE© modeling tool also enables the user to study classified missile profiles. The inclusion of these particular models would further help to qualify the MHT in this field of application.

Future work may additionally consider the injection of surveillance data from space-based infrared (SBIR) technology to discriminate lower-stage components, decoys and debris. The outputs from the IMPULSE© General Write Utility already include the thrust information of the missile throughout its flight. The table in [Ref. 20 pp. 126-127] shows the spectral intensity of a plume as a function of thrust magnitude. A MATLAB script may be written to relate the thrust to spectral intensity from this table.

Lastly, a sensor network should consider the propagation delays in sharing information between sensors as well as the time required to forward information to higher echelon processing command and control nodes. The effects of this physical delay on the MHT used in this research may be studied and discussed.

APPENDIX A. REFERENCE GUIDE TO IMPULSE TOOLS

A. OPERATION OF IMPULSE©

The IMPULSE missile database used in this work is unclassified in nature. However, a database containing real-world threat models and flight parameters is available and can be readily imported to enhance the study of future work. The intent is of this appendix is to cover the basic operation of IMPULSE© in defining several missile launches with boost phase profiles that imply intercontinental flight.

1. Main GUI

The following discussion assumes a clean installation of MATLAB R14 SP1 and IMPULSE versions 1.0. The IMPULSE graphic user interface (GUI) is initialized by double clicking on the IMPULSE V1.0 shortcut in the main MATLAB directory. Clicking the icon also loads MATLAB and sets the appropriate working directories. Figure 30 shows the main window the user is presented [7].

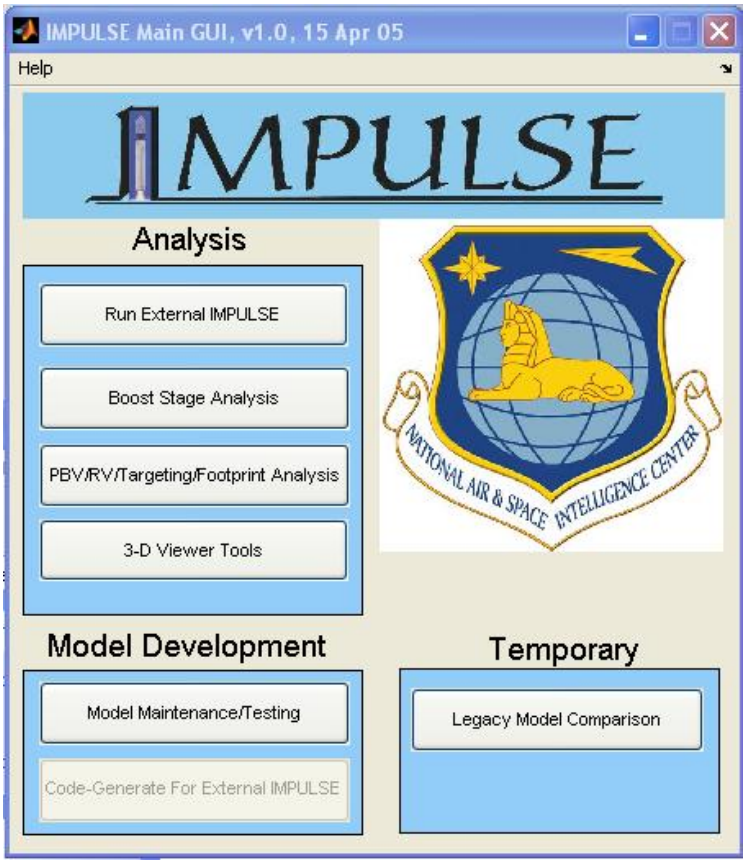


Figure 30. IMPULSE© main GUI.

Within the *Analysis* section, of concern is the "Boost Stage Analysis" segment. This choice will enable the user to select or create a missile model and define initial scenario parameters.

Upon selection of the "Boost Stage Analysis" tool, the user is presented with the window as shown in Figure 31. This is the main interface used to define each missile's physical parameters as well as in the launch scenario. In the "Select Missile" block, the "SampleUnclassModels" is chosen via the pull-down tab. Similarly, the user then selects "(U) BOOST Unclassified Sample." One point to note is if a classified model was added to the missile data-base upon installation, it would be made available as a selection option.

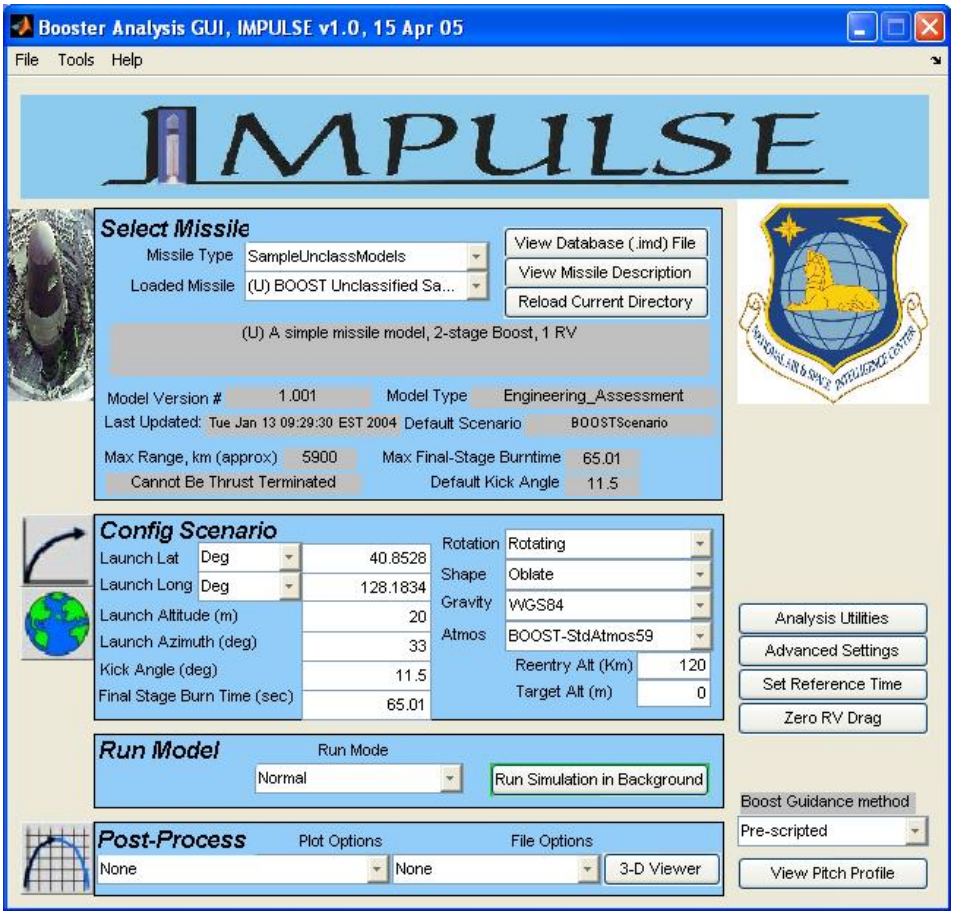


Figure 31. IMPULSE Booster Analysis GUI: Parameters reflect Ballistic Missile 1 entries for launch from No-Dong missile facility.

The GUI immediately populates the MATLAB workspace with the appropriate values to simulate the unclassified missile. Additionally, the parameters: *Max Range (km)*, *Final-Stage Burn Time*, and *Default Kick Angle* are predetermined and displayed in the GUI for review. These values can only be edited with the IMPULSE model maintenance utility. This topic is left for the reader to further investigate as it was not a tool reviewed during this study.

The final step in the setup process is to initiate the "Run Simulation in Background." The IMPULSE software now uses its motion model engine to determine the numerous dynamics of the missile throughout its flight. The MATLAB workspace is updated to reflect simple parameters, such as velocity, acceleration, flight angle of attack, heading, latitude, longitude and altitude. More complex parameters are also calculated and used to characterize the missile's flight path. Additionally, IMPULSE computes effects of Earth gravitational acceleration, vehicle mass decay—as the engines consume fuel, atmospheric density and drag on the vehicle. More parameters are computed but, again, left for the reader to further investigate. Essentially, we have achieved the goal of modeling a ballistic missile with a high-fidelity flight profile and most—if not all—laws of physics have been considered.

Once the "Run Simulation" is initiated, the user waits for the software to compute the profile of the missile's flight. This process requires approximately three minutes on a Pentium 4 processor with a clock rate of 2.4 GHz for a single missile flight to be completed. The simulation writes all parameters to matrices in the MATLAB workspace. The "Post Process" block is used next to export the information to files for further analysis.

2. Generating Output Files for Later Analysis

The final step requires the user to output the pertinent data to a text file for use in user defined MATLAB scripts or functions. Most importantly, they need to also be saved for later analysis sessions. To accomplish this task, the user selects "File Options: General Data Extract" in the "Post-Process" block illustrated in Figure 32.

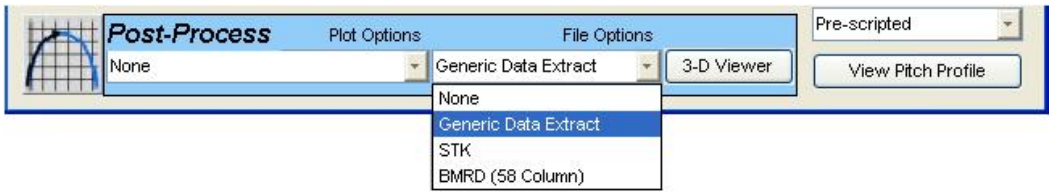


Figure 32. Post-Process general data extract.

The General Table Write Utility window, shown in Figure 33, now appears and the users may select the appropriate information to be exported. The figure depicts the selection of RB1 (Booster 1) from the "Object" selection field. The following parameters were chosen for output: Time, Geodetic Latitude (Radians), Geodetic Longitude (Radians), Pierce Point Altitude (km) and Thrust Magnitude (N). Next, a discussion on the graphical presentation of the missile flight profile is provided.

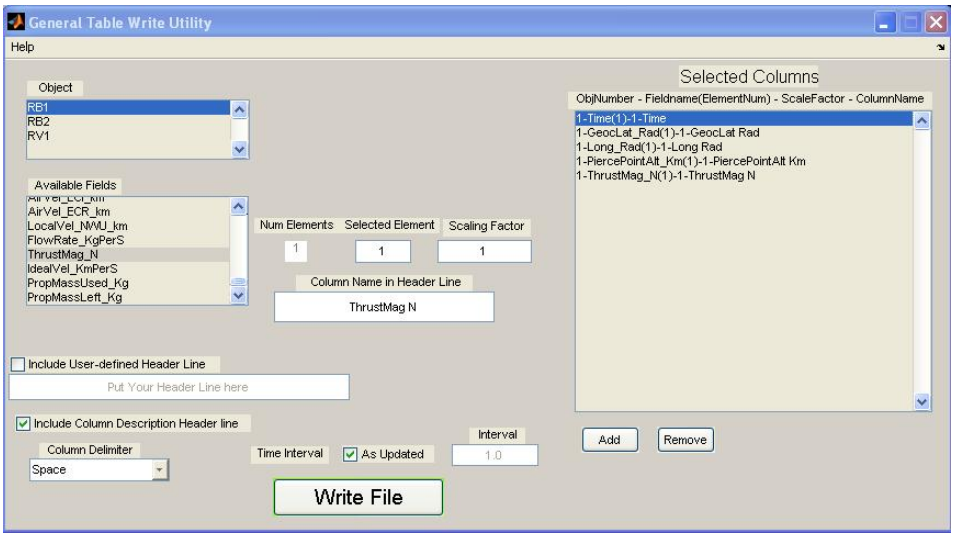


Figure 33. General Table Write Utility.

****** IMPORTANT ******

Ensure that the "Include Column Description Header Line" box and the "As Updated" boxes are UNCHECKED. Also, the interval must match the Kalman filter update interval. In this case an update value of 1 second is used as shown in Figure 34.

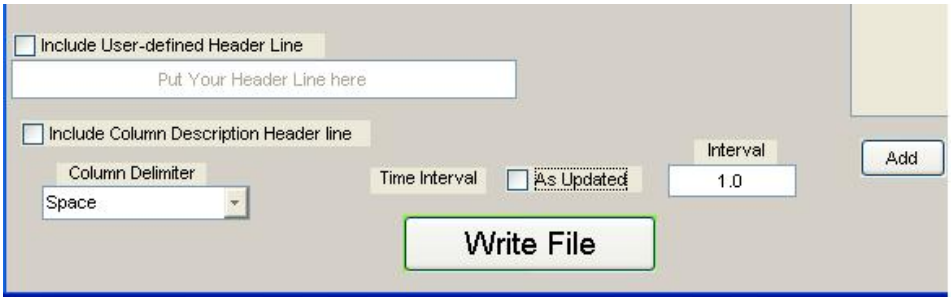


Figure 34. General Write Utility.

B. IMPULSE VISUALIZATION TOOL

Another utility included in the IMPULSE package is the Blue Marble View (BMV) visualization tool as depicted in Figure 35. The GUI for the viewer is initialized by selecting "3-D viewer" in the main IMPULSE window. The BMV tool enables the user to visually check the flight trajectory of the missile. Data contained in the workspace of MATLAB (after the "Run Simulation" phase) is used by this visualization tool to generate animations of the flight on a detailed globe.

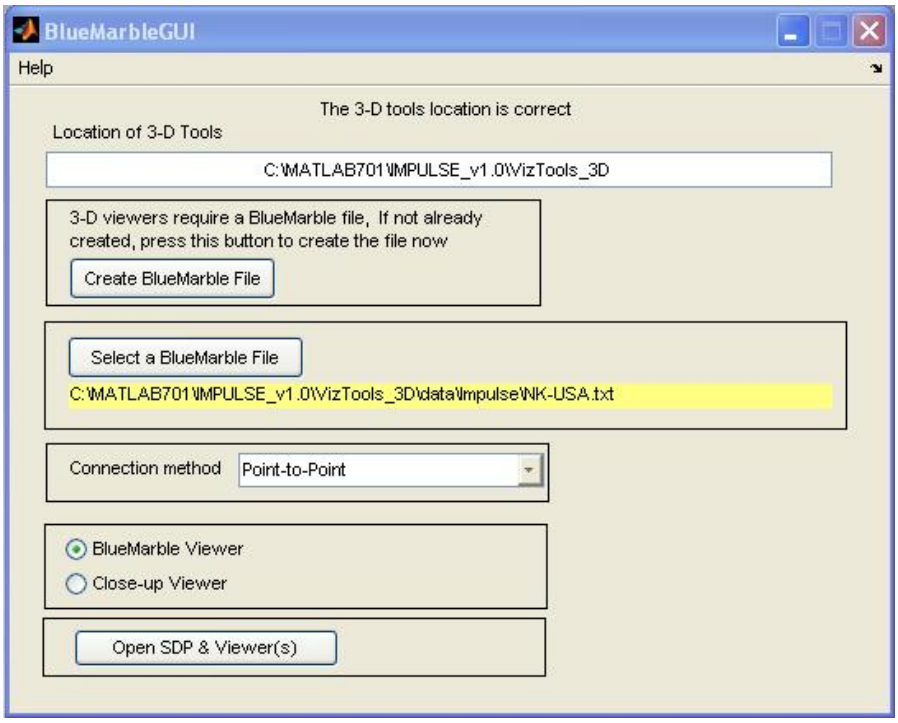


Figure 35. Blue Marble GUI/

The user simply selects the “Create BlueMarble File” tab and saves the file to the user’s directory of choice. On the other hand, if a BMV file already exists, then “Select BlueMarble” file is chosen. The user next ensures that the “connection” is set to “point-to-point” for the Simulation Data Player (SDP) viewer. The SDP appears as depicted in Figure 36. The user now selects the BMV created in the step outlined above. In order for the SDP to communicate with the BMV tool, the “Network” must also be set to point-to-point in the respective tab. The BMV then appears and the simulation of the missile’s flight plays out for the user.

Figure 36. Simulation Data Player.

Figure 37 depicts the simultaneous launch of three ballistic missiles, each respective boost stage track as well as each missile’s ground track.

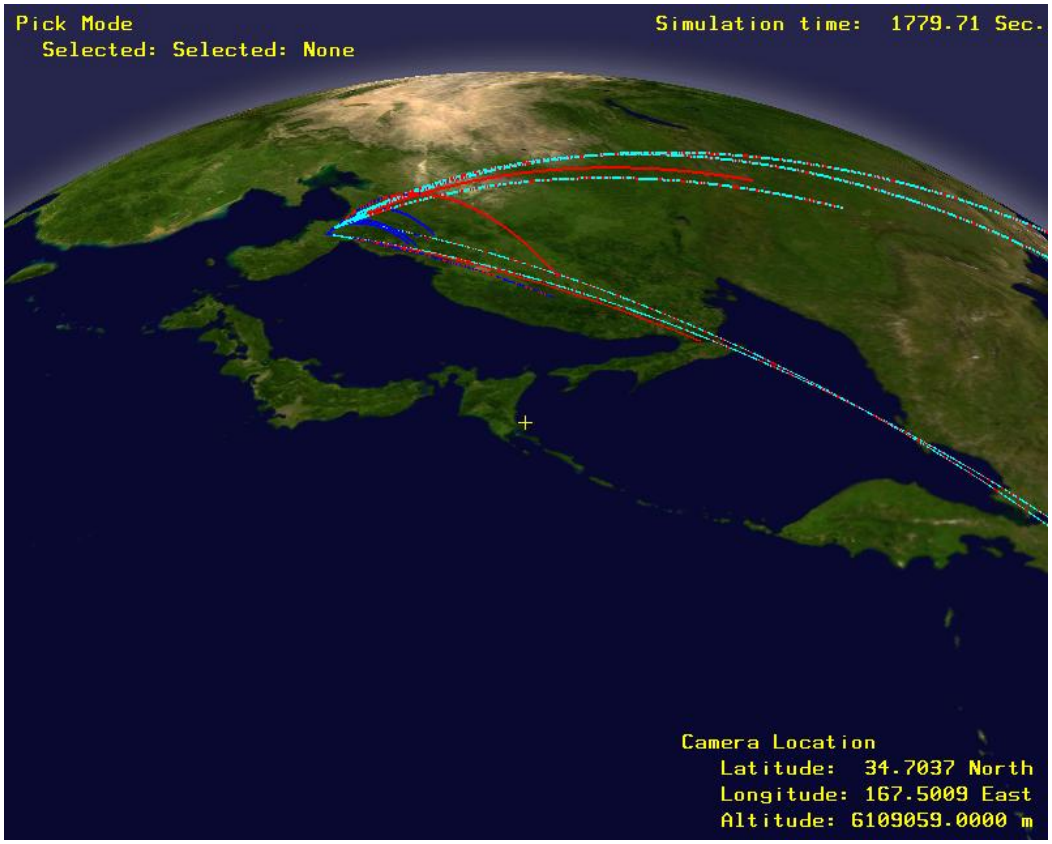


Figure 37. Multiple ballistic missile launch simulation viewed with BMV.

C. BALLISTIC MISSILE MODEL AND LAUNCH SCENARIO PARAMETERS

This section summarizes the ballistic missile profile implemented in IMPULSE well as the parameters set to define the launch scenarios. The parameters for *Ballistic Missile 1* was presented in Chapter II. The following tables reflect parameters for BM2 through BM6.

Parameter	**Value**
Missile Type	“SampleUnclassModels”
Loaded Missile	“(U) BOOST Unclassified Sample”
Launch Facility	*No-Dong alternate location*
Launch Latitude,	N 40°48'02"
Launch Longitude	E 129°17'18"
Launch Altitude	20 m
Launch Azimuth	35°
Kick Angle	11.5°
Final Stage Burn Time	65.01 (default)
Rotation [Earth]	Rotating
Shape [of Earth]	Oblate
Gravity	WGS84

Table 11. Ballistic Missile 2 scenario parameters used in the IMPULSE boost phase analysis GUI. Parameters reflect entries for launch from *No-Dong* missile facility.

Parameter	**Value**
Missile Type	“SampleUnclassModels”
Loaded Missile	“(U) BOOST Unclassified Sample”
Launch Facility	*Toksong-gun*
Launch Latitude,	N 40°25'00"
Launch Longitude	E 128°10'00"
Launch Altitude	100 m
Launch Azimuth	29°
Kick Angle	11.5°
Final Stage Burn Time	65.01 (default)
Rotation [Earth]	Rotating
Shape [of Earth]	Oblate
Gravity	WGS84

Table 12. Ballistic Missile 3 scenario parameters used in the IMPULSE boost phase analysis GUI. Parameters reflect entries for launch *Toksong-gun* missile facility.

Parameter	**Value**
Missile Type	“SampleUnclassModels”
Loaded Missile	“(U) BOOST Unclassified Sample”
Launch Facility	*Yongo-Dong*
Launch Latitude,	N 42°11'47"
Launch Longitude	E 130°11'48"
Launch Altitude	100 m
Launch Azimuth	31 °
Kick Angle	11.5 °
Final Stage Burn Time	65.01 (default)
Rotation [Earth]	Rotating
Shape [of Earth]	Oblate
Gravity	WGS84

Table 13. Ballistic Missile 4 scenario parameters used in the IMPULSE boost phase analysis GUI. Parameters for launch from *Yongo-Dong* missile facility.

Parameter	**Value**
Missile Type	“SampleUnclassModels”
Loaded Missile	“(U) BOOST Unclassified Sample”
Launch Facility	*Mayang*
Launch Latitude,	N 40°00'14"
Launch Longitude	E 128°11'04"
Launch Altitude	100 m
Launch Azimuth	35 °
Kick Angle	11.5 °
Final Stage Burn Time	65.01 (default)
Rotation [Earth]	Rotating
Shape [of Earth]	Oblate
Gravity	WGS84

Table 14. Ballistic Missile 5 scenario parameters used in the IMPULSE boost phase analysis GUI. Parameters reflect entries for launch from *Mayang* missile facility.

Parameter	**Value**
Missile Type	“SampleUnclassModels”
Loaded Missile	“(U) BOOST Unclassified Sample”
Launch Facility	*Mayang alternate location*
Launch Latitude,	N 40°13'40"
Launch Longitude	E 128°36'40"
Launch Altitude	0 m
Launch Azimuth	37°
Kick Angle	11.5°
Final Stage Burn Time	65.01 (default)
Rotation [Earth]	Rotating
Shape [of Earth]	Oblate
Gravity	WGS84

Table 15. Ballistic Missile 6 scenario parameters used in the IMPULSE boost phase analysis GUI. Parameters reflect entries for launch from *Mayang* missile facility.

APPENDIX B. KALMAN FILTER EQUATIONS

Appendix C provides information on the Kalman filter (KF) utilized in this study. This material is available through the hard work of a previous study. In particular, it is excerpted from Chapter III of [9] in which the author examined algorithms for tracking single ballistic missile fly-outs. The work sought to compare the advantages and disadvantage of several different tracking algorithms to include the KF.

A. DISCRETE TIME KALMAN FILTER

The Kalman filter estimates a state vector (*next-state* prediction) based on the knowledge of past measurements. When used in missile tracking, the Kalman filter equations are used to estimate present and future target kinematic quantities such as: positions, velocities, and accelerations. The object's—the ballistic missile's—motion model in discrete form is given by

$$x_{k+1} = F_k x_k + \omega_k$$

where x_{k+1} is the *n* dimensional missile *next*-state vector, F_k is the known state transition matrix, and ω_k is the plant noise associated with the target. The plant noise ω_k is assumed to be zero mean, white and Gaussian with known covariance Q_k. The measurement process is as follows:

$$z_k = H_k x_k + \nu_k$$

where the measurements are linear combinations of the state variables, which are corrupted by the addition of uncorrelated measurement noise, ν_k. The variable z_k designates the sensor measurement at scan *k*. The matrix H_k is a observation matrix to select desired elements from x_k . The measurement noise, v_k , is also assumed to be zero mean, white and Gaussian with known covariance R_k [21].

To initiate the Kalman algorithm, the first state estimate, $\hat{x}_0$, and its associated covariance, P_0, are assumed to be known priori. The algorithm loops over the measurement and then updates the measurement at each measurement time. The process of *updating* the state estimate when a new measurement is obtained can be broken down

into two steps: *prediction* and *correction*. *Prediction* refers to the estimation of the state vector to the next measurement scan, e.g., $k + 1$. In this process, the state estimate is given by

$$\hat{x}_{k+1|k} = F_k \hat{x}_{k|k} + \omega_k$$

and the associated covariance is

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_k$$

where T denotes transpose. *Correction* refers to updating—correcting—the state estimate and associated covariance based on the new measurement, using the correction equations,

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}\left[\tilde{z}_{k+1}\right]$$

whereK_{k+1} K_{k+1} represents the Kalman gain and $\tilde{z}_{k+1}$, the innovation. The two are defined as

$$K_{k+1} = P_{k+1|k} H_{k+1}^T \left[\Sigma_{k+1}\right]^{-1}$$

$$\tilde{z}_{k+1} = z_{k+1} - H_{k+1}\hat{x}_{k+1|k}$$

where Σ_{k+1} is the innovation (or residual) covariance given by

$$\Sigma_{k+1} = H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1}$$

as presented earlier in Equation 1.7. And the covariance update equation is

$$P_{k+1|k+1} = \left(I - K_{k+1} H_{k+1}\right) P_{k+1|k}$$

where I is an identity matrix.

The combined set of *prediction* and *correction* equations constitutes the discrete time Kalman filter. The preceding information is provided as a link to understanding the development of the EKF tracking algorithm [21][17].

B. EXTENDED KALMAN FILTER

In applications involving nonlinear dynamics or nonlinear measurement relationships, the extended Kalman filter (EKF) is used. The measurement from the sensor, i.e., radar measurements in range, bearing and elevation, are nonlinear; therefore, the EKF is appropriate in our ballistic missile tracking algorithm. The evaluation of the

Jacobians of the state transition and the measurement equations (the partial derivatives of the F and H matrices) is the main difference between the Kalman simplified and extended form [21].

In a system with nonlinearities in the dynamics of the measurement process, a similar framework as in a linear system is used. Considering the following nonlinear system equations,

$$x_{k+1} = f_k\left(x_k\right) + \omega_k$$
$$z_k = h_k x_k + \nu_k$$

where $f_k(x_k)$ is the nonlinear dynamics equation, and $h_k x_k$ is the nonlinear measurement equation. The noise processes ν and ω_k are assumed to be white Gaussian, uncorrelated and mutually independent; the following statements result

$$E\left[\nu_k\right] = 0$$
$$E\left[\nu_k \nu'_k\right] = Q_k \cdot \delta_{k1}$$

where δ_{k1} is the Kronecker delta function,

$$E\left[\omega_k\right] = 0$$
$$E\left[\omega_k \omega'_1\right] = R_k \cdot \delta_{k1}$$

with no cross correlation such that

$$0 = E\left[\nu_k \omega'_1\right] = E\left[\nu_k x'_0\right] = E\left[\omega_k x'_0\right] \quad \forall \; k,1$$

In order to determine the EKF prediction and correction equations, the nonlinear system of equations $f_k\left(x_k\right)$ and $h_k\left(x_k\right)$ must first be linearized. This is accomplished by a series expansion of the nonlinear dynamics and of the nonlinear measurement equations. To obtain the predicted state $\hat{x}_{k+1|k}$, the nonlinear function is expanded in a Taylor series around the latest estimate, $\hat{x}_{k|k}$, with terms up to the first order to obtain a first order EKF. The first order Taylor series expansions are required for the dynamic process and for the measurement process, and thus the matrices F_k and H_k must be determined. Let F_k represent the gradient of f_k evaluated at the most recent estimate, $\hat{x}_{k|k}$,

$$F_k \equiv \left. \frac{\partial f_k(x)}{\partial x} \right|_{x=\hat{x}(k|k)}$$

and $H_{\Gamma,k}$ as a gradient of h_k evaluated at the most recent estimate, $\hat{x}_{k|k}$,

$$H_k \equiv \left. \frac{\partial h_k(x)}{\partial x} \right|_{x=\hat{x}(k|k-1)} \tag{B.1}$$

The Taylor series expansions about the estimates are as follows

$$f_k(x_k) = f_k\left(\hat{x}_{k|k}\right) + F_k\left(x_k - \hat{x}_{k|k}\right) + \ldots$$

and

$$h_k(x_k) = h_k\left(\hat{x}_{k|k-1}\right) + H_k\left(x_k - \hat{x}_{k|k-1}\right) + \ldots$$

Then, the approximate system of equations, neglecting the higher order terms are

$$\begin{aligned} x_{k+1} &= f_k(x_k) + \omega_k \\ &= \left(f_k\left(\hat{x}_{k|k}\right) + F_k\left(x_k - \hat{x}_{k|k}\right)\right) + \omega_k \\ &= F_k x_k + f_k\left(\hat{x}_{k|k}\right) - F_k \hat{x}_{k|k} + \omega_k \end{aligned}$$

and

$$\begin{aligned} z_k &= h_k(x_k) + \nu_k \\ &= \left(h_k\left(\hat{x}_{k|k-1}\right) + H_k\left(x_k - \hat{x}_{k|k-1}\right)\right) + \nu_k \\ &= H_k x_k + h_k\left(\hat{x}_{k|k}\right) - H_k \hat{x}_{k|k} + \nu_k \end{aligned}$$

By using the above relationships, the approximate linearized system of equations are,

$$\begin{aligned} x_{k+1} &= F_k x_k + \omega_k + u_k \\ z_k &= H_k x_k + \nu_k + y_k \end{aligned}$$

with the deterministic terms

$$u_k \equiv f_k\left(\hat{x}_{k|k}\right) - F_k \hat{x}_{k|k}$$

$$y_k \equiv h_k\left(\hat{x}_{k|k-1}\right) - H_k \hat{x}_{k|k-1}$$

The Kalman filter *prediction* and *correction* steps for these approximate equations are as follows:

Prediction: In the state estimate, substitute $\hat{x}$ for x, include the deterministic terms and drop the zero mean noise.

$$\begin{aligned} x_{k+1|k} &= F_k \hat{x}_{k|k} + u_k \\ &= F_k \hat{x}_{k|k} + \left[f_k\left(\hat{x}_{k|k}\right) - F_k \hat{x}_{k|k} \right] \\ &= f_k\left(\hat{x}_{k|k}\right) \end{aligned}$$

The covariance prediction is a linear Gaussian update of the noise terms,

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_k$$

Correction:

$$\begin{aligned} \hat{z}_{k-1} &= H_k \hat{x}_{k|k-1} + y_k \\ &= H_k \hat{x}_{k|k-1} + \left[h_k\left(\hat{x}_{k|k-1}\right) - H_k \hat{x}_{k|k-1} \right] \\ &= h_k\left(\hat{x}_{k|k-1}\right) \end{aligned}$$

Hence, the state update equation is

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \left[\tilde{z}_k \right]$$

with innovation

$$\tilde{z}_k \equiv z_k - \hat{z}_{k|k-1}$$

and

$$K_k = P_{k|k-1} H^T \left[H_k P_{k|k-1} H^T + R \right]^{-1}$$

The covariance update equation using the gradient matrices is,

$$P_{k|k} = \left(I - K_k H_k \right) P_{k|k-1}$$

with the equivalent Joseph form [21]

$$P_{k|k} = \left(I - K_k H_k \right) P_{k|k-1} \left(I - K_k H_k \right)^T + K_k R_k K_k^T$$

C. EKF IN TRACKING BALLISTIC MISSILES

In this section, a ballistic missile tracking algorithm is developed utilizing the extended Kalman form. The system equations are the standard tracking equations

$$\begin{aligned} \hat{x}_{k+1} &= F_k \hat{x}_k + G_k + \omega_k \\ z_k &= h_k x_k + \nu_k \end{aligned}$$

where the missile state vector is given as

$$x_k = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ y \\ \dot{y} \\ \ddot{y} \\ z \\ \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix}$$

The term F_k is the linear state transition matrix:

$$F_k = \begin{bmatrix} 1 & \Delta & \frac{\Delta^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \Delta & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta & \frac{\Delta^2}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta & \frac{\Delta^2}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The gravity matrix, G_k, which accounts for the acceleration in $\ddot{z}$ due to gravity with $g = 9.8,\ m \cdot s^{-2}$, is

$$G_k = -g \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{\Delta^2}{2} \\ \Delta \\ 0 \end{bmatrix}$$

and ω_k is the plant noise with covariance Q_k

$$Q_k = q^2 \times \begin{bmatrix} \frac{\Delta^5}{20} & \frac{\Delta^4}{8} & \frac{\Delta^3}{6} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\Delta^4}{8} & \frac{\Delta^3}{3} & \frac{\Delta^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\Delta^3}{6} & \frac{\Delta^2}{2} & \Delta & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\Delta^5}{20} & \frac{\Delta^4}{8} & \frac{\Delta^3}{6} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\Delta^4}{8} & \frac{\Delta^3}{3} & \frac{\Delta^2}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\Delta^3}{6} & \frac{\Delta^2}{2} & \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\Delta^5}{20} & \frac{\Delta^4}{8} & \frac{\Delta^3}{6} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\Delta^4}{8} & \frac{\Delta^3}{3} & \frac{\Delta^2}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\Delta^3}{6} & \frac{\Delta^2}{2} & \Delta \end{bmatrix}$$

where Δ is the sampling interval and q^2 is a scaling factor used to account for un-modeled target maneuver accelerations, and ν_k is the measurement noise with covariance

$$R_k = \begin{bmatrix} \sigma_R^2 & 0 & 0 \\ 0 & \sigma_A^2 & 0 \\ 0 & 0 & \sigma_E^2 \end{bmatrix} \tag{B.2}$$

The standard deviations as defined in [9] were not used in this thesis. Each value was set to zero for this work. The filter performance was improved in this case.

Although the missile dynamics in this system are linear, the measurement process is nonlinear. The sensor platform observes the missile positions through measurements in range, azimuth and elevation relative to the sensor.

Let

$$h_k = \begin{bmatrix} \text{Range} \\ \text{Azimuth} \\ \text{Elevation} \end{bmatrix}$$

where

$$\text{range} = \sqrt{x^2 + y^2 + z^2} = \sqrt{x_1^2 + y_4^2 + z_7^2}$$

$$\text{azimuth} = \tan^{-1}\left[\frac{y}{x}\right] = \tan^{-1}\left[\frac{x_4}{x_1}\right]$$

$$\text{elevation} = \tan^{-1}\left[\frac{z}{r'}\right]$$

with

$$r' = \sqrt{x^2 + y^2} = \sqrt{x_1^2 + y_4^2}$$

These measurement equations are nonlinear and therefore must be converted using a series expansion of the measurement equation h_k. Applying the definition of the $H_{\Gamma,k}$ matrix, as in Equation B.1, the gradient of h_k is determined as follows:

$$H_{\Gamma,k} = \begin{bmatrix} \frac{\partial r(x)}{\partial x_1} & \frac{\partial r(x)}{\partial x_2} & \frac{\partial r(x)}{\partial x_3} & \frac{\partial r(x)}{\partial x_4} & \frac{\partial r(x)}{\partial x_5} & \frac{\partial r(x)}{\partial x_6} & \frac{\partial r(x)}{\partial x_7} & \frac{\partial r(x)}{\partial x_8} & \frac{\partial r(x)}{\partial x_9} \\ \frac{\partial a(x)}{\partial x_1} & \frac{\partial a(x)}{\partial x_2} & \frac{\partial a(x)}{\partial x_3} & \frac{\partial a(x)}{\partial x_4} & \frac{\partial a(x)}{\partial x_5} & \frac{\partial a(x)}{\partial x_6} & \frac{\partial a(x)}{\partial x_7} & \frac{\partial a(x)}{\partial x_8} & \frac{\partial a(x)}{\partial x_9} \\ \frac{\partial e(x)}{\partial x_1} & \frac{\partial e(x)}{\partial x_2} & \frac{\partial e(x)}{\partial x_3} & \frac{\partial e(x)}{\partial x_4} & \frac{\partial e(x)}{\partial x_5} & \frac{\partial e(x)}{\partial x_6} & \frac{\partial e(x)}{\partial x_7} & \frac{\partial e(x)}{\partial x_8} & \frac{\partial e(x)}{\partial x_9} \end{bmatrix}$$

which simplifies to

$$H_{\Gamma,k} = \begin{bmatrix} \frac{x_1}{\sqrt{x_1^2+x_4^2+x_7^2}} & 0 & 0 & \frac{x_4}{\sqrt{x_1^2+x_4^2+x_7^2}} & 0 & 0 & \frac{x_7}{\sqrt{x_1^2+x_4^2+x_7^2}} & 0 & 0 \\ \frac{-x_4}{x_1^2+x_4^2} & 0 & 0 & \frac{-x_4}{x_1^2+x_4^2} & 0 & 0 & 0 & 0 & 0 \\ \frac{-x_1 \cdot x_7}{\sqrt{x_1^2+x_4^2}\left(x_1^2+x_4^2+x_7^2\right)} & 0 & 0 & \frac{-x_4 \cdot x_7}{\sqrt{x_1^2+x_4^2}\left(x_1^2+x_4^2+x_7^2\right)} & 0 & 0 & \frac{\sqrt{x_1^2+x_4^2}}{x_1^2+x_4^2+x_7^2} & 0 & 0 \end{bmatrix} \quad \text{(B.3)}$$

Finally, the approximate—linearized—system of equations for the state predict and measurement are

$$\hat{x}_{k+1} = F_k \hat{x}_k + G_k + \omega_k$$
$$z_k = H_k x_k + \nu_k ,$$

respectively. In the simulation, the EKF is implemented in MATLAB and the matrices developed in this section are applied.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. CODE FLOW CHART

The process flows for the MATLAB® scripts are outlined in this appendix.

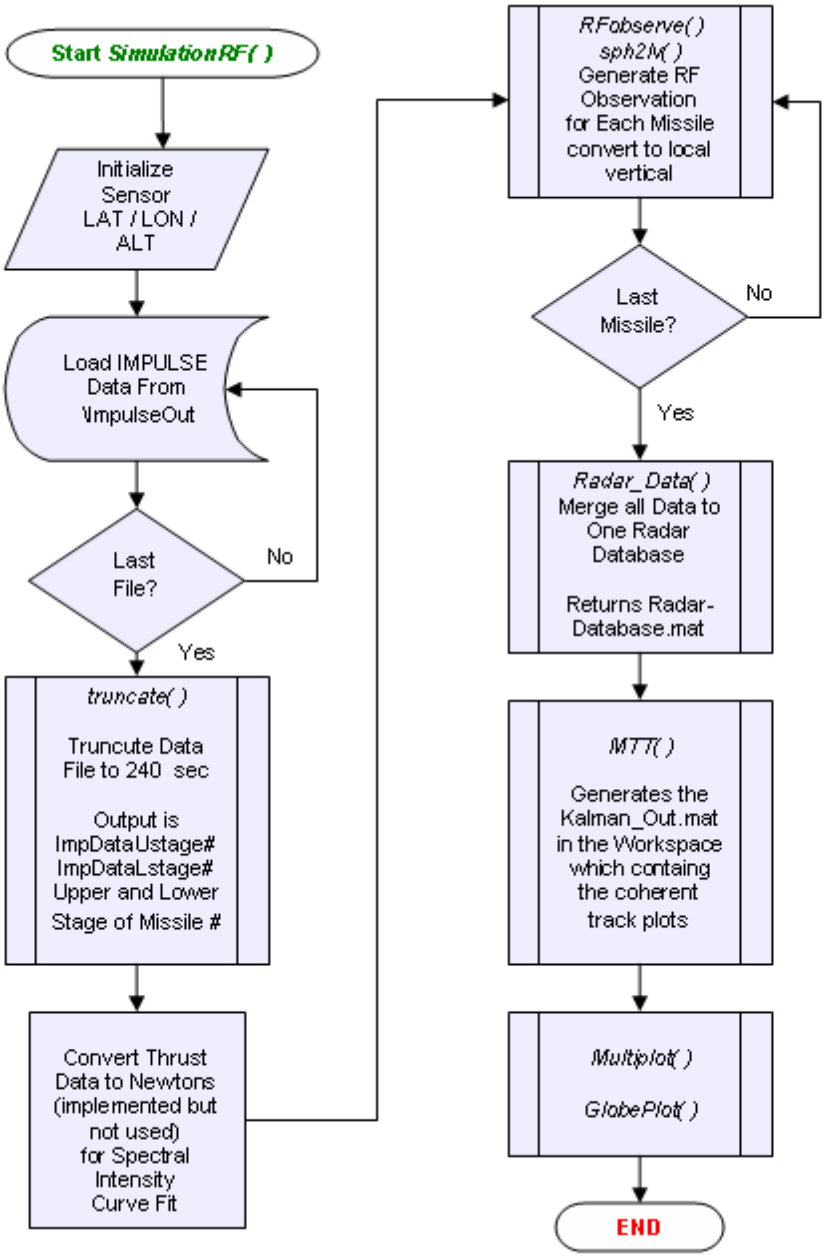


Figure 38. Flowchart SimulationRF1( ) and SimulationRF2( )

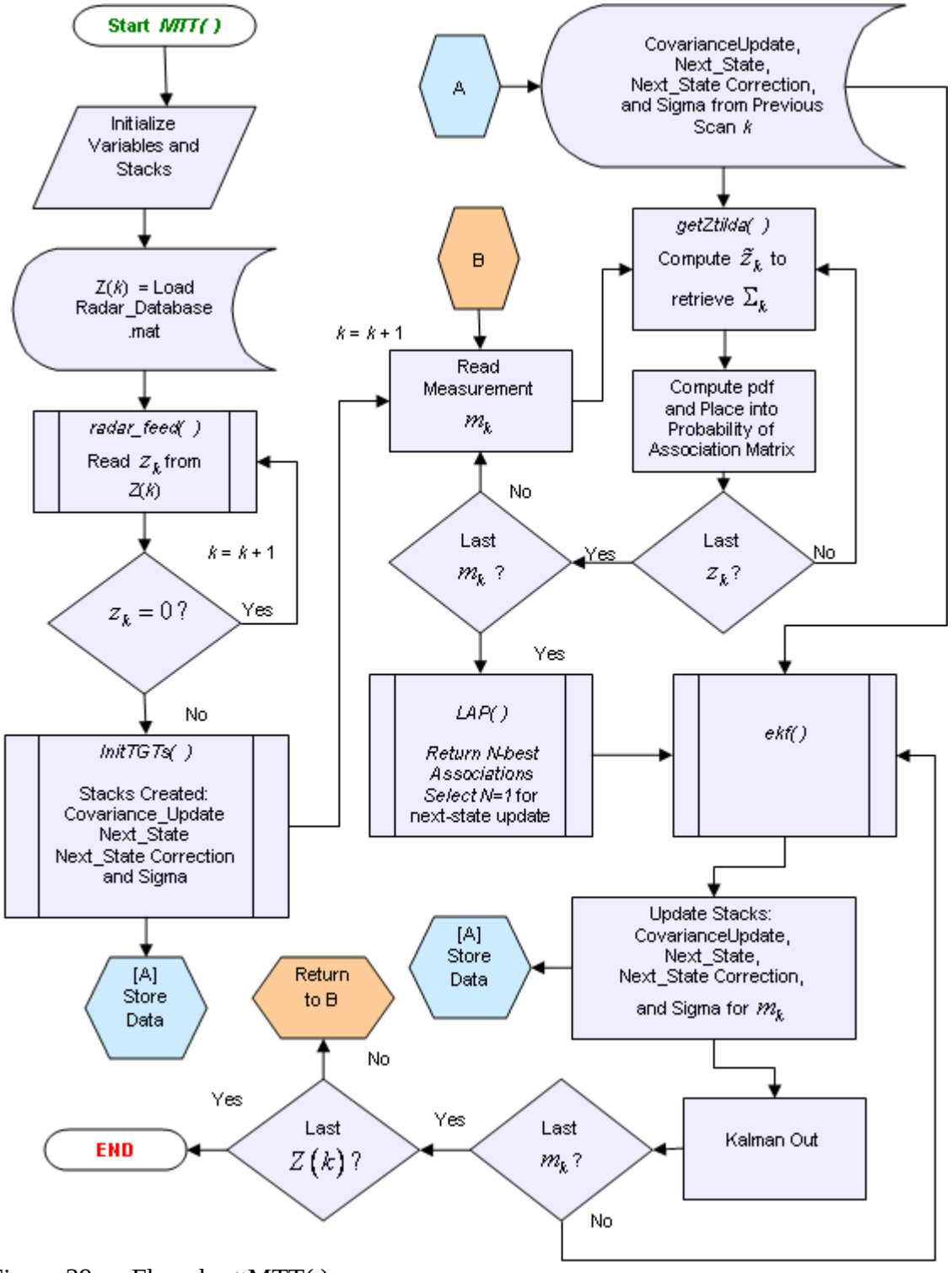


Figure 39. Flowchart MTT( )

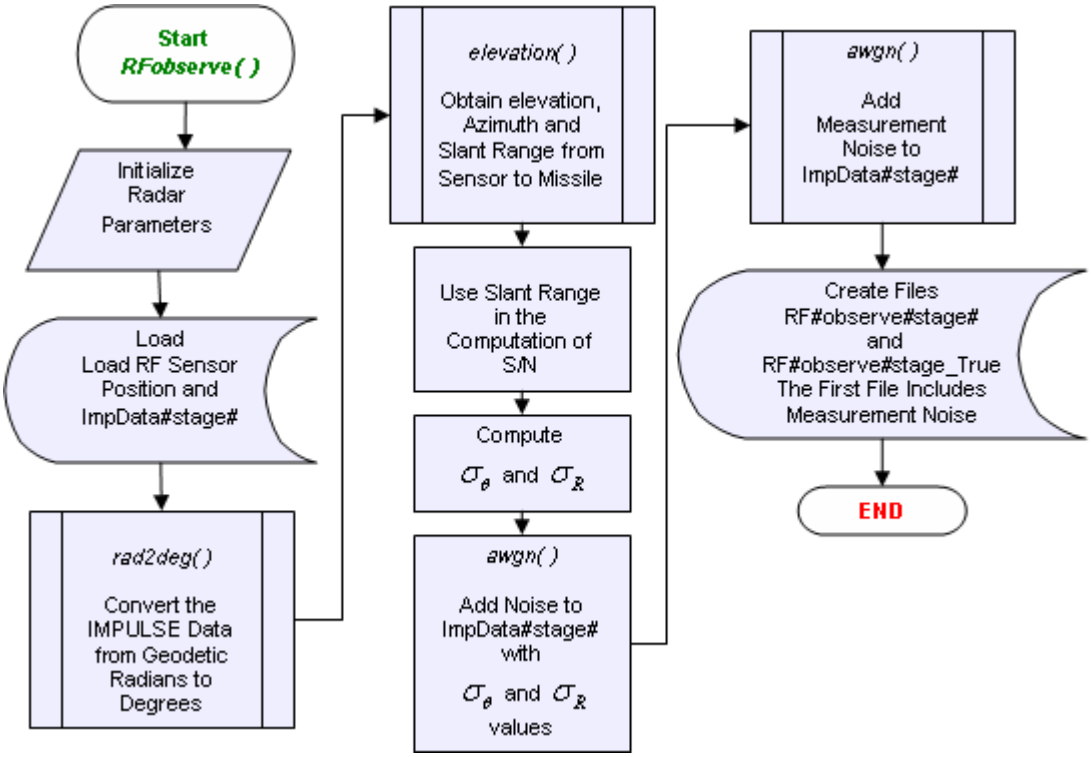


Figure 40. RFObserve( )

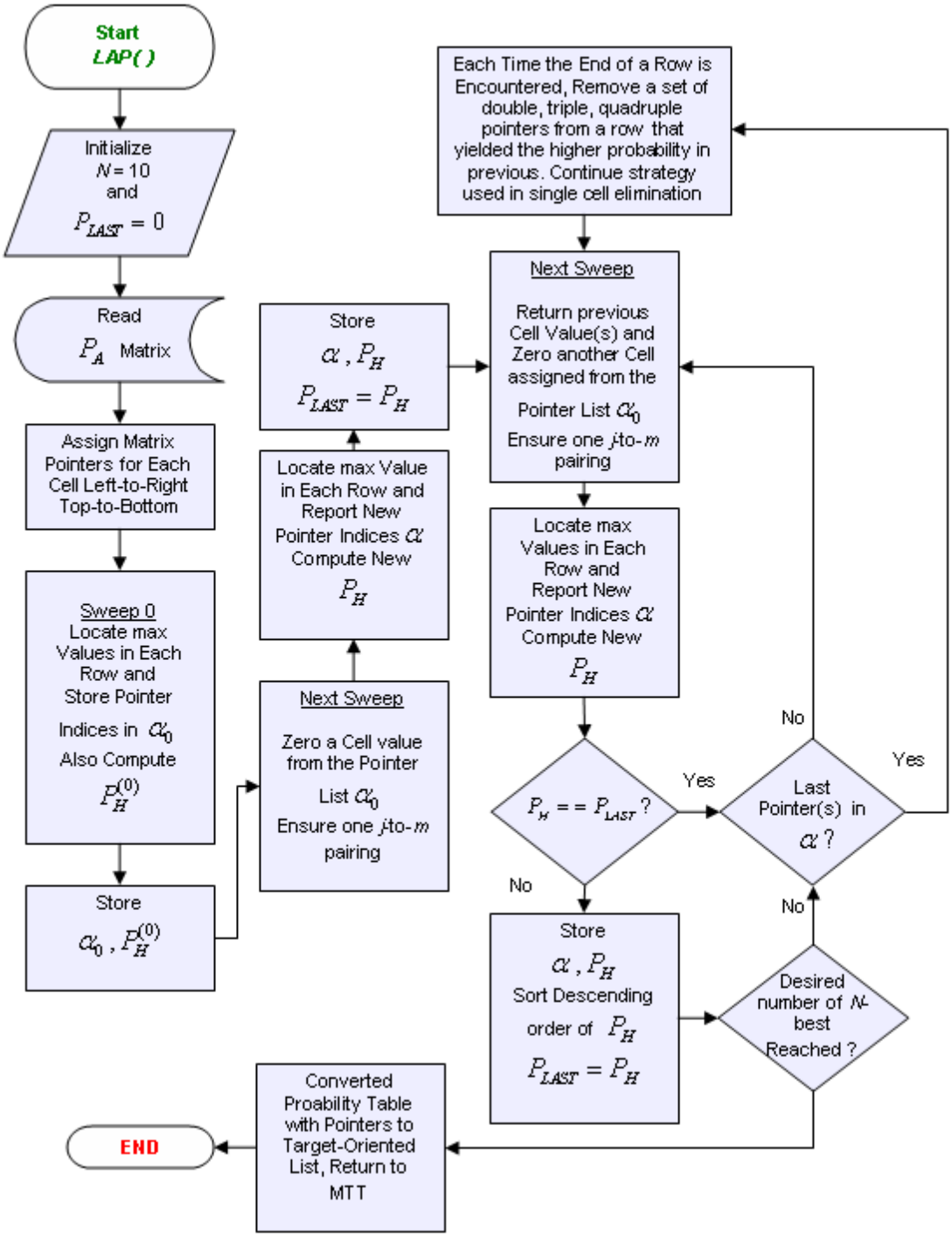


Figure 41. Flow Diagram for the Linear Assignment Problem

LIST OF REFERENCES

1. Lehner, R. (2006). In *MDA News Release*. Retrieved April 7, 2006 from http://www.mda.mil/mdalink/pdf/06news0016.pdf.

2. Oberg, J., "Missiles for all: The New Global Threat," IEEE Spectrum, Vol. 36, No. 3, pp. 20-28, 1999.

3. National Defense Industrial Association (NDIA), "T-Map Models," three-page handout, Third Simulation Based Acquisition Conference, May 2001.

4. Danchick, R., Newnam, G.E., *"A Fast Method for Finding the Exact N-best Hypotheses for Multitarget Tracking,"* IEEE Transactions on Aerospace and Electronic Systems, pp. 555-560, Vol. 29, No 2, 1993.

5. Serway, R. A. and Jewett, J.W., *Physics for Scientists and Engineers*, 6th ed., Brooks & Cole Pub. Co., 2004.

6. Kubilay Uzun, "Requirements and Limitations of Boost–Phase Ballistic Missile Intercept Systems," Master's Thesis, Naval Postgraduate School, Monterey, California, September 2004.

7. National Air & Space Intelligence Center (NASIC), "IMPULSE© User's Guide for MATLAB," 16-page document, April 2005.

8. Kursad Yildiz, "Electronic Attack and Sensor Fusion Techniques for Boost-Phase Defense against Multiple Ballistic Threat Missiles," Master's Thesis, Naval Postgraduate School, Monterey, California, June 2005.

9. Antonio P. San Jose, "Theatre Ballistic Missile Defense-Multisensor Fusion, Targeting and Tracking Techniques," Master's Thesis, Naval Postgraduate School, Monterey, California, March 1998.

10. Kolawole, M.O., *Radar Systems, Peak Detection and Tracking*, Newnes, Burlington, Massachusetts, 2002.

11. Mahafza, B.R., *Radar Systems Analysis and Design Using MATLAB*, Chapman & Hall/CRC, Boca Raton, Florida, 2000.

12. Edde, B., *Radar: Principles, Technology, Applications*, Prentice Hall PTR, Upper Saddle, New Jersey, 1993.

13. Filippo, N., *Introduction to Electronic Defense Systems*, Artech House, Norwood, Massachusetts, 2001.

14. D.B. Reid, “An Algorithm for tracking multiple targets,” IEEE Transaction on Automatic Control, Vol. AC-24, No. 6, December 1979.

15. Nagarajan, V., Chidambara, M.R., Sharma, R.N., “New Approach to Improved Detection and Tracking Performance in Track-While-Scan Radars,” *IEE Proceedings*, Vol. 134, Pt. F, No. 1, February, 1987.

16. Cox, I.J., Hingorani, S.L., “An Efficient Implementation of Reid’s Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 2, February 1996.

17. Bar-Shalom, Y., and Li, X., *Estimation and Tracking: Principles, Techniques, and Software*, Artech House, Inc., Norwood, Massachusetts, 1993.

18. Hutchins, R.G., EC3310 Class Notes, Naval Postgraduate School, Monterey, California, 1997.

19. Stone, LD., Barlow C.A., and Corwin, T.L., *Bayesian Multiple Target Tracking*, Artch House, Maryland, 1999.

20. Simmons, F.S., *Rocket Exhaust Plume Phenomonology*, Aerospace Press, El Segundo, California, 2000.

21. Blackman, S., *Multiple-Target tracking with Radar Application*, American Institute of Aeronautics and Astronautics, Inc., Washington, D.C., 1994.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dr. Murali Tummala
   Department of Electrical and Computer Engineering
   Monterey, California

4. Dr. Phillip E Pace
   Department of Electrical and Computer Engineering
   Monterey, California

5. Marine Corps Representative
   Naval Post Graduate School
   Monterey, California

6. Director, Training and Education, MCCDC Code C46
   Quantico, Virginia

7. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia

8. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)
   Camp Pendleton, California

9. Mr. Jeff Cole
   National Air and Space Intelligence Center
   Wright-Patterson AFB, Ohio

10. Ms. Carolyn Meiss
    National Air and Space Intelligence Center
    Wright-Patterson AFB, Ohio

11. Mr. Brian Sizemore
    National Air and Space Intelligence Center
    Wright-Patterson AFB, Ohio

12. Dr. Butch Caffall
Missile Defense Agency
Washington, DC

13. LTC Jason Stine
Missile Defense Agency
Washington, DC

14. LTC Thomas Cook
Naval Postgraduate School
Monterey, California

15. Dr. Bret Michael
Naval Postgraduate School
Monterey, California

16. Ms. Deborah Stiltner
Missile Defense National Team
Crystal City, Virginia

17. Mr. Mark Thornton
Missile Defense National Team
Huntsville, Alabama

18. Mr. Tim Trapp
Missile Defense National Team
Crystal City, Virginia

19. Capt Bert Rakdham
United States Marine Corps
Monterey, California