

Tactical and Strategic Missile Guidance

Fourth Edition

Paul Zarchan
MIT Lincoln Laboratory
Lexington, Massachusetts

Volume 199
PROGRESS IN
ASTRONAUTICS AND AERONAUTICS
A Volume in the AIAA Tactical Missile Series

Paul Zarchan, Editor-in-Chief
MIT Lincoln Laboratory
Lexington, Massachusetts

Published by the
American Institute of Aeronautics and Astronautics, Inc.
1801 Alexander Bell Drive, Reston, Virginia 20191-4344



To
Maxine, Adina, Ari and Ronit

MATLAB® is a registered trademark of The MathWorks Inc.

Copyright © 2002 by the American Institute of Aeronautics and Astronautics, Inc. Printed in the United States of America. All rights reserved. Reproduction or translation of any part of this work beyond that permitted by Sections 107 and 108 of the U.S. Copyright Law without the permission of the copyright owner is unlawful. The code following this statement indicates the copyright owner's consent that copies of articles in this volume may be made for personal or internal use, on condition that the copier pay the per-copy fee (\$2.00) plus the per-page fee (\$0.50) through the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, Massachusetts 01923. This consent does not extend to other kinds of copying, for which permission requests should be addressed to the publisher. Users should employ the following code when reporting copying from the volume to the Copyright Clearance Center:

1-56347-497-2/97 \$2.50 + .50

Data and information appearing in this book are for informational purposes only. AIAA is not responsible for any injury or damage resulting from use or reliance, nor does AIAA warrant that use or reliance will be free from privately owned rights.

ISBN 1-56347-497-2

Progress in Astronautics and Aeronautics

Editor-in-Chief

Paul Zarchan

MIT Lincoln Laboratory

Editorial Board

David A. Bearden

The Aerospace Corporation

Philip D. Hattis

Charles Stark Draper Laboratory

John D. Binder

The MathWorks, Inc.

Richard C. Lind

University of Florida

Steven A. Brandt

U.S. Air Force Academy

Richard M. Lloyd

Raytheon Electronics Company

Fred R. DeJarnette

North Carolina University

Ahmed K. Noor

NASA Langley Research Center

L. S. "Skip" Fletcher

NASA Ames Research Center

Albert T. Piccirillo

Institute for Defense Analyses

Ben T. Zinn

Georgia Institute of Technology



Table of Contents

Acknowledgments	xvii
Preface to the Fourth Edition	xix
Introduction	xxi
Chapter 1. Numerical Techniques	1
Introduction	1
Laplace Transforms and Differential Equations	1
Numerical Integration of Differential Equations	3
Z Transforms and Difference Equations	6
References	10
Chapter 2. Fundamentals of Tactical Missile Guidance	11
Introduction	11
What Is Proportional Navigation?	12
Simulation of Proportional Navigation in Two Dimensions	12
Two-Dimensional Engagement Simulation	15
Linearization	20
Linearized Engagement Simulation	22
Important Closed-Form Solutions	24
Proportional Navigation and Zero Effort Miss	27
Summary	28
References	28
Chapter 3. Method of Adjoints and the Homing Loop	31
Introduction	31
Homing Loop	31
Single Time Constant Guidance System	32
How to Construct an Adjoint	34
Adjoint Mathematics	37
Adjoints for Deterministic Systems	38
Deterministic Adjoint Example	40
Adjoint Closed-Form Solutions	43
Normalization	48
Summary	50
References	50
Chapter 4. Noise Analysis	51
Introduction	51
Basic Definitions	51

Gaussian Noise Example	54
Computational Issues	58
More Basic Definitions	60
Response of Linear System to White Noise	61
Low-Pass-Filter Example	61
Adjoints for Noise-Driven Systems	65
Shaping Filters and Random Processes	66
Example of a Stochastic Adjoint	70
Closed-Form Solution for Random Target Maneuver	75
Summary	76
References	77
Chapter 5. Covariance Analysis and the Homing Loop	79
Background	79
Theory	79
Low-Pass Filter Example	80
Numerical Considerations	81
Homing Loop Example	82
Acceleration Adjoint	91
Summary	94
References	94
Chapter 6. Proportional Navigation and Miss Distance	95
Introduction	95
System Order	95
Design Relationships	96
Optimal Target Evasive Maneuvers	104
Practical Evasive Maneuvers	106
Saturation	108
Parasitic Effects	110
Thrust Vector Control	116
Summary	118
References	118
Chapter 7. Digital Fading Memory Noise Filters in the Homing Loop	119
Introduction	119
Fading Memory Filters	119
Fading Memory Filter in Homing Loop	120
Mixed Continuous Discrete Adjoint Theory	127
Replace n by $N - n$ in the Arguments of All Variable Coefficients	127
Using Adjoints to Evaluate Filter Performance	130
Some Properties of Fading Memory Filters	134
Estimating Target Maneuver	136
Summary	140
References	140
Chapter 8. Advanced Guidance Laws	143
Introduction	143

Review of Proportional Navigation	143
Augmented Proportional Navigation	145
Derivation of Augmented Proportional Navigation	149
Influence of Time Constants	152
Optimal Guidance	155
Summary	161
References	161
Chapter 9. Kalman Filters and the Homing Loop	163
Introduction	163
Theoretical Equations	163
Application to Homing Loop	164
Kalman Gains	167
Numerical Examples	168
Experiments with Optimal Guidance	178
Summary	183
References	183
Chapter 10. Other Forms of Tactical Guidance	185
Introduction	185
Proportional Navigation Command Guidance	185
Beam Rider Guidance	193
Command to Line-of-Sight Guidance	201
Summary	204
References	205
Chapter 11. Tactical Zones	207
Introduction	207
Velocity Computation	207
Drag	210
Acceleration	214
Gravity	216
Summary	219
References	219
Chapter 12. Strategic Considerations	221
Introduction	221
Background	221
Gravitational Model	221
Polar Coordinate System	228
Closed-Form Solutions	233
Hit Equation	239
Flight Time	243
Summary	244
References	245
Chapter 13. Boosters	247
Introduction	247
Review	247

Staging	249
Booster Numerical Example	251
Gravity Turn	255
Summary	261
Reference	261
Chapter 14. Lambert Guidance	263
Introduction	263
Statement of Lambert’s Problem	263
Solution to Lambert’s Problem	264
Numerical Example	267
Speeding Up Lambert Routine	270
Booster Steering	274
General Energy Management (GEM) Steering	281
Summary	289
References	289
Chapter 15. Strategic Intercepts	291
Introduction	291
Guidance Review	291
Ballistic Engagement Simulation	293
Boosting Target Considerations	305
Summary	316
Reference	316
Chapter 16. Miscellaneous Topics	317
Introduction	317
Gravity Compensation	317
Predictive Guidance	320
Booster Estimation with Range and Angle Measurements	330
Pulsed Guidance	342
Chapter 17. Ballistic Target Properties	353
Introduction	353
Ballistic Target Model	353
Ballistic Target Experiments	354
Closed-Form Solutions for Ballistic Targets	359
Missile Aerodynamics	362
Intercepting a Ballistic Target	364
Summary	371
References	372
Chapter 18. Extended Kalman Filtering and Ballistic Coefficient Estimation	373
Introduction	373
Theoretical Equations	373
Differential Equation for One-Dimensional Ballistic Target	375
Extended Kalman Filter for One-Dimensional Ballistic Target	376
Numerical Example	379

Summary	387
References	387
Chapter 19. Ballistic Target Challenges	389
Introduction	389
Miss Distance Due to Noise	389
Fifth-Order Binomial Guidance System Miss Distances	393
Minimum Guidance System Time Constant	397
Missile Turning Rate Time Constant	397
Checking Minimum Guidance System Time Constant Constraints	399
Miss Due to Noise for Aircraft and Ballistic Targets	403
Summary	405
References	405
Chapter 20. Multiple Targets	407
Introduction and Background	407
Development of a Linear Model	407
Single Time Constant Guidance System	415
Higher-Order Guidance System Dynamics	422
Acceleration Saturation	426
Summary	431
References	431
Chapter 21. Weaving Targets	433
Introduction and Background	433
Weave Maneuver in Single Time Constant Guidance System	433
Closed-Form Solutions for Miss Distance	440
Higher-Order Guidance System Dynamics	444
Acceleration Saturation	448
Reducing the Time Constant to Improve Performance	451
Advanced Guidance Techniques to Improve Performance	453
Summary	460
References	460
Chapter 22. Representing Missile Airframe with Transfer Functions	461
Introduction	461
Force and Moment Equations	461
Airframe Simulation	465
Linearization of the Airframe	469
Numerical Example	472
Experiments	478
Summary	482
References	482
Chapter 23. Introduction to Flight Control Design	483
Introduction	483
Open-Loop Flight-Control System	483
Guidance System Interactions	489

Software Details	643
Integration Example	644
Pursuit Guidance	645
Sensitivity of Optimal Guidance to Time to Go Errors	649
Alternative Formulation of Radome Effects	652
Another Way of Generating Random Numbers	654
Simulating an Impulse	655
Different Guidance System Distributions	659
Fading Memory Filters for Booster Estimation	663
Adjoint of Discrete Inputs	672
Sampling Experiments	675
Brute Force Frequency Response	678
Why Stability Margins Are Important	683
Minimum Energy Trajectories	689
References	695
Appendix B. Converting Programs to C	697
Appendix C. Converting Programs to MATLAB®	703
Appendix D. True BASIC	715
Reference	720
Appendix E. Units	721
Listing Index	723
Index	727

Rate Gyro Flight-Control System	490
Open-Loop Transfer Function	496
Time Domain Verification of Open-Loop Results	500
Simplified Expression for Open-Loop Crossover Frequency	504
Summary	506
References	506
Chapter 24. Three-Loop Autopilot	507
Introduction	507
Three-Loop Autopilot Configuration	507
Open-Loop Analysis	507
Closed-Loop Analysis	510
Experiments with Flight Condition	523
Guidance System Analysis	527
Summary	539
References	539
Chapter 25. Trajectory Shaping Guidance	541
Introduction	541
Problem Setup	541
Using the Schwartz Inequality for Trajectory Shaping Guidance	543
Alternate Form of Trajectory Shaping Guidance Law	547
Testing Trajectory Shaping Guidance in the Linear World	548
Closed-Form Solutions	554
Nonlinear Results	560
Summary	568
References	569
Chapter 26. Filtering and Weaving Targets	571
Introduction	571
Review of Original Three-State Linear Kalman Filter	571
Four-State Weave Kalman Filter	579
Miss Distance Analysis	592
Extended Kalman Filter	595
Summary	612
References	612
Chapter 27. Predictor–Corrector Guidance	613
Introduction	613
Surface-to-Surface Missile Problem	613
Predictor–Corrector Guidance With Surface-to-Surface Intercept Problem	620
Roll Angle or Rate Guidance Problem	627
Predictor–Corrector Guidance With Roll-Rate Problem	631
Summary	641
References	641
Appendix A. Tactical and Strategic Missile Guidance Software	643
Introduction	643



Acknowledgments

Although many of the devices discussed in *Tactical and Strategic Missile Guidance* work in a vacuum, I do not. In this section I would like to take the opportunity to acknowledge those individuals who contributed in some way to the content and publication of this text.

First, I would like to thank Dr. Donald C. Fraser, former Executive Vice President of The Charles Stark Draper Laboratory, Inc. (CSDL) for recommending me as an author for the AIAA Progress in Astronautics and Aeronautics series. He knew of my desire to write a book on missile guidance and through his efforts the dream is now a reality. During the last 16 years, Mr. John M. Elwell, former Guidance, Navigation & Control Director at CSDL, provided me with numerous technical opportunities that expanded my missile guidance background. The technical thread running throughout the text is based on those technical opportunities. The highly competent staff and superb technical environment of CSDL provided the atmosphere that not only made this project possible but also fun. In particular, I would like to thank CSDL technical staff members Dr. Owen Deutsch, Mr. Steven L. Nelson, and Dr. Richard E. Phillips for our many valuable technical discussions. Much of the material in the strategic interceptor chapters was a consequence of our valuable technical interchanges. Some of the material in several of the chapters pertaining to tactical missile guidance was based on knowledge gained during my employment at Raytheon. Here, I not only learned tactical interceptor guidance and control fundamentals from the people who first made radar homing missiles a reality, but also learned of and developed an appreciation for the hard work involved in converting theory to something that works.

Next, I would like to acknowledge those associated with AIAA responsible for the inception and completion of this project. I would like to thank Dr. Allen E. Fuhs, former AIAA Progress in Astronautics and Aeronautics Editorial Board Member, for inviting me to participate in the Progress Series and then helping me formulate an outline from which a manuscript could be written. His kind and constructive review of the initial draft led to numerous changes, all of which improved the overall quality of the text. Mr. John Newbauer, former AIAA Administrator of Scientific Publications, provided me with excellent examples of the standards expected of a Progress Series volume and, in addition, helped outline the preface. Mrs. Jeanne Godette, former AIAA Director of Book Publications, and her very competent staff not only taught me a great deal about grammar but, in addition, moved this project forward very rapidly.

Finally, I wish to express my love and gratitude to my wife Maxine for having patience and understanding, while our home fell into a state of disrepair, for not complaining while I put in the many hours required in writing this text.



Preface to the Fourth Edition

The fourth edition has three new chapters on important topics related to missile guidance, additional examples in Appendix A, and an additional appendix. The first new chapter (Chapter 25) shows how the guidance problem can be reformulated so that a new guidance law can be derived that both hits the target using minimum energy and, in addition, travels on a desired trajectory. This type of guidance is important in applications in which the trajectory must be shaped for either sensor or lethality considerations. Intentional or unintentional spiraling maneuvers on the part of a tactical ballistic missile can also make it particularly difficult for a pursuing missile to hit. The third edition discussed ways in which performance against weaving or spiraling targets could be improved from a guidance point of view. The second new chapter (Chapter 26) explores the various filtering options that can be used to estimate the necessary states for using advanced guidance techniques. As with other material in the text, this chapter uses a step-by-step approach in explaining and deriving the various filtering options. The third new chapter (Chapter 27) approaches the guidance problem from a totally different perspective and one that is often used by the space community. This new chapter shows how a predictor-corrector type of method can be applied to a problem in which other, more conventional guidance laws are also applicable. It is demonstrated how this different technique allows the designer to put additional constraints on the guidance problem. These three new chapters are based on material from the AIAA Continuing Education short course "Fundamentals of Tactical and Strategic Missile Guidance" that I have been teaching since 1990. The new material in this edition was omitted from previous editions because of page limitation and time constraints. The new topics are treated from both analytical and simulation points of view so that readers with different backgrounds and learning styles can benefit from the new material.

As with the third edition, the FORTRAN, C, and MATLAB[®] source code is still included with the book on a CD-ROM that can be used by both Macintosh and IBM-compatible computers. In addition, as a special feature of the fourth edition, the True BASIC language equivalents of the FORTRAN source code listings are now also included on the same CD-ROM. This low-cost compiler is appropriate for those individuals on a limited budget because, at the time of publication, the student edition is free and the full-featured version is only \$35.

Appendix A still serves as a user's guide for the software and has been expanded to include several more interesting examples of how the source code can be modified to explore issues beyond the scope of the text. The new examples in Appendix A were designed to also be of interest to those readers who have no interest in the source code. Appendix D has been added to show the interested reader how the FORTRAN source code was converted to True BASIC.

On a personal note, it continues to be very gratifying for me to learn that many people working with or having to learn about missile guidance have found *Tactical and Strategic Missile Guidance* useful. Over the years, many readers have contacted me and asked questions when the book's material was not clear to them. Material in the existing chapters has been clarified so that all will benefit from my

interaction with the readers. It is still my hope that this fourth edition, with its new chapters and additional software, will be of value not only to new readers, but will also be worthwhile to those who have already read the previous editions.

Paul Zarchan
September 2002

Introduction

The requirements for tactical missile guidance systems were born at the end of World War II as a result of the highly effective kamikaze attacks on U.S. vessels. After the war it was clear that naval guns using unguided shells were not adequate for shooting down hostile aircraft making suicidal attacks against U.S. ships. To counter kamikaze-like threats, the U.S. Navy initiated the development of the Lark guided missile in 1944. Approximately six years of hard work was required before the first successful intercept, made by the Lark against an unmanned aircraft, occurred on Dec. 2, 1950. A very readable history of this significant event, told by one of the engineering participants, can be found in a work by M. W. Fossier ("The Development of Radar Homing Missiles," *Journal of Guidance, Control, and Dynamics*, Vol. 7, No.6, 1984, pp. 641–651).

Operational endoatmospheric tactical missile systems, using the same guidance principles as those used by the Lark, have now been in existence for approximately five decades. These missiles use guidance concepts that work well not only against stationary or predictable targets but also are effective against responsive threats (i.e., aircraft executing evasive maneuvers) whose future position is highly uncertain. In the tactical arena current guidance law technology appears to be adequate if the flight time is long compared to the effective time constant of the guidance system and if the missile enjoys a considerable acceleration advantage over the target. In fact, it is not uncommon for a tactical missile to have an acceleration advantage of more than five against an aircraft target. This advantage is more than adequate for a successful intercept with current guidance law technology.

Strategic ballistic missiles are different from tactical guided missiles because they travel much longer distances and are designed to intercept stationary targets whose location is known precisely. A tactical homing missile acquires the target with a seeker and then guides all the way to intercept, whereas a strategic interceptor does not require a seeker since all of its guidance is in the boost phase. Since the boost phase represents only a small fraction of the total flight of a strategic ballistic missile, the interceptor glides most of the way toward the stationary target. In this type of strategic application, precise instrumentation is necessary so that the interceptor can steer to the correct position and velocity states at the end of the boost phase. With the correct states the missile will be able to glide ballistically, without further corrective maneuvers, toward the target.

The concept of long-range strategic ballistic missiles was also born during World War II with the highly effective V-2 rocket developed by German scientists at Peenemünde. The first combat round, launched on Sept. 6, 1944, represented the culmination of nearly 14 years of work (G. P. Kennedy, *Rockets, Missiles, and Spacecraft of the National Air and Space Museum*, Smithsonian Institution Press, Washington, DC, 1983). The range of the V-2 was only a few hundred miles, and a 2000-lb warhead compensated somewhat for its lack of accuracy. After World War II some of the German scientists from Peenemünde were brought to the United States, along with 80 missiles, in order to advance the V-2 technology. Guidance accuracy was improved by more than an order of magnitude by incorporating precise inertial sensors into the guidance system design. The short-range Redstone missile became the first U.S. ballistic missile with a highly accurate inertial guidance system. More precise and smaller inertial sensor hardware and improved

guidance software were achieved with the U.S. medium-range Jupiter missile. An eyewitness account of this advance in strategic missile technology can be found in a work by W. Haeussermann ("Developments in the Field of Automatic Guidance and Control of Rockets," *Journal of Guidance, Control, and Dynamics*, Vol. 4, No. 3, 1981, pp. 225–239). Today's highly accurate strategic ballistic missiles travel at intercontinental ranges at near-orbital speeds.

In newer systems, such as those being considered by the Ballistic Missile Defense Organization (BMDO), interceptors will have to make use of and improve on yesterday's tactical and strategic missile technology. Tomorrow's interceptors will have to fly strategic distances against moving and possibly accelerating targets whose future position is highly uncertain. In these applications it is not sufficient to apply only ballistic missile technology. Some type of homing guidance system is required after the interceptor boost phase to take out the inevitable errors. However, it is not clear whether tactical guidance law technology is appropriate for the homing phase of these strategic interceptors. Unlike tactical endoatmospheric interceptors, which maneuver by generating lift, the newer exoatmospheric interceptors require fuel for the missile divert engines in order to maneuver or divert in response to guidance commands. If all of the divert fuel is consumed, the interceptor cannot maneuver. In addition, fuel is at a premium, because, according to the rocket equation, interceptor weight grows exponentially with fuel weight. Therefore, unlike tactical endoatmospheric missiles, the newer exoatmospheric interceptors require guidance laws that minimize fuel consumption. In addition, because of practical limits on achievable divert engine thrust-to-weight ratios, the newer interceptors may no longer enjoy the tactical missile's acceleration advantage over the target. In fact, strategic exoatmospheric interceptors may be working at an *acceleration disadvantage* against a booster target!

With this perspective, this book attempts to lay the foundation for meeting today's new challenges. The principles of both tactical and strategic missile guidance are presented in a common language, notation, and perspective. The mathematics, arguments, and examples presented in the text are intended to be nonintimidating so that the designers working in the tactical world will be able to understand and appreciate the difficulty of the strategic problem and vice versa. Numerous examples are presented to illustrate all of the concepts presented in the text. In later chapters of the book, examples are presented showing elementary ways in which tactical and strategic guidance principles can be combined. In this way the potential for a cross-fertilization of ideas, which is necessary for today's challenges in guidance technology, is increased.

The book is written with both the expert and novice in mind. Proven methods of guidance in both the tactical and strategic world are presented from several points of view. The guidance laws are usually first derived mathematically, then explained from a heuristic point of view, and finally a numerical example, along with a FORTRAN listing, is included to prove that the guidance law performs as expected. The numerical example is usually chosen to highlight the strengths and weaknesses of the guidance approach. The expert or more interested reader, using the FORTRAN listing with a suitable compiler, cannot only get a deeper understanding of the principles involved but can also explore issues beyond the scope of the text. All of the graphs presented in the text were generated with the FORTRAN listings.

It might appear from glancing at the many block diagrams scattered throughout the text that the subject matter requires a control system background. *This is not true!* The block diagrams are offered as a pictorial description of differential equations. The reader can always ignore the block diagrams and just read the differential equations. In fact, relief from block diagrams can be found in Chapters 11–19. The only assumption made in the text is that the reader has either an engineering or physics background. All of the numerical and mathematical techniques used in the text are explained in earlier chapters.

Chapter 1 presents the basis of most of the numerical techniques used throughout the text. The utility of Laplace transforms in solving and representing differential equations is explained. It is then shown how differential equations can be solved using the second-order numerical Runge–Kutta integration technique. FORTRAN code for a representative example is presented so that the reader can better understand how numerical integration is actually applied. The code is written so that it is easy to apply to problems involving other differential equations without changing more than a few lines of code. In fact, most of this sample code is repeated throughout the text when the differential equations representing the system under consideration change.

Chapter 2 introduces the reader to the most important and widely used tactical missile guidance law: proportional navigation. A simplified engagement simulation is developed in order to show how proportional navigation operates and why it is an effective guidance law. Next, linearization techniques are used to get a deeper understanding of how proportional navigation works and to form an analytical foundation from which guidance theory can be advanced. Closed-form solutions for the required missile acceleration due to heading error and target maneuver disturbances are derived so that the reader can see how these important errors influence total system performance. It is then demonstrated that the derived closed-form solutions are not only analytically convenient but are in fact accurate indicators of the expected system performance.

Chapter 3 introduces one of the most important methods used in analyzing tactical missile guidance systems: the method of adjoints. The rules for constructing an adjoint are presented, and the necessary mathematics required to understand why adjoints are useful are included in this chapter. A numerical example involving a missile guidance system is used both as a practical application of the theory and to ensure that the reader fully understands how to apply the adjoint technique. The numerical results of the adjoint example are compared to traditional simulation results so that the reader can appreciate both the power and utility of this very elegant technique. New closed-form solutions are derived using adjoints so that the reader can also begin to appreciate the relationship between miss distance, the guidance system time constant, and the error source.

Chapter 4 reviews all of the necessary theory so that the reader can understand how measurement noise and random phenomenon influence the performance of a missile guidance system. Basic definitions are reviewed, and numerical examples are presented showing the reader how theory is used in practice. The method of adjoints, which was introduced in Chapter 3 for deterministic systems, is extended so that a missile guidance system can be analyzed if the error sources are random. A numerical example is presented showing how the method of adjoints can be used to obtain statistical performance projections in only *one computer run*. These results

are compared to multiple-run Monte Carlo projections (i.e., repeated simulation trials with ensemble averaging of the resultant output data).

Another popular computerized analytical technique, known as covariance analysis, is introduced in Chapter 5. It is shown that covariance analysis can also be used to yield exact statistical performance projections of a missile guidance system in the presence of random error sources in one computer run. The numerical requirements for the successful implementation of covariance analysis are discussed and compared to the adjoint method. Chapter 5 also shows how the adjoint technique can be extended to yield acceleration as well as miss distance information.

Chapter 6 presents most of the important properties of a proportional navigation guidance system. It is shown that an accurate dynamic model of the guidance system is crucial in obtaining accurate miss distance performance projections. Normalized design curves are developed for a high-order canonical guidance system so that an engineer can quickly estimate system performance given a minimum amount of information. The concept of an optimal target evasive maneuver is developed. It is shown how one can compute the maximum miss distance that a maneuvering target can induce. This type of information is important in determining the vulnerability of a guidance system. Finally, the influence of saturation and parasitic effects on system performance are considered. Design curves are presented showing how to size the missile to target acceleration advantage so that the interceptor will be effective. It is also shown in this chapter that parasitic effects such as radome place fundamental limits on the attainable speed and gain of the guidance system.

The missile seeker provides a noisy measurement of the line-of-sight angle. Chapter 7 demonstrates how simple digital fading memory filters can be used as part of a missile guidance system to provide an estimate of the line-of-sight angle and rate from the noisy measurement. Adjoint theory is again extended to handle systems with continuous and digital parts so that a mixed continuous-discrete missile guidance system can be analyzed efficiently. Some of the more important properties of fading memory filters are illustrated via numerical examples. This chapter also shows that, if range measurements are available, it is possible to estimate target acceleration with a fading memory filter.

Chapter 8 reviews proportional navigation concepts and sets up the mathematical foundation so that more advanced tactical guidance laws can be derived. Augmented proportional navigation is derived from some properties of the Schwartz inequality. It is then shown, via a numerical example, how guidance system time constants can degrade miss distance performance of both proportional and augmented proportional navigation. Recognizing the cause of the miss distance degradation, a new guidance law is derived. It is demonstrated that, as long as the missile guidance system dynamics is known, the new guidance law can effectively eliminate miss distance degradation due to a guidance system time constant.

Chapter 9 introduces a class of optimal digital noise filters known as Kalman filters. Using the theoretical Kalman filtering equations, the chapter develops, in detail, a digital Kalman filter that is very useful for missile guidance system applications. It is shown, in detail, how a Kalman filter can be used in conjunction with an optimal guidance law to improve system performance and to relax missile acceleration requirements. Various experiments are conducted in order to illustrate important filtering and guidance concepts.

Chapter 10 introduces other forms of tactical guidance. A command guidance implementation of proportional navigation is compared to a homing guidance implementation in terms of system noise propagation. Beam riding and command to line-of-sight methods of guidance are introduced and compared to proportional navigation. It is shown that, although the performance of these new methods of guidance are geometry-dependent, they can be made to work rather effectively.

Chapter 11 wraps up the discussion of tactical interceptors with consideration of the missile's operational zone. Reach considerations, based on the rocket equation and drag effects, are presented. Gravitational effects, which were previously neglected, are offered as another phenomenon that further limits the zone. Numerical examples are used to illustrate important effects considered in this chapter.

Chapter 12 introduces strategic interceptor concepts from a tactical point of view. A gravitational model, based on Newton's law of universal gravitation, is developed for strategic flight. Comparisons between a flat-Earth gravitational model and a strategic gravitational model are made. Although strategic engagement simulation models are presented in the text in a Cartesian Earth-centered coordinate system, a polar coordinate system is also introduced so that important closed-form solutions can be derived. Key formulas for velocity and flight time for an impulsive ballistic missile to travel a fixed distance, given an initial flight-path angle, are developed using the polar coordinate system. A Cartesian Earth-centered simulation is used to confirm the analytical results.

Chapter 13 shows how preliminary strategic booster sizing can be done with the rocket equation. Simplified booster sizing examples are presented in order to clarify the concepts. The rocket equation is extended so that the virtues of staging can be illustrated via a numerical example. Finally, the gravity turn maneuver is introduced as the simplest possible steering method a booster can employ in traveling from its launch point to a desired destination.

Starting from the closed-form solutions derived in Chapter 12, the concepts of Lambert steering, which is fundamental to booster and spacecraft steering, are developed in Chapter 14. A simple to understand but numerically inefficient way of solving Lambert's problem is derived. A numerical example is presented showing how the numerical solution to Lambert's problem can be implemented. A novel use of the secant method is demonstrated to speed up the solution to Lambert's problem by more than two orders of magnitude! It is then shown how the implemented solution can be modified with a simple feedback scheme to steer an interceptor, during its boost phase, to its intended target. Another subset of Lambert steering, known as general energy management (GEM) steering, is also derived and demonstrated. A numerical example highlighting the similarities and differences between Lambert and GEM steering is presented.

Chapter 15 shows elementary but fundamental methods of combining the tactical missile guidance concepts of Chapters 2–10 with the strategic notions of Chapters 12–14. Unifying numerical examples are used to illustrate the strengths and weaknesses of the combined approach. Previously derived closed-form solutions for the required missile acceleration to hit a target by a tactical interceptor are converted to strategic lateral divert formulas. Nonlinear strategic engagement simulation results are used to show that the divert formulas for prediction error, apparent target maneuver, and guidance law are not only useful because of their simplicity but are in fact accurate indicators of strategic interceptor requirements.

Chapter 16 presents some additional concepts that are very important to the strategic world. It is shown how compensating for known gravity effects in the guidance law can considerably reduce interceptor lateral divert requirements. Next, predictive guidance is introduced as the ultimate guidance law. It is shown that, if accurate a priori information exists concerning the target, then predictive guidance can be used to substantially alleviate interceptor lateral divert requirements. All of the homing guidance concepts introduced have assumed that strategic interceptor divert engines were effectively throttleable. A pulsed guidance law is developed assuming that guidance commands can only be issued with a few discrete burns. The performance of the pulsed guidance law is compared to that of proportional navigation.

In Chapter 17, the focus of the text switches to endoatmospheric ballistic targets. Closed-form solutions are derived and validated, based on the properties of ballistic targets, showing how the magnitude of endoatmospheric ballistic target deceleration varies with speed, altitude, and re-entry angle. It is demonstrated that unless advanced guidance techniques are used, the high deceleration levels of a ballistic target make it difficult to hit under all engagement conditions.

Advanced guidance techniques and fire control logic for endoatmospheric intercepts require knowledge of the target's ballistic coefficient. Using a simplified extended Kalman filter as an example, the challenges of estimating a target's ballistic coefficient are demonstrated in Chapter 18. Common filter design pitfalls and their engineering fixes are illustrated in easy-to-understand examples. Guidelines for making an extended Kalman filter robust to large initialization errors are presented.

Chapter 19 integrates many of the text's concepts to further explain why endoatmospheric ballistic targets are challenging. Formulas are derived showing the geometry dependence of noise-induced miss distance. The dependence of the minimum achievable guidance system time constant on radome slope, geometry, and missile aerodynamic properties is shown. Numerical examples are presented in order to highlight missile performance differences for both aircraft and ballistic target threats.

When two aircraft are flying in close formation, they both appear within the pursuing interceptor's seeker beam. When the missile is close enough to the aircraft, one of the targets will drop out of the seeker beam, and it will appear to the missile as if the target has moved instantaneously. Large miss distances may result because of insufficient remaining homing time. Chapter 20 develops normalized design curves to both illustrate and quantify the multiple target problem. Rules of thumb are developed relating the necessary ratio of the time left for homing after resolution has taken place to the guidance system time constant.

It is well known that the barrel roll or weave maneuver can cause large miss distances. Because tactical ballistic missiles can spiral or weave into resonance as they re-enter the atmosphere due to either mass or configurational asymmetries, the weave maneuver is of particular interest to the guidance system designer. Chapter 21 studies the influence of the target weave maneuver on a proportional navigation guidance system. This chapter demonstrates how the target weave frequency and amplitude, the missile guidance system time constant, effective navigation ratio, and acceleration capability all play an important role in determining system performance. Methods for improving system performance are also discussed.

To design a flight-control system the interceptor airframe equations must first be linearized. Chapter 22 shows how the nonlinear missile force and moment equations are related to the geometry of the missile airframe. A simple method for linearizing the force and moment equations is introduced so that transfer functions can be derived for the missile airframe. It is shown that the transfer function approximation to the airframe is an excellent match to reality for small angles of attack.

Chapter 23 illustrates how the flight-control system interacts with the guidance system. Both the open-loop and rate gyro flight-control systems are discussed in detail. It is shown that the open-loop flight-control system has the dynamics of the bare airframe and is therefore not usually acceptable in radar homing applications because of its low damping. It is demonstrated that the rate gyro flight-control system improves the system damping by using a sensor and the principles of feedback.

Chapter 24 demonstrates that by using an accelerometer and rate gyro with the flight-control system, the system damping, time constant, and open-loop crossover frequency can be controlled independently. Controlling the system damping ensures that the guidance system is not overly sensitive to radome slope effects at the high altitudes. Selecting the system time constant means that there will be adequate performance against maneuvering targets. Controlling the open-loop crossover frequency ensures that the design will be robust and not overly sensitive to unmodeled high frequency dynamics. A unifying example is presented to demonstrate how typical guidance system tradeoffs are conducted. The method of adjoints is used to show how miss distance error budgets can be generated, actuator requirements set, and missile acceleration requirements derived.

In all the guidance work done thus far, the goal has been to hit the target using the least amount of energy. In some applications, in addition to hitting the target, it may also be desirable to shape the missile trajectory near impact. For example, in antitank or antiballistic missile applications we may want to have the missile approach the target at certain strike angles to improve lethality. Chapter 25 shows how the guidance problem can be reformulated so that a new guidance law can be derived that hits the target using minimum energy and, in addition, travels on the desired trajectory.

The text has already shown that advanced guidance laws derived in Chapter 21 can be used to improve system performance against spiraling targets. Chapter 26 explores the various filtering options that can be used to estimate the weaving target states required for advanced guidance techniques. After a brief review of existing filtering techniques for the weaving target problem, it is assumed that the target weave frequency is known and an optimal linear four-state weave Kalman filter, which estimates both target acceleration and jerk, is both derived and evaluated. It is then assumed that the target weave frequency is not known but must be estimated along with target acceleration and jerk using an extended five-state Kalman filter. The chapter compares the various candidate Kalman filter options and appropriate guidance laws in terms of both performance and robustness.

Finally, for completeness, in Chapter 27 an alternative guidance approach used by the space community is developed and compared with the more traditional guidance approaches of the text. It is first shown how the predictor-corrector method can be applied to a problem in which other, more conventional guidance

laws are also applicable. The various guidance approaches are compared in terms of performance and robustness. A problem is also selected in which the direct application of a proportional navigation type guidance law is not possible. The strengths and weaknesses of predictor-corrector guidance approach are demonstrated for an unusual example.

This text attempts to present many of the important guidance principles involved in enabling an interceptor to hit its intended target. The utilities of these principles are explained and demonstrated with pictures, equations, and computer code. However, missing from the text is the intensity, challenge of the unknown, and plain hard work that it takes in going from a paper design to something that not only flies but also meets the system objectives. In light of the new challenges facing today's guidance engineer, it is very appropriate to quote from the work by M. W. Fossier:

On joining Raytheon as a young engineer, I found an intensity of spirit that I had never experienced before. I felt myself carried along in what seemed almost a crusade, sharing a burning commitment to succeed against an immense challenge.

As a result of this shared feeling, each triumph led to a broad-based feeling of great elation. I still vividly recall the first flight test by Lark against a low-flying drone aircraft in 1951 over the ocean at Point Mugu. The intercept was a relatively short range and was the first to be in full view of the handful of observers permitted on the beach. When the missile homed unerringly to a spectacular direct hit on the drone, the human explosion matched the one in the air. The formal celebration ended in the wee hours of the morning, but the emotional wave lasted for months.

On the other hand, I can still feel the despair that resulted when a technician inadvertently connected B+ (250 V) to the filament string, blowing out every tube in an early Sparrow being prepared for flight test. At that time, the best flight test engineers we had took about a month of 12-hour days to check out a missile. The resulting delay was felt at a personal level in every corner of the organization.

For years I attributed this intensity to the relative youth of most of the participants. However, I found that age has served mainly to mute their exuberance, but not their intensity, and I was forced to seek another source. My current view is that the intensity was (and is) a result of the intellectual challenge of the unknown. It is the characteristic of the engineering profession that there are always new problems waiting in the wings to replace the old ones as they are solved. The constant element is the challenge itself, which demands that we do our utmost and rewards us only when we do.

Numerical Techniques

Introduction

THE numerical techniques introduced in this chapter involve the use of Laplace transforms for manipulating and displaying differential equations and numerical integration for solving the differential equations. These techniques form the basis of all of the numerical methods used throughout the text. A numerical example will be presented that will illustrate a practical application of the use of Laplace transforms and numerical integration. Another example will be presented showing how z transforms can be used to both represent difference equations and get their solution.

Laplace Transforms and Differential Equations

Transform methods are often useful because certain operations in one domain are different and often simpler than operations in the other domain. For example, ordinary differential equations in the time domain become algebraic expressions in the s domain after being Laplace transformed. In control system engineering Laplace transforms are used both as a shorthand notation and as a method for solving linear differential equations. In this text we will frequently use Laplace transform notation to represent subsystem dynamics in tactical missile guidance systems.

If we define $F(s)$ as the Laplace transform of $f(t)$, then the Laplace transform has the following definition:

$$F(s) = \int_0^{\infty} f(t)e^{-st} dt$$

With this definition it is easy to show that a summation in the time domain is also a summation in the Laplace transform or frequency domain. For example, if $f_1(t)$ and $f_2(t)$ have Laplace transforms $F_1(s)$ and $F_2(s)$, respectively, then

$$\mathcal{L}[f_1(t) \pm f_2(t)] = F_1(s) \pm F_2(s)$$

Again, using the definition of the Laplace transform, it is easy to show that differentiation in the time domain is equivalent to frequency multiplication in the Laplace transform domain, or

$$\mathcal{L}\left(\frac{df(t)}{dt}\right) = sF(s) - f(0)$$

where $f(0)$ is the initial condition on $f(t)$. The Laplace transform of the n th derivative of a function is given by

$$\mathcal{L}\left(\frac{d^n f(t)}{dt^n}\right) = s^n F(s) - s^{n-1} f(0) - s^{n-2} \frac{df(0)}{dt} - \dots$$

From the preceding equation we can see that, for zero initial conditions, the n th derivative in the time domain is equivalent to a multiplication by s^n in the Laplace transform domain.

Laplace transforms can also be used to convert the input-output relationship of a differential equation to a shorthand notation called a transfer function representation. For example, given the second-order equation

$$\frac{d^2 y(t)}{dt^2} + 2 \frac{dy(t)}{dt} + 4y(t) = x(t)$$

with zero initial conditions, or

$$\frac{dy(0)}{dt} = 0, \quad y(0) = 0$$

we can find the same differential equation in the Laplace transform domain to be

$$s^2 Y(s) + 2s Y(s) + 4Y(s) = X(s)$$

Combining like terms in the preceding equation to get a fractional relationship between the output and input, known as a transfer function, yields

$$\frac{Y(s)}{X(s)} = \frac{1}{s^2 + 2s + 4}$$

Similarly, given a transfer function, we can go back to the differential equation form. Consider the second-order transfer function

$$\frac{Y(s)}{X(s)} = \frac{1 + 2s}{1 + 2s + s^2}$$

We know that, according to the chain rule, the transfer function can be expressed as

$$\frac{Y(s)}{X(s)} = \frac{E(s)}{X(s)} \frac{Y(s)}{E(s)}$$

Therefore, we can break the relationship into the following two equivalent transfer functions:

$$\frac{E(s)}{X(s)} = \frac{1}{1 + 2s + s^2}, \quad \frac{Y(s)}{E(s)} = 1 + 2s$$

Cross multiplication results in

$$s^2 E(s) + 2s E(s) + E(s) = X(s)$$

and

$$2s E(s) + E(s) = Y(s)$$

Converting the first equation to the time domain yields the second-order differential equation

$$\frac{d^2e(t)}{dt^2} + 2 \frac{de(t)}{dt} + e(t) = x(t)$$

and converting the second equation yields the output relationship

$$y(t) = 2 \frac{de(t)}{dt} + e(t)$$

The implication from the transfer function notation is that the initial conditions on the second-order differential equation are zero, or

$$\frac{de(0)}{dt} = 0, \quad e(0) = 0$$

Often we will use Laplace transform notation and, for shorthand, drop the functional dependence on s in the notation [i.e. F is equivalent to $F(s)$]. Similarly, when we are in the time domain, the functional dependence on t will often be dropped [i.e., f is equivalent to $f(t)$]. In addition, block diagrams and program listings will frequently use the overdot notation to represent time derivatives. With this notation, each overdot represents a derivative. For example,

$$\dot{y} = \frac{dy}{dt}, \quad \ddot{y} = \frac{d^2y}{dt^2}, \quad \dddot{y} = \frac{d^3y}{dt^3}, \quad \text{etc.}$$

Therefore, converting

$$\frac{d^2e(t)}{dt^2} + 2 \frac{de(t)}{dt} + e(t) = x(t)$$

to the overdot notation yields

$$\ddot{e} + 2\dot{e} + e = x$$

Occasionally, we shall either convert time functions to Laplace transforms or vice versa, by inspection. Some common transfer functions,¹ along with their time domain equivalents, appear in Table 1.1. A more extensive listing of inverse Laplace transforms can be found in Ref. 1.

Numerical Integration of Differential Equations

Throughout this text we will be simulating both linear and nonlinear ordinary differential equations. Because, in general, these equations have no closed-form solutions, it will be necessary to resort to numerical integration techniques to solve or simulate these equations. Many numerical integration techniques² exist for solving differential equations. However, we shall use the second-order Runge–Kutta technique throughout the text because it is simple to understand, easy to program, and, most importantly, yields accurate answers for all of the examples presented in this text.

The second-order Runge–Kutta numerical integration procedure is easy to state. Given a first-order differential equation of the form

$$\dot{x} = f(x, t)$$

Table 1.1 Common inverse Laplace transforms

$F(s)$	$f(t)$
$\frac{K}{s}$	K
$\frac{K}{s^n} (n = 1, 2, \dots)$	$\frac{Kt^{n-1}}{(n-1)!}$
$\frac{K}{(s-a)^n} (n = 1, 2, \dots)$	$\frac{Kt^{n-1}e^{at}}{(n-1)!}$
$\frac{K}{s^2 + a^2}$	$\frac{K \sin(at)}{a}$
$\frac{Ks}{s^2 + a^2}$	$K \cos(at)$
$\frac{K}{(s-a)^2 + b^2}$	$\frac{Ke^{at} \sin(bt)}{b}$
$\frac{K(s-a)}{(s-a)^2 + b^2}$	$Ke^{at} \cos(bt)$

where t is time, we seek to find a recursive relationship for x as a function of time. With the second-order Runge–Kutta numerical technique, the value of x at the next integration interval h is given by

$$x_{K+1} = x_K + \frac{hf(x, t)}{2} + \frac{hf(x, t+h)}{2}$$

where the subscript K represents the last interval and $K + 1$ represents the new interval. From the preceding expression we can see that the new value of x is simply the old value of x plus a term proportional to the derivative evaluated at time t and another term with the derivative evaluated at time $t + h$.

The integration step size h must be small enough to yield answers of sufficient accuracy. A simple test, commonly practiced among engineers, is to find the appropriate integration step size by experiment. As a rule of thumb, the initial step size is chosen to be several times smaller than the smallest time constant in the system under consideration. The step size is then halved to see if the answers change significantly. If the new answers are approximately the same, the larger integration step size is used to avoid excessive computer running time. If the answers change substantially, then the integration interval is again halved and the process is repeated.

To see how the Runge–Kutta technique can be applied to a practical example, let us consider the problem of finding the step response of one of the second-order networks from Table 1.1. Consider the sinusoidal transfer function

$$\frac{Y}{X} = \frac{\omega}{s^2 + \omega^2}$$

where x is the input, Y the output, ω the natural frequency of the second-order

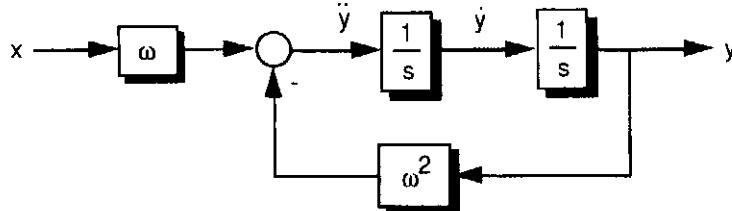


Fig. 1.1 Block diagram representation of second-order system.

network, and s the Laplace transformation notation for a derivative. Cross multiplying the numerator and denominator of the transfer function and solving for the highest derivative, as was shown in the previous section, yields the following second-order differential equation:

$$\ddot{y} = \omega x - \omega^2 y$$

where the double overdot represents two differentiations. This second-order differential equation can be represented in block diagram form as shown in Fig. 1.1. In this diagram each $1/s$ represents an integration. The outputs of each integrator are sometimes called states and are y and y dot respectively.

If x is a step input in Fig. 1.1, we can find the response y exactly using Laplace transform techniques. Recall from Table 1.1 that $1/s$ represents a step function in the Laplace transform domain. Therefore we can express the output y in the Laplace transform domain as

$$Y(s) = \frac{\omega}{s(s^2 + \omega^2)}$$

Expanding the preceding expression using partial fraction expansion yields

$$Y(s) = \frac{1}{\omega} \left[\frac{1}{s} - \frac{s}{s^2 + \omega^2} \right]$$

The inverse Laplace transform of $Y(s)$ produces y in the time domain or $y(t)$. The output can be found by using Table 1.1 obtaining

$$y = \frac{1}{\omega} (1 - \cos \omega t)$$

To check the preceding theoretical closed-form solution for y , a simulation involving numerical integration was written based on the system of Fig. 1.1. A FORTRAN listing of the simulation, using the second-order Runge-Kutta integration techniques, appears in Listing 1.1. We can see from the listing that the second-order differential equation, or derivative information, appears after label 200. We come to this label twice during the integration interval: once to evaluate the derivative at time t and once to evaluate the derivative at time $t + h$. We can also see from Listing 1.1 that every 0.01 s we print out the output along with the closed-form solution. In this particular example the natural frequency ω of the second-order system is 20 rad/s.

We can see from Listing 1.1 that the integration step size h is 0.001 s. Because the simulation time is 1 s, the ratio of the simulation time to the step size is 1000. This means that 2000 passes are made to the differential equations. The resultant

Listing 1.1 FORTRAN simulation of second-order system

```

INTEGER STEP
OPEN (1, STATUS='UNKNOWN',FILE='DATFIL')
W=20.
T=0.
S=0.
Y=0.
YD=0.
X=1.
H=.001
10 IF(T>1.) GOTO 999
S=S+H
YOLD=Y
YDOLD=YD
STEP=1
GOTO 200
66 STEP=2
Y=Y+H*YD
YD=YD+H*YDD
T=T+H
GOTO 200
55 CONTINUE
YD=.5* (YOLD+Y+H*YD)
Y=.5* (YDOLD+YD+H*YDD)
IF (S<.009999) GOTO 10
S=0.
YTHERY=(1.-COS (W*T))/W
WRITE (*,97)T,Y,YTHERY
WRITE(1,97)T,Y,YTHERY
GOTO 10
97 FORMAT (3F10.5)
200 CONTINUE
YDD=W*X-W*W*Y
IF(STEP-1) 66,66,55
999 CONTINUE
PAUSE
CLOSE(1)
END

```

system transient response, due to a step input ($x = 1$), is shown in Fig. 1.2. We can see that the simulation output agrees exactly with the closed-form solution.

Z Transforms and Difference Equations³

We have already shown that Laplace transforms are a useful way of representing differential equations. In this text we shall also want to simulate difference equations. Z transforms can also be used as an engineering shorthand for representing the difference equations. Later in this section we will also show how Z transforms can be used to solve difference equations and sometimes check simulation results.

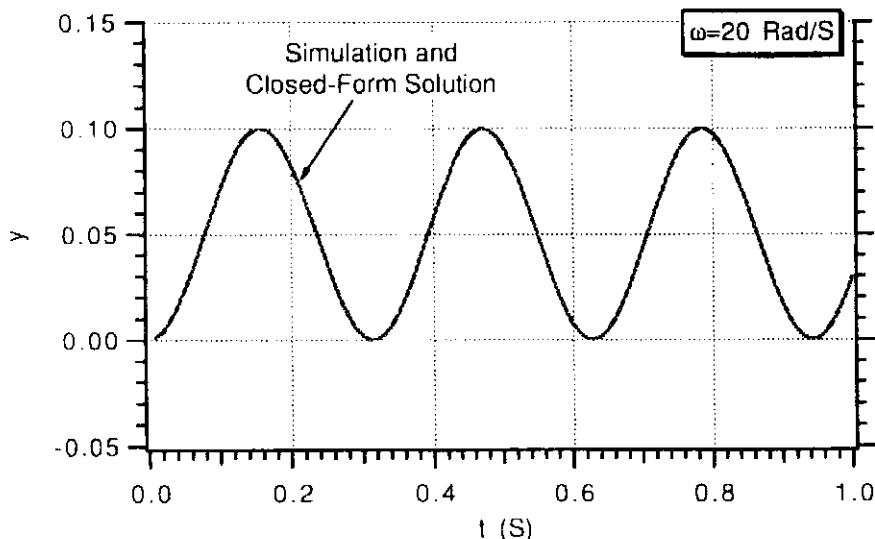


Fig. 1.2 Numerically integrating differential equations yields same results as closed-form solution.

If we define $F(z)$ as the Z transform of $f(n)$, then the Z transform has the following definition:

$$F(z) = \sum_{n=0}^{\infty} f(n)z^{-n}$$

With this definition it is easy to show that a summation in the time or n domain is also a summation in the Z transform domain. For example, if $f_1(n)$ and $f_2(n)$ have Z transforms $F_1(z)$ and $F_2(z)$, respectively, then

$$Z[f_1(n) \pm f_2(n)] = F_1(z) \pm F_2(z)$$

One can show that the Z transform of a signal at time $n + 1$ is a multiplication of the function in the Z transform domain by z . The Z transform of the signal at time n according to

$$Z(f_{n+1}) = zF(z) - zf(0)$$

where $f(0)$ is an initial condition. Often we will be working with systems having zero initial conditions.

A list of some common Z transforms can be found in Table 1.2.

Table 1.2 Z transforms of common functions

Function	Z transform
$\delta(n)$	1
1	$z/(z - 1)$
a^n	$z/(z - a)$
n	$z/(z - 1)^2$
$\sin \omega n T_s$	$z \sin \omega T_s / (z^2 - 2z \cos \omega T_s + 1)$

From Table 1.2 we can see that there is a relationship between the sampling time T_s and time t given by

$$t = nT_s$$

To illustrate how Z transforms can be used to solve difference equations, let us consider a numerical example involving the fading memory filters we will be working with in Chapter 7. The simplest fading memory filter can be expressed as the difference equation

$$y_{n+1} = y_n + G(x_{n+1} - y_n)$$

where y is the filter estimate or output, x the filter input or measurement, and G the filter gain. For the first-order fading memory filter, the filter gain is a designer chosen number between zero and unity. We can find the filter response to a step input (i.e., $x_{n+1} = 1$) by observing from Table 1.2 that the Z transform of a unit step function or constant is given by

$$Z(1) = z/(z - 1)$$

Therefore taking the Z transform of both sides of the difference equation yields

$$zY = Y + G \left(\frac{z}{z - 1} - Y \right)$$

If we bring all the terms in Y to the left-hand side of the equation, we get

$$Y(z - 1 + G) = Gz/(z - 1)$$

Solving for Y produces

$$Y = \frac{Gz}{(z - 1)(z - a)}$$

where

$$a = 1 - G$$

Using a partial fraction expansion on the solution for Y yields

$$\frac{G}{(z - 1)(z - a)} = \frac{G}{1 - a} \left[\frac{1}{z - 1} - \frac{1}{z - a} \right]$$

Therefore by multiplying both sides of the preceding equation by z , we obtain

$$\frac{Gz}{(z - 1)(z - a)} = \frac{G}{1 - a} \left[\frac{z}{z - 1} - \frac{z}{z - a} \right]$$

Using Table 1.2 to find the inverse Z transform of the preceding expression yields

$$y_n = \frac{G}{1 - a}(1 - a^n)$$

Substitution of the value of a in the preceding expression yields the closed-form solution for y as

$$y_n = 1 - (1 - G)^n$$

Listing 1.2 Difference equation simulation

```

OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
G=.5
X=1.
TS=.1
Y=0.
T=0.
N=0.
YTHERORY=1.-(1.-G)**N
WRITE(*,97)T,Y,YTHERORY
WRITE(1,97)T,Y,YTHERORY
97 FORMAT(3F10.3)
DO 10 N=1,20
Y=Y+G*(X-Y)
T=N*TS
YTHERORY=1.-(1.-G)**N
WRITE(*,97)T,Y,YTHERORY
WRITE(1,97)T,Y,YTHERORY
10 CONTINUE
PAUSE
CLOSE(1)
END

```

We now have an exact expression for the filter output as a function of the number of measurements n . To test the accuracy of the preceding closed-form solution for y , a simulation of the original difference equation was written and appears in Listing 1.2. We can see from the listing that unlike the previous simulation, numerical integration is not required. In this simulation we are simply solving the difference equation at each iteration of the “do loop” to get a new value for y . As we can see from the listing, the simulation solves the difference equation 20 times. The closed-form solution for y is also calculated at each iteration in order to check the validity of the simulation.

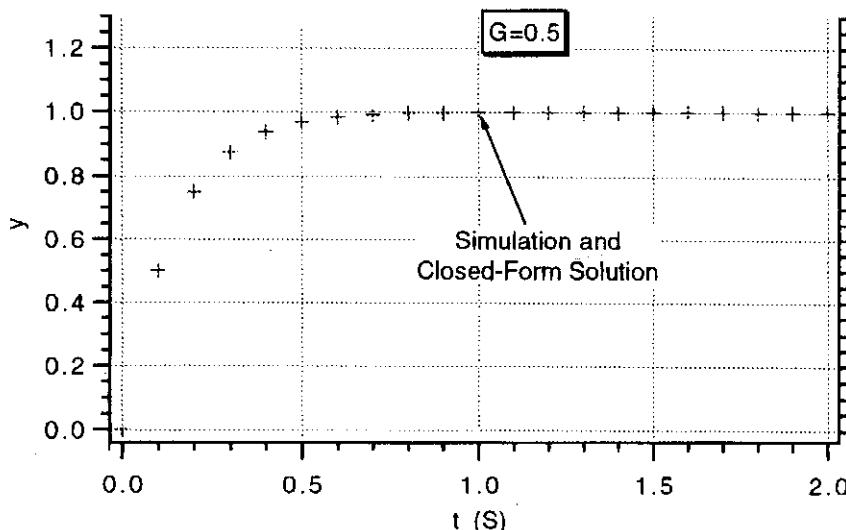


Fig. 1.3 Difference equation simulations results agree with closed-form solution.

We can see from Fig. 1.3 that the filter output eventually matches the filter input. The amount of time it takes the filter output to reach 63% of its steady-state value is the filter time constant. Varying the filter gain G will change the time constant of the fading memory filter. We can also see from Fig. 1.3 that the simulation results agree with the closed-form solution.

References

- ¹Selby, S. M., *Standard Mathematical Tables—Twentieth Edition*, Chemical Rubber Co., Cleveland, OH, 1972.
- ²Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., *Numerical Recipes: The Art of Scientific Computation*, Cambridge Univ. Press, London, 1986.
- ³Schwarz, R., and Friedland, B., *Linear Systems*, McGraw-Hill, New York, 1965.

Fundamentals of Tactical Missile Guidance

Introduction

TACTICAL guided missiles apparently had their origin in Germany. For example, the Hs. 298 was one of a series of German air-to-air guided missiles developed by the Henschel Company during World War II.¹ A high-thrust first stage accelerated the missile from the carrier aircraft, whereas a low-thrust, long-burning sustainer maintained the vehicle's velocity. The Hs. 298, which was radio-controlled from the parent aircraft, was to be released either slightly above or below the target. Apparently the height differential made it easier to aim and guide the missile. This first air-to-air missile weighed 265 lb and had a range of nearly 3 miles. On December 22, 1944, three missiles were test flown from a JU 88G aircraft. All three tests resulted in failure. Although 100 of these air-to-air missiles were manufactured, none was used in combat.

The Rheintochter (R-1) was a surface-to-air missile also developed in Germany during World War II.¹ This unusual looking two-stage radio-controlled missile weighed nearly 4000 lb and had three sets of plywood fins: one for the booster and two for the sustainer. Eighty-two of these missiles flew before production was halted in December 1944. The missile was ineffective because Allied bombers, which were the R-1's intended target, flew above the range (about 20,000 ft) of this surface-to-air missile.

Although proportional navigation was apparently known by the Germans during World War II at Peenemünde, no applications on the Hs. 298 or R-1 missiles using proportional navigation were reported.² The Lark missile, which had its first successful test in December 1950, was the first missile to use proportional navigation. Since that time proportional navigation guidance has been used in virtually all of the world's tactical radar, infrared (IR), and television (TV) guided missiles.³ The popularity of this interceptor guidance law is based upon its simplicity, effectiveness, and ease of implementation. Apparently, proportional navigation was first studied by C. Yuan and others at the RCA Laboratories during World War II under the auspices of the U.S. Navy.⁴ The guidance law was conceived from physical reasoning and equipment available at that time. Proportional navigation was extensively studied at Hughes Aircraft Company⁵ and implemented in a tactical missile using a pulsed radar system. Finally, proportional navigation was more fully developed at Raytheon and implemented in a tactical continuous wave radar homing missile.⁶ After World War II, the U.S. work on proportional navigation was declassified and first appeared in the *Journal of Applied Physics*.⁷ Mathematical derivations of the "optimality" of proportional navigation came more than 20 years later.⁸

Keeping with the spirit of the origins of proportional navigation, we shall avoid mathematical proofs in this chapter on deriving the guidance law, but shall, instead, concentrate first on proving to the reader that the guidance technique works. Next we shall investigate some properties of the guidance law that we shall both observe and derive. Finally, we shall show how this classical guidance law provides the foundation for more advanced techniques of interceptor guidance.

What Is Proportional Navigation?

Theoretically, the proportional navigation guidance law issues acceleration commands, perpendicular to the instantaneous missile-target line-of-sight, which are proportional to the line-of-sight rate and closing velocity. Mathematically, the guidance law can be stated as

$$n_c = N' V_c \dot{\lambda}$$

where n_c is the acceleration command (in ft/s^2), N' a unitless designer chosen gain (usually in the range of 3–5) known as the effective navigation ratio, V_c the missile-target closing velocity (in ft/s), and λ the line-of-sight angle (in rad). The overdot indicates the time derivative of the line-of-sight angle or the line-of-sight rate.

In tactical radar homing missiles using proportional navigation guidance, the seeker provides an effective measurement of the line-of-sight rate, and a Doppler radar provides closing velocity information. In tactical IR missile applications of proportional navigation guidance, the line-of-sight rate is measured, whereas the closing velocity, required by the guidance law, is “guesstimated.”

In tactical endoatmospheric missiles, proportional navigation guidance commands are usually implemented by moving fins or other control surfaces to obtain the required lift. Exoatmospheric strategic interceptors use thrust vector control, lateral divert engines, or squibs to achieve the desired acceleration levels.

Simulation of Proportional Navigation in Two Dimensions

To better understand how proportional navigation works, let us consider the two-dimensional, point mass missile-target engagement geometry of Fig. 2.1. Here we have an inertial coordinate system fixed to the surface of a flat-Earth model (i.e., the 1 axis is downrange and the 2 axis can either be altitude or crossrange). Using the inertial coordinate system of Fig. 2.1 means that we can integrate components of the accelerations and velocities along the 1 and 2 directions without having to worry about additional terms due to the Coriolis effect. In this model it is assumed that both the missile and target travel at constant velocity. In addition, gravitational and drag effects have been neglected for simplicity.

We can see from the figure that the missile, with velocity magnitude V_M , is heading at an angle of $L + HE$ with respect to the line of sight. The angle L is known as the missile lead angle. The lead angle is the theoretically correct angle for the missile to be on a collision triangle with the target. In other words, if the missile is on a collision triangle, no further acceleration commands are required for the missile to hit the target. The angle HE is known as the heading error. This angle represents the initial deviation of the missile from the collision triangle.

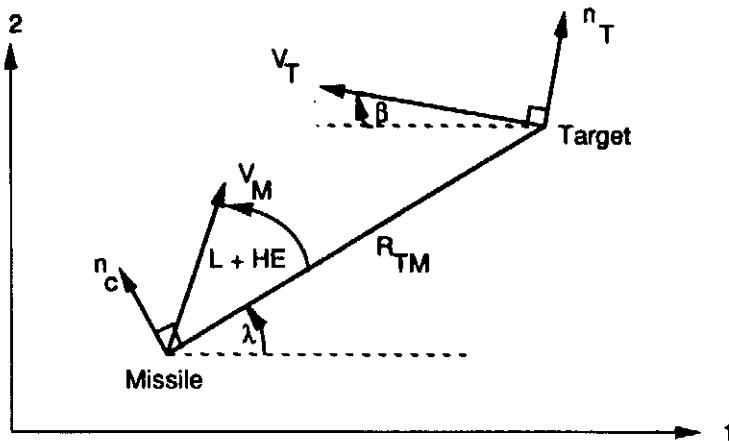


Fig. 2.1 Two-dimensional missile-target engagement geometry.

In Fig. 2.1 the imaginary line connecting the missile and target is known as the line of sight. The line of sight makes an angle of λ with respect to the fixed reference, and the length of the line of sight (instantaneous separation between missile and target) is a range denoted R_{TM} . From a guidance point of view, we desire to make the range between missile and target at the expected intercept time as small as possible (hopefully zero). The point of closest approach of the missile and target is known as the miss distance.

The closing velocity V_c is defined as the negative rate of change of the distance from the missile to the target, or

$$V_c = -\dot{R}_{TM}$$

Therefore, at the end of the engagement, when the missile and target are in closest proximity, the sign of V_c will change. In other words, from calculus we know that the closing velocity will be zero when R_{TM} is a minimum (i.e., the function is either minimum or maximum when its derivative is zero). The desired acceleration command n_c , which is derived from the proportional navigation guidance law, is perpendicular to the instantaneous line of sight.

In our engagement model of Fig. 2.1, the target can maneuver evasively with acceleration magnitude n_T . Since target acceleration n_T in the preceding model is perpendicular to the target velocity vector, the angular velocity of the target can be expressed as

$$\dot{\beta} = \frac{n_T}{V_T}$$

where V_T is the magnitude of the target velocity. The components of the target velocity vector in the Earth or inertial coordinate system can be found by integrating the differential equation given earlier for the flight-path angle of the target β and substituting in

$$V_{T1} = -V_T \cos \beta$$

$$V_{T2} = V_T \sin \beta$$

Target position components in the Earth fixed coordinate system can be found by directly integrating the target velocity components. Therefore, the differential

equations for the components of the target position are given by

$$\dot{R}_{T1} = V_{T1}$$

$$\dot{R}_{T2} = V_{T2}$$

Similarly, the missile velocity and position differential equations are given by

$$\dot{V}_{M1} = a_{M1}$$

$$\dot{V}_{M2} = a_{M2}$$

$$\dot{R}_{M1} = V_{M1}$$

$$\dot{R}_{M2} = V_{M2}$$

where a_{M1} and a_{M2} are the missile acceleration components in the Earth coordinate system. To find the missile acceleration components, we must first find the components of the relative missile-target separation. This is accomplished by first defining the components of the relative missile-target separations by

$$R_{TM1} = R_{T1} - R_{M1}$$

$$R_{TM2} = R_{T2} - R_{M2}$$

We can see from Fig. 2.1 that the line-of-sight angle can be found, using trigonometry, in terms of the relative separation components as

$$\lambda = \tan^{-1} \frac{R_{TM2}}{R_{TM1}}$$

If we define the relative velocity components in Earth coordinates to be

$$V_{TM1} = V_{T1} - V_{M1}$$

$$V_{TM2} = V_{T2} - V_{M2}$$

we can calculate the line-of-sight rate by direct differentiation of the expression for line-of-sight angle. After some algebra we obtain the expression for the line-of-sight rate to be

$$\dot{\lambda} = \frac{R_{TM1} V_{TM2} - R_{TM2} V_{TM1}}{R_{TM}^2}$$

The relative separation between missile and target R_{TM} can be expressed in terms of its inertial components by application of the distance formula, as

$$R_{TM} = (R_{TM1}^2 + R_{TM2}^2)^{\frac{1}{2}}$$

Since the closing velocity is defined as the negative rate of change of the missile target separation, it can be obtained by differentiating the preceding equation, yielding

$$V_c = -\dot{R}_{TM} = \frac{-(R_{TM1} V_{TM1} + R_{TM2} V_{TM2})}{R_{TM}}$$

The magnitude of the missile guidance command n_c can then be found from the definition of proportional navigation, or

$$n_c = N' V_c \lambda$$

Since the acceleration command is perpendicular to the instantaneous line of sight, the missile acceleration components in Earth coordinates can be found by trigonometry using the angular definitions from Fig. 2.1. The missile acceleration components are

$$a_{M1} = -n_c \sin \lambda$$

$$a_{M2} = n_c \cos \lambda$$

We have now listed all of the differential equations required to model a complete missile-target engagement in two dimensions. However, some additional equations are required for the initial conditions on the differential equations in order to complete the engagement model.

A missile employing proportional navigation guidance is not fired at the target but is fired in a direction to lead the target. The initial angle of the missile velocity vector with respect to the line of sight is known as the missile lead angle L . In essence we are firing the missile at the expected intercept point. We can see from Fig. 2.1 that, for the missile to be on a collision triangle (missile will hit target if both continue to fly along a straight-line path at constant velocities), the theoretical missile lead angle can be found by application of the law of sines, yielding

$$L = \sin^{-1} \frac{V_T \sin(\beta + \lambda)}{V_M}$$

In practice, the missile is usually not launched exactly on a collision triangle, since the expected intercept point is not known precisely. The location of the intercept point can only be approximated because we do not know in advance what the target will do in the future. In fact, that is why a guidance system is required! Any initial angular deviation of the missile from the collision triangle is known as a heading error HE . The initial missile velocity components can therefore be expressed in terms of the theoretical lead angle L and actual heading error HE as

$$V_{M1}(0) = V_M \cos(L + HE + \lambda)$$

$$V_{M2}(0) = V_M \sin(L + HE + \lambda)$$

Two-Dimensional Engagement Simulation

To witness and understand the effectiveness of proportional navigation, it is best to simulate the guidance law and test its properties under a variety of circumstances. A two-dimensional missile-target engagement simulation was set up using the differential equations derived in the previous section. The simulation inputs are the initial location of the missile and target, speeds, flight time, and effective navigation ratio. The user can vary the level of the two error sources considered: target maneuver and heading error.

A FORTRAN tactical missile-target engagement simulation appears in Listing 2.1. We can see from the listing that the missile and target differential equations are solved using the second-order Runge-Kutta numerical integration technique. As was the case in the second-order system simulation of Chapter 1, the differential

Listing 2.1 Two-dimensional tactical missile-target engagement simulation

```

DATA VM,VT,XNT,HEDEG,XNP/3000.,1000.,0.,-20.,4./
DATA RM1,RM2,RT1,RT2/0.,10000.,40000.,10000./
XNCLIM=966.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
BETA=0.
VT1=-VT*COS(BETA)
VT2=VT*SIN(BETA)
HE=HEDEG/57.3
T=0.
S=0.
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
XLAM=ATAN2(RTM2,RTM1)
XLEAD=ASIN(VT*SIN(BETA+XLAM)/VM)
THET=XLAM+XLEAD
VM1=VM*COS(THET+HE)
VM2=VM*SIN(THET+HE)
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
10 IF(VC<0.) GOTO 999
IF(RTM<1000.)THEN
    H=.0002
ELSE
    H=.01
ENDIF
BETAOLD=BETA
RT1OLD=RT1
RT2OLD=RT2
RM1OLD=RM1
RM2OLD=RM2
VM1OLD=VM1
VM2OLD=VM2
STEP=1
GOTO 200
66 STEP=2
BETA=BETA+H*BETAD
RT1=RT1+H*VT1
RT2=RT2+H*VT2
RM1=RM1+H*VM1
RM2=RM2+H*VM2
VM1=VM1+H*AM1
VM2=VM2+H*AM2
T=T+H
GOTO 200

```

(Contd.)

Listing 2.1 (Continued)

```

55  CONTINUE
    BETA=.5*(BETAOLD+BETA+H*BETAD)
    RT1=.5*(RT1OLD+RT1+H*VT1)
    RT2=.5*(RT2OLD+RT2+H*VT2)
    RM1=.5*(RM1OLD+RM1+H*VM1)
    RM2=.5*(RM2OLD+RM2+H*VM2)
    VM1=.5*(VM1OLD+VM1+H*AM1)
    VM2=.5*(VM2OLD+VM2+H*AM2)
    S=S+H
    IF(S<.09999)GOTO 10
    S=0.
    RT1K=RT1/1000.
    RT2K=RT2/1000.
    RM1K=RM1/1000.
    RM2K=RM2/1000.
    WRITE(*,97)T,RT1K,RT2K,RM1K,RM2K,XNC/32.2
    WRITE(1,97)T,RT1K,RT2K,RM1K,RM2K,XNC/32.2
    GOTO 10
97   FORMAT(6F10.3)
200  CONTINUE
        RTM1=RT1-RM1
        RTM2=RT2-RM2
        RTM=SQRT(RTM1**2+RTM2**2)
        IF(RTM2.GE.0.)THEN
            RTMP=RTM
        ELSE
            RTMP=-RTM
        ENDIF
        VTM1=VT1-VM1
        VTM2=VT2-VM2
        VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
        XLAM=ATAN2(RTM2,RTM1)
        XLAMD=(RTM1*VTM2-RTM2*VTM1)/(RTM*RTM)
        XNC=XNP*VC*XLAMD
        IF(XNC>XNCLIM)XNC=XNCLIM
        IF(XNC <-XNCLIM)XNC=-XNCLIM
        AM1=-XNC*SIN(XLAM)
        AM2=XNC*COS(XLAM)
        VT1=-VT*COS(BETA)
        VT2=VT*SIN(BETA)
        BETAD=XNT/VT
        IF(STEP-1)66,66,55
999  CONTINUE
        WRITE(*,97)T,RT1K,RT2K,RM1K,RM2K,XNC/32.2
        WRITE(*,*)RTMP
        PAUSE
        CLOSE(1)
        END

```

equations appear after label 200. All computation in the simulation is performed in single-precision arithmetic. The integration step size is fixed for most of the flight ($H = 0.01$ s) but is made smaller near the end of the flight ($H = 0.0002$ s when $R_{TM} < 1000$ ft) to accurately capture the magnitude of the miss distance. The program is terminated when the closing velocity changes sign, because this means that the separation between the missile and target is a minimum. At this time the missile-target separation is the miss distance: We can see from the preceding equations that the miss distance will always be positive since it is calculated from the distance formula. Later on we shall be interested in the location of the missile with respect to the target at the point of closest approach. Therefore simple logic with RTMP has been included after statement label 200 to say that the miss is considered to be positive if the target is above the missile at intercept ($RTM2 > 0$) and negative if the opposite occurs. We can see from the listing that errors can be introduced by changing values in the data statements. Status of the missile and target location, along with acceleration and separation information, is displayed every 0.1 s. Note that the missile acceleration is written to a file DATFIL in units of gravity.

A sample case was run in which the only disturbance was a 20-deg heading error ($HEDEG = -20$). Sample trajectories for effective navigation ratios of 4 and 5 are depicted in Fig. 2.2. We can see from the figure that initially the missile is flying in the wrong direction because of the heading error. Gradually the guidance law forces the missile to home on the target. The larger effective navigation ratio enables the missile to remove the initial heading error more rapidly, thus causing a much tighter trajectory. In both cases, proportional navigation appears to be an effective guidance law because the missile hits the target (near zero miss distance with the simulation).

The resultant missile acceleration histories, displayed in Fig. 2.3, for both cases are somewhat different. The quicker removal of heading error in the higher effective navigation ratio case ($N' = 5$) results in larger missile accelerations at the beginning of the flight and lower accelerations near the end of the flight. In both cases the acceleration profiles for the required missile acceleration to take out the

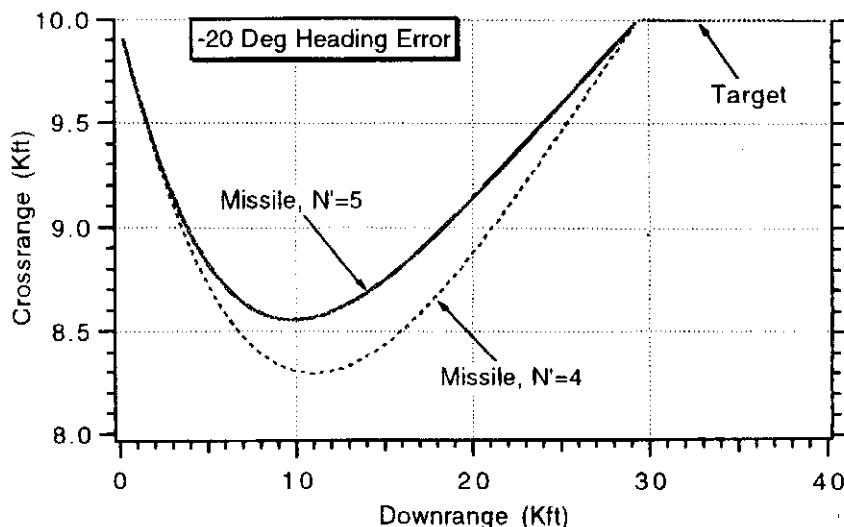


Fig. 2.2 Increasing effective navigation ratio causes heading error to be removed more rapidly.

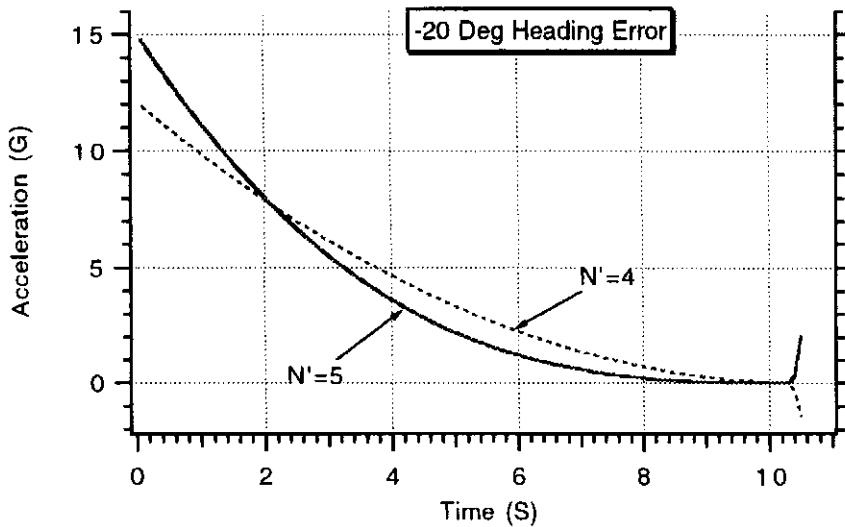


Fig. 2.3 Increasing effective navigation ratio causes more acceleration initially.

heading error and to hit the target is monotonically decreasing and zero at the end of the flight. Thus, a property of a proportional navigation guidance system is to start taking out heading error as soon as possible but also gradually throughout the *entire* flight. In Chapter 16 we shall study a guidance system that tries to remove the entire heading error immediately. By increasing the effective navigation ratio, we are allowing the missile to take out heading error more rapidly.

Another sample case was run in which the only disturbance was a 3-g target maneuver ($XNT = 96.6$, $HEDEG = 0$). In this scenario the missile and target are initially on a collision triangle and flying along the downrange component of the Earth fixed coordinate system (cross-range velocity components of both interceptor and target are zero). Therefore, the target velocity vector is initially along the line of sight, and at first all 3 g of the target acceleration are perpendicular to the line of sight. As the target maneuvers, the magnitude of the target acceleration perpendicular to the line of sight diminishes due to the turning of the target. Sample missile-target trajectories for this case with effective navigation ratios of 4 and 5 are depicted in Fig. 2.4. We can see that the higher effective navigation ratio causes the missile to lead the target slightly more than the lower navigation ratio case.

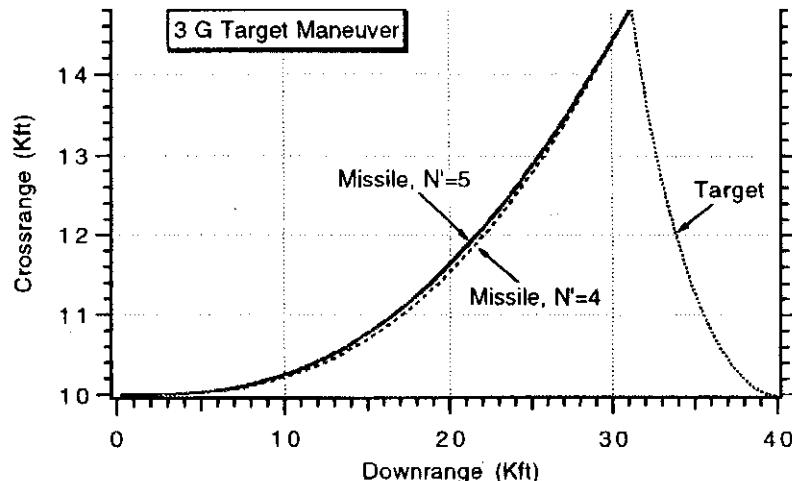


Fig. 2.4 Proportional navigation works against maneuvering target.

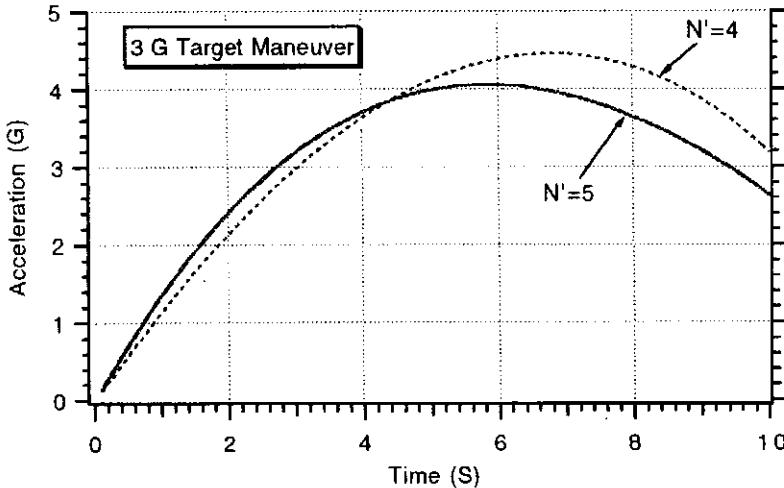


Fig. 2.5 Higher navigation ratio yields less acceleration to hit maneuvering target.

Otherwise the trajectories are virtually identical. In both cases, the proportional navigation guidance law enabled the missile to hit the maneuvering target.

However, Fig. 2.5 shows that there are significant differences between the acceleration profiles for the maneuvering target case. Although both acceleration profiles are virtually monotonically increasing for the entire flight, the higher effective navigation ratio requires less acceleration capability of the missile. In addition, we can see that the peak acceleration required by the missile to hit the target is significantly higher than the maneuver level of the target ($3\ g$).

In both simulation examples we have seen the effectiveness of proportional navigation guidance. First we saw that proportional navigation is able to hit a target, even if it is initially launched in the wrong direction by $20\ deg$. Then we observed that the guidance law was also effective in hitting a maneuvering target. In both cases certain acceleration levels were required of the missile in order for it to hit the target. The levels were dependent on the type of error source and the effective navigation ratio. If the missile does not have the acceleration required by the guidance law, a miss will result.

Linearization

Thus far our understanding of the effectiveness of proportional navigation has come from the numerical simulation results of the two-dimensional engagement simulation. It is critical for the analysis, understanding, and development of design relationships to temporarily depart from the nonlinear missile-target simulation and develop a simpler model. Therefore, we will linearize the two-dimensional engagement model in the hope of gaining more understanding. This does not mean that we will abandon the nonlinear engagement model. In fact, we will always use the nonlinear engagement model to verify the insights generated by powerful analytical techniques to be used on the linearized engagement model.

The linearization of the missile-target geometry can easily be accomplished if we define some new relative quantities as shown in Fig. 2.6. Here y is the relative separation between the missile and target perpendicular to the fixed reference.

The relative acceleration (difference between missile and target acceleration) can be written by inspection of Fig. 2.6 as

$$\ddot{y} = n_T \cos \beta - n_c \cos \lambda$$

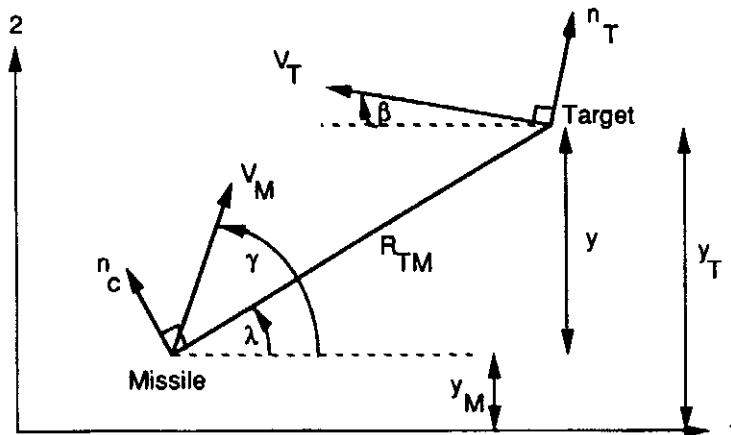


Fig. 2.6 Engagement model for linearization.

If the flight-path angles are small (near head-on or tail chase case), the cosine terms are approximately unity, and the preceding equation becomes

$$\ddot{y} = n_T - n_c$$

Similarly, the expression for the line-of-sight angle can also be linearized using the small-angle approximation, yielding

$$\lambda = y/R_{TM}$$

For a head-on case we can approximate the closing velocity as

$$V_c = V_M + V_T$$

whereas in a tail chase case the closing velocity can be approximated as

$$V_c = V_M - V_T$$

Therefore, in a linearized analysis we will treat the closing velocity as a positive constant. Since closing velocity has also been previously defined as the negative derivative of the range from the missile to target, and since the range must go to zero at the end of the flight, we can also linearize the range equation with the time-varying relationship

$$R_{TM} = V_c(t_F - t)$$

where t is current time and t_F the total flight time of the engagement. Note that t_F is also now a constant. The quantity $t_F - t$ is the time to go until the end of the flight. Therefore, the range from the missile to the target is also the closing velocity multiplied by the time to go until intercept. Since range goes to zero at the end of the flight by definition, we must reexamine the definition of miss distance. The linearized miss distance is taken to be the relative separation between missile and target y at the end of the flight, or

$$\text{Miss} = y(t_F)$$

Since the linearized miss is not obtained from the distance formula, it is only an approximation to the actual miss. However, we shall soon see that the miss distance approximation is very accurate.

Linearized Engagement Simulation

In the previous section we developed linearized equations for the missile-target engagement. In this section we will see if the resultant linearized equations give performance projections that have trends similar to those of the nonlinear engagement equations. If they do not, then there is no point in developing design relationships based on a meaningless model. If they do, then there may be a point for the interested reader to continue reading this text!

The linearized proportional navigation engagement FORTRAN simulation appears in Listing 2.2. In this simulation the flight time t_F is an input rather than output. We can see from the listing that the simulation only consists of two differential equations: one for relative velocity and the other for relative acceleration. These differential equations are also solved using the second-order Runge-Kutta

Listing 2.2 Linearized engagement simulation

```

DATA VC,XNT,Y,VM,HEDEG,TF,XNP/4000.,0.,0.,3000.,-20.,10.,4./
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
YD=-VM*HEDEG/57.3
T=0.
H=.01
S=0.
10 IF(T>(TF-.0001))GOTO 999
YOLD=Y
YDOLD=YD
STEP=1
GOTO 200
66 STEP=2
Y=Y+H*YD
YD=YD+H*YDD
T=T+H
GOTO 200
55 CONTINUE
Y=.5*(YOLD+Y+H*YD)
YD=.5*(YDOLD+YD+H*YDD)
S=S+H
IF(S<.09999)GOTO 10
S=0.
WRITE(*,97)T,Y,YD,XNC/32.2
WRITE(1,97)T,Y,YD,XNC/32.2
GOTO 10
97 FORMAT(4F10.3)
200 CONTINUE
TGO=TF-T+.00001
XLAMD=(Y+YD*TGO)/(VC*TGO*TGO)
XNC=XNP*VC*XLAMD
YDD=XNT-XNC
IF(STEP-1)66,66,55
999 CONTINUE
PAUSE
CLOSE(1)
END

```

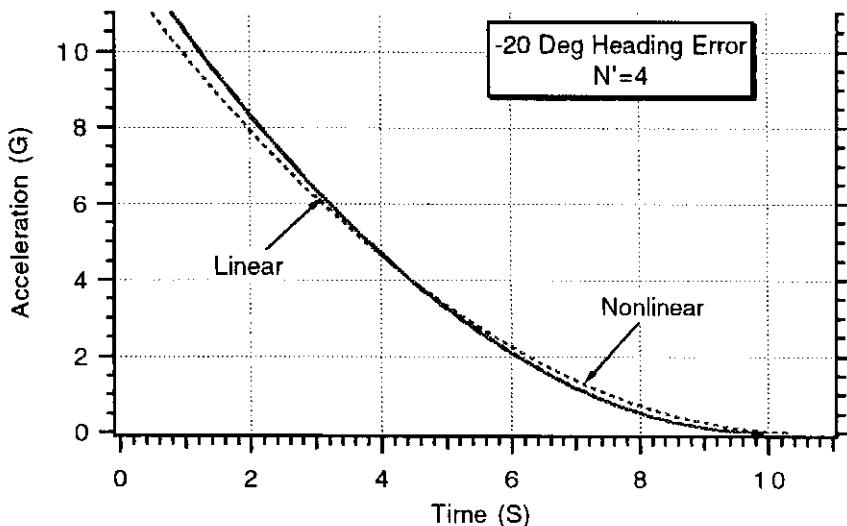


Fig. 2.7 Linearized engagement model yields accurate performance projections for heading error disturbance.

numerical integration technique. The linearized differential equations appear in the listing after label 200. Unlike the nonlinear engagement simulation, the integration step size in the linear simulation can be kept fixed for the entire flight ($H = 0.01$ s). The program is stopped when the current time equals the flight time. Nominally the program is set up without errors. Errors can be introduced by changing values in the data statements. The status of the relative position and velocity, along with missile acceleration information, is displayed every 0.1 s.

To verify that the linearized engagement model is a reasonable approximation to the nonlinear engagement model, cases that were run for the nonlinear engagement model were repeated using the simulation of Listing 2.2. A sample run was made with the linearized engagement model in which the only disturbance was a -20 -deg heading error ($HEDEG = -20$). In this case the effective navigation ratio was 4. Acceleration profile comparisons for both the linear and nonlinear engagement models are presented in Fig. 2.7. The figure clearly shows that, even for a relatively large heading error disturbance, the resultant acceleration profiles are virtually indistinguishable. Thus, the linearized model is an excellent approximation to the nonlinear engagement model in the case of a heading error disturbance.

Another sample run was made with the linear engagement model; this time with a 3-g target maneuver disturbance. Figure 2.8 shows that this time the linearized model overestimates the missile acceleration requirements. The reason for the discrepancy is that the linear model assumes that the target acceleration magnitude, perpendicular to the line of sight, is always the same and equal to the magnitude of the maneuver. In reality, as the target maneuvers, the component of acceleration perpendicular to the line of sight decreases because the target is turning. Therefore, the nonlinear acceleration requirements due to a maneuvering target are somewhat less than those predicted by the linearized engagement model. However, it is important to note that the linear engagement model accurately predicts the monotonically increasing trend (for most of the flight) for the missile acceleration profile due to a target maneuver.

At this point we can conclude that the linearized engagement model yields performance projections of sufficient accuracy to make it worthwhile to proceed

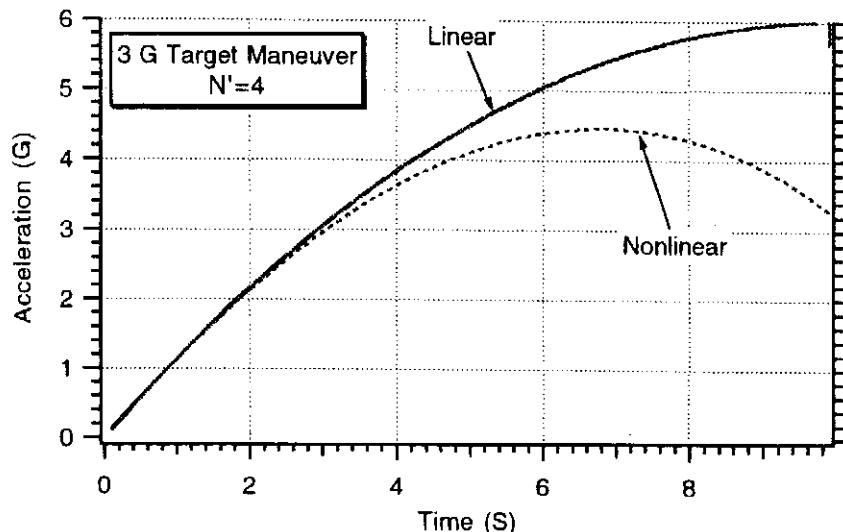


Fig. 2.8 Linear model overestimates missile acceleration due to target maneuver.

with the development of design relationships. We will test the validity of those relationships throughout the text in a variety of environments.

Important Closed-Form Solutions

The linearization of the engagement model is important for two reasons. First, with a linear model, powerful computerized techniques such as the method of adjoints (described in Chapters 3 and 4) can be used to analyze the missile guidance system both statistically and deterministically in one computer run. With this technique, error budgets are automatically generated so that key system drivers can be identified and a balanced guidance system design can be achieved. The linear model is also important because, under special circumstances, closed-form solutions can be obtained. These solutions can be used as system sizing aids. In addition, the form of the solutions will suggest how key parameters influence system performance.

Let us consider obtaining closed-form solutions for the two important cases we have already considered in both the linear and nonlinear engagement simulations. The first case is the missile acceleration required to remove a heading error, and the second case is the missile acceleration required to hit a maneuvering target. In the absence of target maneuver the relative acceleration (target acceleration minus missile acceleration) can be expressed as

$$\ddot{y} = -N' V_c \dot{\lambda}$$

Integrating the preceding differential equation once yields

$$\dot{y} = -N' V_c \lambda + C_1$$

where C_1 is the constant of integration. Substitution of the linear approximation to the line-of-sight angle in the preceding expression yields the following time-varying first-order differential equation:

$$\frac{dy}{dt} + \frac{N' y}{t_F - t} = C_1$$

Since a first-order differential equation of the form

$$\frac{dy}{dt} + a(t)y = h(t)$$

has the solution⁹⁻¹²

$$y = \exp\left[-\int_0^t a(T) dT\right] \left\{ \int_0^t h(n) \exp\left[\int_0^n a(T) dT\right] dn + C_2 \right\}$$

we can solve the linearized trajectory differential equation exactly. Note that the first constant of integration C_1 is contained in $h(t)$ while the second constant of integration C_2 appears in the preceding equation. Both constants of integration can be found by evaluating initial conditions on y and its derivative. Let us assume that the initial condition on the first state is zero, or

$$y(0) = 0$$

and that the initial condition on the second state is related to the heading error by

$$\dot{y}(0) = -V_M HE$$

where V_M is the missile velocity and HE the heading error in radians. Under these circumstances, after much algebra, we find that the closed-form solution for the missile acceleration due to heading error is given by

$$n_c = \frac{-V_M HE N'}{t_F} \left(1 - \frac{t}{t_F}\right)^{N'-2}$$

where t_F is the flight time and N' the effective navigation ratio. We can see that the magnitude of the initial acceleration is proportional to the heading error and missile velocity and inversely proportional to the flight time. Doubling the velocity or heading error will double the initial missile acceleration, whereas doubling the flight time or time available for guidance will halve the initial missile acceleration. In addition, the closed-form solution for the miss distance $y(t_F)$ is zero. In other words, as long as the missile has sufficient acceleration capability, there is no miss due to heading error!

The closed-form solution for the missile acceleration response due to heading error is displayed in normalized form in Fig. 2.9. We can see that higher effective navigation ratios require more acceleration at the beginning of flight than at the end of the flight and less acceleration as the flight progresses. From a system sizing point of view, the designer usually wants to ensure that the acceleration capability of the missile is adequate at the beginning of flight so that saturation can be avoided. For a fixed missile acceleration capability, Fig. 2.9 shows how requirements are placed on minimum guidance or flight time and maximum allowable heading error and missile velocity.

Similarly, if the only disturbance is a target maneuver, the appropriate second-order differential equation becomes

$$\ddot{y} = -N' V_c \dot{\lambda} + n_T$$

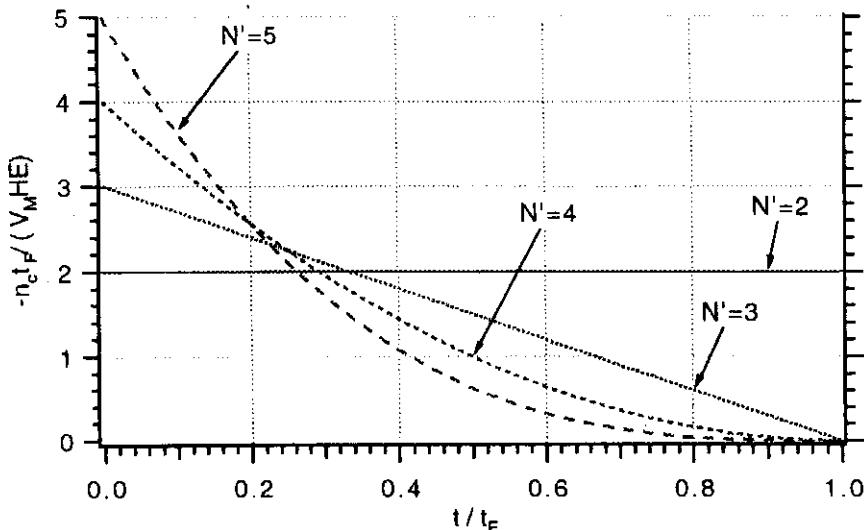


Fig. 2.9 Normalized missile acceleration due to heading error for proportional navigation guidance.

with initial conditions

$$y(0) = 0$$

$$\dot{y}(0) = 0$$

After conversion to a first-order differential equation and much algebra, the solution can be found to be

$$n_c = \frac{N'}{N' - 2} \left[1 - \left(1 - \frac{t}{t_F} \right)^{N'-2} \right] n_T$$

It appears that something “magical” happens to the acceleration when the effective navigation ratio is two. Application of L’Hopital’s rule eliminates the division by zero in the preceding formula and indicates that

$$\lim_{N' \rightarrow 2} n_c = -2 \ln \left(\frac{t_F - t}{t_F} \right)$$

This is approximately the same solution as if we simply let $N' = 2.01$ or $N' = 1.99$ in the original closed-form solution for the acceleration as a function of the effective navigation ratio. As with the heading error case, the closed-form solution indicates that the miss distance due to target maneuver is exactly zero!

Unlike the heading error case, missile acceleration due to maneuver is independent of flight time and missile velocity and only depends on the magnitude of the maneuver and the effective navigation ratio. Doubling the maneuver level of the target doubles the missile acceleration requirements.

The closed-form solution for the missile acceleration response due to target maneuver is displayed in normalized form in Fig. 2.10. We can see that higher effective navigation ratios relax the acceleration requirements at the end of the flight. Unlike the heading error response, the missile acceleration required to hit a maneuvering target increases as the flight progresses. From a system sizing point of view, the designer must ensure that the acceleration capability of the missile is

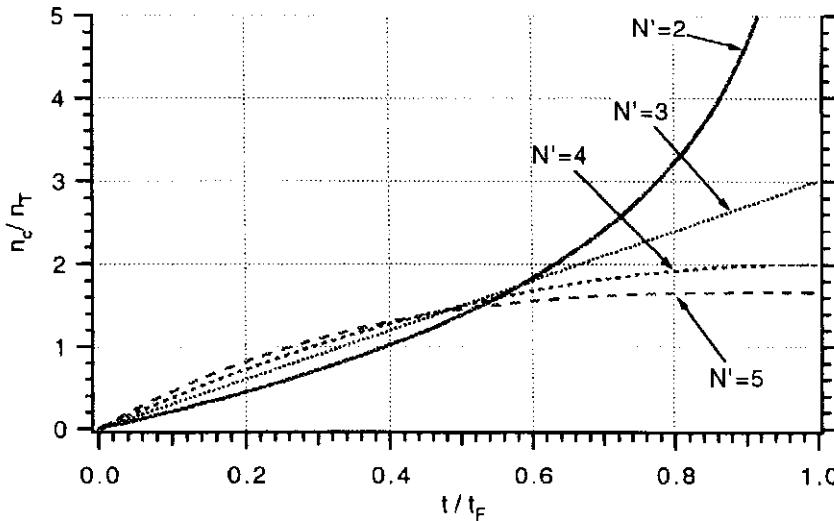


Fig. 2.10 Normalized missile acceleration due to target maneuver for proportional navigation guidance.

adequate at the end of flight so that saturation can be avoided so that the missile can hit the target.

Proportional Navigation and Zero Effort Miss

Thus far we have seen from simulation results and closed-form solutions that proportional navigation appears to be effective, but we do not know why. Although it is possible to construct geometric arguments showing that it is very logical to issue acceleration commands proportional to the line-of-sight rate (i.e., zero line-of-sight rate means we are on a collision triangle and therefore no further commands are necessary), it is not obvious what is happening. The concept of zero effort miss is not only useful in explaining proportional navigation but is also useful in deriving and understanding more advanced guidance laws.

We can define the zero effort miss to be the distance the missile would miss the target if the target continued along its present course and the missile made no further corrective maneuvers. Therefore, if the target does not maneuver, the two components, in the Earth fixed coordinate system, of the zero effort miss can be expressed in terms of the previously defined relative quantities as

$$ZEM_1 = R_{TM1} + V_{TM1}t_{go}$$

$$ZEM_2 = R_{TM2} + V_{TM2}t_{go}$$

where t_{go} is the time to go until intercept. Thus, we can see that in this case the zero effort miss is just a simple prediction (assuming constant velocities and zero acceleration) of the future relative separation between missile and target. From Fig. 2.1 we can see that the component of the zero effort miss that is perpendicular to the line of sight ZEM_{PLOS} can be found by trigonometry and is given by

$$ZEM_{PLOS} = -ZEM_1 \sin \lambda + ZEM_2 \cos \lambda$$

Expansion and simplification of the preceding equation yields

$$ZEM_{PLOS} = \frac{t_{go}(R_{TM1}V_{TM2} - R_{TM2}V_{TM1})}{R_{TM}}$$

Comparing the preceding expression to the expression for line-of-sight rate, we can see that the line-of-sight rate can be expressed in terms of the component of the zero effort miss perpendicular to the line of sight or

$$\dot{\lambda} = \frac{ZEM_{PLOS}}{R_{TM} t_{go}}$$

If we assume that the relative separation between missile and target and closing velocity are approximately related to the time to go by

$$R_{TM} = V_c t_{go}$$

then the proportional navigation guidance command can be expressed in terms of the zero effort miss perpendicular to the line sight as

$$n_c = \frac{N' ZEM_{PLOS}}{t_{go}^2}$$

Thus, we can see that the proportional navigation acceleration command that is perpendicular to the line of sight is not only proportional to the line-of-sight rate and closing velocity but is also proportional to the zero effort miss and inversely proportional to the square of time to go. We shall later see that this is a very powerful concept, since the zero effort miss can be computed by a variety of methods, including the on-line numerical integration of the assumed nonlinear differential equations of the missile and target.

Summary

In this chapter we have developed and shown the results of a simple two-dimensional proportional navigation missile-target engagement simulation. Results have shown that the proportional navigation law is effective in a variety of cases. Linearization of the nonlinear missile-target geometry was shown to be an accurate approximation to the actual geometry. Closed-form solutions were derived, based on the linearized geometry, for the missile acceleration requirements due to heading error and target maneuver. From these solutions it was shown how the effective navigation ratio influences system performance. Finally, the concept of zero effort miss was introduced, and it was shown how the proportional navigation guidance law could be expressed in terms of this concept. In later chapters we shall develop more advanced guidance laws based upon the zero effort miss concept.

References

¹Kennedy, G. P., *Rockets, Missiles and Spacecraft of the National Air and Space Museum*, Smithsonian Institution Press, Washington, DC, 1983.

²Benecke, T., and Quick, A. W. (eds.), "History of German Guided Missile Development," *Proceedings of AGARD First Guided Missile Seminar*, 1956.

³Nesline, F. W., and Zarchan, P., "A New Look at Classical Versus Modern Homing Guidance," *Journal of Guidance and Control*, Vol. 4, Jan.–Feb. 1981, pp. 78–85.

⁴Yuan, C. L., "Homing and Navigation Courses of Automatic Target-Seeking Devices," RCA Labs., Rept. PTR-12C, Princeton, NJ, Dec. 1942.

⁵Bennett, R. R., and Mathews, W. E., "Analytical Determination of Miss Distance for Linear Homing Navigation Systems," Hughes Aircraft Co., TN-260, Culver City, CA, March 1952.

⁶Fossier, M. W., "The Development of Radar Homing Missiles," *Journal of Guidance, Control, and Dynamics*, Vol. 7, Nov.-Dec 1984, pp. 641-651.

⁷Yuan, C. L., "Homing and Navigation Courses of Automatic Target-Seeking Devices," *Journal of Applied Physics*, Vol. 19, Dec. 1948, pp. 1122-1128.

⁸Bryson, A. E., and Ho, Y. C., *Applied Optimal Control*, Blaisdell, Waltham, MA, 1969.

⁹Regan, F. J., *Re-Entry Vehicle Dynamics*, AIAA Education Series, AIAA, New York, 1984.

¹⁰Locke, A. S., *Guidance*, D. Van Nostrand, Toronto, 1955.

¹¹Jerger, J. J., *System Preliminary Design*, Van Nostrand, Princeton, NJ, 1960.

¹²Garnell, P., and East, D. J., *Guided Weapon Control System*, Pergamon, Oxford, 1977.



Method of Adjoints and the Homing Loop

Introduction

ALTHOUGH direct simulation is always used in evaluating missile system designs, the adjoint technique has historically been the main computerized analysis and design tool used in tactical missile guidance system design. The adjoint technique goes back at least to Vito Volterra,¹ circa 1870, and was used particularly by the ballisticians in connection with their theoretical studies of artillery hit dispersions.² The adjoint was popularized by Laning and Battin in the 1950s.³

The adjoint technique is based on the system impulse response and can be used to analyze linear time-varying systems such as the missile homing loop. With the adjoint method, exact performance projections of any quantity at a particular time and information showing how all disturbance terms contribute to the performance are available.^{4,5} In other words, error budgets are automatically generated with the adjoint technique. Although this technique has mainly been used in missile guidance system design and analysis, its application potential is much broader.

In this chapter we shall show how to construct an adjoint model from a missile guidance system homing loop. Numerical examples will be presented that demonstrate the power and utility of the adjoint approach. Performance projection comparisons will be made from nonlinear engagement simulation results and adjoint solutions.

Homing Loop

It is convenient to take the linearized engagement equations of Chapter 2 and draw them in block diagram form as is shown in Fig. 3.1. This type of block diagram is known as a homing loop because it is drawn as a feedback control system. In this diagram missile acceleration is subtracted from target acceleration to form a relative acceleration. After two integrations we have relative position, which at the end of the flight is the miss distance. A division by range (or the closing velocity multiplied by the time to go until intercept) yields the geometric line-of-sight angle where the time to go is defined as

$$t_{go} = t_F - t$$

The missile seeker, which is represented in Fig. 3.1 as a perfect differentiator, attempts to track the target. Effectively the seeker takes the derivative of the geometric line-of-sight angle, thus providing a measurement of the line-of-sight rate. The noise filter must process the noisy line-of-sight rate measurement of the

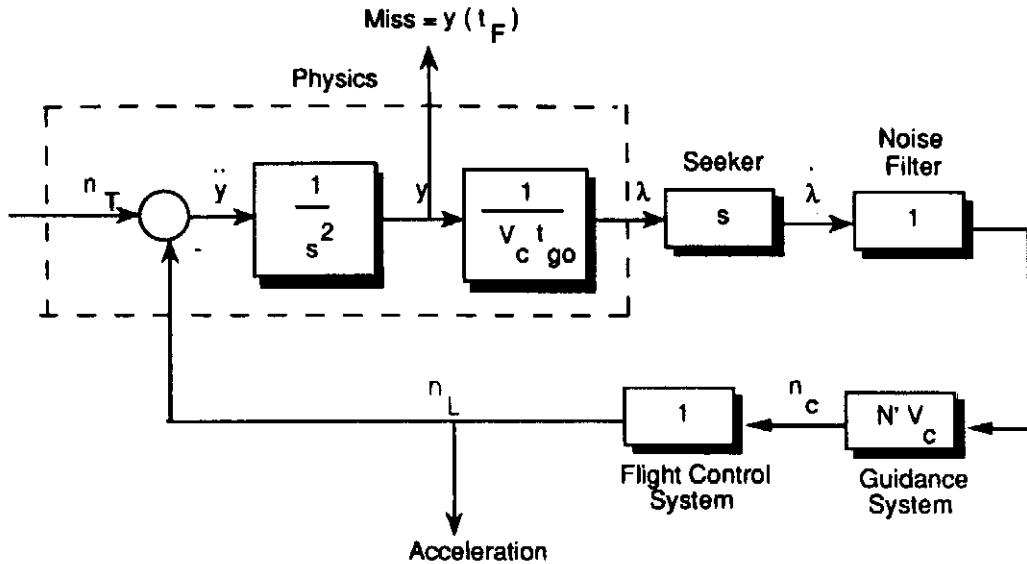


Fig. 3.1 Simplest possible proportional navigation guidance homing loop.

seeker and provide an estimate of the line-of-sight rate. A guidance command is generated, based on the proportional navigation guidance law, from the noise filter output. In tactical aerodynamic missiles the flight-control system (which is represented by unity gain in Fig. 3.1) must, by moving control surfaces, cause the missile to maneuver in such a way that the achieved acceleration matches the desired acceleration.

Figure 3.1 presents the simplest possible proportional navigation homing loop. In this perfect homing loop, models of the seeker, noise filter, guidance, and flight-control systems have been considered to be perfect and without dynamics. Such a block diagram is known as a zero-lag guidance system. The miss distance will always be zero in a zero-lag proportional navigation homing loop.

Guidance system lags or subsystem dynamics will cause miss distance. As long as the lags can be represented by either linear differential or difference equations, the homing loop will still remain linear and more powerful methods of analysis, such as the method of adjoints, can be used to determine system performance and behavior.

Single Time Constant Guidance System

Thus far, in our homing loop analysis, the missile has always hit the target. The strength of proportional navigation is that, in the absence of acceleration saturation effects, zero miss distance can be achieved if there are no lags within the homing loop. If the flight-control system dynamics were modeled as a single lag, or

$$\frac{n_L}{n_c} = \frac{1}{1 + sT}$$

where n_L is the achieved missile acceleration, n_c the commanded missile acceleration, and T the flight-control system time constant. Note that the relative acceleration equation in Fig. 3.1 would also have to be modified to

$$\ddot{y} = n_T - n_L$$

To determine how flight-control system time constant influences miss distance, a massive simulation experiment was conducted. Both the linear and nonlinear engagement simulations, developed in Chapter 2, were run for many different flight times. Each simulation trial had a 1-s flight-control system time constant ($T = 1$ s), an effective navigation ratio of 4, a -20 -deg heading error, and a different flight time. The flight times ranged from 0.1 s to 10 s in steps of 0.1 s.

In the linearized model the miss distance $y(t_F)$ can be either positive or negative. Recall that we have already included a few lines of extra code in the nonlinear engagement simulation to also determine if the miss is positive or negative. A positive miss means that the target is above the missile, whereas a negative miss means the opposite.

The miss distance results for each run representing a different flight time, for both the linear and nonlinear engagement models, were recorded. Figure 3.2 displays miss distance as a function of flight time for both the linear and nonlinear engagement simulation results. First, the figure shows that a 1-s flight-control system time constant can have a profound influence on the miss distance. In order for the miss distance to be negligible, the flight time must be large when compared to the flight-control system time constant. In addition, we can see from Fig. 3.2 that the linearized engagement model and nonlinear engagement model results are in close proximity, which demonstrates that the linearized model accurately captures the interaction between guidance system dynamics (flight-control system time constant) and miss distance for the heading error disturbance.

Another important disturbance is target maneuver. We saw in Chapter 2 that the linearization of the engagement model in the case of target maneuver was not as accurate as it was for heading error. It is important to determine if the inaccuracy in linearization leads to false conclusions concerning system performance. Both the linear and nonlinear engagement simulations were run for many different flight times, each run having a 1-s flight-control system time constant, an effective navigation ratio of 4, and a 3-g target maneuver. Figure 3.3 displays miss distance as a function of the flight time. Again we can see that the miss due to a constant target maneuver is only negligible if the flight time is much larger than the flight-control

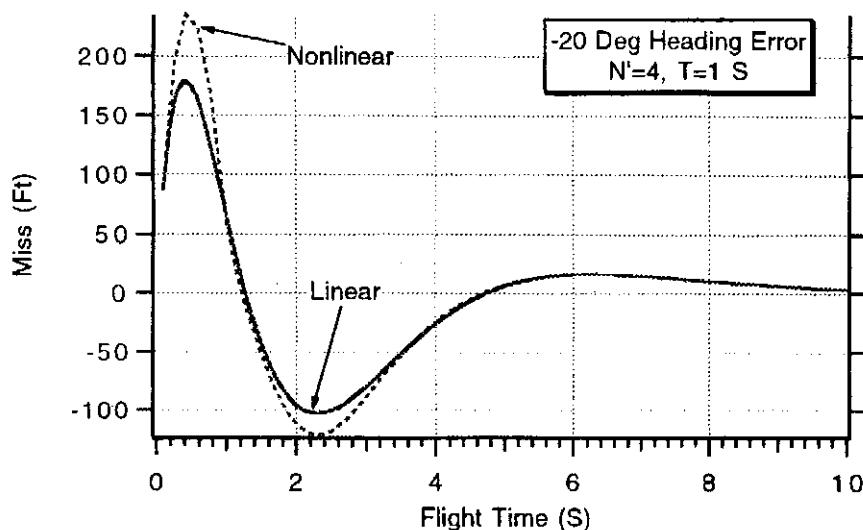


Fig. 3.2 Both models show that heading error miss approaches zero as flight time increases.

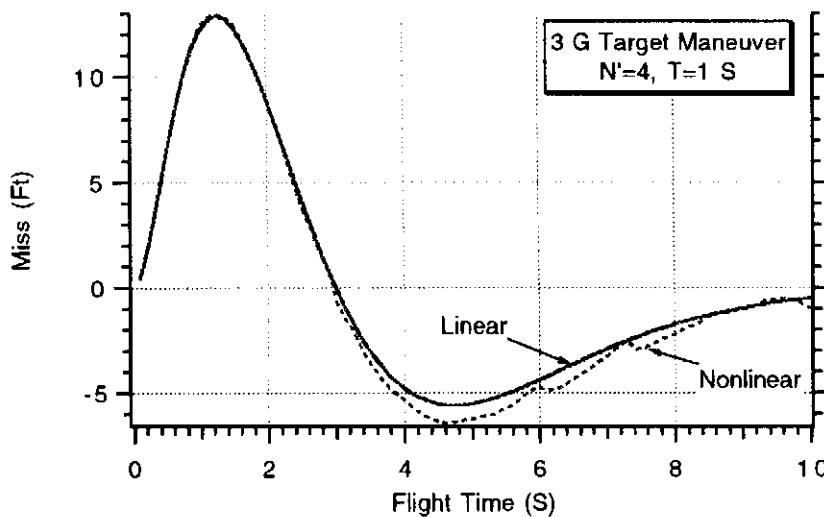


Fig. 3.3 Linear model accurately captures relationship between miss and flight time.

system time constant. In addition, Fig. 3.3 shows that the linearized guidance system model accurately captures the effect of flight control-system time constant on miss distance. The jaggedness in the nonlinear results is due to the approximate way in which the miss distance is computed. At the end of the flight, the integration step size is reduced to 0.0002 s. This means that, for the case considered, where the closing velocity is 4000 ft/s, the nonlinear miss distance computation is only good to 0.8 ft ($4000 * 0.0002$).

In this section we have shown the very important result that, when the homing loop has guidance system dynamics, the linearized guidance system model yields very accurate miss distance performance projections for both heading error and target maneuver disturbances. Therefore, techniques that depend on the linearized engagement model for miss distance projections should also be accurate.

How to Construct an Adjoint

In this section we will see how the miss distance results of Figs. 3.2 and 3.3, which were generated from many simulation trials, can be obtained in one computer run using the method of adjoints. However, we must first learn how to construct an adjoint model from the original system.

For every linear deterministic system there exists an adjoint system that can be constructed from the original system, given in block diagram form, by application of the following rules³⁻⁶:

Rule 1: Convert All System Inputs to Impulses

To construct an adjoint we must have impulsive inputs in the original system. Since impulsive inputs may not exist in the original system, block diagram manipulation of the actual inputs of the original system may be necessary. For example, deterministic inputs can be converted to impulsive inputs by judicious use of integrators. Figure 3.4 shows that a step input and an integrator-driven impulse are equivalent at the integrator output. The figure also shows that an initial condition is equivalent at the integrator output to an integrator with an impulsive input.

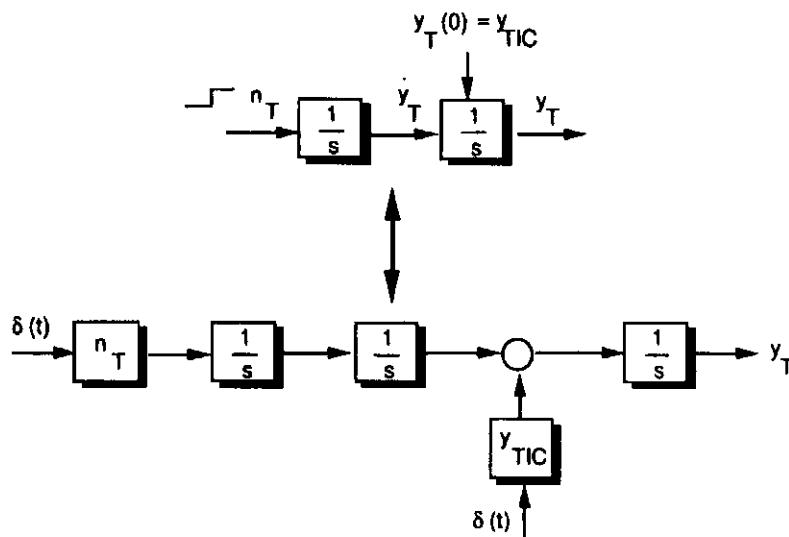


Fig. 3.4 Steps and initial conditions can be replaced by impulses.

Rule 2: Replace t by $t_F - t$ in the Arguments of All Time-Varying Coefficients

Often a linear system has a gain that can be expressed as a function of time, either in analytical or tabular form. Care must be taken with time-varying gains when applying the method of adjoints. Figure 3.5 shows, by example, how both a time-varying gain and a gain expressed as a tabular function of time can be converted to the adjoint domain. Notice that the adjoint of a table, which is a function of time, is the same table with the gains reversed. Otherwise, gains that are a function of time simply have t replaced by $t_F - t$ when the adjoint is taken, where t_F is the final time or time of flight.

Rule 3: Reverse All Signal Flow, Redefining Nodes as Summing Junctions and Vice Versa

Figure 3.6 shows how summing junctions and nodes are converted in going from the original to the adjoint system. Note that all original system inputs become adjoint outputs and vice versa. This simple relationship between the two systems

System Function \ System Function	Original System	Adjoint System																				
Time Varying Gain	$K(t) = at + b$ $K(t) = \frac{1}{a(t_F - t) + b}$	$K(t_F - t) = a(t_F - t) + b$ $K(t_F - t) = \frac{1}{at + b}$																				
Table	<table border="1"> <thead> <tr> <th>t</th> <th>K</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>8</td> </tr> <tr> <td>1</td> <td>4</td> </tr> <tr> <td>2</td> <td>3</td> </tr> <tr> <td>3</td> <td>9</td> </tr> </tbody> </table>	t	K	0	8	1	4	2	3	3	9	<table border="1"> <thead> <tr> <th>t</th> <th>K</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>9</td> </tr> <tr> <td>1</td> <td>3</td> </tr> <tr> <td>2</td> <td>4</td> </tr> <tr> <td>3</td> <td>8</td> </tr> </tbody> </table>	t	K	0	9	1	3	2	4	3	8
t	K																					
0	8																					
1	4																					
2	3																					
3	9																					
t	K																					
0	9																					
1	3																					
2	4																					
3	8																					

Fig. 3.5 Taking the adjoint of a time-varying gain.

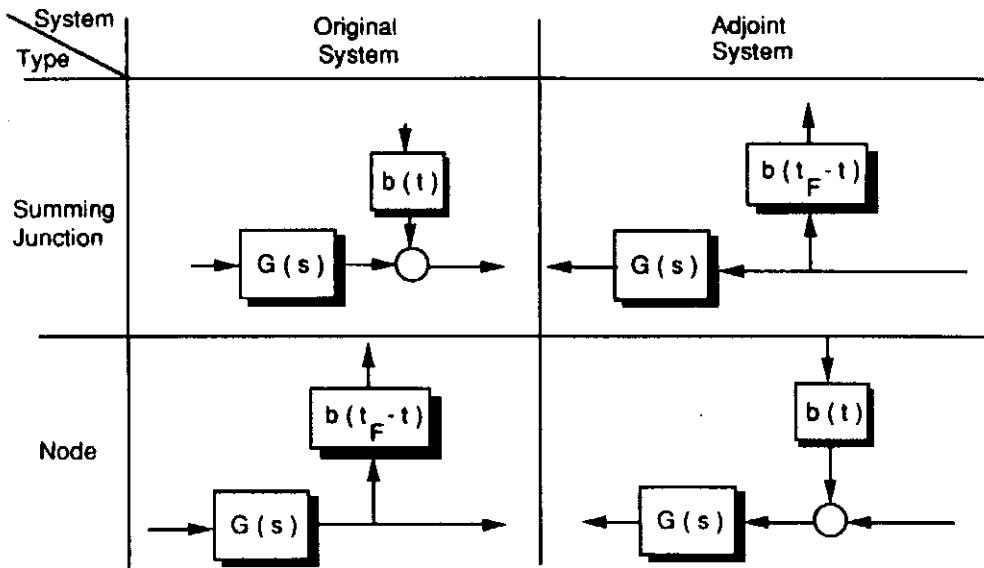


Fig. 3.6 Adjoints redefine branch points and nodes.

enables one to take an adjoint by first drawing the original block diagram and then using tracing paper to construct the adjoint model.

Figure 3.7 presents an example of a single-lag, proportional navigation homing loop in which a step target maneuver disturbance has been converted to an impulsive input by the use of an extra integrator. In addition, a heading error initial condition has also been converted to an impulsive input. The output of interest is the miss distance or $y(t_F)$. A simulation of this block diagram will yield y as a function of time $y(t)$, and the last value of y will be the miss distance $y(t_F)$. To find the miss due to a target maneuver disturbance, we would have to set the heading error disturbance to zero, and to find the miss due to an initial heading error, the target maneuver disturbance would have to be set to zero.

The adjoint of this homing loop, obtained by following the rules for constructing an adjoint, is shown in Fig. 3.8. Here the original output of interest [miss distance or $y(t_F)$] becomes an impulsive input to the adjoint system, and the two original

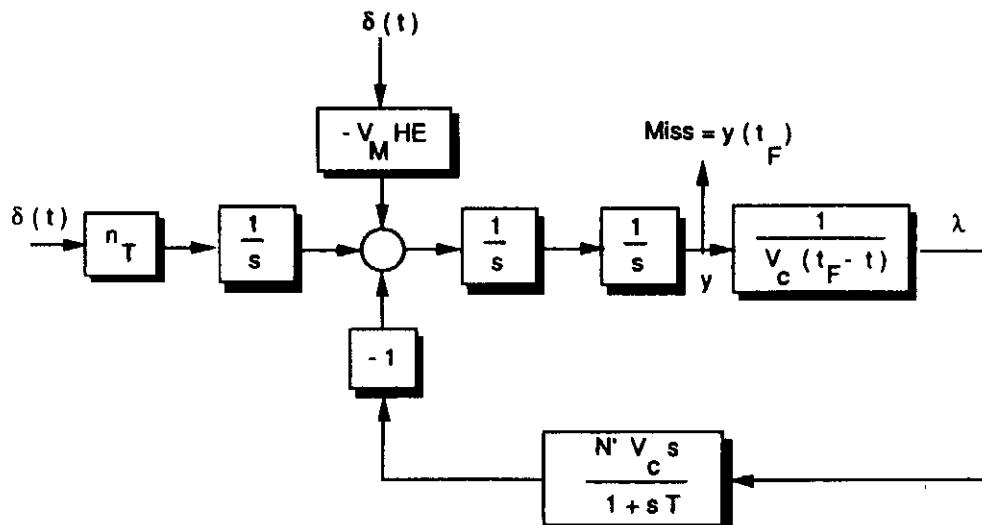


Fig. 3.7 Single-lag proportional navigation homing loop.

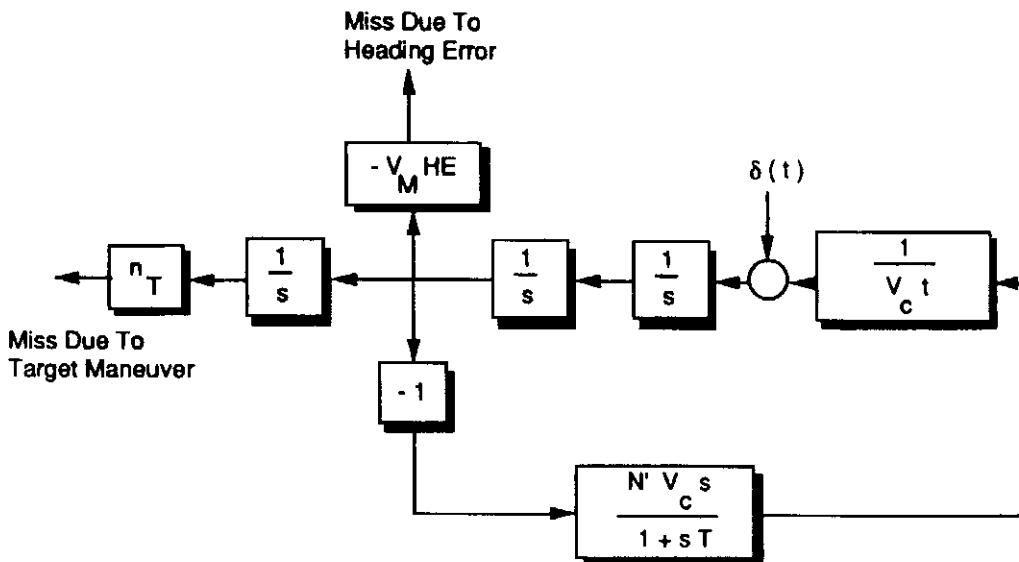


Fig. 3.8 Adjoint of homing loop.

system inputs (target maneuver and heading error) become two adjoint outputs. A simulation of the adjoint block diagram will yield y as a function of flight time $y(t_F)$. This means that in an adjoint simulation we obtain the miss distances due to both a step target maneuver and initial heading error for various flight times—all obtained in *one* computer run!

Adjoint Mathematics

The impulse response of the adjoint system h^* and the impulse response of the original system h are related by

$$h^*(t_F - t_I, t_F - t_O) = h(t_O, t_I)$$

where t_I and t_O are the impulse application and observation times, respectively, of the original system. This equation means that applying an impulse at time t_I and observing the output at time t_O of the original system is equivalent to applying an impulse to the adjoint system at time $t_F - t_O$ and observing the adjoint output at time $t_F - t_I$. The importance of this fundamental relationship becomes more apparent when it is desired to observe the impulse response of the original system at time t_F due to various impulse application times. This means that in order to generate $h(t_F, t_I)$ it becomes necessary to simulate the system response for each impulse application time as shown in Fig. 3.9.

However, since the observation time is the final time ($t_O = t_F$), only one adjoint response need be generated since the fundamental adjoint relationship simplifies to

$$h^*(t_F - t_I, 0) = h(t_F, t_I)$$

Therefore, an impulse applied at any time t_I and observed only at the final time t_F in the original system is equivalent to applying an impulse at time zero in the adjoint system and monitoring the output at time $t_F - t_I$. Figure 3.10 shows that the adjoint impulse response is identical to the impulse response of the original system, except that it is generated backwards!

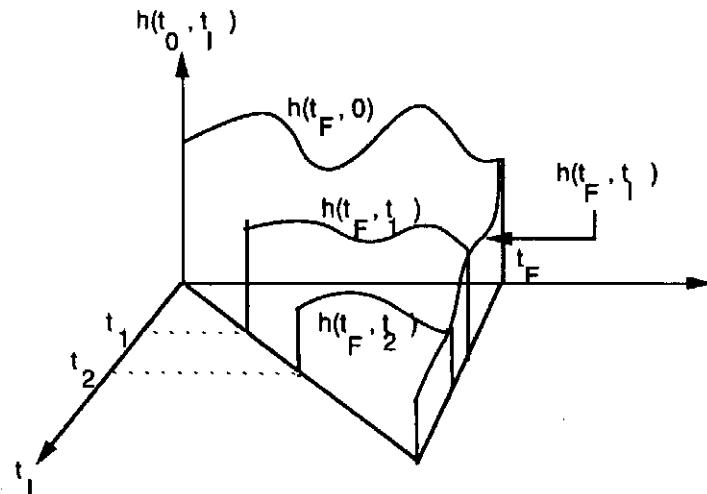


Fig. 3.9 Impulse response of original system for different application times.

Adjoints for Deterministic Systems

To fully understand the utility of adjoints, let us consider the convolution integral, or

$$y(t) = \int_{-\infty}^t x(\tau)h(t, \tau) d\tau$$

where x is the system input and h the system impulse response. For physically realizable (noncausal) systems, this integral becomes

$$y(t) = \int_0^t x(\tau)h(t, \tau) d\tau$$

A step input of magnitude a changes the preceding equation to

$$y(t) = a \int_0^t h(t, \tau) d\tau$$

Therefore, this equation states that the step response of a system can be found by integrating the impulse response. A closer examination reveals that this revelation

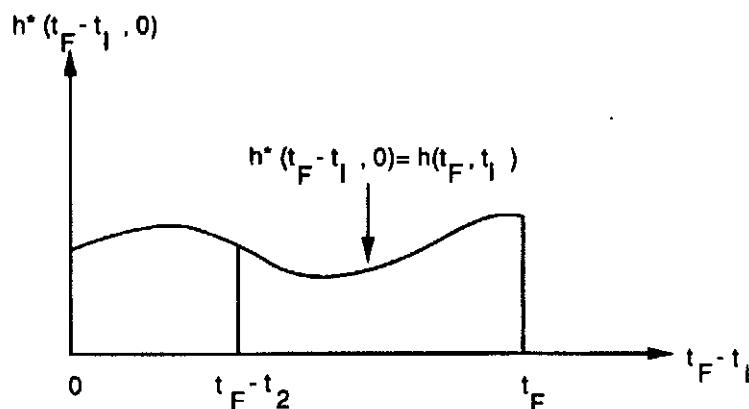


Fig. 3.10 Impulse response of adjoint system is related to impulse response of original system.

is of no practical utility because the variable of integration is with respect to τ . This means that many impulse responses, each with a different application time, would have to be generated. Then the results would have to be saved and then integrated—just to get a system step response! Of course, it would be much easier to avoid the convolution integral and to just simulate the system with the step input and observe the output in order to get the system step response.

Let us now see if the method of adjoints can be useful in the case where the system has a step input. We can substitute the fundamental relationship between the original and adjoint system impulse responses into the convolution integral, yielding

$$y(t) = a \int_0^t h^*(t_F - \tau, t_F - t) d\tau$$

Variables can be changed according to

$$x = t_F - \tau$$

$$dx = -d\tau$$

Hence, we obtain

$$y(t) = a \int_{t_F-t}^{t_F} h^*(x, t_F - t) dx$$

If the observation time of interest is the final time ($t = t_F$), the preceding relationship simplifies to

$$y(t_F) = a \int_0^{t_F} h^*(x, 0) dx$$

Note that the integration in the adjoint system is with respect to the observation time rather than the impulse application time. This means that the original system step response output at time t_F can be obtained by using an impulsive input in the adjoint system at time zero and then integrating the output. The resultant one computer run adjoint solution obtains the step response value at the final time for *all* final time values! For time-invariant systems the original system step response, for all final times, could also have been obtained in one computer run. However, for time-varying systems, many original system computer runs would have been required to obtain the same information as that provided by the adjoint response.

To see further benefits from the method of adjoints, let us consider many step input disturbances to the original system as shown in Fig. 3.11. Here, not only would many computer runs be required to find the system step response for different flight times, but also information showing how each disturbance contributed to the total output would require many more computer runs. To generate this type of error budget in the original system for N disturbances, N computer runs would be required (each run only having that disturbance), and then superposition could be invoked (add up all responses) to get the total step response. However, original system inputs become adjoint outputs; thus, only one adjoint run would be required to get the total step response value at the final time and to also automatically generate an error budget. Only the adjoint outputs (original system inputs) have to be monitored, as shown in Fig. 3.11, and superposition allows the total output to

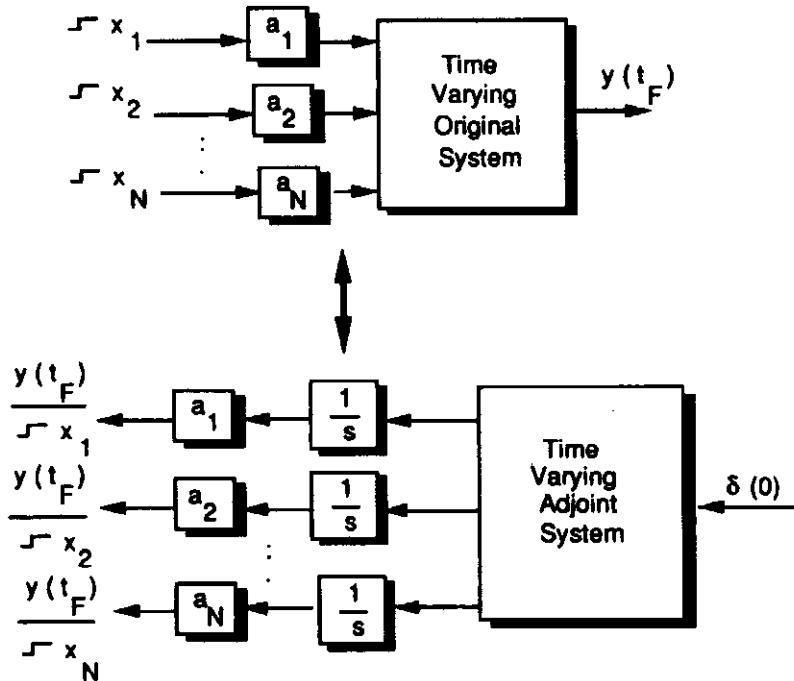


Fig. 3.11 Equivalence between adjoint and original systems for deterministic step disturbances.

be expressed as

$$y(t_F) = \frac{y(t_F)}{\int x_1} + \frac{y(t_F)}{\int x_2} + \cdots + \frac{y(t_F)}{\int x_N}$$

Thus, when there are many deterministic disturbances to the original system, one adjoint computer run yields the system response for all final times, along with a detailed error budget showing how each disturbance influenced total system performance.

Deterministic Adjoint Example

To demonstrate the practical utility of adjoint theory for a system with deterministic inputs, let us reconsider the proportional navigation homing loop example of Fig. 3.7. After following the rules for constructing an adjoint, we obtain the detailed adjoint model shown in Fig. 3.12.

All of the integrator inputs and outputs are marked in Fig. 3.12 for the purpose of understanding the adjoint simulation. With the exception of the two adjoint outputs, none of the quantities shown in the adjoint model has any physical meaning. The impulse required for adjoint initiation can also be represented as an initial condition on the x_3 integrator. The output x_1 is the miss distance sensitivity (multiply by n_T to get miss) due to a step target maneuver, whereas the output x_2 is related to the miss sensitivity (multiply by $-V_M HE$) due to an initial heading error.

Listing 3.1 presents the FORTRAN adjoint simulation of Fig. 3.12. Here the four differential equations of Fig. 3.12 (which appear after label 200 in listing) are integrated using the second-order Runge-Kutta method with an integration step size of 0.01 s. The integration step size is small enough in this sample program to get fairly accurate answers. Note that in this example the effective navigation ratio

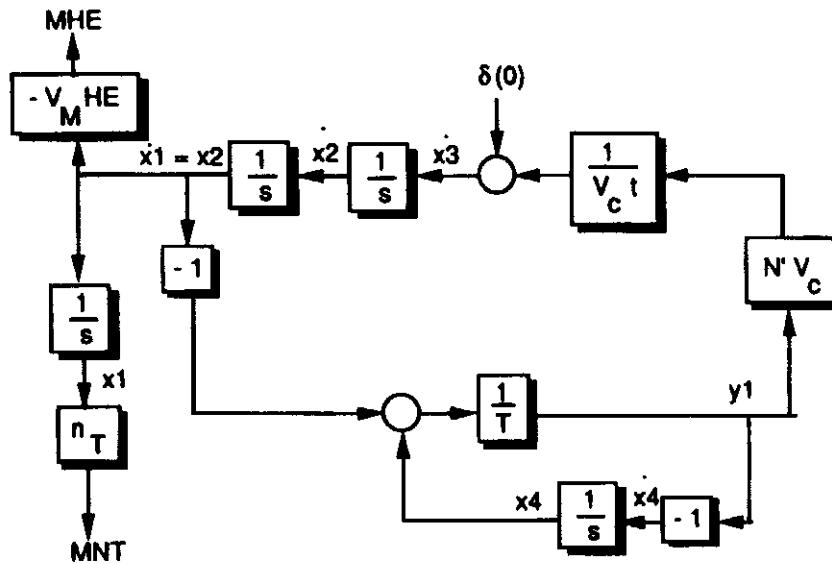


Fig. 3.12 Adjoint simulation model of single-lag guidance system.

is four, the guidance time constant is 1 s, there is -20 deg of heading error, and the maneuver level is 96.6 ft/s^2 or 3 g .

The adjoint output due to a 3-g step target maneuver appears in Fig. 3.13 along with results from the linearized engagement simulation (run for many different flight times). The figure shows that the adjoint simulation yields accurate miss distance projections for many different flight times in one computer run. Since the adjoint is linear, we can apply superposition to the results. Doubling the target maneuver acceleration level doubles the miss. The abscissa of Fig. 3.13 can either be interpreted as flight time or the time to go until intercept at which the target initiates its maneuver. Therefore, in this example, a 3-g maneuver causes nearly 12 ft of miss if the flight time is only 1 s or if the maneuver occurs at 1 s to go before intercept. The figure also shows that long flights (flight time large compared to guidance system time constant), or flights with maneuver initiation occurring at the beginning of flight (large time to go), result in virtually zero miss.

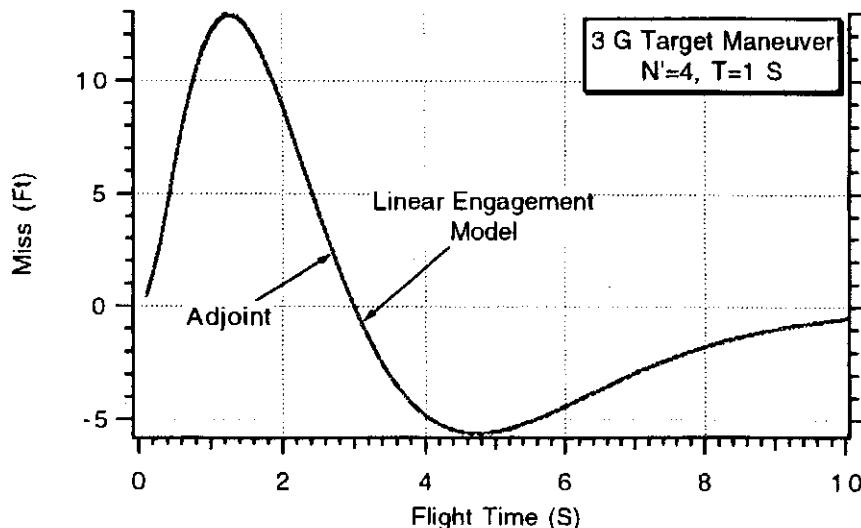


Fig. 3.13 Adjoint yields accurate miss distance information for all flight times in one run.

**Listing 3.1 Single-lag adjoint with second-order
Runge-Kutta integration**

```

INTEGER STEP
DATA XNT,XNP,TAU,TF,VM,HEDEG/ 96.6,4.,1.,10.,3000.,-20./
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
T=0.
S=0.
TP=TP+.00001
X1=0
X2=0
X3=1
X4=0
H=.01
HE=HEDEG/57.3
10 IF(TP>(TF-.00001))GOTO 999
S=S+H
X1OLD=X1
X2OLD=X2
X3OLD=X3
X4OLD=X4
STEP=1
GOTO 200
66 STEP=2
X1=X1+H*X1D
X2=X2+H*X2D
X3=X3+H*X3D
X4=X4+H*X4D
TP=TP+H
GOTO 200
55 CONTINUE
X1=(X1OLD+X1)/2+.5*H*X1D
X2=(X2OLD+X2)/2+.5*H*X2D
X3=(X3OLD+X3)/2+.5*H*X3D
X4=(X4OLD+X4)/2+.5*H*X4D
IF(S<.09999)GOTO 10
S=0.
XMNT=XNT*X1
XMHE=-VM*HE*X2
WRITE(*,97)TP,XMNT,XMHE
WRITE(1,97)TP,XMNT,XMHE
GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
X1D=X2
X2D=X3
Y1=(X4-X2)/TAU
TGO=TP+.00001
X3D=XNP*Y1/TGO

```

(Contd.)

Listing 3.1 (Continued)

```

X4D=-Y1
IF(STEP-1)66,66,55
999  CONTINUE
PAUSE
CLOSE(1)
END

```

Another adjoint output, from the same simulation trial, also yields miss distance information for a heading error disturbance. These results, along with linearized engagement simulation results (run for many different flight times) appear in Fig. 3.14. We can see that the adjoint also yields accurate performance projections in the case of the heading error disturbance.

Therefore, we can see that a great deal of information is available from one adjoint run representing a system with many deterministic disturbances. Miss distance sensitivity data for a variety of disturbances and for all flight times can be obtained in a single adjoint run. We can also tell from the adjoint output when the guidance system is most sensitive to an error source. In addition, the shape of the adjoint output also provides information concerning system behavior. For example, if the adjoint output for a deterministic input does not approach zero as the flight time increases, then we know that the missile cannot guide effectively.

Adjoint Closed-Form Solutions^{7,8}

If there are no dynamics in a proportional navigation homing loop, then the resultant miss distance should always be zero. Guidance system dynamics cause miss distance. Under special circumstances it is possible to obtain closed-form solutions for the miss distance when there are dynamics in the homing loop. These closed-form solutions can be used to gain insight into the effectiveness of homing

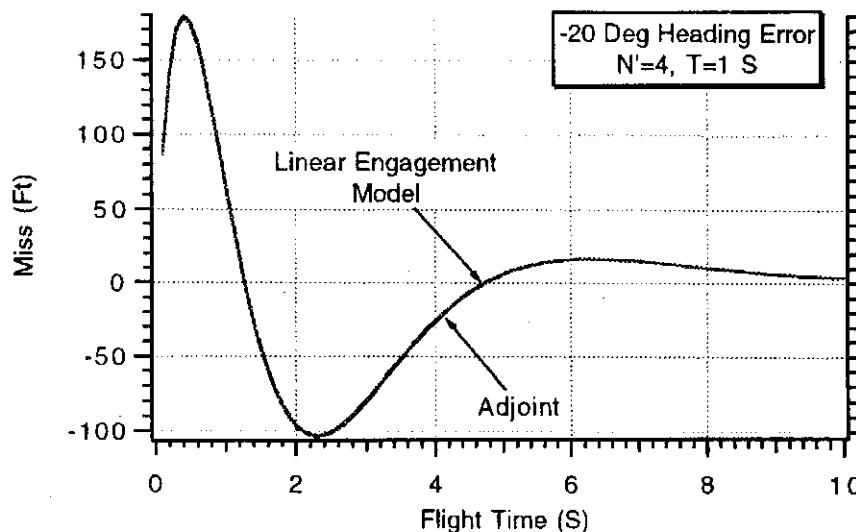


Fig. 3.14 Adjoint also yields accurate heading-error-induced miss information for all flight times in one run.

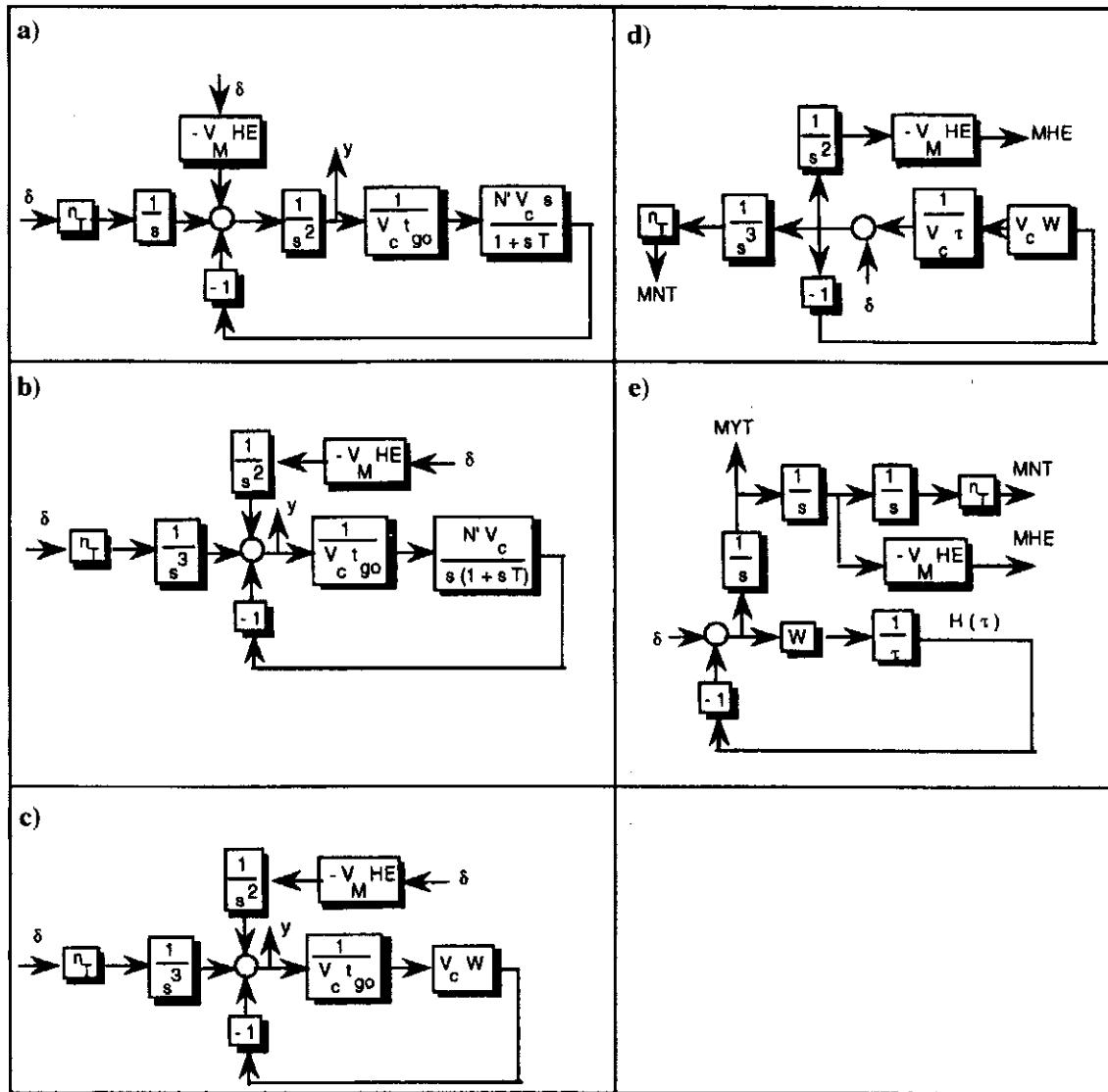


Fig. 3.15 Methodology in getting generalized homing loop adjoint. a) Single time constant proportional navigation guidance system; b) block diagram manipulation; c) replace guidance transfer function with W ; d) take adjoint for miss distance; e) rearrange block diagram.

and also be used to check the accuracy of computerized adjoints. In addition, we shall see later that the normalization factors developed from simpler systems are also valid for more complex systems.

Consider the single time constant proportional navigation guidance system shown in Fig. 3.15a. In this guidance system there are two disturbances (i.e., target maneuver and heading error) that are represented as impulsive inputs so that we can easily take the adjoint later. For convenience, we can use block diagram manipulation to remove the two integrations from the main loop and place them before the two inputs. After some work we obtain Fig. 3.15b.

For conceptual purposes we may be interested in guidance system dynamics other than first order. Therefore we will use shorthand notation and express the guidance system dynamics as W . For a single-lag guidance system W can be expressed as

$$W_{\text{Single Lag}} = \frac{N'}{s(1+sT)}$$

and in a perfect or zero-lag guidance system W is given by

$$W_{\text{Zero Lag}} = \frac{N'}{s}$$

In general W can represent guidance system dynamics of any order.

Using W to represent the guidance system dynamics, a more generalized proportional navigation guidance system is displayed in Fig. 3.15c. The miss distance adjoint of the generalized guidance system can be obtained by inspection (i.e., following the rules of adjoints outlined in this chapter) and applying an impulse at time zero where y was in the original system. The resultant generalized adjoint block diagram appears in Fig. 3.15d. Using block diagram manipulation and reorienting the figure yields the final generalized adjoint shown in Fig. 3.15e. We can see that the miss due to a step target maneuver MNT is after three integrations while the miss due to heading error MHE (i.e., step in target velocity) is after two integrations. Therefore MYT , which is after one integration, must represent a step in target displacement. This important error source will be investigated in detail in Chapter 20.

For convenience, $H(\tau)$ is also indicated as an adjoint signal of interest. From the convolution integral we can relate the adjoint output to the input by

$$H(\tau) = \frac{1}{\tau} \int W(x)[\delta(\tau - x) - H(\tau - x)] dx$$

Converting from the time to the frequency domain (Laplace transform notation), we can express the preceding relationship as

$$\frac{-dH(s)}{ds} = W(s)[1 - H(s)]$$

because of a Laplace transform identity. Recall also that

$$\frac{d}{ds}[1 - H(s)] = \frac{-dH(s)}{ds}$$

For convenience let us allow $H(s)$ to be replaced by H and $W(s)$ to be replaced by W . Therefore, substitution yields

$$\int \frac{d(1 - H)}{1 - H} = \int W ds$$

Finally, integrating both sides of the preceding equations yields the important conclusion

$$1 - H = c \exp \left(\int W ds \right)$$

where c is a constant of integration. We can evaluate c by first recognizing from Fig. 3.15e that a miss due to a unit step in target displacement MYT can be expressed in Laplace transform notation as

$$MYT(s) = \frac{1 - H(s)}{s}$$

We know that in the time domain the miss due to a unit step target displacement at flight time zero is unity. Therefore, using the initial value theorem, which relates

the time domain with the Laplace transform domain, we can say that

$$MYT(0) = 1 = \lim_{s \rightarrow \infty} s \left(\frac{1 - H}{s} \right)$$

Therefore c is chosen to make

$$\lim_{s \rightarrow \infty} \left[c \exp \left(\int W ds \right) \right] = 1$$

Since the simplest possible guidance system has at least one integration, the constant of integration becomes unity ($c = 1$).

To demonstrate that we have enough analytical tools to find closed-form solutions under special circumstances, let us find the miss due to a step target maneuver for a single-lag guidance system. As mentioned before, the guidance system transfer function in this case is

$$W(s) = \frac{N'}{s(1 + sT)}$$

where N' is the effective navigation ratio and T the guidance system time constant. Since

$$1 - H = \exp \left(\int W ds \right)$$

we get after integration

$$1 - H(s) = \left[s \left/ \left(s + \frac{1}{T} \right) \right. \right]^{N'}$$

Therefore, the miss due to a step maneuver of magnitude n_T is given by

$$\frac{MNT}{n_T}(s) = \frac{1 - H(s)}{s^3} = \frac{1}{s^3} \left[s \left/ \left(s + \frac{1}{T} \right) \right. \right]^{N'}$$

For an effective navigation ratio of 4, the miss, in Laplace transform notation, is given by

$$\left. \frac{MNT}{n_T} \right|_{N'=4}(s) = \frac{s}{(s + 1/T)^4}$$

Taking the inverse Laplace transform of the preceding expression yields the miss due to a step target maneuver in the adjoint time domain as

$$\left. \frac{MNT}{n_T} \right|_{N'=4}(\tau) = \tau^2 e^{-\tau/T} \left(0.5 - \frac{\tau}{6T} \right)$$

where τ is adjoint time and can be interpreted as either time of flight (t_F) or time to go at which the maneuver occurs.

The single-lag adjoint simulation of Listing 3.1 was run for a case in which the target maneuver level was 3 g, the guidance system time constant was 1 s, and the effective navigation ratio was 4. Figure 3.16 displays the adjoint simulation results along with the closed-form solution. We can see from the figure that both

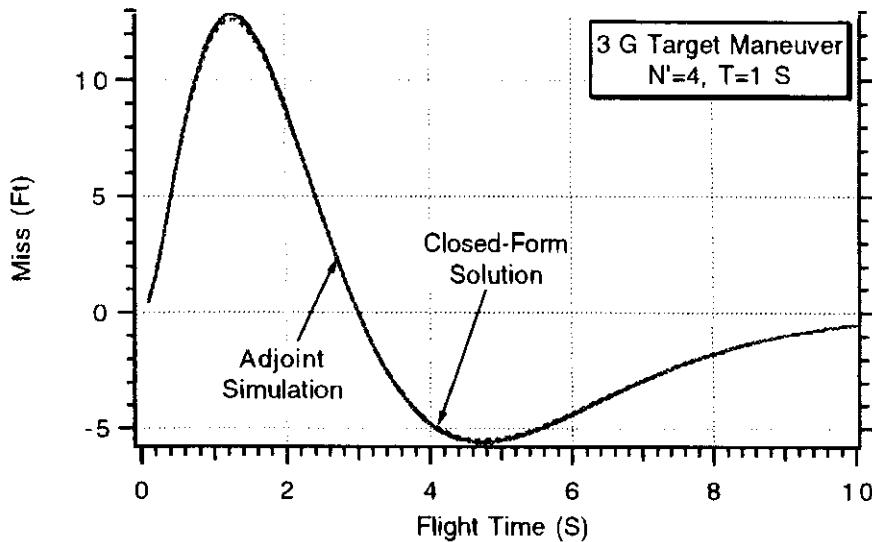


Fig. 3.16 Closed-form solution agrees with computerized adjoint for step target maneuver.

the adjoint results and closed-form solution results are virtually identical, which proves that it is possible for theory and simulation to agree.

The nonlinear engagement simulation of Chapter 2 was rerun for the same case (3-g target maneuver, $N' = 4$, and $T = 1$ s) for many different flight times. The nonlinear results and the closed-form solution results are compared in Fig. 3.17. We can see that the miss distance projections from the closed-form solution are in excellent agreement with the nonlinear results.

Similarly, a closed-form expression can be found for the miss due to an initial heading error. The generalized formula for the heading error miss is given by

$$\frac{MHE}{HE}(s) = \frac{-V_M[1 - H(s)]}{s^2}$$

Following a similar procedure to that of the target maneuver case and assuming

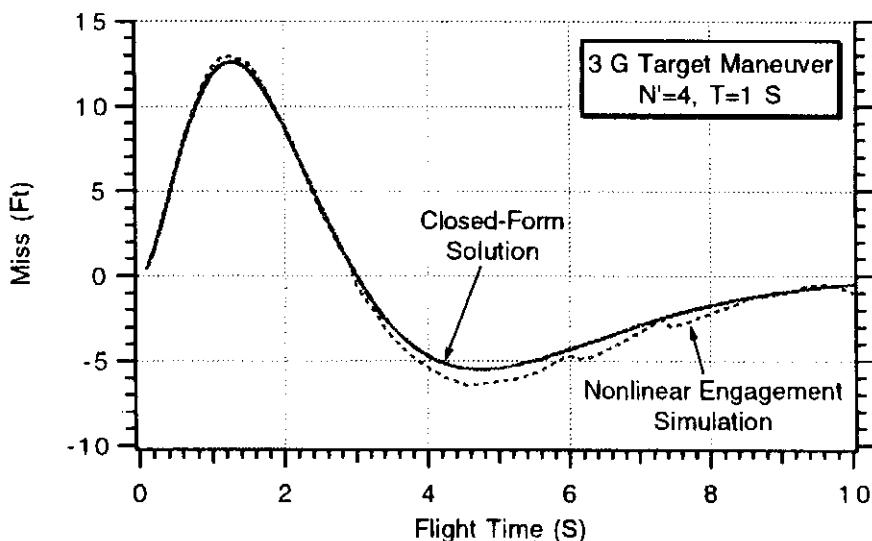


Fig. 3.17 Miss distance formula and nonlinear simulation results are in close agreement.

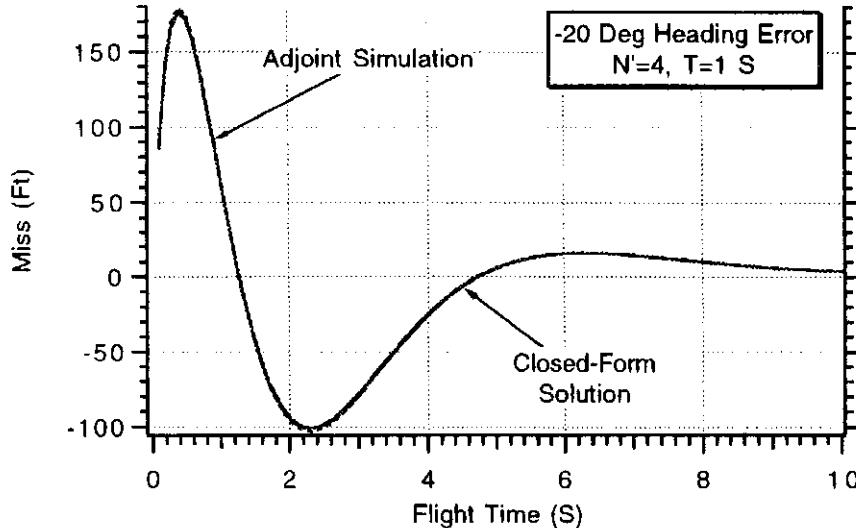


Fig. 3.18 Closed-form solution agrees with computerized adjoint for heading error.

an effective navigation ratio of 4 for a single-lag guidance system, we obtain a closed-form solution for the heading error miss:

$$\frac{\text{MHE}}{\text{HE}} \Big|_{N'=4}(\tau) = -V_M \tau e^{-\tau/T} \left(1 - \frac{\tau}{T} + \frac{\tau^2}{6T^2} \right)$$

The adjoint simulation results for a -20 -deg heading error and the closed-form solution results are displayed in Fig. 3.18. Again we can see that the adjoint simulation results are in complete agreement with the closed-form solution.

Normalization

In the previous section we showed how closed-form solutions for the heading error and target maneuver could be derived for a single time constant guidance system. Specific solutions were derived for the case in which the effective navigation ratio was 4. Following the same procedure outlined in the previous section, closed-form solutions were derived for the miss due to a target maneuver when the effective navigation ratio varied between 3 and 5. The solutions are

$$\frac{\text{Miss}}{n_T} \Big|_{N'=3} = 0.5 t_F^2 e^{-t_F/T}$$

$$\frac{\text{Miss}}{n_T} \Big|_{N'=4} = t_F^2 e^{-t_F/T} \left(0.5 - \frac{t_F}{6T} \right)$$

$$\frac{\text{Miss}}{n_T} \Big|_{N'=5} = t_F^2 e^{-t_F/T} \left(0.5 - \frac{t_F}{3T} + \frac{t_F^2}{24T^2} \right)$$

where n_T is the maneuver level of the target (in ft/s^2), t_F is the flight time (in s), T is the guidance system time constant (in s), and Miss is the miss distance (in ft).

In a similar way, miss distance formulas can be derived for heading error in a single time constant guidance system. The formulas are

$$\begin{aligned}\left. \frac{\text{Miss}}{-V_M \text{HE}} \right|_{N'=3} &= t_F e^{-t_F/T} \left(1 - \frac{t_F}{2T} \right) \\ \left. \frac{\text{Miss}}{-V_M \text{HE}} \right|_{N'=4} &= t_F e^{-t_F/T} \left(1 - \frac{t_F}{T} + \frac{t_F^2}{6T^2} \right) \\ \left. \frac{\text{Miss}}{-V_M \text{HE}} \right|_{N'=5} &= t_F e^{-t_F/T} \left(1 - 1.5 \frac{t_F}{T} + \frac{t_F^2}{2T^2} - \frac{t_F^3}{24T^3} \right)\end{aligned}$$

where V_M is the missile velocity (in ft/s) and HE the heading error (in rad).

The closed-form solutions for both the target maneuver and heading error miss can be normalized for conciseness. For example, Fig. 3.19 displays the target maneuver miss sensitivity, in normalized form, for various effective navigation ratios. From the normalization factor it becomes obvious that, as the ratio of the flight time to the guidance system time constant (t_F/T) becomes large, the miss eventually goes to zero. From the normalization on the ordinate it becomes obvious that, for a given ratio of flight time to guidance time constant, doubling the guidance system time constant quadruples the miss!

Figure 3.20 displays the heading error miss sensitivity, in normalized form, for various effective navigation ratios. In this case too, it is obvious from the figure that, as the ratio of the flight time to the guidance system time constant (t_F/T) becomes large, the miss eventually goes to zero. From the normalization on the ordinate, we can see that, for a given ratio of flight time to guidance time constant, doubling the guidance system time constant only doubles the heading error miss.

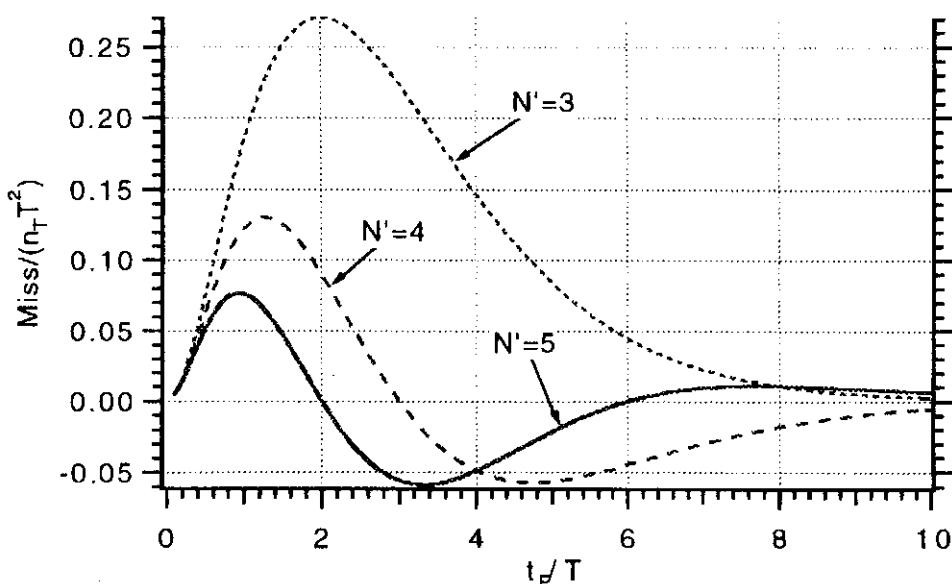


Fig. 3.19 Normalized miss due to target maneuver for single time constant guidance system.

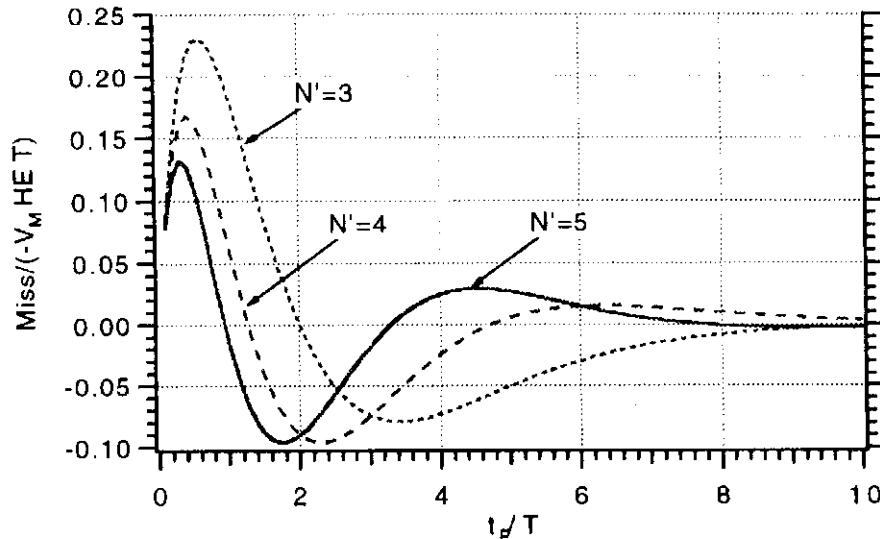


Fig. 3.20 Normalized miss due to heading error for single time constant guidance system.

Summary

In this chapter we have seen the power and accuracy of linearization. First we showed that the method of adjoints could be applied to the linearized homing loop. The adjoint technique permitted us to obtain miss distance performance projections as a function of flight time, in error budget form, for many inputs in a single adjoint computer run. The method was shown to be accurate when compared to linear performance projections obtained by massive simulation. It was also shown how the adjoint method could be used to derive miss distance formulas. These closed-form solutions agreed closely with results obtained by massive simulation trials of the nonlinear engagement simulation.

References

- ¹ Goldsmith, J. L., "A Discussion of the Adjoint Technique of System Analysis," Raytheon, Memo SDD-74-525, Bedford, MA, Oct. 1974.
- ² Bliss, G. A., *Exterior Ballistics*, Univ. of Chicago Press, Chicago, 1925.
- ³ Laning, J. H., and Battin, R. H., *Random Processes in Automatic Control*, McGraw-Hill, New York, 1956.
- ⁴ Zarchan, P., "Complete Statistical Analysis of Nonlinear Missile Guidance Systems—SLAM," *Journal of Guidance and Control*, Vol. 2, Jan.–Feb. 1979, pp. 71–78.
- ⁵ Peterson, E. L., *Statistical Analysis and Optimization of Systems*, Wiley, New York, 1961.
- ⁶ Zarchan, P., "Comparison of Statistical Digital Simulation Methods," *Advisory Group for Aerospace Research and Development*, AGARDograph No. 273, July 1988, pp. 2-1–2-16.
- ⁷ Travers, P., "Interceptor Dynamics," unpublished lecture notes, Raytheon, circa 1971.
- ⁸ Bennett, R. R., and Mathews, W. E., "Analytical Determination of Miss Distance for Linear Homing Navigation Systems," Hughes Aircraft Co., TM-260, Culver City, CA, March 1952.

Noise Analysis

Introduction

THE concept of noise is important to the guidance system engineer. For example, in a radar homing tactical missile the seeker measurement of the line-of-sight rate signal, required for the implementation of proportional navigation guidance, is not perfect but is corrupted by noise. To extract the signal from the measurement, an understanding of the concept of noise and its various properties are mandatory. In addition, in order to evaluate system performance in the presence of noise, we must first know how to simulate noise and then how to conduct and interpret experiments with repeated simulation trials. The concepts developed and illustrated in this chapter will be used throughout the text for filtering and evaluating system performance in the presence of noise or other random phenomenon.

Basic Definitions¹

In this section we will depart from our usual guidance discussions and start by defining some important quantities related to random variables. Since random variables have unknown specific values, they are usually quantified according to their statistical properties. One of the most important statistical properties of any random function x is its probability density function $p(x)$. This function is a measure of the likelihood of occurrence of each value of x and is defined such that

$$p(x) \geq 0$$

and

$$\int_{-\infty}^{\infty} p(x) dx = 1$$

This means that there is a probability that x will occur, and it is certain that the value of x is somewhere between plus and minus infinity. The probability that x is between a and b can be expressed in terms of the probability density function as

$$\text{Prob}(a \leq x \leq b) = \int_a^b p(x) dx$$

Another important quantity related to random variables is the distribution function. A distribution function $P(x)$ is the probability that a random variable is less than or equal to x . Therefore, if the probability density function is known, the distribution

function can be found by integration as

$$P(x) = \int_{-\infty}^x p(u) du$$

The mean or expected value of x is defined by

$$m = E(x) = \int_{-\infty}^{\infty} xp(x) dx$$

Therefore, the mean can also be thought of as the first moment of x . We can also think of the mean value of x as the sum (integral) of all values of x , each being weighted by its probability of occurrence. It can be shown that, if random variables x_1, \dots, x_n are independent, then the expectation of the sum is the sum of the expectations, or

$$E(x_1 + x_2 + \dots + x_n) = E(x_1) + E(x_2) + \dots + E(x_n)$$

The second moment or mean squared value of x is defined as

$$E(x^2) = \int_{-\infty}^{\infty} x^2 p(x) dx$$

Therefore, the rms of x can be obtained by taking the square root of the preceding equation, or

$$\text{rms} = [E(x^2)]^{1/2}$$

The variance of x , σ^2 is defined as the expected squared deviation of x from its mean value. Mathematically, the variance can be expressed as

$$\sigma^2 = E\{[x - E(x)]^2\} = E(x^2) - E^2(x)$$

We can see that the variance is the difference between the mean squared value of x and the square of the mean of x . If we have independent random variables x_1, \dots, x_n then the variance of the sum can be shown to be the sum of the variances, or

$$\sigma^2 = \sigma_1^2 + \sigma_2^2 + \dots + \sigma_n^2$$

The square root of the variance σ is also known as the standard deviation. In general, the rms value and standard deviation are not the same unless the random process under consideration has a zero mean.

An example of a probability density function is the uniform distribution, which is depicted in Fig. 4.1. With this probability density function all values of x between a and b are equally likely to occur. An important practical example of the uniform distribution, which should be familiar to any engineer who has programmed on a personal computer, is the BASIC language random number generator (RND). The BASIC RND statement supplies a uniformly distributed random number, on each call, between 0 and 1. Soon we will see how random numbers with different probability density functions can be constructed from random numbers following

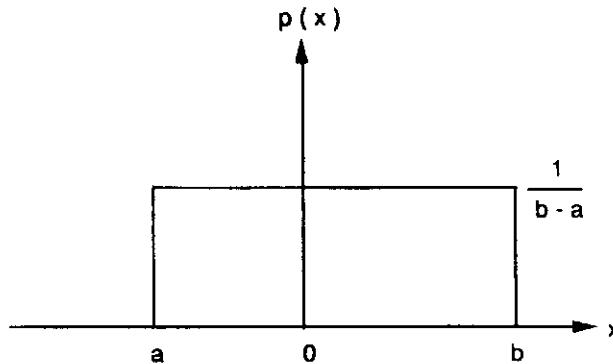


Fig. 4.1 Uniform probability distribution.

the uniform distribution. From our previous definitions we can see that the mean value of a uniform distribution is

$$m = \int_{-\infty}^{\infty} xp(x) dx = \frac{1}{b-a} \int_a^b x dx = \frac{b+a}{2}$$

This makes sense, since the expected or mean value is halfway between a and b . The variance of a uniform distribution can also be found from our previous definitions and can be shown to be

$$\sigma^2 = E(x^2) - m^2 = \frac{b^3 - a^3}{3(b-a)} - \left(\frac{b+a}{2}\right)^2 = \frac{(b-a)^2}{12}$$

This means that, if the random numbers from a uniform distribution vary from 0 to 1, the mean of the resultant set of numbers should be 1/2 and the variance should be 1/12. We will use this property of a uniform distribution later in this chapter for constructing random numbers with different probability density functions.

Another important probability density function is the Gaussian or normal distribution. In the missile homing loop we shall often treat the sensor noise disturbances as having a Gaussian distribution. The probability density function for this distribution is shown in Fig. 4.2 and is given by the formula

$$p(x) = \exp\left[-\frac{(x-m)^2}{2\sigma^2}\right] / [\sigma(2\pi)^{0.5}]$$

where m and σ are parameters. By using our basic definitions it is easy to show that the expected or mean value of a Gaussian distribution is given by

$$E(x) = m$$

and its variance is

$$E(x^2) - m^2 = \sigma^2$$

Therefore, m and σ in the expression for the Gaussian probability density function correspond to the mean and standard deviation, respectively.

We can see from Fig. 4.2 that this bell-shaped distribution is virtually zero after three standard deviations ($\pm 3\sigma$). Integration of the probability density function, to find the distribution function, shows that there is a 68% probability that the Gaussian random variable is within one standard deviation ($\pm\sigma$) of the mean, 95%

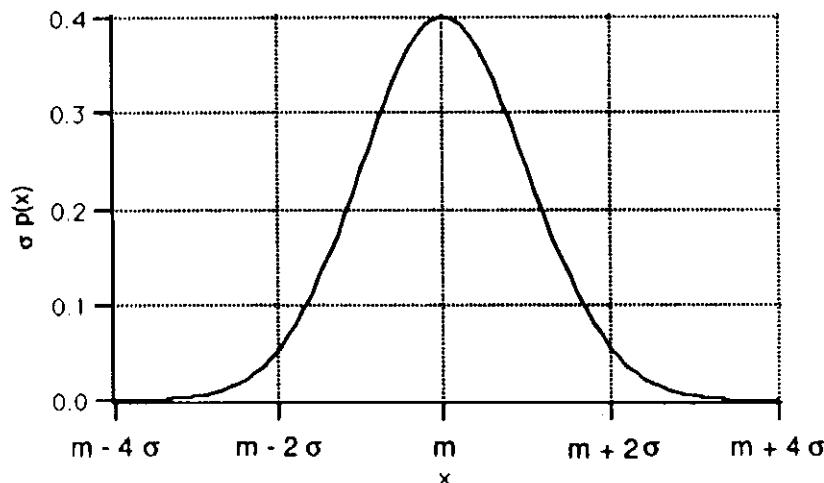


Fig. 4.2 Gaussian or normal probability density function.

probability it is within two standard deviations of the mean, and 99% probability that it is within three standard deviations of the mean.

It can be shown that the resultant probability density function of a sum of Gaussian distributed random variables is also Gaussian. In addition, under certain circumstances, it can also be shown that the sum of independent random variables, regardless of individual density function, tends toward Gaussian as the number of random variables gets larger (an illustration of this phenomenon will be illustrated in the next section). That is in fact why so many random variables are Gaussian distributed.

Gaussian Noise Example

To simulate noise or random events we have to know how to generate, via the computer, pseudorandom numbers with the appropriate probability density function. The FORTRAN language does not come with a random number generator. However, many microcomputer implementations of FORTRAN provide extensions from which noise, with the desired probability density function, can be constructed. For example, in Macintosh Absoft FORTRAN, the statement Random() produces a uniformly distributed integer between ± 32768 . It can be shown from the central limit theorem that the addition of many uniformly distributed variables produces a Gaussian distributed variable.

The first step in constructing random numbers with the desired probability density function is to normalize the uniform noise generator so that random numbers between -0.5 and 0.5 are produced. The second step is to add 12 uniformly distributed random variables in order to obtain a zero-mean Gaussian variable with unity standard deviation (since the variance of one uniformly distributed random variable is $1/12$, the variance of 12 must be 1). The FORTRAN listing for the generation of 100 Gaussian-distributed random numbers with zero mean and unity variance is shown in Listing 4.1.

Figure 4.3 displays the values of each of the 100 random numbers, generated via the program of Listing 4.1, in graphic form. A quick glance at the plot indicates that the data appear to have approximately zero mean. The standard deviation of the data can be "eyeballed" by looking at the maximum excursions (99% chance

Listing 4.1 Gaussian random number generator

```

GLOBAL DEFINE
INCLUDE 'quickdraw.inc'
END
INTEGER SUM
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
N=100
DO 10 I=1,N
SUM=0
DO 14 J=1,12
IRAN=Random()
SUM=SUM+IRAN
14 CONTINUE
X=SUM/65536.
WRITE(*,97)I,X
WRITE(1,97)I,X
10 CONTINUE
40 CONTINUE
97 FORMAT(I3,2X,F10.3)
PAUSE
END

```

that data is within the 3σ values) and using the simplified relationship

$$\sigma_{\text{approx}} = (\text{peak to peak})/6 = 4/6 = 0.67$$

Thus, the eyeballed value of σ does not quite meet the theoretical expectations of unity standard deviation.

To get an idea of the resultant probability density function of the computer-generated 100 random numbers, another FORTRAN program was written and appears in Listing 4.2. Essentially each random number is placed in a bin in order to calculate the frequency of occurrence and hence the probability density

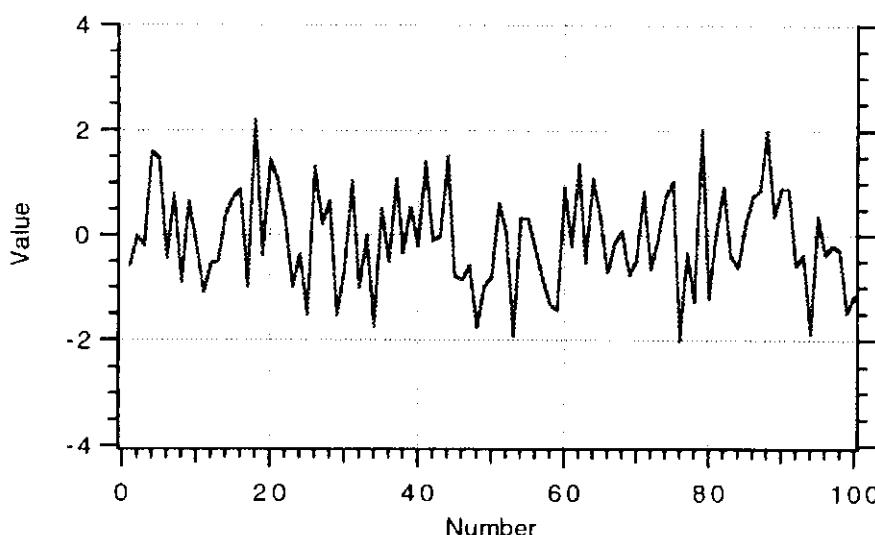


Fig. 4.3 One hundred random numbers with Gaussian distribution.

Listing 4.2 FORTRAN program used to generate probability density function

```

GLOBAL DEFINE
    INCLUDE 'quickdraw.inc'
END
INTEGER BIN,SUM
DIMENSION H(2000),X(2000)
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
XMAX=6.
XMIN=-6.
RANGE=XMAX-XMIN
TMP=1./SQRT(6.28)
N=100
BIN=50
DO 10 I=1,N
SUM=0
DO 14 J=1,12
IRAN=Random()
SUM=SUM+IRAN
14 CONTINUE
X(I)=SUM/65536.
10 CONTINUE
DO 20 I=1,BIN
H(I)=0
20 CONTINUE
DO 30 I=1,N
K=INT(((X(I)-XMIN)/RANGE)*BIN)+.99
IF(K<1)K=1
IF(K>BIN)K=BIN
H(K)=H(K)+1
30 CONTINUE
DO 40 K=1,BIN
PDF=(H(K)/N)*BIN/RANGE
AB=XMIN+K*RANGE/BIN
TH=TMP*EXP(-AB*AB/2.)
WRITE(*,97)AB,PDF,TH
WRITE(1,97)AB,PDF,TH
40 CONTINUE
97 FORMAT(3F10.3)
PAUSE
CLOSE(1)
END

```

function.² Also included in the listing, for comparative purposes, is the theoretical formula for the probability density function of a zero-mean, unity variance, Gaussian distribution.

Figure 4.4 presents the calculated probability density function in graphic form. Superimposed on the figure is a plot of the theoretical Gaussian distribution. The figure indicates that, with a sample size of only 100 random numbers, it is not immediately obvious that the computer-generated probability density function follows a Gaussian distribution.

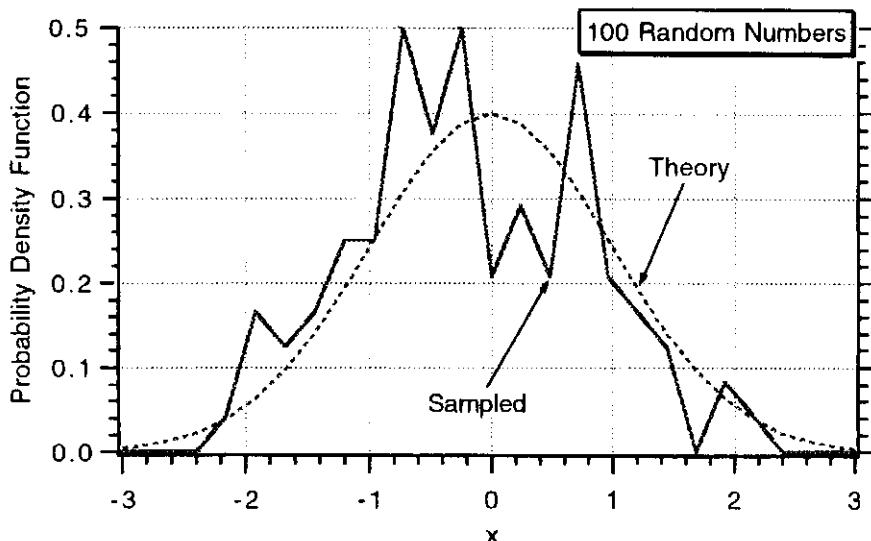


Fig. 4.4 Sampled Gaussian distribution does not closely follow theory for 100 random numbers.

Increasing the sample size from 100 random numbers to 1000 random numbers will clarify the “goodness” of the computer-generated random numbers. Figure 4.5 displays each of the 1000 random numbers generated. The figure demonstrates that, although the mean of the random numbers is still about zero, we now have larger excursions (numbers vary between $\pm 3\sigma$) and the approximate value of the standard deviation is

$$\sigma_{\text{approx}} = (\text{peak to peak})/6 = 6/6 = 1$$

which is the theoretically correct value.

Finally, Fig. 4.6 now shows that when the sample size is increased to 1000 numbers the resultant probability density function closely follows the theoretical bell-shaped curve. Thus, we have seen how a Gaussian distribution can be constructed from the summation of 12 uniformly distributed random variables. This is a practical application of the central limit theorem. In practice, to save on computer running time, we can add fewer than 12 uniformly distributed numbers to get a

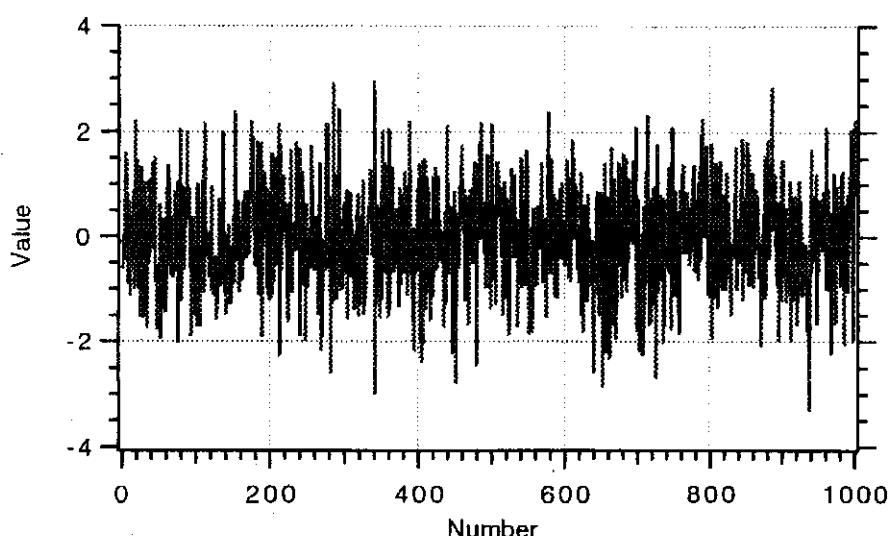


Fig. 4.5 One thousand random numbers with Gaussian distribution.

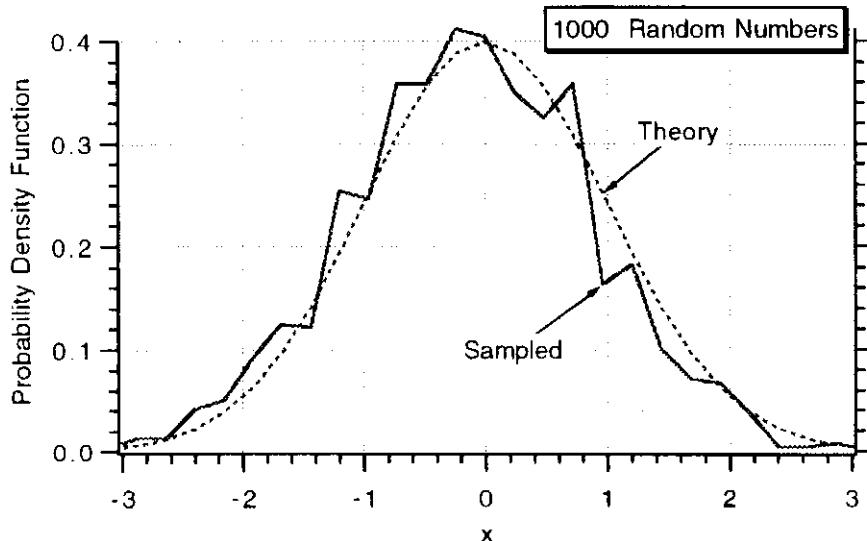


Fig. 4.6 Sampled Gaussian distribution more closely follows theory for 1000 random numbers.

Gaussian-distributed random number. In the rest of the text we will only add six uniformly distributed numbers to get the desired Gaussian distribution.

Computational Issues

Often, from simulation outputs, we wish to compute some of the basic random variable properties (i.e., mean, variance, etc.). Stated more mathematically, we wish to compute these basic random variable properties from a finite set of data x_i when only n samples are available. The discrete equivalent of the previously presented formulas for basic random variable properties are presented in the following equations:

$$\text{mean} = \sum_{i=1}^n x_i / n$$

$$\text{mean square} = \sum_{i=1}^n x_i^2 / (n - 1)$$

$$\text{standard deviation} = \left\{ \left[\sum_{i=1}^n (x_i - \text{mean})^2 \right] / (n - 1) \right\}^{1/2}$$

We can see from these equations that integrals from the theoretical or continuous formulas have been replaced with summations in their discrete equivalents. In order for the theoretical and calculated random variable properties to be equal, the number of samples in the discrete computations must be infinite. Since the sample size is finite, the discrete or calculated formulas are approximations. In fact, the answers generated from these formulas have statistics of their own.

Recognizing that simulation outputs based upon random inputs can vary, the Monte Carlo approach³ will often be used in this text to obtain system performance. The Monte Carlo method is approximate and is simply repeated simulation trials plus postprocessing of the resultant data in order to do ensemble averaging (using

Listing 4.3 Program for computing sampled standard deviation

```

GLOBAL DEFINE
INCLUDE 'quickdraw.inc'
END
INTEGER SUM
DIMENSION Z(100)
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
Z1=0.
DO 10 I=1,100
SUM=0
DO 14 J=1,12
IRAN=Random()
SUM=SUM+IRAN
14 CONTINUE
X=SUM/65536.
Z(I)=X
Z1=Z(I)+Z1
XMEAN=Z1/I
10 CONTINUE
SIGMA=0.
Z1=0.
DO 20 I=1,100
Z1=(Z(I)-XMEAN)**2+Z1
IF(I.EQ.1)THEN
SIGMA=0.
ELSE
SIGMA=SQRT(Z1/(I-1))
ENDIF
WRITE(*,97)I,SIGMA
WRITE(1,97)I,SIGMA
20 CONTINUE
97 FORMAT(I4,2X,F10.3)
PAUSE
CLOSE(1)
END

```

the preceding formulas) to get the mean and standard deviation. Usually a large number of simulation trials are required in order to provide confidence in the accuracy of the results. However, because of its simplicity and generality, the Monte Carlo approach is probably the most popular computerized method of statistical analysis.

To demonstrate that our computed statistics are not precise and in fact are random variables with statistics, a FORTRAN simulation of the Gaussian noise was generated, and Listing 4.3 shows the computation of the sampled standard deviation. The number of i samples used in the program computation was made a parameter in the study and varied from 1 to 100.

Figure 4.7 shows that the computed standard deviation (actual standard deviation is unity) obtained from the FORTRAN program is a function of the sample size used. Large errors in the standard deviation estimate occur when there are less than

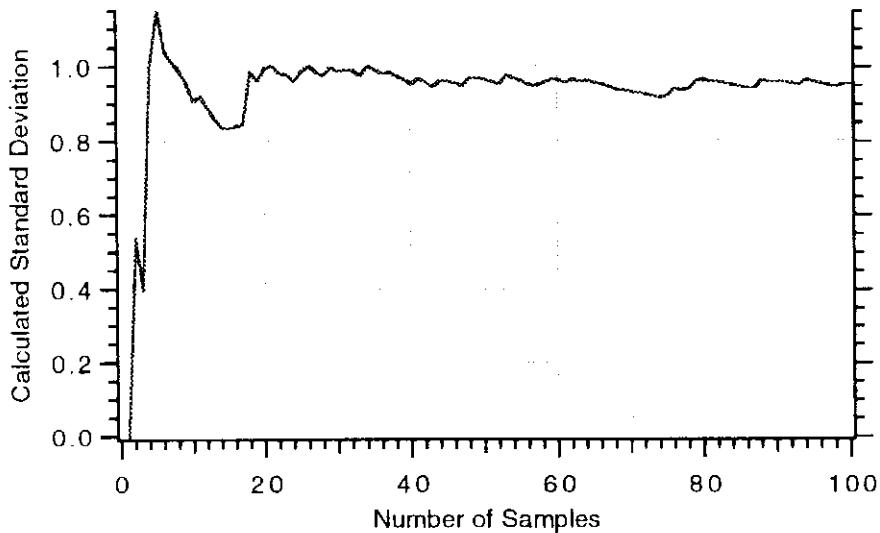


Fig. 4.7 Large errors occur when only a few samples are taken.

20 samples. The accuracy of the computation improves significantly when many samples are used in computing the standard deviation. In this example, we need more than 100 samples for the computed standard deviation to be within 5% of the theoretical value of unity. When we begin to evaluate system performance, when the inputs are random, we will take this information into account in determining how many simulation trials (Monte Carlo runs) will be required to get reasonably accurate results. Usually we will consider 50 runs to be sufficient in the tradeoff between computer running time and numerical accuracy.

More Basic Definitions¹

A few more definitions are required before we can build up the tools required for the analysis of noise-driven systems. Thus far we have discussed the second-order statistics of random processes. However, in practice, we are limited to even less information than that given by the probability density function. Often, only the first moment of these random processes is measured. One such moment is the autocorrelation function, which is defined by

$$\phi_{xx}(t_1, t_2) = E[x(t_1)x(t_2)]$$

The Fourier transform of the autocorrelation function is called the power spectral density and is defined as

$$\Phi_{xx} = \int_{-\infty}^{\infty} \phi_{xx}(\tau) e^{-j\omega\tau} d\tau$$

where the power spectral density, using these definitions, has dimensions of unit squared per Hertz. In all of the statistical work presented throughout this text, the power spectral density will have those units.

One simple and useful form for the power spectral density is that of white noise, in which the power spectral density is constant, or

$$\Phi_{xx} = \Phi_0 \quad (\text{white noise})$$

The autocorrelation function for white noise is a delta function given by

$$\phi_{xx} = \Phi_0 \delta(\tau) \quad (\text{white noise})$$

Although white noise is not physically realizable, it can be used to serve as an invaluable approximation to situations in which a disturbing noise is wideband compared to the system bandwidth. In addition, white noise is useful for analytical operations because of the impulsive nature of the autocorrelation function (it makes integrals disappear).

Response of Linear System to White Noise

Often we are interested in finding the response of a linear system to noise. If the system is linear, with impulse response $h(t, \tau)$, the output $y(t)$, can be expressed in terms of the input $x(t)$ via the convolution integral, or

$$y(t) = \int_{-\infty}^t x(\tau)h(t, \tau) d\tau$$

Squaring both sides of the preceding equation yields

$$y^2(t) = \int_{-\infty}^t x(\tau_1)h(t, \tau_1) d\tau_1 \int_{-\infty}^t x(\tau_2)h(t, \tau_2) d\tau_2$$

If $x(t)$ is random, we can take expectations of both sides, or

$$E[y^2(t)] = \int_{-\infty}^t \int_{-\infty}^t h(t, \tau_1)h(t, \tau_2) E[x(\tau_1)x(\tau_2)] d\tau_1 d\tau_2$$

In addition, if the input $x(t)$ is white noise with power spectral density Φ , the double integral of the preceding equation can be simplified because of the impulsive nature of the autocorrelation function, or

$$E[x(\tau_1)x(\tau_2)] = \Phi\delta(\tau_1 - \tau_2)$$

Substitution yields

$$E[y^2(t)] = \Phi \int_{-\infty}^t h^2(t, \tau) d\tau$$

Therefore, the mean square response of a linear system driven by white noise with power spectral density Φ (where Φ has the dimensions of unit²/Hz) is proportional to the integral of the square of the impulse response. The preceding equation is a general relationship and is valid for both time-varying and time-invariant linear systems driven by white noise.

Low-Pass-Filter Example

To illustrate the utility of the mean square response equation, let us find the response of a low-pass filter to white noise as shown in Fig. 4.8. Here the input x is

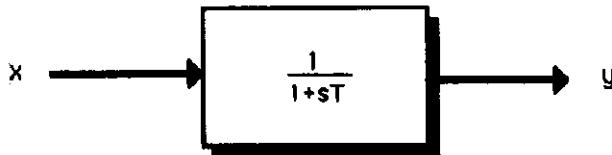


Fig. 4.8 Low-pass filter with white noise input.

white noise with power spectral density Φ . Since this system is time-invariant and physically realizable (noncausal), the fundamental noise relationship simplifies to

$$E[y^2(t)] = \Phi \int_0^t h^2(\tau) d\tau$$

We can find the system impulse response in the preceding integral by first recognizing that the transfer function of the low-pass filter, shown in Fig. 4.8, is given by

$$H(s) = \frac{1}{1 + sT}$$

Therefore, its inverse Laplace transform is

$$h(t) = \mathcal{L}^{-1}[H(s)] = \frac{e^{-t/T}}{T}$$

Substitution yields

$$E[y^2(t)] = \frac{\Phi}{T^2} \int_0^t e^{-2\tau/T} d\tau$$

Evaluation of the upper and lower limits results in the final answer:

$$E[y^2(t)] = \frac{\Phi(1 - e^{-2t/T})}{2T}$$

In the steady state, the exponential term drops out, yielding

$$E[y^2(\infty)] = \frac{\Phi}{2T}$$

We can write a FORTRAN program to simulate the problem and to see how the theoretical results and simulation results agree. To do this we must be able to simulate Gaussian white noise. We already know how to simulate Gaussian random numbers. Since the Gaussian distributed random numbers are independent, the resultant Gaussian random numbers will look white to the low-pass filter if its bandwidth is much greater than the filter bandwidth. In a simulation of the continuous system of Fig. 4.8, the equivalent Gaussian noise generator is called every integration interval h . Since the integration interval is always chosen to be at least several times smaller than the smallest system time constant T ($h \ll T$ in order to get correct answers with numerical integration techniques), the noise will look white to the system.

The standard deviation of the pseudowhite noise (actual white noise has infinite standard deviation) is related to the desired white noise spectral density Φ and

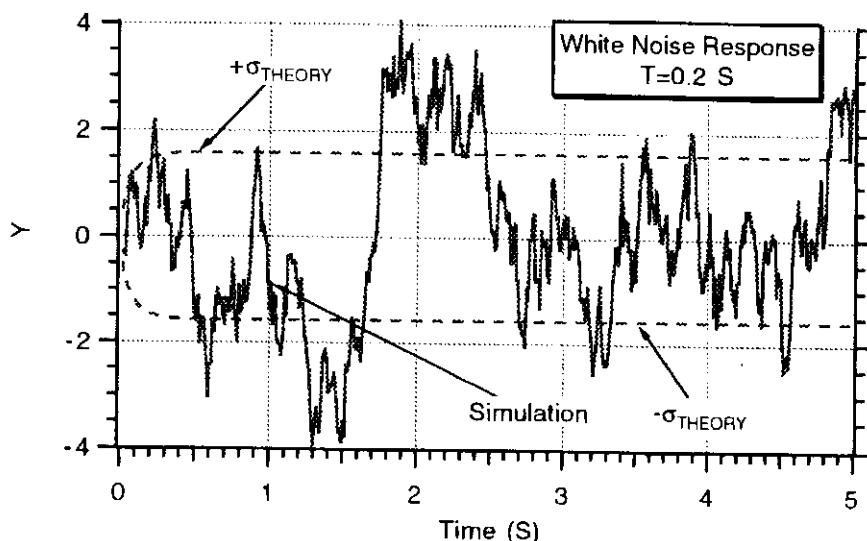


Fig. 4.9 Low-pass filter output agrees with theory.

integration interval h according to³

$$\sigma = \sqrt{\frac{\Phi}{h}}$$

where Φ has dimensions of units squared per Hertz. The FORTRAN simulation listing of this white-noise-driven low-pass filter is shown in Listing 4.4. It is important to note that this listing will run as is on any Macintosh microcomputer using the Abssoft FORTRAN compiler. Other computers and FORTRAN compilers will use slightly different methods for generating uniformly distributed random variables. We can see from the listing that the Gaussian noise with unity standard deviation is modified to get the desired pseudowhite noise spectral density ($\Phi = 1$). The approximate white noise enters the system every integration interval. A sample output for a correlation time of 0.2 s is shown in Fig. 4.9. Also shown in the listing and figure are the theoretical results obtained from the previously derived formula for the output standard deviation of a white-noise-driven low-pass filter, which is

$$\sigma_{\text{theory}} = \pm \sqrt{\frac{\Phi[1 - e^{-2t/T}]}{2T}}$$

We can see from this figure that the simulation results, based upon the FORTRAN listing, agree with theory in the sense that the simulation results lie within the $\pm\sigma$ bounds approximately 68% of the time. Therefore, we can say that the experimental and theoretical results are in agreement.

Increasing the correlation time constant increases the smoothing action of the low-pass filter. Figure 4.10 shows the filter output when the correlation time constant is increased from 0.2 s to 1 s. Here we can see that the larger filter time constant not only provides more filtering action but also tends to correlate the noise. In other words, the randomness of the noise is starting to disappear as the filter time constant increases. Again, this figure shows that the simulated results appear to be within the $\pm\sigma$ bounds about 68% of the time.

It is important to note that both the simulated time domain results and the theoretical second-order statistical results provided invaluable visual information. The use of both theory and simulation can be used to not only verify results but also to provide a deeper understanding of the processes involved.

Listing 4.4 Simulation of low-pass filter driven by white noise

```

GLOBAL DEFINE
INCLUDE 'quickdraw.inc'
END
INTEGER STEP
DATA TAU,PHI/.2,1./
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
T=0.
H=.01
SIG=SQRT(PHI/H)
Y=0.
10 IF(T>4.999)GO TO 999
CALL GAUSS(X,SIG)
YOLD=Y
STEP=1
GOTO 200
66 STEP=2
Y=Y+H*YD
T=T+H
GOTO 200
55 CONTINUE
Y=(YOLD+Y)/2.+.5*H*YD
SIGPLUS=SQRT(PHI*(1.-EXP(-2.*T/TAU))/(2.*TAU))
SIGMINUS=-SIGPLUS
WRITE(*,97)T,Y,SIGPLUS,SIGMINUS
WRITE(1,97)T,Y,SIGPLUS,SIGMINUS
GOTO 10
97 FORMAT(4F10.3)
200 CONTINUE
YD=(X-Y)/TAU
IF(STEP-1)66,66,55
999 CONTINUE
PAUSE
CLOSE(1)
END

SUBROUTINE GAUSS(X,SIG)
INTEGER SUM
SUM=0
DO 14 J=1,12
IRAN=Random()
SUM=SUM+IRAN
14 CONTINUE
X=SUM/65536.
X=X*SIG
RETURN
END

```

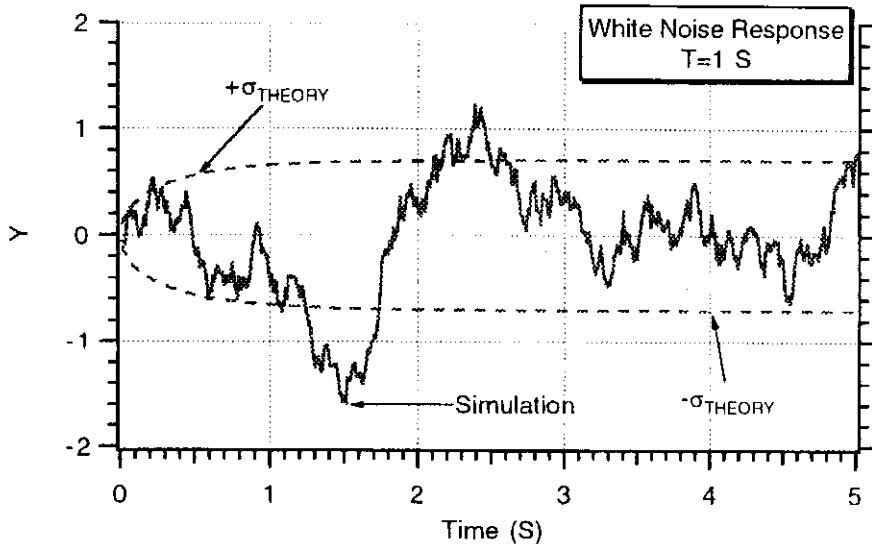


Fig. 4.10 Increasing filter time constant reduces the noise transmission.

Adjoints for Noise-Driven Systems^{4,5}

In the previous chapter we saw that the method of adjoints could be very useful in analyzing linear time-varying deterministic systems. We shall now demonstrate that adjoints can also be of great utility in analyzing linear time-varying systems driven by white noise. It was shown before that the mean square response of a linear time-varying system driven by white noise is given by

$$E[y^2(t)] = \Phi \int_0^t h^2(t, \tau) d\tau$$

where Φ is the white noise power spectral density and $h(t, \tau)$ the impulse response of the linear time-varying system. In this case τ is the impulse application time and t the observation time. The previous section presented a simple example demonstrating the practical utility of this equation. However, for time-varying systems this equation is not as useful because the integration is with respect to the impulse application time τ . As with the deterministic case, this means that many computer runs would have to be made, each having a different impulse application time, in order to evaluate the preceding equation.

If we go back to the fundamental relationship between the original and adjoint systems, we can say that

$$E[y^2(t)] = \Phi \int_0^t h^*[(t_F - \tau, t_F - t)]^2 d\tau$$

After making the substitution

$$x = t_F - \tau$$

$$dx = -d\tau$$

we obtain

$$E[y^2(t)] = \Phi \int_{t_F-t}^{t_F} [h^*(x, t_F - t)]^2 dx$$

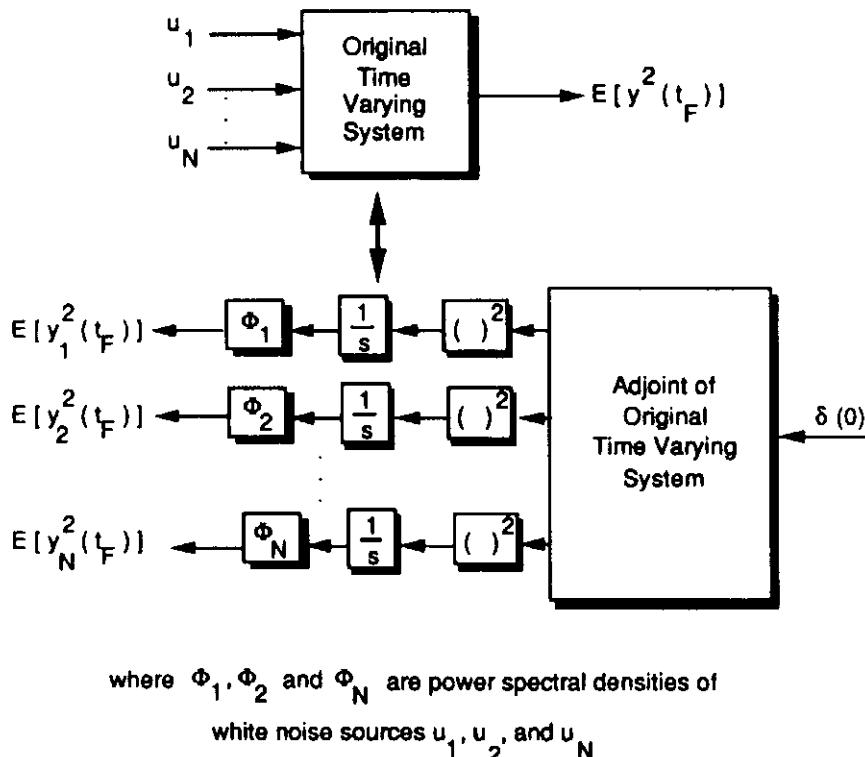


Fig. 4.11 Equivalence between adjoint and original systems for stochastic inputs.

If the final time is of interest ($t = t_F$), the preceding equation can be rewritten as

$$E[y^2(t_F)] = \Phi \int_0^{t_F} [h^*(x, 0)]^2 dx$$

Since the integration is now with respect to the observation time, this new equation is quite useful. Therefore, we can find the mean square response of a linear, time-varying system driven by white noise by squaring and integrating the output of the impulsively driven adjoint system *in one computer run!*

The benefits of the adjoint approach become even more dramatic when we consider many white noise inputs to the original system as shown in Fig. 4.11.

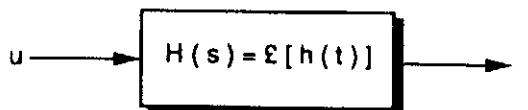
As with the deterministic inputs, white noise inputs to the original system become outputs in the adjoint system. Therefore, by superposition, one adjoint run yields an exact statistical analysis of the noise-driven system plus a statistical error budget showing how each white noise error source contributed to the total performance projection. In Fig. 4.11 the total mean square response is computed from

$$E[y^2(t)] = E[y_1^2(t)] + E[y_2^2(t)] + \dots + E[y_N^2(t)]$$

Shaping Filters and Random Processes^{6,7}

Thus far we have seen the importance of target maneuver on system performance. In this section we will show how shaping filters can be used to accurately represent aircraft evasive maneuvers. The purpose of the shaping filter approach is to allow us to use efficient and effective means of performance analysis such as the method of adjoints.

The concept of "shaping filter" has been used for many years in the analysis of physical systems because it allows a system with a random input to be replaced



where u is white noise with power spectral density Φ_u
and

$$\Phi_u = p_T(t)$$

Fig. 4.12 Shaping network representation of deterministic signal with random starting time.

by an augmented system (the original system plus the shaping filter) excited only by white noise. An example of this is the replacement of the random telegraph signal by white noise through a simple lag network. This approach is generally applied to problems where mean square values of outputs are of prime importance. In such cases only second-order statistics are important, and rather complex input processes can sometimes be represented by very simple shaping filters. This is due to the fact that random processes that have the same mean and autocorrelation are mathematically equivalent. This is true even though the associated probability density functions of the random processes may be quite different. In other words, a random phenomenon and its shaping filter equivalent are indistinguishable as far as their second-order statistics are concerned. The concept of shaping filter can also be applied to the statistical representation of signals with known form but random starting time. Consider a signal of known form $h(t)$ with random starting time so that the resultant signal $x(t)$ is given by

$$x(t) = h(t - T) u(t - T)$$

where the probability density function of T is given by $p_T(t)$, and $u(t)$ is the unit step function. Note that, although $h(t)$ is deterministic, $x(t)$ is random because of the random starting time. It can be shown that the white-noise-driven shaping network of Fig. 4.12 has the same mean and autocorrelation functions as those of the preceding equation. Here we can see that the white noise has a power spectral density equal to the probability density function of the random starting time and that the inverse Laplace transform of the shaping filter is equal to the deterministic signal.

The output of the shaping network and the actual random process are equivalent in terms of second-order statistics. If either process is passed through a linear physical system, the outputs would be indistinguishable if second-order statistics are being observed (i.e., mean square values).

Consider a step target maneuver which has a starting time that is uniformly distributed over the flight time. Mathematically speaking, the maneuver can be modeled as a constant signal of magnitude n_T , which starts at time T , or

$$x(t) = n_T u(t - T)$$

where $u(t - T)$ is a unit step function defined by

$$\begin{aligned} u(t - T) &= 0 && \text{for } t < T \\ &= 1 && \text{otherwise} \end{aligned}$$

Let us assume that the initiation of the maneuver is equally likely to occur anywhere during the flight. More precisely, we can say that the starting time T is

uniformly distributed over the flight time t_F , with probability density function

$$\begin{aligned} p_T(t) &= 1/t_F && \text{for } 0 \leq t \leq t_F \\ &= 0 && \text{otherwise} \end{aligned}$$

Therefore, the autocorrelation function of this signal with random starting time is given by

$$\phi_{xx}(t_1, t_2) = \int_{-\infty}^{\infty} x(t_1)x(t_2)p_T(T) dT$$

or

$$\phi_{xx}(t_1, t_2) = \int_0^{t_F} n_T u(t_1 - T)n_T u(t_2 - T) \frac{dT}{t_F}$$

Assuming that

$$0 < t_1 < t_2 < t_F$$

the autocorrelation function simplifies to

$$\phi_{xx}(t_1, t_2) = \frac{n_T^2}{t_F} \int_0^{t_1} dT$$

It is important to note that the output autocorrelation function of a linear time-invariant system with impulse response $h(t)$ driven by white noise can be expressed as

$$\phi_{yy}(t_1, t_2) = \int_{-\infty}^{t_1} h(t_1 - \tau_1) \int_{-\infty}^{t_2} h(t_2 - \tau_2) \phi_{uu}(\tau_1, \tau_2) d\tau_1 d\tau_2$$

The autocorrelation function of the white noise input is

$$\phi_{uu}(\tau_1, \tau_2) = \Phi_u(\tau_1)\delta(\tau_1 - \tau_2)$$

where the spectral density of the white noise Φ_u is a function of time. Substitution of the white noise autocorrelation function into the preceding integral equation eliminates one of the integrals. After some manipulations and assuming $t_1 < t_2$, we obtain

$$\phi_{yy}(t_1, t_2) = \int_{-\infty}^{t_1} \Phi_u(\tau_1)h(t_1 - \tau_1)h(t_2 - \tau_1) d\tau_1$$

If the spectral density takes on values of

$$\begin{aligned} \phi_u(t) &= \phi_u, && 0 \leq t \leq t_F \\ &= 0 && \text{otherwise} \end{aligned}$$

and we assume that

$$0 < t_1 < t_2 < t_F$$

then we can say that

$$\phi_{yy}(t_1, t_2) = \Phi_u \int_0^{t_1} h(t_1 - \tau_1)h(t_2 - \tau_1) d\tau_1$$

Therefore, we can say that the two different expressions for the autocorrelation

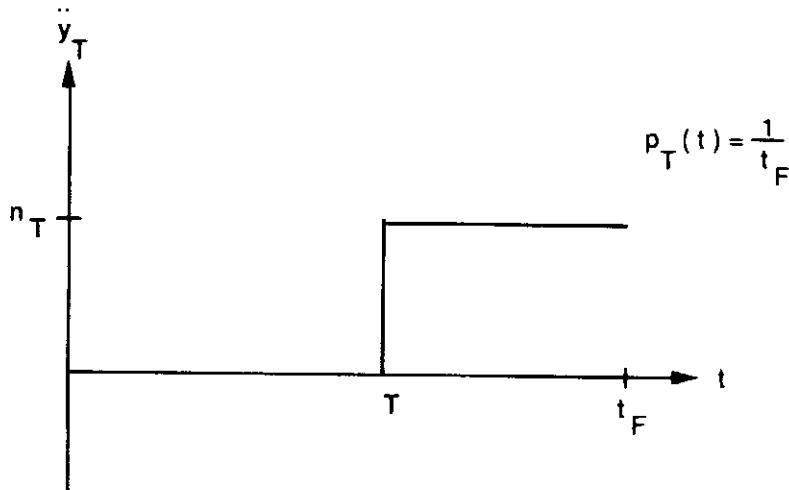


Fig. 4.13 Step maneuver with uniformly distributed starting time.

function are equivalent if

$$\Phi_u = n_T^2 / t_F$$

and

$$h(t) = 1$$

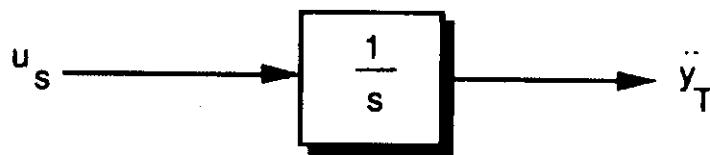
In summary, a step maneuver of amplitude n_T , whose starting time is uniformly distributed over the flight time t_F has the same autocorrelation function as a linear network with transfer function

$$H(s) = \frac{1}{s}$$

driven by white noise with power spectral density

$$\begin{aligned} \Phi_u &= n_T^2 / t_F & 0 < t < t_F \\ &= 0 & \text{otherwise} \end{aligned}$$

The uniformly distributed step maneuver is shown in Fig. 4.13 and its shaping filter equivalent is shown in Fig. 4.14.



u_s is White Noise With Power Spectral Density

$$\Phi_s = \frac{n_T^2}{t_F}$$

Fig. 4.14 Shaping filter equivalent of random starting time step maneuver.

Example of a Stochastic Adjoint

To show how the adjoint can also be used to analyze linear systems driven by stochastic or random disturbances, let us revisit the single-lag proportional navigation homing loop. However, this time we will consider a target maneuver with a random starting time (starting time that is uniformly distributed over the flight time) as the error source. A single-lag proportional navigation homing loop with the stochastic input is shown in Fig. 4.15.

In Fig. 4.15 the target maneuver is a constant from flight to flight (either plus or minus n_T). However, on a given flight its initiation time is equally likely to occur anywhere during the flight (uniformly distributed over the flight time). A FORTRAN Monte Carlo simulation of Fig. 4.15 with the random target maneuver appears in Listing 4.5. We can see from the listing that there are two main loops. The outer loop varies the flight time from 1 to 10 s in increments of 1 s. The inner loop performs 50 sets of runs on a particular case. In each of these cases the starting time of the maneuver is chosen from a uniform distribution. After each Monte Carlo set, the standard deviation and mean of the 50 miss distances are computed according to the formulas developed in this chapter.

A case was run for the single-time-constant guidance system in which the time constant was set to 1 s and the effective navigation ratio was set to 3. Fifty-run Monte Carlo sets for 10 different flight times were run for this single case, which actually encompassed a total of 500 runs (50×10). The standard deviation of the miss for each flight time was calculated and is displayed as a function of flight time in Fig. 4.16. We can see that the miss distance is small for both small and large flight times. The miss is small for short flight times because the miss distance does not have enough time to develop. At the larger flight times there is a good chance that the target maneuver initiation time will be at a long time to go, relative to the guidance system time constant, and will therefore induce a smaller miss distance.

We have seen in the previous section that the shaping filter equivalent for a uniformly distributed target maneuver is white noise, with spectral density Φ_s , through an integrator. The spectral density of the white noise is related to the maneuver level and the flight time according to

$$\Phi_s = \frac{n_T^2}{t_F}$$

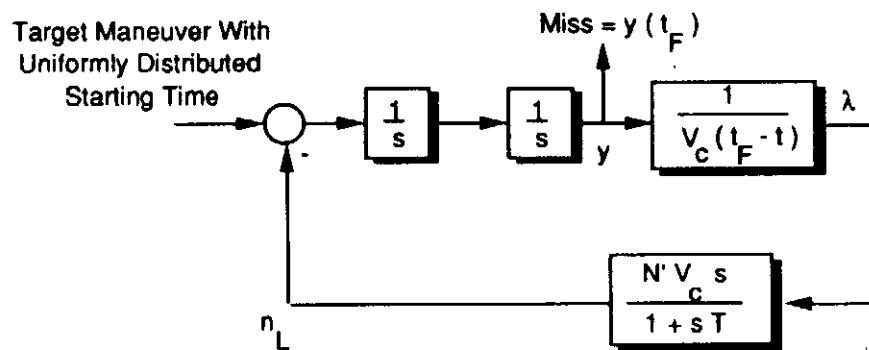


Fig. 4.15 Single-lag homing loop with stochastic inputs.

Listing 4.5 Simulation of homing loop with random target maneuver

```

GLOBAL DEFINE
    INCLUDE 'quickdraw.inc'
END
DIMENSION Z(1000)
INTEGER RUN
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
VC=4000.
XNT=96.6
VM=3000.
XNP=3.
TAU=1.
RUN=50
106 CONTINUE
DO 60 TF=1,10
Z1=0.
DO 20 I=1,RUN
CALL UNIFORM(SUM)
TSTART=TF*SUM
CALL UNIFORM(PZ)
PZ=PZ-.5
IF(PZ>0.)THEN
    COEF=1.
ELSE
    COEF=-1.
ENDIF
Y=0.
YD=0.
T=0.
H=.01
S=0.
XNC=0.
XNL=0.
10 IF(T>(TF-.0001))GOTO 999
IF(T<TSTART)THEN
    XNT=0.
ELSE
    XNT=COEF*96.6
ENDIF
YOLD=Y
YDOLD=YD
XNOLD=XNL
STEP=1
GOTO 200
66 STEP=2
Y=Y+H*YD
YD=YD+H*YDD
XNL=XNL+H*XNLD
T=T+H

```

(Contd.)

Listing 4.5 (Continued)

```

      GOTO 200
55   CONTINUE
      Y=.5*(YOLD+Y+H*YD)
      YD=.5*(YDOLD+YD+H*YDD)
      XNL=.5*(XNLOLD+XNL+H*XNLD)
      S=S+H
      GOTO 10
200  CONTINUE
      TGO=TF-T+.00001
      RTM=VC*TGO
      XLAMD=(RTM*YD+Y*VC)/(RTM**2)
      XNC=XNP*VC*XLAMD
      XNLD=(XNC-XNL)/TAU
      YDD=XNT-XNL
      IF(STEP-1)66,66,55
999   CONTINUE
      Z(I)=Y
      Z1=Z(I)+Z1
      XMEAN=Z1/I
20    CONTINUE
      SIGMA=0.
      Z1=0.
      DO 50 I=1,RUN
      Z1=(Z(I)-XMEAN)**2+Z1
      IF(I.EQ.1)THEN
          SIGMA=0.
      ELSE
          SIGMA=SQRT(Z1/(I-1))
      ENDIF
50   CONTINUE
      WRITE(*,97)TF,SIGMA,XMEAN
      WRITE(1,97)TF,SIGMA,XMEAN
60   CONTINUE
97    FORMAT(3F10.3)
      PAUSE
      CLOSE(1)
      END

      SUBROUTINE UNIFORM(SUM)
      IRAN=Random()
      SUM=IRAN/65536.+5
      RETURN
      END

```

An adjoint model can be constructed from the original system by following the rules for constructing stochastic adjoints.^{4,5} The only additional rule for stochastic systems is that all stochastic inputs to the original system must be modeled as white noise inputs, which then become outputs in the adjoint system. Since the input to the original system can be modeled as white noise through an integrator,

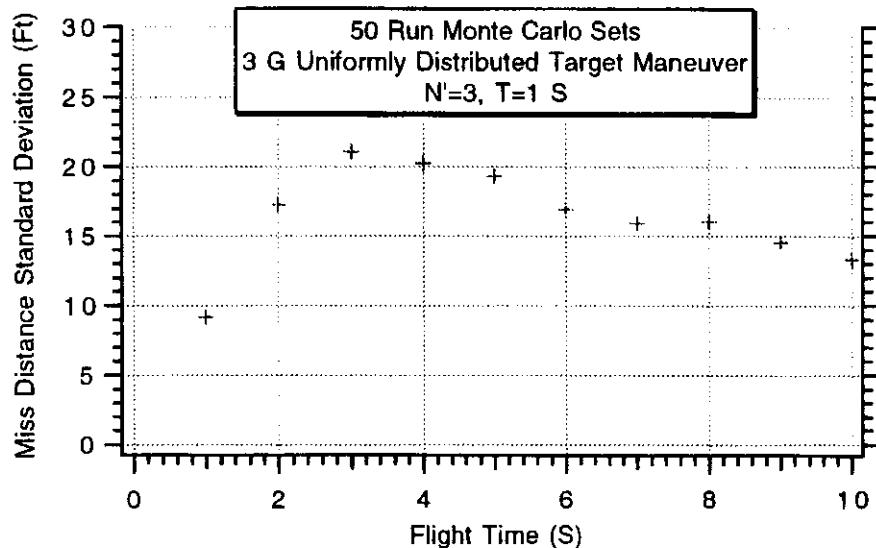


Fig. 4.16 Monte Carlo results for uniformly distributed target maneuver.

the adjoint model will reverse the signal flow and square and integrate the output. The resultant adjoint model is shown in Fig. 4.17. The impulsive input can be replaced by an initial condition of unity on integrator x_3 .

An adjoint FORTRAN simulation of the model in Fig. 4.17 appears in Listing 4.6. The adjoint program only has to run once to find the standard deviation of the miss as a function of the flight time.

The adjoint simulation was run using the input parameters shown in Listing 4.6. The adjoint results for this example are shown in Fig. 4.18. Superimposed on the plot are the Monte Carlo results previously generated. We can see that the adjoint and Monte Carlo results are in close proximity, thus experimentally confirming the shaping filter approach and demonstrating the utility of stochastic adjoints. One

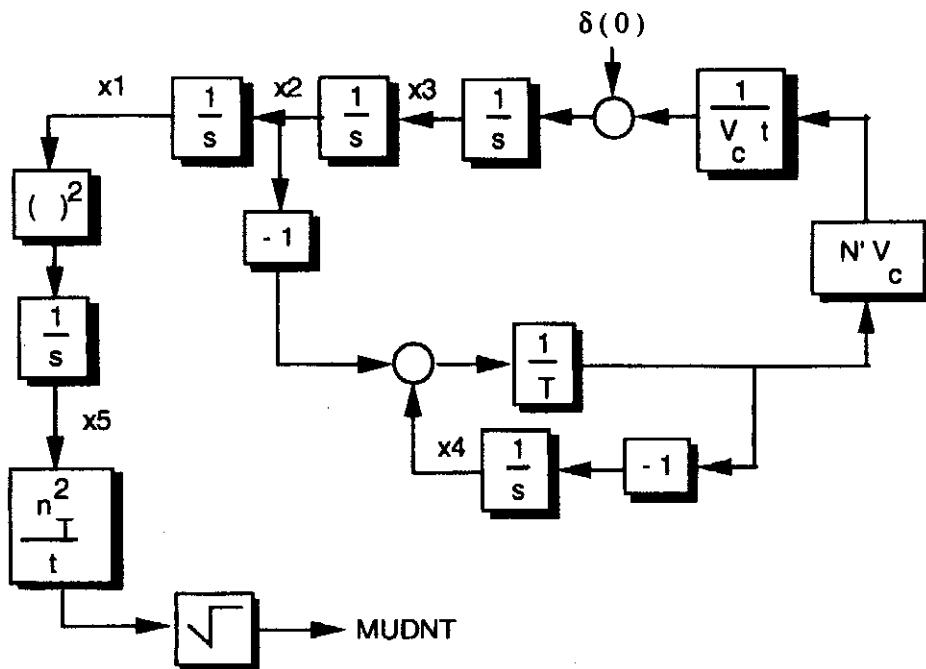


Fig. 4.17 Adjoint model for stochastic example.

**Listing 4.6 Adjoint model using shaping
filter approach**

```

INTEGER STEP
DATA XNT,XNP,TAU,TF/ 96.6,3,1,10/
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
T=0.
S=0.
TP=T+.00001
X1=0
X2=0
X3=1
X4=0
X5=0.
H=.01
10 IF(TP>(TF-.00001))GOTO 999
S=S+H
X1OLD=X1
X2OLD=X2
X3OLD=X3
X4OLD=X4
X5OLD=X5
STEP=1
GOTO 200
66 STEP=2
X1=X1+H*X1D
X2=X2+H*X2D
X3=X3+H*X3D
X4=X4+H*X4D
X5=X5+H*X5D
TP=TP+H
GOTO 200
55 CONTINUE
X1=(X1OLD+X1)/2+.5*H*X1D
X2=(X2OLD+X2)/2+.5*H*X2D
X3=(X3OLD+X3)/2+.5*H*X3D
X4=(X4OLD+X4)/2+.5*H*X4D
X5=(X5OLD+X5)/2+.5*H*X5D
IF(S<.099999)GOTO 10
S=0.
XMUDNT=XNT*SQRT(X5/TGO)
WRITE(*,97)TP,XMUDNT
WRITE(1,97)TP,XMUDNT
GOTO 10
97 FORMAT(2F10.3)
200 CONTINUE
X1D=X2
X2D=X3
Y1=(X4-X2)/TAU
TGO=TP+.00001
X3D=XNP*Y1/TGO

```

(Contd.)

Listing 4.6 (Continued)

```

X4D=-Y1
X5D=X1*X1
IF(STEP=1)66,66,55
999  CONTINUE
PAUSE
CLOSE(1)
END

```

single adjoint run gave results that were the equivalent of ten 50-run Monte Carlo sets.

Closed-Form Solution for Random Target Maneuver

If we closely investigate Fig. 4.17 we can see that the miss due to a target maneuver with uniformly distributed starting time can be found by squaring the sensitivity due to a step target maneuver and then integrating and taking the square root of the resultant adjoint signal. If we define the miss due to a step target maneuver as MNT , the miss due to a uniformly distributed target maneuver can be expressed mathematically as

$$MUDNT = n_T \sqrt{\frac{1}{t_F} \int_0^{t_F} MNT^2 d\tau}$$

where n_T is the target maneuver level and t_F the flight time. For a single-lag guidance system with an effective navigation ratio of 3, we have already shown in Chapter 3 that the miss due to a step target maneuver is given by

$$MNT|_{N'=3} = 0.5 t_F^2 e^{-t_F/T}$$

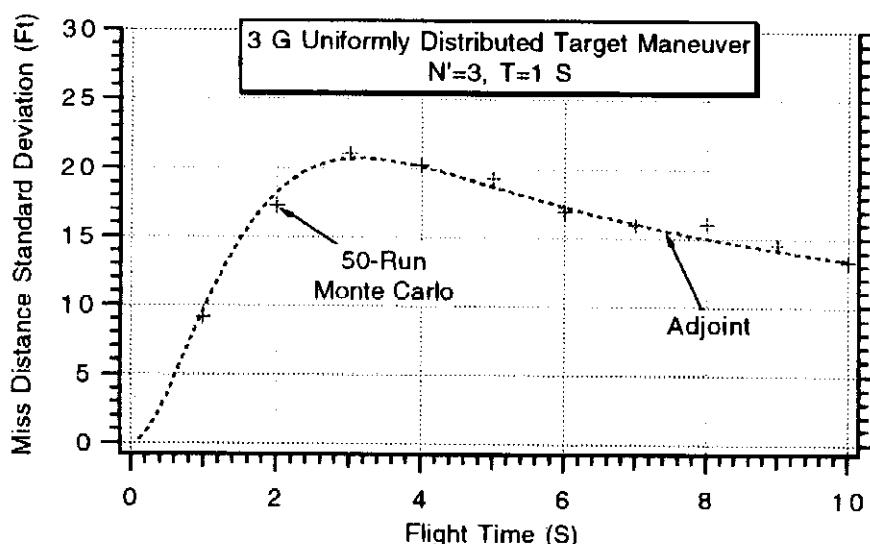


Fig. 4.18 Shaping filter adjoint and Monte Carlo results are in close agreement.

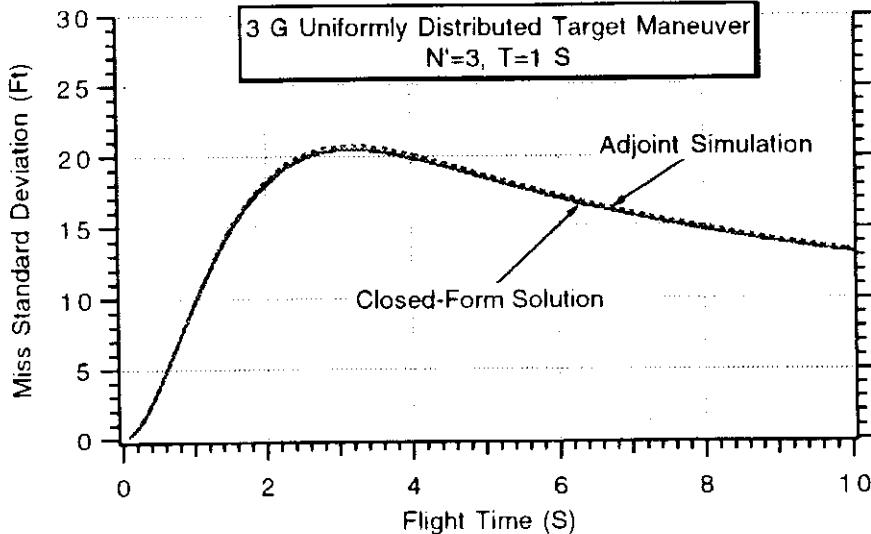


Fig. 4.19 Closed-form solution and adjoint simulation results agree.

where T is the guidance system time constant. Substitution of the step target maneuver solution into the expression for the uniformly distributed target maneuver yields

$$MUDNT|_{N'=3} = n_T \sqrt{\frac{1}{t_F} \int_0^{t_F} 0.25 \tau^4 e^{-2\tau/T} d\tau}$$

After integration by parts and much algebra we obtain the closed-form expression, valid for an effective navigation ratio of 3, for the uniformly distributed target maneuver:

$$MUDNT|_{N'=3} = \frac{n_T}{4} \sqrt{\frac{T^5}{t_F} [3 - e^{-2x}(2x^4 + 4x^3 + 6x^2 + 6x + 3)]}$$

where x is defined as normalized time, or

$$x = \frac{t_F}{T}$$

Figure 4.19 displays the closed-form solution for the case in which the guidance system time constant is 1 s. Superimposed on the plot are the previously presented adjoint results for a uniformly distributed target maneuver. We can see from the close proximity of the two curves that both solutions are in close agreement.

Summary

Starting from basic definitions of random variables, we have shown how to simulate random phenomena and properly interpret the simulation results. Throughout this chapter we have shown two ways of doing problems: the theoretical way, which only works under certain circumstances, and simulation, which is always valid. Numerical examples have been presented that not only demonstrate that theory and simulation agree but also show how each method offers new insights. Finally, it was shown how the method of adjoints can be extended to evaluate system performance in the presence of random disturbances.

References

- ¹Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.
- ²Zarchan, P., "Engineering Tips For Plotting," *MacTutor*, Vol. 4, Feb. 1988, pp. 78-82.
- ³Zarchan, P., "Comparison of Statistical Digital Simulation Methods," *Advisory Group For Aerospace Research and Development*, AGARDograph No. 273, July 1988, pp. 2-1-2-16.
- ⁴Laning, J. H., and Battin, R. H., *Random Processes in Automatic Control*, McGraw-Hill, New York, 1956.
- ⁵Zarchan, P., "Complete Statistical Analysis of Nonlinear Missile Guidance Systems—SLAM," *Journal of Guidance and Control*, Vol. 2, Jan.-Feb. 1979, pp. 71-78.
- ⁶Zarchan, P., "Representation of realistic evasive maneuvers by the use of shaping filters," *Journal of Guidance and Control*, Vol. 2, July-Aug. 1979, pp. 290-295.
- ⁷Fitzgerald, R. J., "Shaping Filters For Disturbances With Random Starting Times," *Journal of Guidance and Control*, Vol. 2, March-April 1979, pp. 152-154.



Covariance Analysis and the Homing Loop

Background

COVARIENCE analysis is another useful computerized tool that can be used to analyze time-varying linear systems driven by random inputs. Covariance analysis, like the adjoint technique, is an exact method of analysis that is restricted to linear systems. With this method, the covariance matrix of the system state vector is propagated as a function of time by the direct integration of a nonlinear matrix differential equation. Exact statistical performance projections of any state or combination of states as a function of time can be obtained with this technique. Covariance analysis is quite popular in problems associated with inertial navigation and optimal estimation. We shall show that the covariance analysis technique can also be used to get exact statistical performance projections in a missile guidance system.

Theory

So far we are accustomed to writing computer programs directly from inspection of the system block diagram. To apply covariance analysis, we must first change our method of operation and convert the system block diagram to state space notation or an equivalent set of first-order differential equations expressed in matrix form.

The dynamics of any linear system driven by white noise inputs can be converted to the following first-order vector differential equation:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{u}(t)$$

where $\mathbf{x}(t)$ is the system state vector, $\mathbf{F}(t)$ is the system dynamics matrix, and $\mathbf{u}(t)$ is a white noise vector with spectral density matrix $\mathbf{Q}(t)$, or

$$\mathbf{Q}(t) = E[\mathbf{u}(t)\mathbf{u}^T(t)]$$

The matrix differential equation for the propagation of the covariance of this general system is^{1,2}

$$\dot{\mathbf{X}}(t) = \mathbf{F}(t)\mathbf{X}(t) + [\mathbf{F}(t)\mathbf{X}(t)]^T + \mathbf{Q}(t)$$

where the covariance matrix $\mathbf{X}(t)$ is related to the state $\mathbf{x}(t)$ according to

$$\mathbf{X}(t) = E[\mathbf{x}(t)\mathbf{x}^T(t)]$$

The diagonal elements of the covariance matrix represent the variances of the state

variables if the disturbance processes are zero mean. The off-diagonal elements of the covariance matrix represent the degree of correlation between the various state variables.

Low-Pass Filter Example

To demonstrate the application of covariance analysis, let us revisit the example of Chapter 4 in which a low-pass filter with a white noise input has been redrawn in block diagram form as shown in Fig. 5.1. In this example the input u_s is white noise with power spectral density Φ_s , T is the time constant of the low-pass filter, and x is the filter output. We want to find the variance of x as a function of time.

By inspection of Fig. 5.1, we can write the first-order differential equation of the low-pass filter in state space form as

$$\dot{x} = -\frac{x}{T} + \frac{u_s}{T}$$

Therefore, for this example, the system dynamic matrix and spectral density matrix are time-invariant scalars and can be written by inspection of the preceding differential equation as

$$F = \frac{-1}{T}$$

$$Q = \frac{\Phi_s}{T^2}$$

The differential equation for the propagation of the covariance simplifies to the linear equation

$$\dot{X} = \frac{-2}{T} X + \frac{\Phi_s}{T^2}$$

The solution to the preceding linear covariance analysis differential equation can be found, using standard differential equation solution techniques, to be

$$X = \frac{\Phi_s}{2T} [1 - e^{-2t/T}]$$

Since X represents the mean square value of x , we recognize that the preceding answer is identical to the answer obtained in Chapter 4 using the impulse response technique.

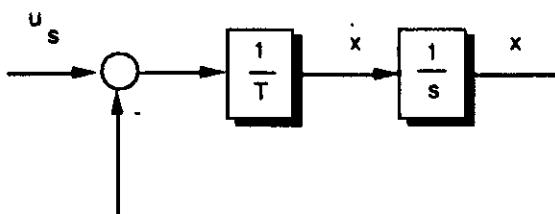


Fig. 5.1 Low-pass filter with white noise input.

Numerical Considerations

In all of the systems simulated in the text, the second-order Runge–Kutta numerical integration method is used to solve the necessary differential equations. Although more accurate numerical integration techniques exist, the second-order Runge–Kutta technique is adequate for getting the correct answers. When the equations associated with covariance analysis are solved numerically, higher order integration methods are required to get the desired accuracy.

Let us again consider the second-order network simulation of Chapter 1 (i.e., Fig. 1.1 and Listing 1.1) in which the second-order Runge–Kutta numerical integration technique was used. In that simulation the integration step size was made very small ($h = 0.001$ s). If we arbitrarily increase the integration step size, we can see from Fig. 5.2 that the accuracy of the answers begin to degrade.

It is apparent from Fig. 5.2 that $h = 0.01$ s is about the largest the integration step size can be made without degrading accuracy. This is not surprising because the natural frequency of the second-order system in this example is 20 rad/s. This means that the approximate time constant of the system under consideration is 0.05 s ($1/20 = 0.05$). With second-order Runge–Kutta numerical integration we must make the integration interval at least five times smaller than the natural frequency of the network in order to get accurate answers. Making the integration step size equal to or larger than 0.02 s means that we will be missing the effect of the systems high bandwidth.

Better accuracy can be achieved with the fourth-order Runge–Kutta numerical integration technique.^{3,4} Given a first-order differential equation of the form

$$\dot{x} = f(x, t)$$

where t is time, we want to find a numerical integration recursive relationship for x as a function of time. With the fourth-order Runge–Kutta numerical integration technique the value of x at the next integration interval h is given by

$$x_{k+1} = x_k + \frac{h}{6} [K_0 + 2K_1 + 2K_2 + K_3]$$

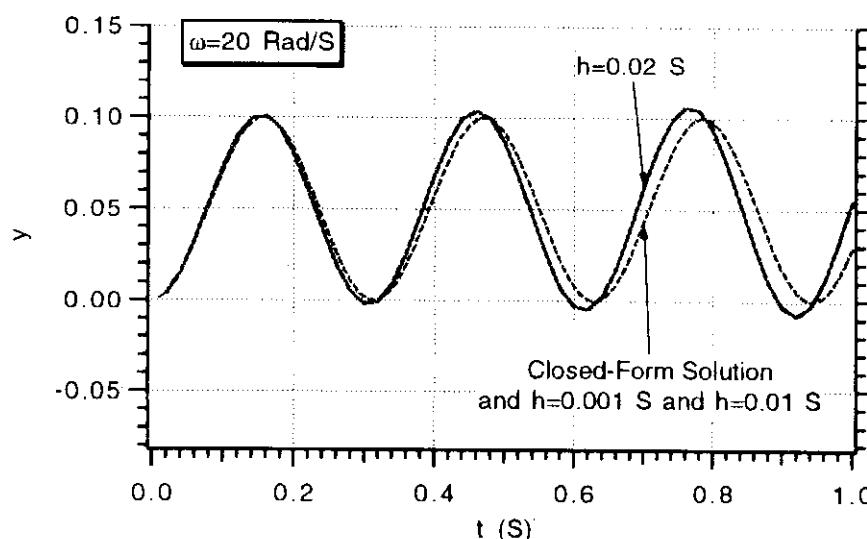


Fig. 5.2 Increasing integration step size eventually degrades accuracy.

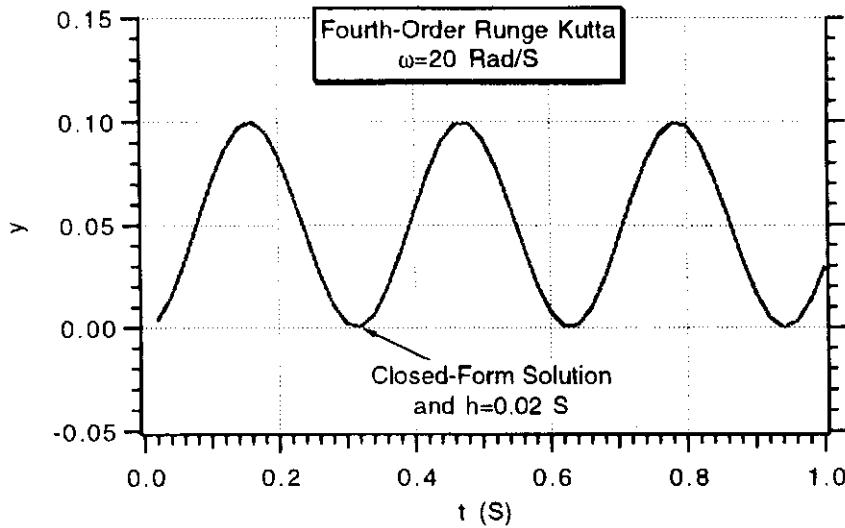


Fig. 5.3 Fourth-order Runge–Kutta integration yields adequate accuracy with larger integration step sizes.

where

$$K_0 = f(x_k, t_k)$$

$$K_1 = f(x_k + 0.5K_0, t + 0.5h)$$

$$K_2 = f(x_k + 0.5K_1, t + 0.5h)$$

$$K_3 = f(x_k + K_2, t + h)$$

From the preceding expressions we can see that the new value of x is simply the old value of x plus terms proportional to the derivative evaluated at various times between t and $t + h$. Using the relationships for the fourth-order Runge–Kutta integration technique, we can write a program to simulate the second-order network as shown in Listing 5.1. We can see that the structure of this program is identical to the one of Listing 1.1 (i.e., differential equations appear after statement label 200) except extra steps have been added to the integration procedure.

Figure 5.3 shows that when we simulate the second-order network with Listing 5.1 using fourth-order rather than second-order Runge–Kutta numerical integration we can use a larger integration step size to get the same accuracy. In this example, a step size of 0.02 s was adequate for getting the correct solution.

Homing Loop Example

To demonstrate the utility of covariance analysis for a more relevant example, let us revisit the single-lag homing loop example of Chapter 4 in which the random error source is a uniformly distributed target maneuver. The homing loop model of Fig. 4.15 is redrawn in Fig. 5.4 for convenience. In Fig. 5.4 the uniformly distributed target maneuver has been replaced by its shaping filter equivalent, which is white noise through an integrator. The spectral density of the white noise input u_S is given by Φ_S , which was shown in Chapter 4 to be

$$\Phi_S = \frac{n_T^2}{t_F}$$

**Listing 5.1 Simulation of second-order system
with fourth-order Runge-Kutta integration technique**

```

INTEGER ORDER,STEP
REAL K0(2),K1(2),K2(2),K3(2)
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
DATA XIN/1./
W=20.
Y=0.
YD=0.
T=0.
H=.01
S=0.
5 IF(T.GE.1.)GOTO 999
S=S+H
YOLD=Y
YDOLD=YD
STEP=1
GOTO 200
40 STEP=2
K0(1)=YD
K0(2)=YDD
TNEW=T+.5*H
Y=YOLD+.5*H*K0(1)
YD=YDOLD+.5*H*K0(2)
GOTO 200
41 STEP=3
K1(1)=YD
K1(2)=YDD
TNEW=T+.5*H
Y=YOLD+.5*H*K1(1)
YD=YDOLD+.5*H*K1(2)
GOTO 200
42 STEP=4
K2(1)=YD
K2(2)=YDD
TNEW=T+H
Y=YOLD+H*K2(1)
YD=YDOLD+H*K2(2)
GOTO 200
43 CONTINUE
K3(1)=YD
K3(2)=YDD
T=TNEW
Y=YOLD+H*(K0(1)+2.*(K1(1)+K2(1))+K3(1))/6.
YD=YDOLD+H*(K0(2)+2.*(K1(2)+K2(2))+K3(2))/6.
IF(S.GE..01999)THEN
S=0.
YTHERORY=(1.-COS(W*T))/W
WRITE(*,97)T,Y,YTHERORY
WRITE(1,97)T,Y,YTHERORY

```

(Contd.)

Listing 5.1 (Continued)

```

END IF
GOTO 5
97 FORMAT(3F10.5)
200 CONTINUE
YDD=W*XIN-W*W*Y
IF(STEP.EQ.1)THEN
  GOTO 40
ELSEIF(STEP.EQ.2)THEN
  GOTO 41
ELSEIF(STEP.EQ.3)THEN
  GOTO 42
ELSE
  GOTO 43
ENDIF
999 CONTINUE
CLOSE(1)
PAUSE
END

```

where n_T is the magnitude of the target maneuver and t_F is the flight time over which the maneuver is equally likely to occur.

To apply covariance analysis to the homing loop of Fig. 5.4, we must convert this block diagram to state space form. To perform the conversion, the homing loop equations must first be expressed as a set of first-order linear differential equations or

$$\ddot{y} = \ddot{y}_T - N' V_c \dot{D} = \ddot{y}_T - \frac{N' V_c}{T} \left[\frac{y}{V_c(t_F - t)} - D \right]$$

$$\ddot{y}_T = u_s$$

$$\dot{D} = \frac{1}{T} \left[\frac{y}{V_c(t_F - t)} - D \right]$$

Since the preceding set of first-order differential equations are functions of the

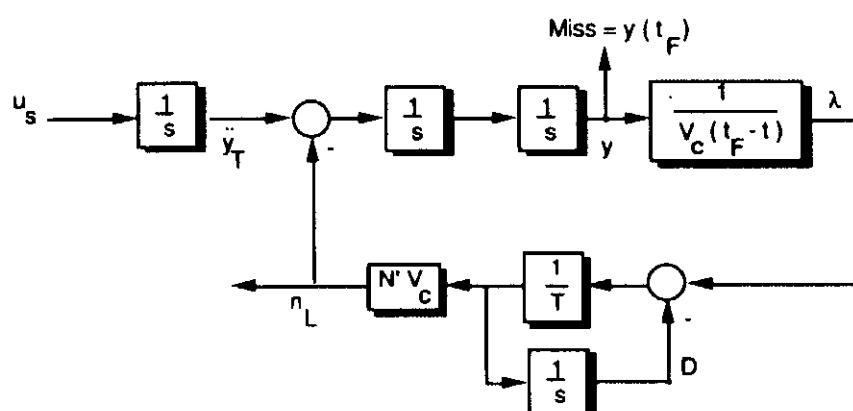


Fig. 5.4 Single-lag homing loop with random target maneuver.

states, they can be written in state space form by inspection as

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \ddot{y}_T \\ \dot{D} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -N' & 0 & 1 & \frac{N' V_c}{T} \\ \frac{-N'}{T(t_F - t)} & 0 & 1 & \frac{N' V_c}{T} \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & \frac{-1}{T} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \ddot{y}_T \\ D \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ u_S \\ 0 \end{bmatrix}$$

By comparing the preceding matrix differential equation with the generalized state space equation we can see that the state vector for this example is

$$\mathbf{x} = \begin{bmatrix} y \\ \dot{y} \\ \ddot{y}_T \\ D \end{bmatrix}$$

and the system dynamic matrix is given by

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -N' & 0 & 1 & \frac{N' V_c}{T} \\ \frac{-N'}{T(t_F - t)} & 0 & 1 & \frac{N' V_c}{T} \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & \frac{-1}{T} \end{bmatrix}$$

From the homing loop state space equation we can also see that $\mathbf{u}(t)$ is

$$\mathbf{u} = \begin{bmatrix} 0 \\ 0 \\ u_S \\ 0 \end{bmatrix}$$

and therefore the spectral density matrix $\mathbf{Q}(t)$ becomes

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \Phi_S & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where Φ_S has been previously defined.

Integration of the covariance analysis nonlinear matrix differential equation yields statistical information for all of the states. For this homing loop example, the standard deviation of the relative trajectory y can be found by taking the square root of the first diagonal element of the covariance matrix X or

$$\sigma_y(t) = \sqrt{X(1, 1)}$$

The source code listing of the homing loop covariance analysis program appears in Listing 5.2. We can see from the listing that both double-precision arithmetic and fourth-order Runge–Kutta integration are used to get the necessary accuracy. Standard transposition, multiplication, and addition matrix subroutines⁴ have been

Listing 5.2 Homing loop covariance analysis program

```

IMPLICIT REAL*8(A-H)
IMPLICIT REAL*8(O-Z)
REAL*8 F(4,4),X(4,4),FX(4,4),FXT(4,4),FXFXT(4,4)
REAL*8 XOLD(4,4),K0(4,4),Q(4,4)
REAL*8 K1(4,4),K2(4,4),K3(4,4),XD(4,4)
REAL*8 A(1,4),AT(4,1),AX(1,4),AXAT(1,1)
INTEGER ORDER,STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
ORDER=4
T=0.
TNEW=T
S=0.
H=.01
XNP=3.
TAU=1.
XNT=96.6
VC=4000.
TF=10.
TGO=TF-T+.00001
PHIS=XNT*XNT/TF
DO 10 I=1,ORDER
DO 10 J=1,ORDER
F(I,J)=0.
X(I,J)=0.
Q(I,J)=0.
10 CONTINUE
F(1,2)=1.
F(2,1)=-XNP/(TAU*TGO)
F(2,3)=1.
F(2,4)=XNP*VC/TAU
F(4,1)=1./(TAU*VC*TGO)
F(4,4)=-1./TAU
Q(3,3)=PHIS
5 IF(T.GE.TF)GOTO 999
DO 20 I=1,ORDER
DO 20 J=1,ORDER
XOLD(I,J)=X(I,J)
20 CONTINUE
STEP=1
GOTO 200
40 STEP=2
DO 50 I=1,ORDER
DO 50 J=1,ORDER
K0(I,J)=XD(I,J)
50 CONTINUE
TNEW=T+.5*H
DO 60 I=1,ORDER
DO 60 J=1,ORDER
X(I,J)=XOLD(I,J)+.5*H*K0(I,J)

```

(Contd.)

Listing 5.2 (Continued)

```

60  CONTINUE
    GOTO 200
41  STEP=3
    DO 70 I=1,ORDER
    DO 70 J=1,ORDER
    K1(I,J)=XD(I,J)
70  CONTINUE
    TNEW=T+.5*H
    DO 75 I=1,ORDER
    DO 75 J=1,ORDER
    X(I,J)=XOLD(I,J)+.5*H*K1(I,J)
75  CONTINUE
    GOTO 200
42  STEP=4
    DO 80 I=1,ORDER
    DO 80 J=1,ORDER
    K2(I,J)=XD(I,J)
80  CONTINUE
    TNEW=T+H
    DO 85 I=1,ORDER
    DO 85 J=1,ORDER
    X(I,J)=XOLD(I,J)+H*K2(I,J)
85  CONTINUE
    GOTO 200
43  DO 90 I=1,ORDER
    DO 90 J=1,ORDER
    K3(I,J)=XD(I,J)
90  CONTINUE
    T=TNEW
    DO 95 I=1,ORDER
    DO 95 J=1,ORDER
    X(I,J)=XOLD(I,J)+H*(K0(I,J)+2.*(K1(I,J)+K2(I,J))+
1          K3(I,J))/6.
95  CONTINUE
    S=S+H
    IF(S.LE..09999)GOTO 5
    S=0.
    A(1,1)=XNP/(TAU*TGO)
    A(1,2)=0.
    A(1,3)=0.
    A(1,4)=-XNP*VC/TAU
    CALL MATMUL(A,1,ORDER,X,ORDER,ORDER,AX)
    CALL MATTRN(A,1,ORDER,AT)
    CALL MATMUL(AX,1,ORDER,AT,ORDER,1,AXAT)
    SIGY=SQRT(X(1,1))
    SIGNL=SQRT(AXAT(1,1))
    WRITE(*,97)T,SIGY,SIGNL/32.2
    WRITE(1,97)T,SIGY,SIGNL/32.2
    GOTO 5

```

(Contd.)

Listing 5.2 (Continued)

```

97  FORMAT(3F12.3)
200 TGO=TF-TNEW+.00001
    F(2,1)=-XNP/(TAU*TGO)
    F(4,1)=1.0/(TAU*VC*TGO)
    CALL MATMUL(FORDER,ORDER,X,ORDER,ORDER,FX)
    CALL MATTRN(FX,ORDER,ORDER,FXT)
    CALL MATADD(FX,ORDER,ORDER,FXT,FXFXT)
    CALL MATADD(FXFXT,ORDER,ORDER,Q,XD)
    IF(STEP.EQ.1)THEN
        GOTO 40
    ELSEIF(STEP.EQ.2)THEN
        GOTO 41
    ELSEIF(STEP.EQ.3)THEN
        GOTO 42
    ELSE
        GOTO 43
    ENDIF
999 CONTINUE
PAUSE
END

SUBROUTINE MATTRN(A,IROW,ICOL,AT)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
REAL*8 A(IROW,ICOL),AT(ICOL,IROW)
DO 105 I=1,IROW
    DO 105 J=1,ICOL
        AT(J,I)=A(I,J)
105 CONTINUE
RETURN
END

SUBROUTINE MATMUL(A,IROW,ICOL,B,JROW,JCOL,C)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
REAL*8 A(IROW,ICOL),B(JROW,JCOL),C(IROW,JCOL)
DO 110 I=1,IROW
    DO 110 J=1,JCOL
        C(I,J)=0.
        DO 110 K=1,ICOL
            C(I,J)=C(I,J)+A(I,K)*B(K,J)
110 CONTINUE
RETURN
END

SUBROUTINE MATADD(A,IROW,ICOL,B,C)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
REAL*8 A(IROW,ICOL),B(IROW,ICOL),C(IROW,ICOL)

```

(Contd.)

Listing 5.2 (Continued)

```

DO 120 I=1,IROW
DO 120 J=1,ICOL
      C(I,J)=A(I,J)+B(I,J)
120   CONTINUE
      RETURN
      END

```

added to conveniently form the covariance matrix differential equations after statement label 200. From Listing 5.2 we can see that the only error source in the guidance system is a 3-g uniformly distributed target maneuver.

The homing loop covariance analysis program of Listing 5.2 was run, and Fig. 5.5 presents the resultant standard deviation of the relative separation between the missile and target [i.e., square root of first diagonal element of covariance matrix represents $\sigma_y(t)$] for the entire 10-s flight. At the end of the flight, the relative separation between the missile and target is the miss distance [i.e., $\sigma_{\text{Miss}} = \sigma_y(t_F)$]. In this example the covariance analysis program indicates that the standard deviation of the miss distance is 13.3 ft, which is identical to the adjoint results of Chapter 4 (i.e., see Fig. 4.18). Unlike the adjoint technique, covariance analysis does not provide miss distance error budget information for all different flight times in a single computer run. However, as can be seen from Fig. 5.5, covariance analysis does provide relative trajectory information at all times for a given flight time. If many random error sources are present, one covariance analysis computer run yields a total statistical performance projection. If learning how each error source contributed to the total performance is desired, additional computer runs must be made—each one run with one error source at a time!

Covariance analysis also has the capability of providing information concerning other quantities in the same computer run. For example, covariance analysis could also show us how the standard deviation of the missile acceleration varies with time

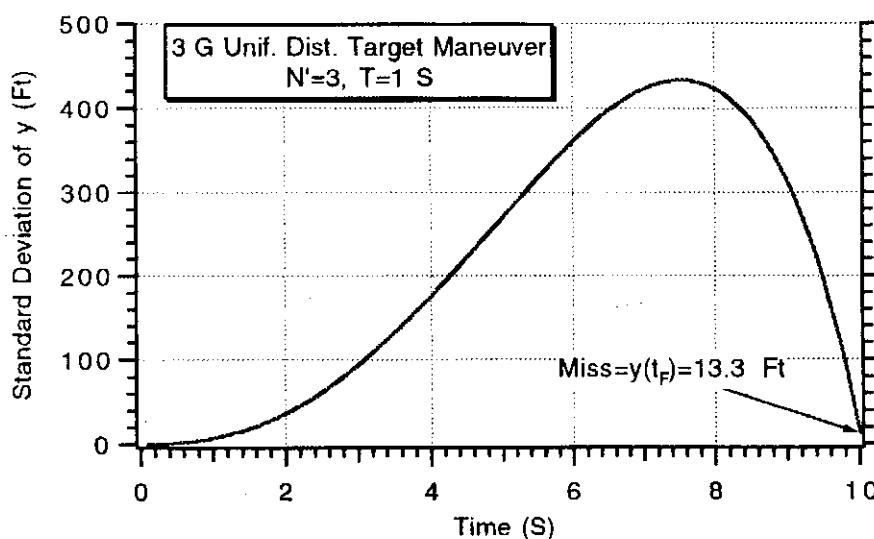


Fig. 5.5 Covariance analysis miss distance results agree with adjoint.

in the same computer run. However, we must first express the missile acceleration as a function of the states. From Fig. 5.4 we can see that the missile acceleration is related to the states according to

$$n_L = \frac{N' V_c}{T} \left[\frac{y}{V_c(t_F - t)} - D \right]$$

or more concisely, in matrix form we can say that

$$n_L = Ax$$

where x is the system state vector, and for this example A is given by

$$A = \begin{bmatrix} \frac{N'}{T(t_F - t)} & 0 & 0 & \frac{-N' V_c}{T} \end{bmatrix}$$

Therefore the variance of the missile acceleration is given by

$$E [n_L n_L^T] = \sigma_{n_L}^2 = AXA^T$$

where X is the covariance matrix and the standard deviation of the acceleration is simply the square root of the preceding expression. From Listing 5.2 we can see that A and the variance of the acceleration are defined before the *WRITE* statement preceding statement label 200. Figure 5.6 displays the resultant missile acceleration profile, using the preceding expression in the covariance analysis program, for the entire flight. We can see that the standard deviation of the missile acceleration is monotonically increasing for the 10-s flight. If we make the Gaussian assumption, we can infer that 68% of the time 15 g (i.e., 483 ft/s²) or 5 times the acceleration of the target is required to avoid acceleration saturation. In more pessimistic terms we can also say that if the missile only has a 15-g capability there is a 32% probability that the missile will acceleration saturate for this example. This example demonstrates that although covariance analysis does not provide all of the information of the adjoint, it does provide extra useful information which can be used to access system performance. In addition, the covariance analysis technique can be used to provide an independent check of the accuracy of an adjoint simulation.

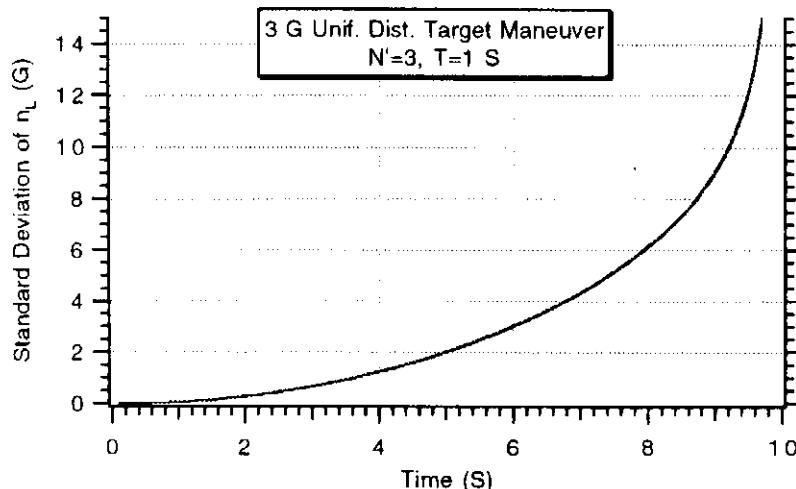


Fig. 5.6 Covariance analysis also provides acceleration profile information.

Acceleration Adjoint

We state in Chapter 3 that the impulse response of the original system and adjoint system are related according to

$$h^*(t_F - t_I, t_F - t_o) = h(t_o, t_I)$$

where h denotes the impulse responses of the original system and h^* is the impulse response of the adjoint system. This important relationship means that putting an impulse into the original system at time t_I and observing the output at time t_o is identical to putting an impulse into the adjoint system at time $t_F - t_o$ and observing the output at time $t_F - t_I$, where t_F is the final time or flight time. In all of the adjoint applications discussed so far, the observation time was always the final time t_F since we were only interested in the miss distance. If all disturbances occur at time zero in the original system but the observation time is not the final time, the fundamental adjoint relationship simplifies to

$$h^*(t_F, t_F - t_o) = h(t_o, 0)$$

The preceding relationship means that applying an impulse to the original system at time zero and observing the output at time t_0 is equivalent to initiating the impulse at time $t_F - t_0$ in the adjoint system and observing the output at time t_F . *In other words, if we would like to develop other types of adjoints it is only necessary to change the impulse application time and the location of the impulse application. The adjoint block diagram remains unchanged!*

Figure 5.7 is the adjoint block diagram of the single-lag homing loop of Fig. 5.4. This adjoint diagram is identical to Fig. 4.17 (adjoint model in Chapter 4) except that it is noted that certain initial conditions are used if a miss distance adjoint is

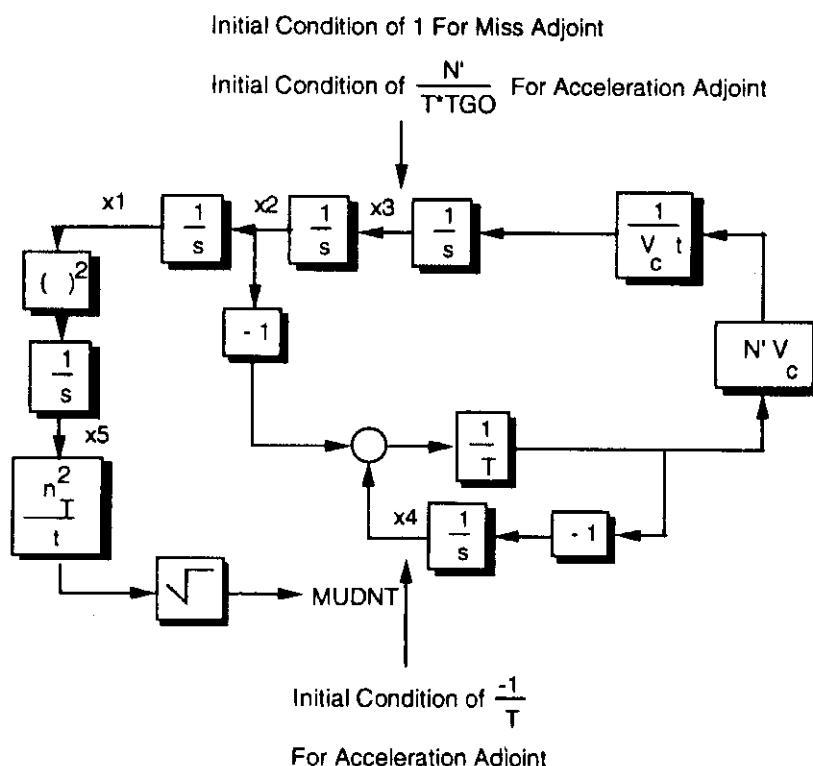


Fig. 5.7 Adjoint model for miss distance and acceleration.

desired and other initial conditions are used for an acceleration adjoint. If a miss distance adjoint is being run, an initial condition of unity is applied at time zero on the x_3 integrator. If an acceleration adjoint is required, initial conditions are applied at time $t_F - t_o$ to the x_3 and x_4 integrators. In this case time $t_F - t_o$ corresponds to the time to go until intercept in which we desire to observe the acceleration in the original system. In other words, if we desire to observe acceleration in the original system at time 8 s for a 10-s flight, that is the same as observing the acceleration at 2 s to go (i.e., $10 - 8 = 2$). Therefore the impulse (or initial conditions on appropriate integrators) is applied at time 2 s in the adjoint system and the output is observed at 10 s in the adjoint system. Changing the observation time in the adjoint system corresponds to observing acceleration at 2 s to go for different flight times in the original system.

The source code listing for the adjoint program, which can be used for both miss distance and acceleration computation, appears in Listing 5.3. In this program $TINT$ represents the time to go in the original system in which we wish to observe the quantity of interest. If we want to compute miss distance, $MISS$ should be set to `TRUE`, and $TINT$ set to zero. For an acceleration adjoint, $MISS$ should be set to `FALSE`, and $TINT$ set to a number representing the time to go at which we want to observe the acceleration. The adjoint program of Listing 5.3 is set to run as an acceleration adjoint in which acceleration levels correspond to 0.5 s to go in the original system ($TINT = 0.5$).

The preceding acceleration adjoint program was run for values of observation time in the original system corresponding to 0.5, 1, and 2 s to go ($TINT = 0.5, 1, 2$), and the results for the three adjoint runs are displayed in Fig. 5.8. We can interpret the abscissa of the plot as either adjoint time or flight time. The curve representing acceleration at 0.5 s to go, $TINT = 0.5$ (labeled $t_{go} = 0.5$), indicates that the standard deviation of the missile acceleration is 12 g at 0.5 s to go for a 10-s flight, approximately 12 g at 0.5 s to go for a 6-s flight, and approximately 4 g for a 2-s flight. The missile acceleration values for a 10-s flight at observation times to go of 0.5, 1, and 2 s (12 g, 9 g, and 6.1 g) respectively agree exactly with the single run covariance analysis results of Fig. 5.6.

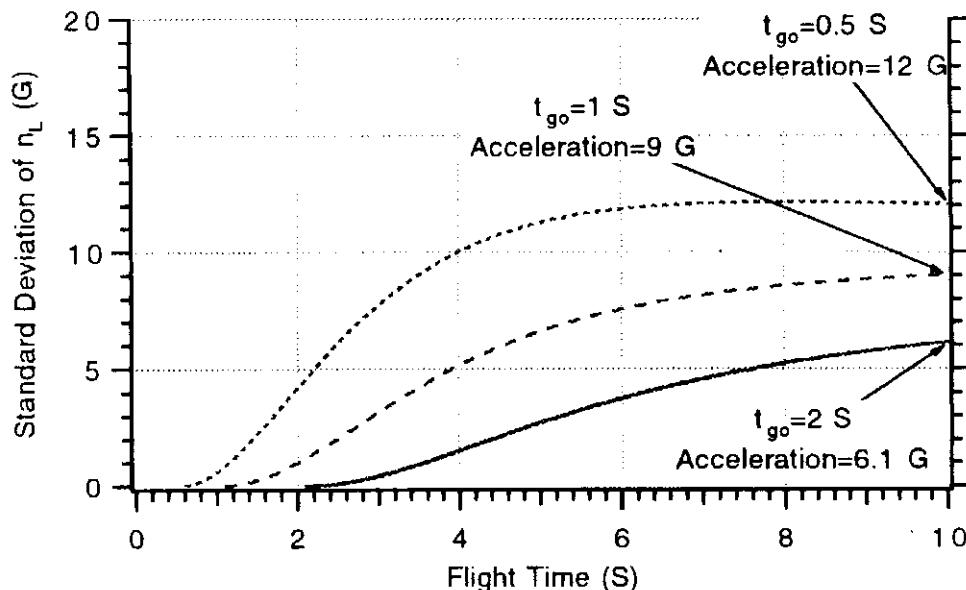


Fig. 5.8 Adjoint and covariance analysis acceleration results agree.

**Listing 5.3 Acceleration and miss distance
adjoint program**

```

INTEGER STEP
LOGICAL MISS
DATA XNT,XNP,TAU,TF/ 96.6,3,1,10/
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
T=0.
S=0.
TINT=.5
MISS=.FALSE.
TP=T+.00001+TINT
X1=0
X2=0
X5=0.
IF(MISS)THEN
    X3=1.
    X4=0.
ELSE
    X3=XNP/(TAU*TINT)
    X4=-1./TAU
ENDIF
H=.01
10 IF(TP>(TF-.00001))GOTO 999
S=S+H
X1OLD=X1
X2OLD=X2
X3OLD=X3
X4OLD=X4
X5OLD=X5
STEP=1
GOTO 200
66 STEP=2
X1=X1+H*X1D
X2=X2+H*X2D
X3=X3+H*X3D
X4=X4+H*X4D
X5=X5+H*X5D
TP=TP+H
GOTO 200
55 CONTINUE
X1=(X1OLD+X1)/2+.5*H*X1D
X2=(X2OLD+X2)/2+.5*H*X2D
X3=(X3OLD+X3)/2+.5*H*X3D
X4=(X4OLD+X4)/2+.5*H*X4D
X5=(X5OLD+X5)/2+.5*H*X5D
IF(S<.099999)GOTO 10
S=0.
XMUDNT=XNT*SQRT(X5/TGO)
IF(MISS)THEN
    WRITE(*,97)TP,XMUDNT
    WRITE(1,97)TP,XMUDNT

```

(Contd.)

Listing 5.3 (Continued)

```

        ELSE
          WRITE(*,97)TP,XMUDNT/32.2
          WRITE(1,97)TP,XMUDNT/32.2
        ENDIF
        GOTO 10
97      FORMAT(2F10.3)
200     CONTINUE
        X1D=X2
        X2D=X3
        Y1=(X4-X2)/TAU
        TGO=TP+.00001
        X3D=XNP*Y1/TGO
        X4D=-Y1
        X5D=X1*X1
        IF(STEP-1)66,66,55
999     CONTINUE
        PAUSE
        CLOSE(1)
        END

```

Summary

In this chapter we have shown how the covariance analysis technique can be applied to a missile guidance system. Double-precision arithmetic and the fourth-order Runge-Kutta numerical integration technique were required in order to obtain performance projections of the desired accuracy. Although covariance analysis techniques do not yield error budget information as does the adjoint technique, exact performance projects can be obtained for all quantities of interest in a single run. It was also shown how the adjoint technique could be extended to yield acceleration as well as miss distance information.

References

- ¹Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.
- ²Bryson, A. E., and Ho, Y. C., *Applied Optimal Control*, Blaisdell, Waltham, MA, 1969.
- ³Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., *Numerical Recipes: The Art of Scientific Computation*, Cambridge, University Press, London, 1986.
- ⁴Wolf, P. M., and Koelling, C. P., *Basic Engineering, Science and Business Programs for the Apple II and IIe*, Bradly Communications Co. Inc., Bowie, MD, 1984.

Proportional Navigation and Miss Distance

Introduction

THE relationship between proportional navigation and miss distance will be investigated more extensively in this chapter. First we will demonstrate, via numerical examples, that it is important to have an accurate guidance system model in order to get performance projections that are meaningful and not overly optimistic. Normalized design curves will be presented that allow an analyst to rapidly predict system behavior given a minimum of information. Curves of this type are invaluable in preliminary system design. The influence of optimal target maneuvers on system performance will be evaluated to highlight potential guidance system weaknesses. Finally, the influence of saturation and parasitic effects will be demonstrated to help the designer place realistic bounds on achievable system performance.

System Order¹

Thus far, the work presented has concerned itself with either a zero- or single-lag guidance system. We have seen that, if the flight time is *not* significantly larger than the guidance system time constant, then the difference between the performance of a zero- and single-lag guidance system can be significant. Both the single- and zero-lag guidance systems are convenient analytical models but do not quite match reality. It is important to determine if a higher-order guidance system representation would influence system performance. To separate time constant and system order effects, it is convenient to use a binomial representation of the guidance system:

$$\frac{n_L}{\lambda} = (N' V_c) \left/ \left[\left(\frac{1 + sT}{n} \right)^n \right] \right.$$

In the preceding representation, T is the effective guidance system time constant and n the system order. If $n = 1$, then the binomial expansion reduces to the single-lag guidance system, which we have already previously studied. This particular form of the binomial representation is useful, although not especially realistic, because an expansion of the guidance system denominator always yields

$$\left(1 + \frac{sT}{n} \right)^n = 1 + sT + \dots$$

which means that T is always the approximate time constant of the guidance system, regardless of system order.

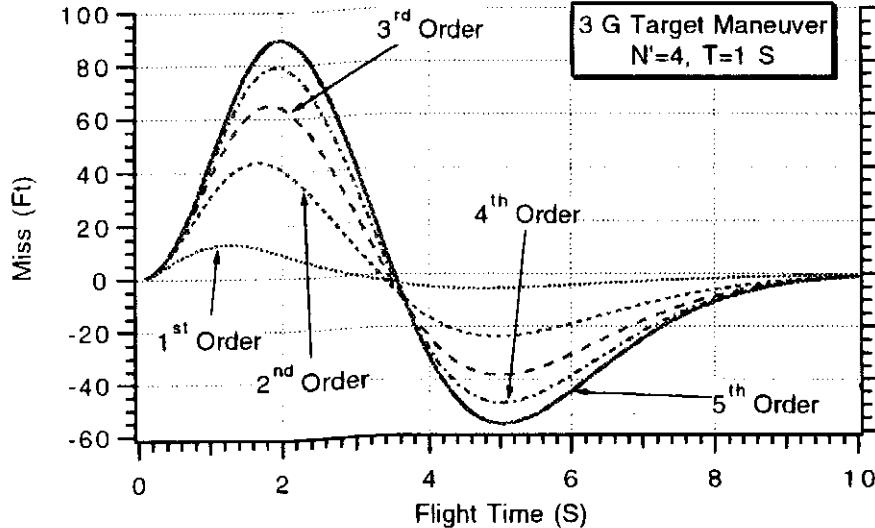


Fig. 6.1 System order has a profound influence on system performance.

Figure 6.1 shows how the miss distance due to a 3-g target maneuver varies with flight time and system order for a binomial guidance system in which the effective navigation ratio is 4 and the effective guidance system time constant is 1 s. We can see that the performance projections resulting from a single-lag guidance system model are a serious underestimate of the influence of target maneuver on miss when the flight time is not an order of magnitude greater than the guidance system time constant. The importance of system order and its influence on system performance becomes less important as system order increases. The experiment conveys the importance of accurately modeling the guidance system (which is generally not a binomial) under consideration if accurate performance projections are required.

An experiment was also conducted to determine if, in the presence of guidance system dynamics, the linearized model of the homing loop still gives accurate performance projections. Fifth-order binomial guidance system models were included in both the linearized and nonlinear engagement simulations. Cases were run for both simulations in the case of a 3-g target maneuver disturbance for various flight times, and the resultant miss distances were monitored. Figure 6.2 shows that the linearized model of the homing loop gives very accurate performance projections. Thus, we can feel confident in using our linearized guidance system model for studies involving a binomial representation of the guidance system.

Normally a missile guidance system is represented by n different time constants for an n -state system. If the time constants are widely separated, then the slowest time constant will usually dictate system performance. If the time constants are closely spaced, one must evaluate the guidance system to get accurate performance projections.

Design Relationships

We have just seen that target maneuver can play a major role in determining missile system performance. Target maneuver can induce very large miss distances if the effective guidance time constant is too large or if the flight time is very short. In addition, we have seen in Chapter 2 that target maneuver induces large missile acceleration levels. This may lead to acceleration saturation, which will significantly further increase the induced miss distance. The purpose of this section

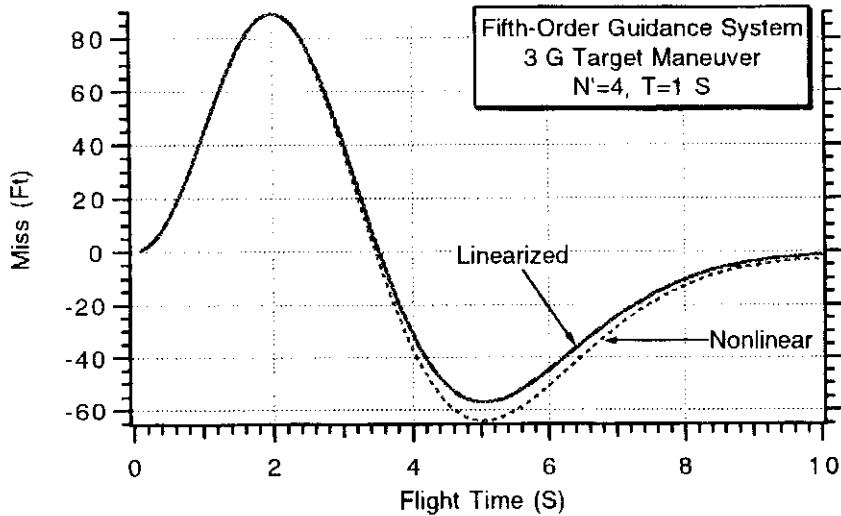


Fig. 6.2 Linearized guidance system model gives very accurate performance projections.

is to quantify the influence of target maneuver on system performance in a form that will be of value to an analyst in preliminary system design.

In the previous section we established that the linearized model of the guidance system gave accurate performance projections in terms of miss distance induced by target maneuver. Performance projections were obtained by running both the linear and nonlinear engagement simulation for many times of flight, and the resultant data were plotted. The same results could have been obtained by making one adjoint run as shown in Fig. 6.3.

In Chapter 3 a closed-form solution was developed for the miss distance induced by target maneuver in a single time constant representation of the guidance system. Although the miss distance formula will change for varying effective navigation ratios and canonic system form and order, the normalization for miss due to target maneuver will be the same. In this section we will use the method of adjoints to develop design curves that may be of use in preliminary system sizing. We will

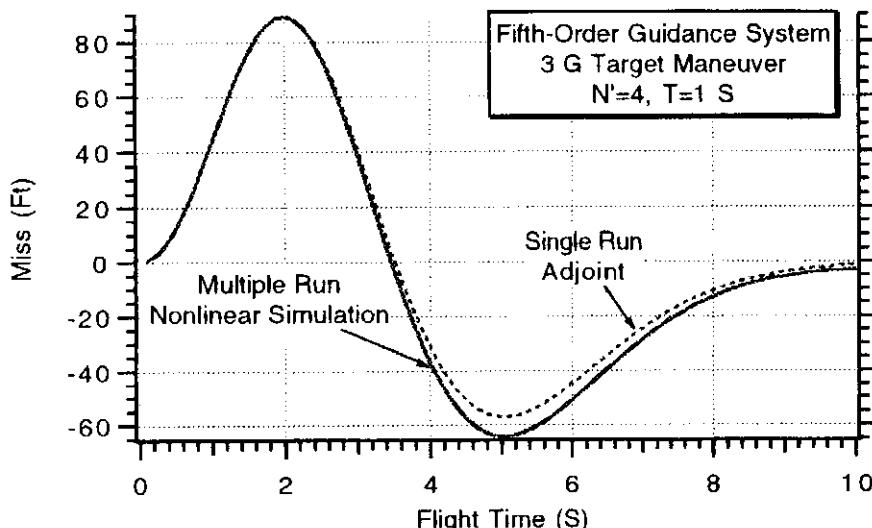


Fig. 6.3 One adjoint run gives the same result as many runs with nonlinear engagement simulation.

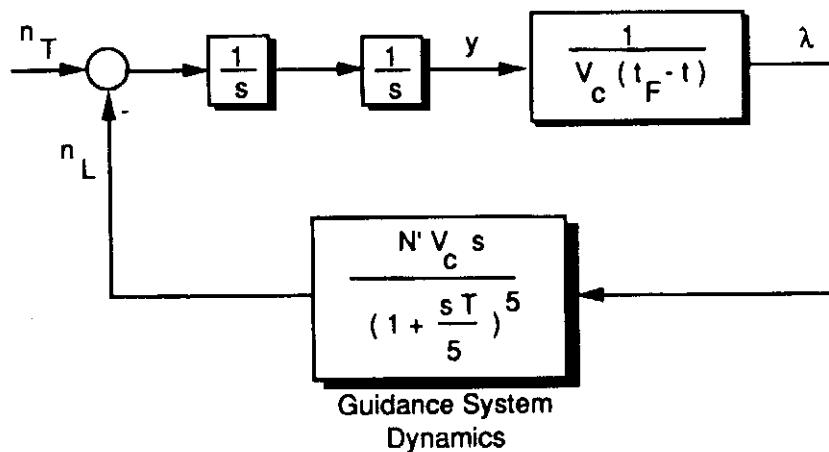


Fig. 6.4 Fifth-order binomial guidance system.

choose a guidance system form that has only one parameter: the guidance system time constant.

The model to be used for the development of normalized design curves is the fifth-order binomial proportional navigation system. This guidance system, which is depicted in Fig. 6.4, has guidance system transfer function

$$\frac{n_L}{\lambda} = \left(N' V_c s \right) \left/ \left[\left(1 + \frac{sT}{5} \right)^5 \right] \right.$$

where T is the guidance system time constant. In this canonic model, one time constant represents the seeker, another represents the noise filter, and the three time constants represents the flight-control system. Hopefully, the simplicity of this model will shed some light on fundamental issues and be of value for other guidance system forms.

Figure 6.5 presents the adjoint model of the fifth-order binomial guidance system. The adjoint model consists of three outputs that are related to three input

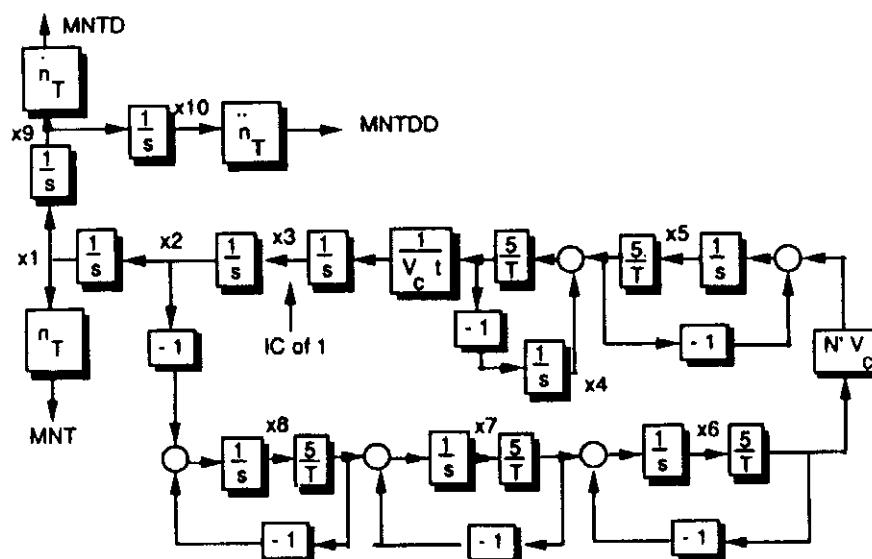


Fig. 6.5 Adjoint of fifth-order binomial guidance system.

disturbances in the original system. The miss due to a step target maneuver is represented by MNT , the miss due to a ramp target maneuver is represented by $MNTD$, and the miss due to a parabolic maneuver is denoted $MNTDD$. In the figure each integrator output is denoted by variables x_1 to x_{10} . The impulse needed to start a miss distance adjoint is represented by a unity initial condition on integrator x_3 .

An adjoint simulation can be derived from the model of Fig. 6.5. Listing 6.1 presents a FORTRAN adjoint program of this fifth-order binomial guidance system. We can see from the listing that the nominal value of the target maneuver is 1 g, the value of target jerk $XNTD$ is 1 g/s, and the value of target yank $XNTDD$ is 1 g/s². As in our other simulations, the differential equations describing the adjoint system can be found after statement label 200. All integrator initial conditions are zero, except for integrator x_3 . We can see from the listing that this integrator has a unity initial condition in order to make a miss distance adjoint. We can also see from the listing that a small number is added to adjoint time so that we can avoid a division by zero. This is a practical way of applying L'Hopital's rule.

The adjoint program is set up to generate normalized results by choosing the guidance system time constant TAU to be unity and the step target maneuver disturbance XNT to be 1 g or 32.2 ft/s². The value of closing velocity is not important, since there is a cancellation of this term in the guidance loop. Normalized adjoint results can be generated by running the program once for a value of unity guidance system time constant. The normalization factors, derived in Chapter 3 for a single-lag guidance system, are also valid for the fifth-order binomial guidance system. Therefore, the adjoint program only has to be rerun for each effective navigation ratio XNP . For example, Fig. 6.6 presents the normalized system response to a step in target acceleration. The abscissa can be interpreted as either normalized time of flight for a step maneuver occurring at the beginning of flight or the normalized time to go at which the disturbance occurs. We can see from Fig. 6.6 that for long normalized flight times the miss approaches zero and for small normalized flight times the miss can be quite large. Increasing the effective navigation ratio tends to reduce the miss for small normalized flight times and increases the miss at the larger normalized flight times.

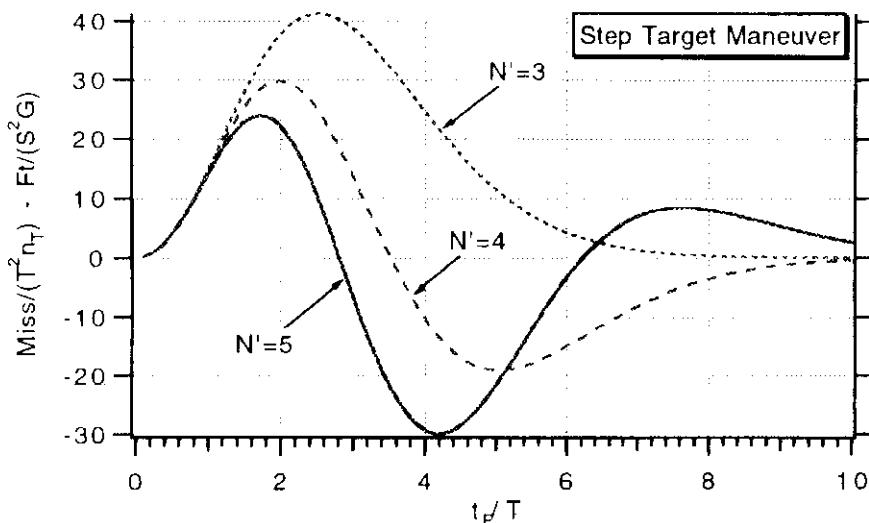


Fig. 6.6 Normalized miss due to step target maneuver.

Listing 6.1 Adjoint of fifth-order binomial guidance system

```

INTEGER STEP
DATA XNT,XNP,TAU,TF,VC/ 32.2,3.,1,10.,4000./
DATA XNTD,XNTDD/32.2,32.2/
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
T=0.
S=0.
TP=T+.00001
X1=0
X2=0
X3=1
X4=0
X5=0.
X6=0.
X7=0.
X8=0.
X9=0.
X10=0.
H=.01
10 IF(TP>(TF-.00001))GOTO 999
S=S+H
X1OLD=X1
X2OLD=X2
X3OLD=X3
X4OLD=X4
X5OLD=X5
X6OLD=X6
X7OLD=X7
X8OLD=X8
X9OLD=X9
X10OLD=X10
STEP=1
GOTO 200
66 STEP=2
X1=X1+H*X1D
X2=X2+H*X2D
X3=X3+H*X3D
X4=X4+H*X4D
X5=X5+H*X5D
X6=X6+H*X6D
X7=X7+H*X7D
X8=X8+H*X8D
X9=X9+H*X9D
X10=X10+H*X10D
TP=TP+H
GOTO 200
55 CONTINUE
X1=(X1OLD+X1)/2+.5*H*X1D
X2=(X2OLD+X2)/2+.5*H*X2D

```

(Contd.)

Listing 6.1 (Continued)

```

X3=(X3OLD+X3)/2+.5*H*X3D
X4=(X4OLD+X4)/2+.5*H*X4D
X5=(X5OLD+X5)/2+.5*H*X5D
X6=(X6OLD+X6)/2+.5*H*X6D
X7=(X7OLD+X7)/2+.5*H*X7D
X8=(X8OLD+X8)/2+.5*H*X8D
X9=(X9OLD+X9)/2+.5*H*X9D
X10=(X10OLD+X10)/2+.5*H*X10D
IF(S<.09999)GOTO 10
S=0.
XMNT=XNT*X1
XMNTD=XNTD*X9
XMNTDD=XNTDD*X10
WRITE(*,97)TP,XMNT,XMNTD,XMNTDD
WRITE(1,97)TP,XMNT,XMNTD,XMNTDD
GOTO 10
97 FORMAT(4F10.3)
200 CONTINUE
X1D=X2
X2D=X3
Y1=5.*(.5*X5/TAU+X4)/TAU
TGO=TP+.00001
X3D=Y1/(VC*TGO)
X4D=-Y1
X5D=-5.*X5/TAU+5.*X6*XNP*VC/TAU
X6D=-5.*X6/TAU+5.*X7/TAU
X7D=-5.*X7/TAU+5.*X8/TAU
X8D=-5.*X8/TAU-X2
X9D=X1
X10D=X9
IF(STEP-1)66,66,55
999 CONTINUE
PAUSE
CLOSE(1)
END

```

To illustrate the use of the normalized miss distance results of Fig. 6.6, let us consider a numerical example. If the guidance time constant is 0.5 s, and the time of flight is 2.5 s, then the normalized flight time is 5, or

$$t_F/T = 2.5/0.5 = 5$$

For an effective navigation ratio of 3, the normalized miss can be read from Fig. 6.6 as 12, or

$$\frac{\text{Miss}}{T^2 n_T} = 12$$

To compute the actual miss distance in this example, we must assume a target maneuver level n_T . With a 4-g maneuver level the actual miss distance becomes

$$\text{Miss} = 12T^2 n_T = 12 * 0.5^2 * 4 = 12 \text{ ft}$$

Increasing the guidance system time constant can substantially influence the miss distance. For example, if we increase the guidance system time constant from 0.5 s to 1 s, the normalized flight time becomes

$$t_F/T = 2.5/1 = 2.5$$

Keeping the effective navigation ratio to 3 yields a new normalized miss of approximately 42, or

$$\frac{\text{Miss}}{T^2 n_T} = 42$$

which means that for a 4-g maneuver the actual miss is

$$\text{Miss} = 42 T^2 n_T = 42 * 1^2 * 4 = 168 \text{ ft}$$

In other words, for this case doubling the guidance system time constant increased the miss distance by more than an order of magnitude!

By integrating the miss due to a step target maneuver in the adjoint program, we can also find the normalized miss due to a ramp maneuver. Figure 6.7 presents normalized miss distance results for a ramp maneuver disturbance. Again, the abscissa has the same interpretation as before. Here we can see that the effective navigation ratio must be greater than 3 for the miss to approach zero for long flight times. This means that, if the actual maneuver is a ramp, we need an effective navigation ratio of more than 3 to hit the target. In addition, we can see from Fig. 6.7 that the normalization on the ordinate is different from that in the case of a step target maneuver.

If we had a ramp maneuver that reached the 4-g level in 2.5 s, then its acceleration rate would be

$$r_T = n_T / t_F = 4/2.5 = 1.6 \text{ g/s}$$

With the same inputs as before (i.e., $T = 0.5$ s), we can read the normalized miss

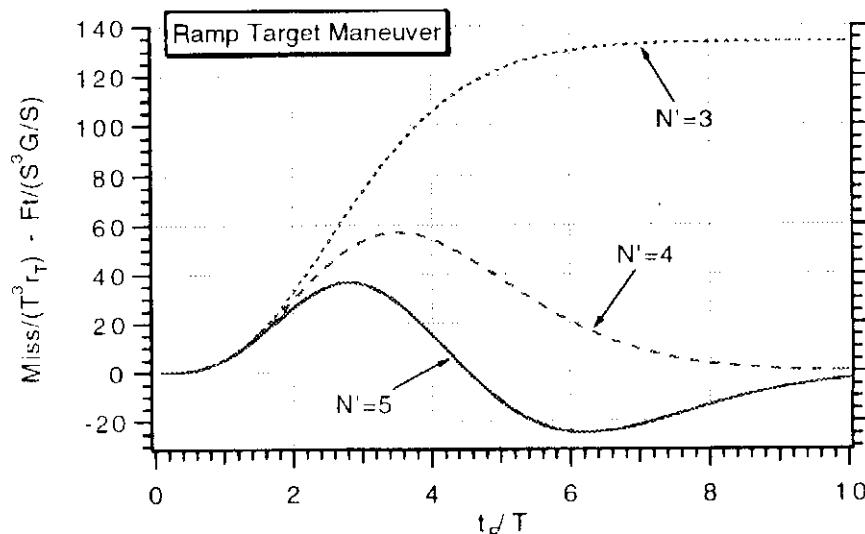


Fig. 6.7 Normalized miss due to ramp target maneuver.

from Fig. 6.7 as

$$\frac{\text{Miss}}{T^3 r_T} = 120$$

Therefore, the actual miss distance is

$$\text{Miss} = 120 T^3 r_T = 120 * 0.5^3 * 1.6 = 24 \text{ ft}$$

We can see that, although the ramp maneuver only reaches the 4-g level by the end of the flight, its influence on miss distance, for this example, is much greater than that of the step maneuver. Increasing the effective navigation ratio to 4 reduces the normalized miss to

$$\frac{\text{Miss}}{T^3 r_T} = 40$$

which reduces the actual miss to

$$\text{Miss} = 40 * 0.5^3 * 1.6 = 8 \text{ ft}$$

This numerical example illustrates the need for larger effective navigation ratios in a proportional navigation guidance system for nonconstant target maneuvers.

Integrating the adjoint ramp maneuver output yields the miss due to a parabolic maneuver. Figure 6.8 presents the normalized miss distance induced by a parabolic target maneuver. Here we can see that an effective navigation ratio of 5 is required for the miss to go to zero for long flight times.

For consistency, let us consider a case in which all numerical values are related to the previous cases considered. If the parabolic maneuver reaches the 4-g level in 2.5 s, the acceleration jerk will be

$$p_T = n_T / t^2 = 4 / 2.5^2 = 0.64 \text{ g/s}^2$$

For the same inputs as before, the miss for a navigation ratio of 3 becomes

$$\text{Miss}|_{N''=3} = 300 T^4 p_T = 300 * 0.5^4 * 0.64 = 12 \text{ ft}$$

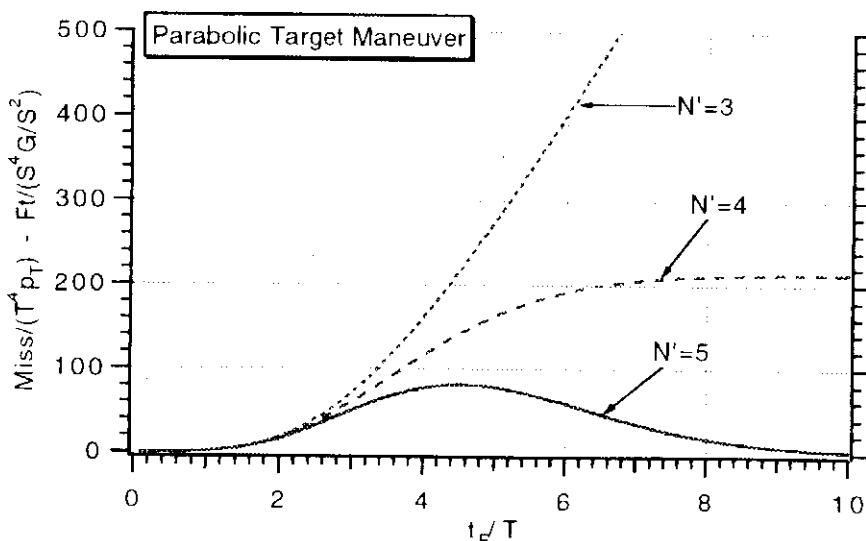


Fig. 6.8 Normalized miss due to parabolic target maneuver.

Increasing the navigation ratio reduces the miss, as we can see from

$$\text{Miss}|_{N'=4} = 150 * 0.5^4 * 0.64 = 6 \text{ ft}$$

$$\text{Miss}|_{N'=5} = 75 * 0.5^4 * 0.64 = 3 \text{ ft}$$

In summary, we can say that increasing the effective navigation ratio and decreasing the guidance system time constant work in the direction of reducing the miss due to target maneuver. We shall see later that increasing the navigation ratio also increases the miss due to noise and parasitic effects. Variable maneuver levels such as ramps and parabolas may cause more miss than constant maneuvers because the navigation ratio may not be set at a high enough level.

Optimal Target Evasive Maneuvers²⁻⁶

We have seen that the effective navigation ratio has a strong influence on missile guidance system performance against all types of target maneuvers. Let us consider the influence of step target maneuver on a fifth-order binomial guidance system in more detail. From a target's point of view an optimal maneuver is one that induces the most miss distance. Figure 6.9 shows that, when the effective navigation ratio is 3, the normalized miss distance curve has a maxima at a normalized flight time of 2.5. Since flight time and time to go are interchangeable for this system, we can interpret the abscissa of Fig. 6.9 as normalized time to go. Therefore, as shown in Fig. 6.9, the target can induce the most miss distance by first executing $+n_T g$ (normalized time to go is large) and then rolling 180 deg at a normalized time to go of 2.5 so that the target will be executing $-n_T g$. As far as the missile is concerned, the target appears to be executing a maneuver of magnitude $2n_T$! The optimality of this maneuver is proven mathematically in Ref. 2 using a combination of optimal control theory and adjoint theory.

We can see from Fig. 6.9 that the induced miss distance caused by this optimal maneuver will be

$$\text{Miss}|_{\text{opt}, N'=3} = 41.4 * (2n_T)^2 T^2 = 82.8 T^2 n_T$$

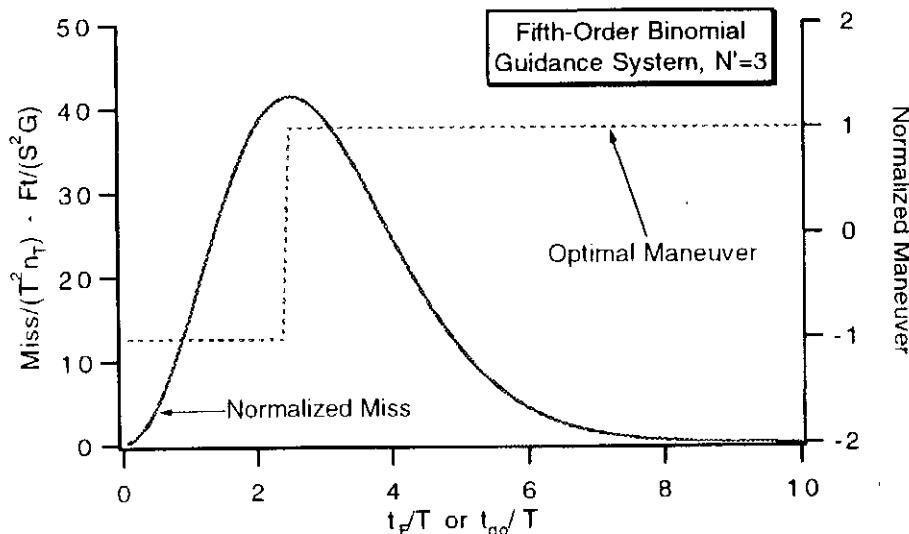


Fig. 6.9 Optimal maneuver policy for effective navigation ratio of 3.

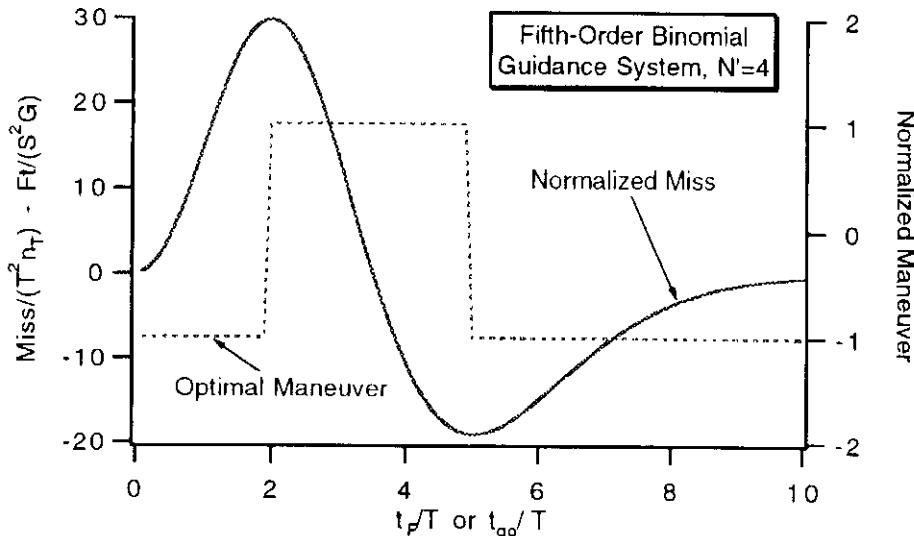


Fig. 6.10 Optimal maneuver policy for effective navigation ratio of 4.

For a maneuver level of 4 g and a guidance system time constant of 0.5 s , the largest miss distance the target maneuver can induce for an effective navigation ratio of 3 is

$$\text{Miss}_{\text{opt}, N'=3} = 82.8 * 0.5^2 * 4 = 82.8 \text{ ft}$$

This miss distance is considerably larger than 12 ft , which was previously obtained with a step target maneuver occurring at a normalized time to go of 5 s .

If the effective navigation ratio is 4, the miss response has maxima indicated in Fig. 6.10. Therefore, in this case, the optimal maneuver policy is for the target to begin with a maneuver level of $-n_T \text{ g}$ until $t_{\text{go}}/T = 5$, then rolling 180 deg in order to execute $+n_T \text{ g}$ and then finally at $t_{\text{go}}/T = 2$, rolling another 180 deg in order to execute $-n_T \text{ g}$. We can see from Fig. 6.10 that the induced miss in this case will be

$$\text{Miss}_{\text{opt}, N'=4} = (19.1 + 29.8) * (2n_T)T^2 = 97.8T^2n_T$$

If the navigation ratio is 5, the miss response has three maxima, as shown in Fig. 6.11. The optimal maneuver strategy is superimposed on this figure, and the optimal induced miss turns out to be

$$\text{Miss}_{\text{opt}, N'=5} = (8.4 + 29.9 + 24) * (2n_T)T^2 = 124.6T^2n_T$$

It is interesting to note that, as we increase the effective navigation ratio, the optimal miss due to a step target maneuver also increases because of the increased number of maxima in the miss distance sensitivity curve.

The concept of an optimal maneuver is useful in that it identifies the largest possible miss distance that the target can induce and possibly aid in the selection of the missile guidance system time constant. Of course, this optimal maneuver assumes unrealistically that the target has precise knowledge of the time to go until intercept and of the missile guidance system dynamics. It is readily apparent from the preceding miss distance curves that the missile guidance time constant must be minimized if the miss distance due to target maneuver is to be kept small. However, we shall see later that noise and parasitic effect place a practical lower limit on the minimum achievable guidance system time constant.

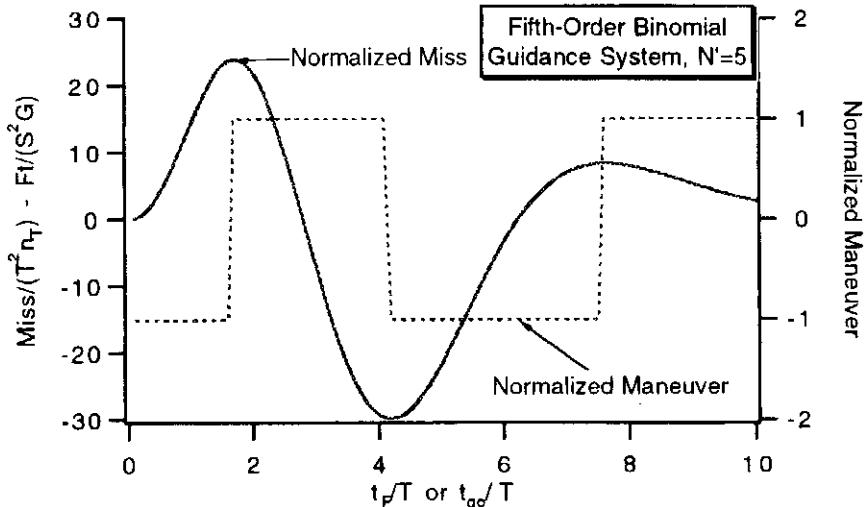


Fig. 6.11 Optimal maneuver policy for effective navigation ratio of 5.

Practical Evasive Maneuvers

In the previous section we showed that optimal target evasive maneuvers could induce very large miss distances if a priori information concerning the missile guidance system was available. In this section we shall demonstrate that when a priori information is lacking, practical periodic target evasive maneuvers can also generate very large miss distances. Two such practical evasive periodic maneuver policies are the barrel roll and the Vertical-S.

The barrel roll can be described in one dimension as a sinusoid with radian frequency ω or period T and amplitude n_T as given by

$$\ddot{y}_T = n_T \sin \omega t = n_T \sin \frac{2\pi}{T} t$$

From the preceding relationship we can see that the barrel roll only yields maximum acceleration levels some of the time. With the Vertical-S maneuver, however, the aircraft is always at maximum acceleration and the sign of the acceleration is periodically reversed by rolling the aircraft through 180 deg. With a theoretically infinite roll rate, this maneuver policy can be approximated by a periodic square wave in one dimension. The barrel roll and Vertical-S maneuver policies do not require information about the missile guidance system. The amplitudes of both target maneuvers are chosen to reflect the maximum acceleration capability of the aircraft, whereas the frequencies of both maneuvers are chosen to be physiologically possible for a human pilot and robust enough to cause any missile guidance system problems.

Both the barrel roll and Vertical-S maneuver policies are illustrated in Fig. 6.12 where, for illustrative purposes, it is assumed that the maneuver amplitude n_T is 4 g and the maneuver frequency ω is 1 rad/s. Figure 6.12 confirms that the effective maneuver period T is 6.28 s.

For comparative purposes, the preceding maneuver policies were evaluated on the fifth-order proportional navigation binomial guidance system. The guidance system time constant was 0.5 s, the target maneuver amplitude was 4 g, and the effective navigation ratio considered was three so that comparisons could be made with the optimal evasive maneuver induced miss distances. With these numbers

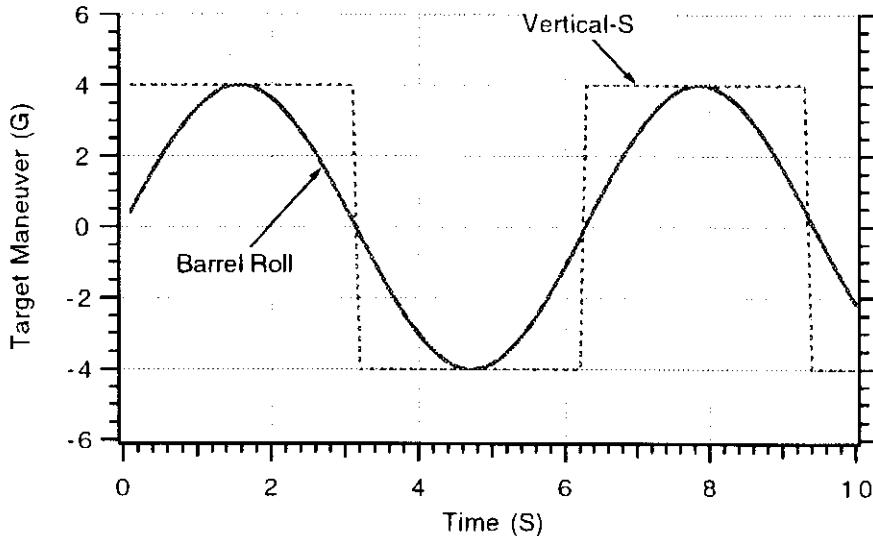


Fig. 6.12 More realistic evasive maneuver policies.

it was shown in the previous section that the optimal miss distance was 82.8 ft. Figure 6.13 shows how the miss distances vary with flight time for both the barrel roll and Vertical-S maneuver policies. It is apparent that the Vertical-S maneuver generates the largest miss distances because the target is always at maximum acceleration. We can also see from Fig. 6.13 that, on the average, the miss distances for both maneuver policies are quite high. For this example the Vertical-S maneuver yields miss distances which approach that of the optimal maneuver when the flight times are 0.5, 1.8, 4.4, or 7.5 s. Of course, if the pilot is not lucky and the flight time is 1.0, 3.4, 6.5, or 9.5 s, the miss distance will be quite small. In these cases the aircraft is hitting the missile rather than the missile hitting the target!

More details on the influence of practical evasive maneuver strategies on miss distance can be found in Ref. 5. In addition, Ref. 5 derives the shaping filter equivalent for many practical maneuvers, so that the method of adjoints can be used to assess system performance in a single computer run.

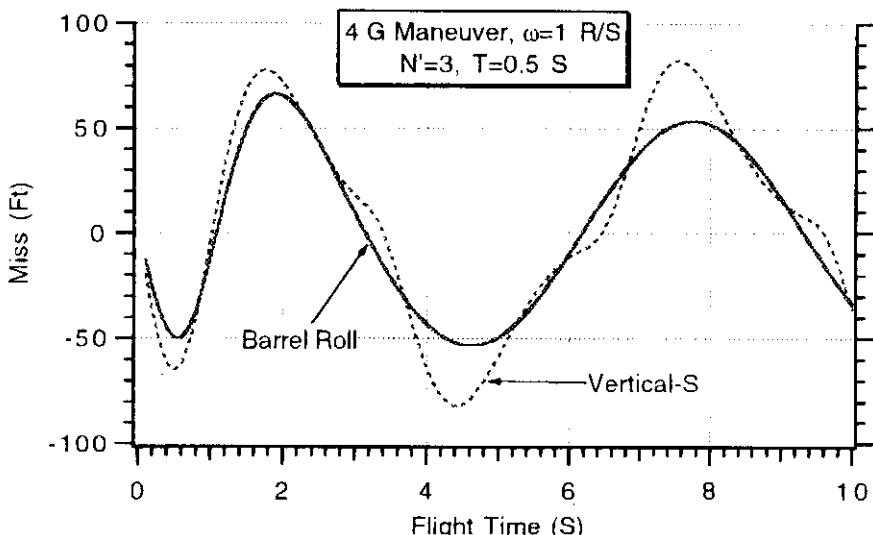


Fig. 6.13 Realistic maneuvers can induce very large miss distances.

Saturation²

Thus far we have seen normalized miss distance curves for a fifth-order proportional navigation binomial guidance system. The results presented have implicitly assumed that the missile had adequate acceleration capability in order to guide and hit the target. If adequate acceleration capability is not available, the missile acceleration saturates, which results in additional miss distance. In endoatmospheric interceptors, angle-of-attack constraints limit maximum achievable accelerations at high altitudes, whereas the missile structure limits achievable acceleration levels at the lower altitudes. The lateral engine thrust-to-weight ratio limits the acceleration level in exoatmospheric interceptors.

The basic homing loop can be modified and made nonlinear to account for acceleration saturation as shown in Fig. 6.14. In this figure the guidance system is also represented as a fifth-order binomial guidance system with guidance time constant T . Two of the time constants are devoted to the seeker and noise filter, and the other three time constants are devoted to the flight-control system. The acceleration limit is on the acceleration command n_c , and the resultant acceleration command is denoted n'_c .

The first question that must be answered again is whether or not linearizing the geometry in the presence of the nonlinear acceleration saturation model is adequate for capturing important miss distance effects. For convenience, let us define the acceleration ratio as the ratio between the missile acceleration limit to the maneuver level of the target. Figure 6.14 will only be linear if the acceleration ratio is infinity.

Figure 6.15 represents the results of running both the nonlinear engagement simulation, with saturation effects modeled, and an engagement simulation of the model shown in Fig. 6.14. In the case considered, the guidance time constant was 1 s, the effective navigation ratio was 4, and the level of the target maneuver was 3 g. The acceleration ratio considered was 3. This means that since the target maneuver level is 3 g, the effective acceleration limit of the missile is 9 g. We can see from Fig. 6.15 that the miss distance results for both the linearized geometry model and nonlinear geometry models are virtually identical. Therefore, we can conclude that the linearized geometry model is adequate for investigating saturation effects. By comparing Fig. 6.15 with the nonsaturation case of Fig. 6.6, we can also conclude

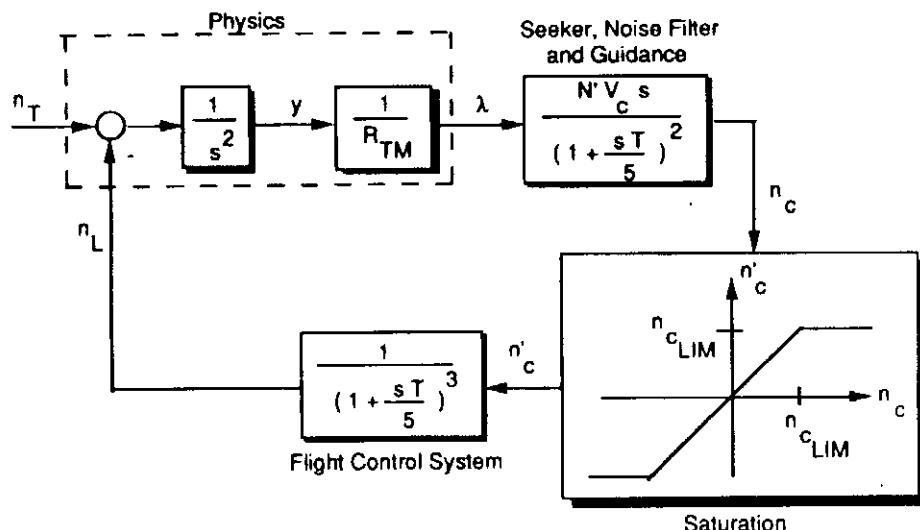


Fig. 6.14 Homing loop with acceleration saturation.

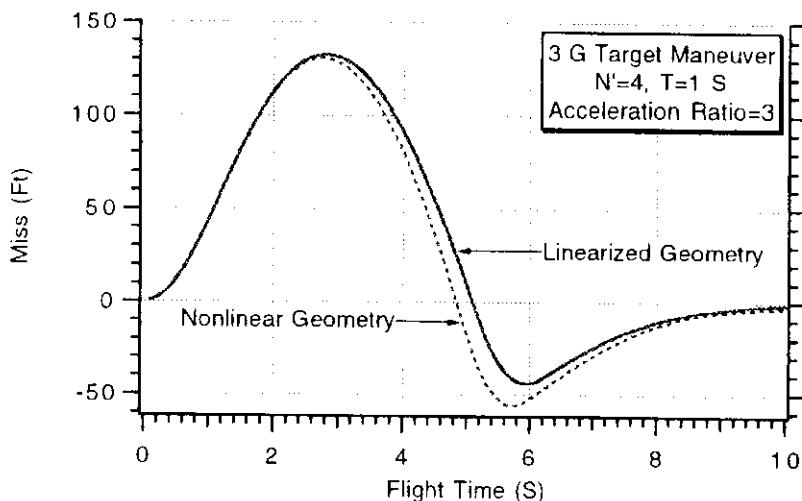


Fig. 6.15 Linearized geometry model is adequate for investigating saturation effects.

that even with a missile-to-target acceleration advantage of 3, considerable miss distance is contributed by saturation, especially for the shorter flight times.

Using the missile to target acceleration capability (n_{CLIM}/n_T) and the normalization factors for miss due to a constant target maneuver, we can derive normalized miss distance curves by the method of brute force (running engagement model with nonlinear saturation effect for many different flight times and noting the miss distance). In other words, we can generate normalized design curves by simulating all of the possibilities. We can then infer performance by making extrapolations from the normalized design curves. Figure 6.16 presents the normalized miss distance due to a step target maneuver when the effective navigation ratio is 3. This figure shows that miss distance always increases with increasing flight time if the acceleration ratio is only 2. For acceleration ratios of 4 or more, the miss is virtually 0 for flight times approximately 10 times greater than the guidance time constant. An acceleration ratio of about 5 closely follows the infinite ratio or linear case.

Let us do a numerical example in order to clarify the use of these curves. If the guidance time constant is 0.5 s and the flight time is 2.5 s, we get a normalized

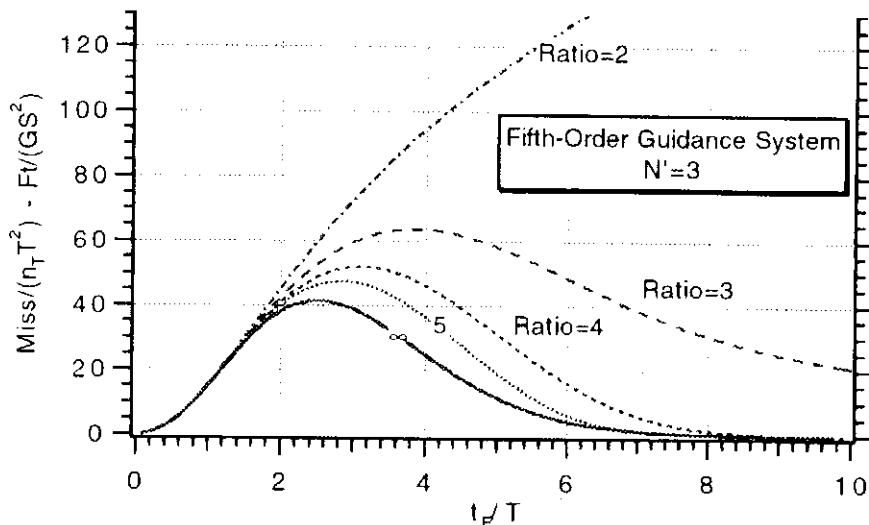


Fig. 6.16 Normalized miss due to target maneuver with saturation effects ($N' = 3$).

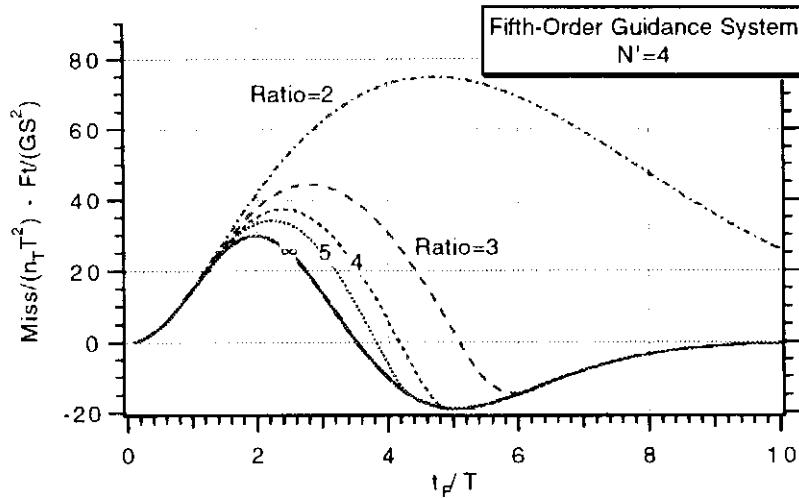


Fig. 6.17 Normalized miss due to target maneuver with saturation effects ($N' = 4$).

flight time of

$$t_F/T = 2.5/0.5 = 5$$

Assuming a target maneuver level of $4 g$, we can then calculate the miss distances for various acceleration limits as

$$\text{Miss}|_{\infty g} = 12.0T^2n_T = 12.0 * 0.5^2 * 4 = 12.0 \text{ ft}$$

$$\text{Miss}|_{20g} = 18.9T^2n_T = 18.9 * 0.5^2 * 4 = 18.9 \text{ ft}$$

$$\text{Miss}|_{16g} = 31.1T^2n_T = 31.1 * 0.5^2 * 4 = 31.1 \text{ ft}$$

$$\text{Miss}|_{12g} = 58.1T^2n_T = 58.1 * 0.5^2 * 4 = 58.1 \text{ ft}$$

$$\text{Miss}|_8g = 112T^2n_T = 112 * 0.5^2 * 4 = 112 \text{ ft}$$

Therefore, we can see that the miss goes up by nearly a factor of 5 from the linear case if the missile-to-target acceleration advantage is only 3 and by a factor of 10 if the acceleration advantage is only 2.

Increasing the effective navigation ratio tends to reduce the acceleration requirements as shown in the normalized curves of Figs. 6.17 and 6.18.

Parasitic Effects⁷⁻¹⁰

Thus far, from all of the results presented it would appear that the guidance system designer has an easy job, since all the graphs indicate that smaller time constants and larger effective navigation ratios appear to improve system performance. Actually, parasitic or unwanted feedback paths within the homing loop will work in the direction of larger time constants and smaller effective navigation ratios to get acceptable performance. One of the most serious unwanted feedback paths is created in tactical radar homing missile applications by the missile radome. The radome causes a refraction or bending of the incoming radar wave, which in turn gives a false indication of the target location. A parameter associated with missile maneuverability, which has a significant interaction with radome effects, is the turning rate time constant T_α . If we consider the basic geometry of Fig. 6.19,

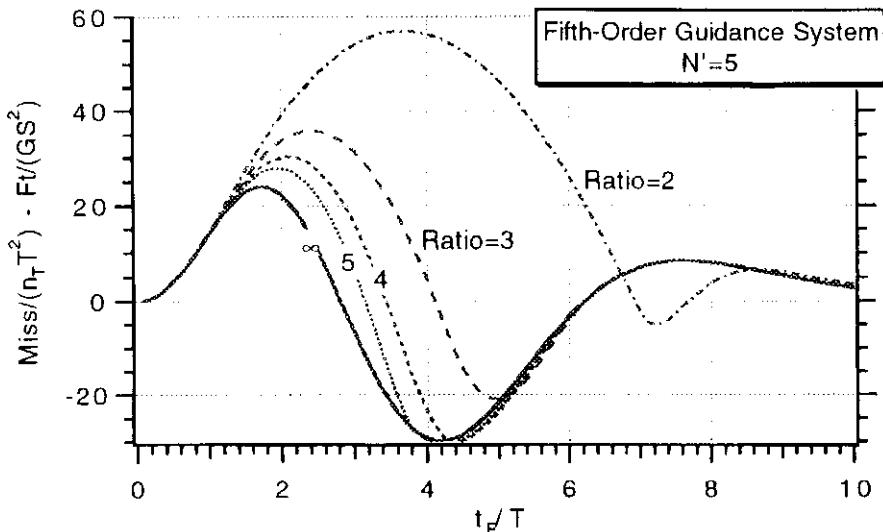


Fig. 6.18 Normalized miss due to target maneuver with saturation effects ($N' = 5$).

the turning rate time constant can be defined as the amount of time it takes to turn the missile flight-path angle γ through an equivalent angle of attack α , or

$$T_\alpha = \frac{\alpha}{\dot{\gamma}}$$

where the angle of attack and the flight-path angle are defined in Fig. 6.19. Generally the turning rate time constant increases with increasing missile altitude and decreasing missile velocity.

To see how the turning rate time constant enters into the homing loop, we must see how it is related to other important quantities. From Fig. 6.19 we can see that the missile pitch angle θ can be expressed as

$$\theta = \gamma + \alpha$$

Taking derivatives of both sides of the equation yields

$$\dot{\theta} = \dot{\gamma} + \dot{\alpha} = \dot{\gamma} + \frac{s\alpha\dot{\gamma}}{\dot{\gamma}}$$

Since the missile acceleration is perpendicular to the missile velocity, we can say that

$$n_L = V_M \dot{\gamma}$$

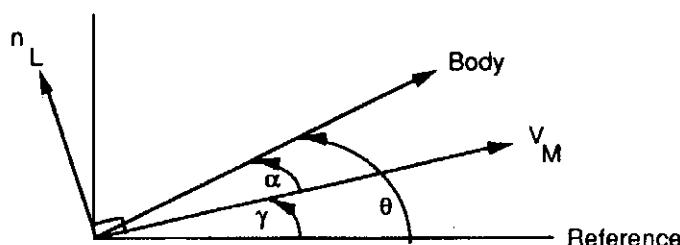


Fig. 6.19 Geometry for turning rate time constant.

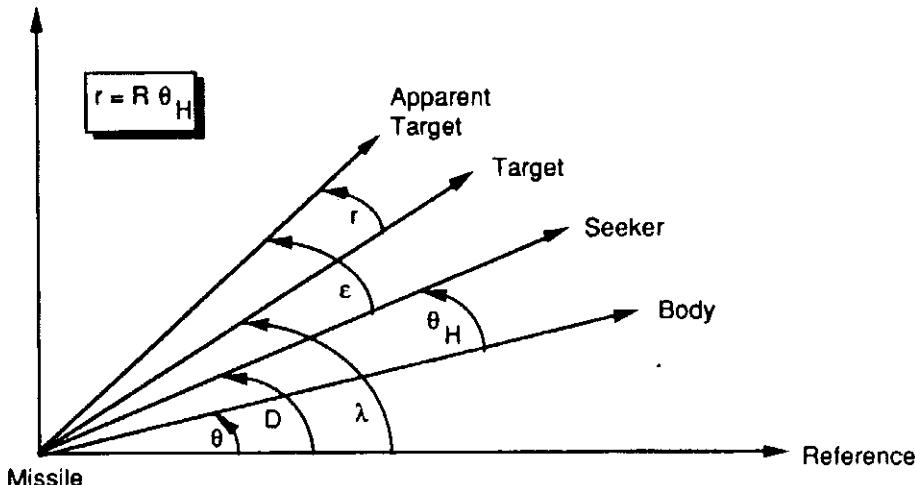


Fig. 6.20 Basic geometry for radome analysis.

Therefore, we can express the missile pitch rate in terms of the missile acceleration as

$$\dot{\theta} = \frac{n_L}{V_M} + s T_\alpha \frac{n_L}{V_M}$$

Dividing both sides by the missile acceleration yields the missile pitch rate transfer function,

$$\frac{\dot{\theta}}{n_L} = \frac{1}{V_M} (1 + T_\alpha S)$$

This aerodynamic transfer function shows that there is a missile body rate whenever the missile is accelerating.

Now we need to see how the missile aerodynamic transfer function interacts with the radome slope. Consider the basic geometry of Fig. 6.20 in which the seeker is not pointed at the actual target because of seeker dynamics and radome effects.

The radome refraction angle r varies with the missile gimbal angle θ_H . For preliminary analysis it is usually assumed that the refraction angle is linearly proportional to the gimbal angle, or

$$r = R\theta_H$$

where R is constant known as the radome slope (see Appendix A for alternative definition). The radome slope is a function of the radome material, radome diameter, and fineness ratio, and the wavelength of the incoming signal. From Fig. 6.20 we can see that it is possible to express the missile boresight error ϵ as

$$\epsilon = \lambda - \theta - \theta_H + r = \lambda - \theta - \theta_H + R\theta_H$$

A block diagram of the homing loop with the radome unwanted feedback path is indicated in Fig. 6.21. We can see that without radome effects ($R = 0$) we would have a fifth-order binomial guidance system transfer function. The missile aerodynamic transfer function¹¹ provides the unwanted feedback path in the guidance system.

Listing 6.2 presents an engagement simulation with the fifth-order binomial model, including radome effects, of Fig. 6.21. The simulation is set to run multiple

Listing 6.2 Engagement simulation with radome effects

```

OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
VC=4000.
XNT=32.2
YIC=0.
VM=3000.
HEDEG=0.
TAU=.5
XNP=3.
TA=0.
R=-.01
DO 20 TF=.1,10.,.1
Y=YIC
YD=-VM*HEDEG/57.3
YDIC=YD
XNL=0.
ELAMDH=0.
X4=0.
X5=0.
TH=0.
THH=0.
T=0.
H=.01
S=0.
10 IF(T>(TF-.0001))GOTO 999
YOLD=Y
YDOLD=YD
XNLOLD=XNL
ELAMDHOLD=ELAMDH
X4OLD=X4
X5OLD=X5
THOLD=TH
THHOLD=THH
STEP=1
GOTO 200
66 STEP=2
Y=Y+H*YD
YD=YD+H*YDD
XNL=XNL+H*XNLD
ELAMDH=ELAMDH+H*ELAMDHD
X4=X4+H*X4D
X5=X5+H*X5D
TH=TH+H*THD
THH=THH+H*THHD
T=T+H
GOTO 200
55 CONTINUE
Y=.5*(YOLD+Y+H*YD)
YD=.5*(YDOLD+YD+H*YDD)
XNL=.5*(XNLOLD+XNL+H*XNLD)
ELAMDH=.5*(ELAMDHOLD+ELAMDH+H*ELAMDHD)

```

(Contd.)

Listing 6.2 (Continued)

```

X4=.5*(X4OLD+X4+H*X4D)
X5=.5*(X5OLD+X5+H*X5D)
TH=.5*(THOLD+TH+H*THD)
THH=.5*(THHOLD+THH+H*THHD)
S=S+H
GOTO 10
200 CONTINUE
TGO=TF-T+.00001
XLAM=Y/(VC*TGO)
EPS=XLAM-TH-THH+R*THH
DD=5.*EPS/TAU
ELAMDHD=5.*(DD-ELAMDH)/TAU
XNC=XNP*VC*ELAMDH
X4D=5.*(XNC-X4)/TAU
X5D=5.*(X4-X5)/TAU
XNLD=5.*(X5-XNL)/TAU
THD=XNL/VM+TA*XNLD/VM
THHD=DD-THD
YDD=XNT-XNL
IF(STEP-1)66,66,55
999 CONTINUE
WRITE(*,97)TF,Y
WRITE(1,97)TF,Y
20 CONTINUE
97 FORMAT(2F10.3)
PAUSE
CLOSE(1)
END

```

cases with the flight time as a parameter so that miss distance sensitivity curves can be generated by brute force. Again, the differential equations representing the guidance system of Fig. 6.21 appear after statement label 200.

To see how the turning rate time constant influences system performance, a case was run for a 1-g target maneuver disturbance in which the guidance system time constant was 0.5 s, the radome slope was -0.01, and the effective navigation ratio was 3. The turning rate time constant was varied from 0 to 10 s. Figure 6.22 shows that when the turning rate time constant is zero the miss distance response is virtually identical to the case in which there are no parasitic paths in the homing loop (compare with Fig. 6.6, for example). When the turning rate time constant is increased to 5 s, the miss distance response begins to become more oscillatory, but the miss distances are still small and tend to zero as the flight time increases. Finally, when the turning rate time constant is increased to 10 s, the miss distance response becomes unstable. Thus, we can see that we have to be concerned about radome effects from both a miss distance and stability point of view.

As was mentioned previously, the magnitude of the effective radome slope is determined by the physical characteristics of the radome and the wavelength of the incoming signal. For a given radome, the guidance designer has only two parameters (i.e., guidance system time constant and effective navigation ratio) under control to get acceptable performance and meet stability requirements. Figure 6.23

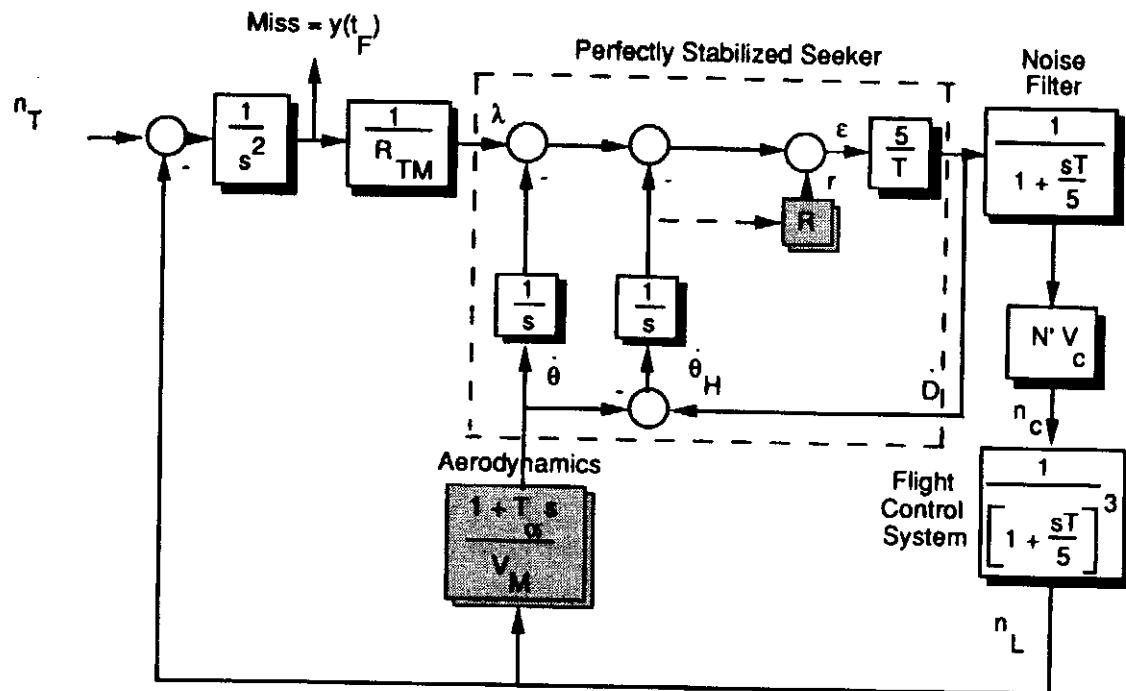


Fig. 6.21 Fifth-order binomial model of guidance system with radome effects.

shows how miss distance due to a 1-g target maneuver varies with flight time for two different values of effective navigation ratio in the presence of a negative radome slope ($R = -0.01$). We can see that the higher effective navigation ratio has a destabilizing effect. This is not unreasonable since we are essentially increasing the guidance system gain. Thus, the guidance system designer desires to keep the effective navigation ratio as small as possible to meet the stability requirements and yet large enough so that homing will be effective.

Figure 6.24 shows that, in the presence of a large effective navigation ratio and negative radome slope, increasing the guidance system time constant from 0.5 s to 0.75 s has a stabilizing effect.

In tactical missile design the guidance system time constant is generally made larger at the higher altitudes because the turning rate time constant is largest at the

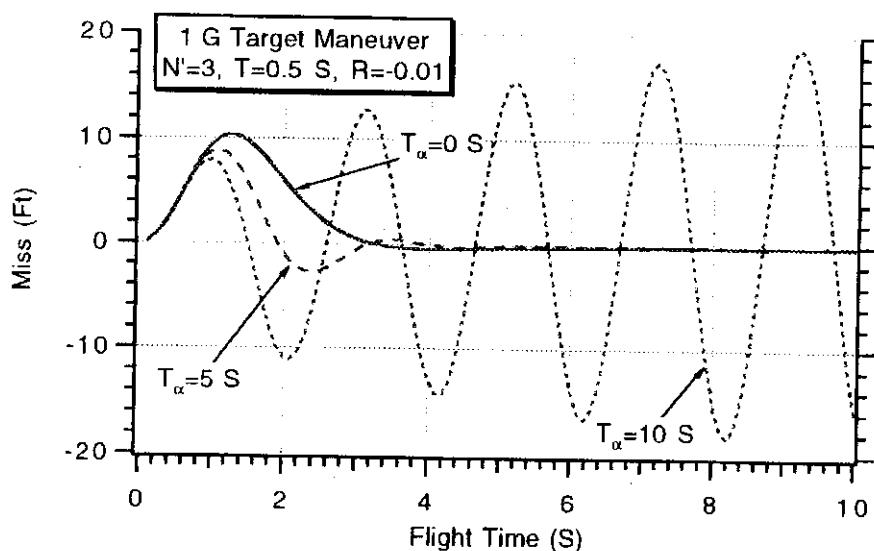


Fig. 6.22 Miss degrades with increasing turning rate time constant.

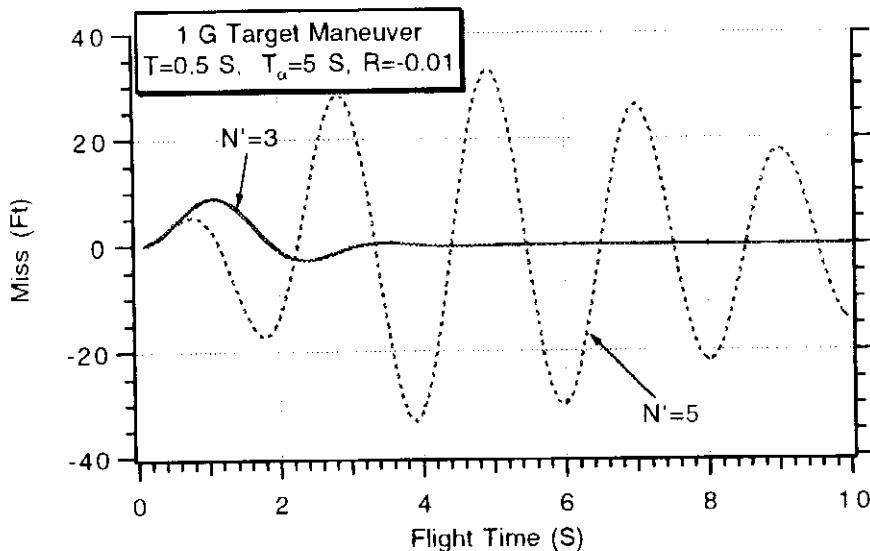


Fig. 6.23 Increasing effective navigation ratio has destabilizing effect in presence of negative radome slope.

higher altitudes. Of course, the penalty for such a decision is that miss distances tend to increase with increasing guidance system time constant. Therefore, the guidance system designer attempts to make the guidance system time constant as small as possible subject to meeting guidance system stability requirements.

Thrust Vector Control

We saw in the previous section that, if the turning rate time constant of a tactical aerodynamic missile was large, radome effects caused stability problems and miss distance deterioration. This problem is not confined to only tactical aerodynamic missiles. Consider a missile that operates outside the atmosphere and uses thrust vector control to maneuver. Figure 6.25 presents a diagram of a thrust vector controlled missile with all important angles indicated. The missile acceleration n_L

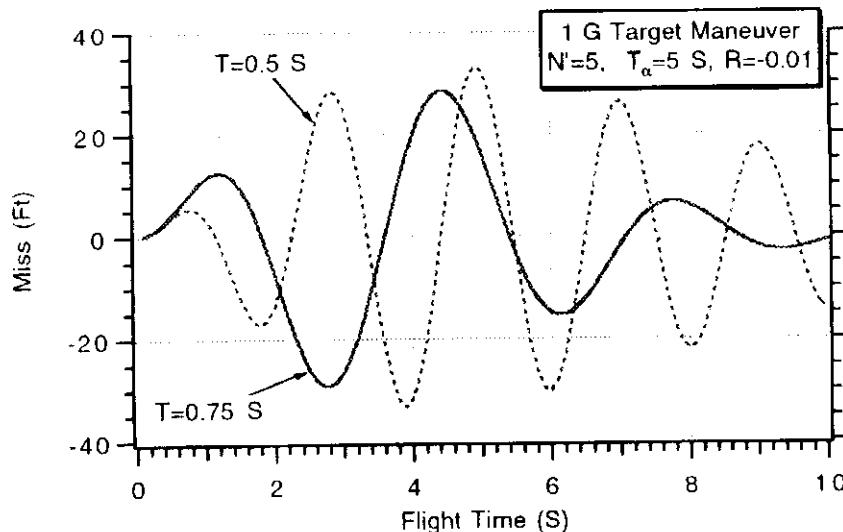


Fig. 6.24 Increasing guidance system time constant has stabilizing effect in presence of negative radome slope.

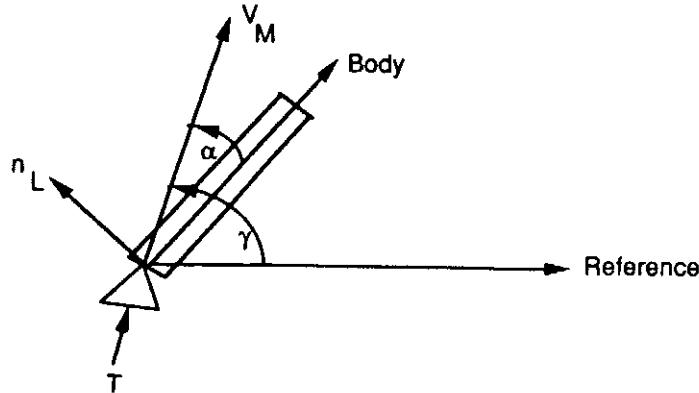


Fig. 6.25 Important angles in thrust vector control.

needed to maneuver in accordance with guidance commands is obtained from the component of the thrust T perpendicular to the missile body.

For simplicity we are neglecting the fact that, if the thrust does not go through the center of gravity, the missile will tumble. The rate of change of the missile flight-path angle γ is related to the missile acceleration and velocity according to

$$\dot{\gamma} = \frac{n_L}{V_M}$$

where n_L is the missile acceleration and V_M the missile velocity. From Fig. 6.25 we can see that the flight-path rate can also be expressed as

$$\dot{\gamma} = \frac{n_L}{V_M} = \frac{Tg \sin \alpha}{WV_M}$$

where T is thrust (in lb), g is the gravitational acceleration (in ft/s²), α is the angle of attack, and W is the missile weight. For small angles of attack we can approximate the flight-path rate to be

$$\dot{\gamma} = \frac{Tg\alpha}{WV_M}$$

Recalling that the turning rate time constant is the ratio of the angle of attack to the flight-path rate, we obtain

$$T_\alpha = \frac{\alpha}{\dot{\gamma}} = \frac{WV_M}{Tg}$$

This means that the effective turning rate time constant for a thrust vector controlled missile is proportional to the missile weight and velocity and inversely proportional to the thrust.

To illustrate the importance of turning rate time constant to a thrust vector controlled missile, let us work a numerical example. Consider a missile traveling at 20,000 ft/s and requiring a 5-deg angle of attack in order to generate 5 g of acceleration. To generate 5 g of acceleration at 5-deg angle of attack, the missile must have a thrust-to-weight ratio given by

$$\frac{T}{W} = \frac{n_L/g}{\alpha} = \frac{5}{5/57.3} = 57.3$$

This means that the effective turning rate time constant is

$$T_\alpha = \frac{V_M}{(T/W)g} = \frac{20,000}{57.3 * 32.2} = 10.8 \text{ s}$$

The turning rate time constant in this example is quite large compared to values indicated in the previous section. However, since the thrust vector controlled missile operates outside the atmosphere, the shape of the missile nose can be made near-hemispherical. This means that the effective radome slope will be close to zero. For a thrust vector controlled missile the guidance system designer must pay close attention to the product of the radome slope and turning rate time constant to ensure adequate stability margins in the resultant design. If the design yields unacceptable stability margins, the guidance system time constant must be increased to yield a workable design.

Summary

In this chapter we have shown how system order, optimal target maneuvers, saturation, and parasitic effects all influence miss distance. Miss distance design curves were presented to aid the guidance system designer in predicting preliminary system performance. These curves could also be used to ensure that the interceptor had adequate acceleration capability. Examples were presented showing the conflicting tradeoffs the guidance system designer must confront in choosing acceptable guidance system parameters.

References

- ¹Nesline, F. W., and Zarchan, P., "Miss Distance Dynamics in Homing Missiles," *Proceedings of AIAA Guidance and Control Conference*, AIAA, New York, Aug. 1984.
- ²Shinar, J., and Steinberg, D., "Analysis of Optimal Evasive Maneuvers Based On a Linearized Two-Dimensional Model," *Journal of Aircraft*, Vol. 14, Aug. 1977, pp. 795-802.
- ³Bennett, R. R., and Mathews, W. E., "Analytical Determination of Miss Distance for Linear Homing Navigation Systems," Hughes Aircraft Co., TM-260, Culver City, CA, March 1952.
- ⁴Travers, P., *Interceptor Dynamics*, unpublished lecture notes, Raytheon Co., circa 1971.
- ⁵Zarchan, P., "Representation of Realistic Evasive Maneuvers by the Use of Shaping Filters," *Journal of Guidance and Control*, Vol. 2, July-Aug. 1979, pp. 290-295.
- ⁶Howe, R. M., "Guidance," *System Engineering Handbook*, edited by R. E. Machol, W. P. Tanner Jr., and S. N. Alexander, McGraw-Hill, New York, 1965, Chap. 19.
- ⁷Nesline, F. W., and Zarchan, P., "Radome Induced Miss Distance in Aerodynamically Controlled Homing Missiles," *Proceedings of AIAA Guidance and Control Conference*, AIAA, New York, Aug. 1984.
- ⁸Youngren, F. R., "Minimizing Boresight Errors in Aerodynamic Radomes," *Electronic Design*, Dec. 20, 1961, pp. 152-157.
- ⁹Peterson, E. L., *Statistical Analysis and Optimization of Systems*, Wiley, New York, 1961
- ¹⁰Eichblatt, E. (ed.), *Test and Evaluation of the Tactical Missile*, Vol. 119, Progress in Astronautics and Aeronautics, AIAA, Washington, DC, 1989, p. 415.
- ¹¹Hemsch, M. J., and Nielsen, J. N. (eds.), *Tactical Missile Aerodynamics*, Vol. 104, Progress in Astronautics and Aeronautics, AIAA, New York, 1986, p. 858.

Digital Fading Memory Noise Filters in the Homing Loop

Introduction

THUS far, we have assumed in our analysis that the geometric line-of-sight rate was available for guidance purposes. Actually, the seeker measurement of the line-of-sight angle is corrupted by noise. Therefore, in order to derive the line-of-sight rate estimate required by proportional navigation guidance, it is necessary to use a digital noise filter in an onboard guidance system. Although we shall later study optimal digital noise filters, we shall first consider simple constant gain filters, known as fading memory filters, to derive the line-of-sight rate estimate. We will investigate, by example, some of the properties of digital fading memory filters and their influence on system performance. Fading memory filters will serve as the foundation for more advanced digital filters, known as Kalman filters.

Fading Memory Filters¹

A simple digital noise filter is known as a fading memory filter. This filter is recursive and weights new measurements more heavily than older measurements. First-, second-, and third-order fading memory filters and their gains are tabulated in recursive form in Table 7.1. We can see from the table that the filter estimate is essentially the old estimate plus a gain times a residual (difference between current measurement and previous estimate). Table 7.1 also shows that the fading memory filter gains are constant and are a function of only one parameter β . This parameter is associated with the memory length of the filter and is a constant between zero and unity. Increasing β tends to decrease the bandwidth of the filter and enables the filter to "remember" more about previous measurements.

We can see from Table 7.1 that the fading memory filter assumes a polynomial model for the actual process. If the polynomial process of the filter is an underestimate of the polynomial degree of the actual process, then there will be a filter truncation error. The lowest order filter does the best job of removing the noise from the signal. However, it also has the potential for having the most truncation error. The filter designer must select the appropriate filter order to trade off filter variance reduction vs truncation error buildup. Fading memory filters are quite popular in radar tracking applications^{2,3} but, as we shall see, can also be made to work in tactical missile homing applications.

Table 7.1 Different order digital fading memory filters

Filter	Gains
$\hat{x}_n = \hat{x}_{n-1} + G[x_n^* - \hat{x}_{n-1}]$	$G = 1 - \beta$
$\hat{x}_n = \hat{x}_{n-1} + \hat{x}_{n-1}T_s + G[x_n^* - (\hat{x}_{n-1} + \hat{x}_{n-1}T_s)]$	$G = 1 - \beta^2$
$\hat{x}_n = \hat{x}_{n-1} + \frac{H}{T_s}[x_n^* - (\hat{x}_{n-1} + \hat{x}_{n-1}T_s)]$	$H = (1 - \beta)^2$
$\hat{x}_n = \hat{x}_{n-1} + \hat{x}_{n-1}T_s + 0.5\hat{x}_{n-1}T_s^2$ + $G[x_n^* - (\hat{x}_{n-1} + \hat{x}_{n-1}T_s + 0.5\hat{x}_{n-1}T_s^2)]$	$G = 1 - \beta^3$
$\hat{x}_n = \hat{x}_{n-1} + \hat{x}_{n-1}T_s + \frac{H}{T_s}[x_n^* - (\hat{x}_{n-1} + \hat{x}_{n-1}T_s$ + $0.5\hat{x}_{n-1}T_s^2)]$	$H = 1.5(1 - \beta)^2(1 + \beta)$
$\hat{x}_n = \hat{x}_{n-1} + \frac{2K}{T_s^2}[x_n^* - (\hat{x}_{n-1} + \hat{x}_{n-1}T_s + 0.5\hat{x}_{n-1}T_s^2)]$	$K = 0.5(1 - \beta)^3$

Fading Memory Filter in Homing Loop

Figure 7.1 shows an example of how a second-order fading memory filter can be included in the homing loop. In this loop the actual line-of-sight angle λ is sampled with noise added every T_s seconds, thus providing an idealized seeker model. Estimates of the line-of-sight angle and rate are made with a digital two-state fading memory filter's measurement of the noisy line-of-sight angle λ_k^* . As was mentioned in Chapter 1, the notation z^{-1} is Z transform notation for a pure delay of T_s seconds. A guidance command is generated, using the proportional navigation

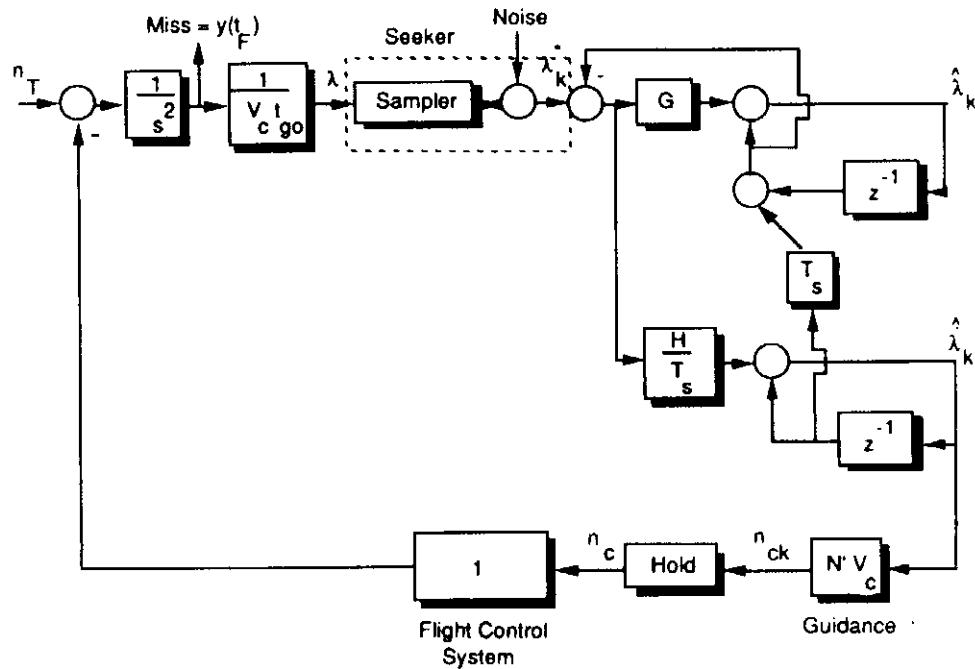


Fig. 7.1 Second-order fading memory filter in homing loop.

guidance law from the estimated line-of-sight rate. The resultant command is passed through a "hold" network that converts the digital signal to a continuous signal for the flight-control system. The diagram shows a unity gain for an idealized representation of the flight-control system.

Listing 7.1 is a FORTRAN engagement simulation of the homing loop shown in Fig. 7.1. Zero-mean Gaussian noise, independent from sample to sample, with standard deviation, SIGNOISE, is added to the measured line-of-sight angle every T_s seconds. We can see from the listing that the program consists of two separate parts. The first part, which represents the real world, consists of differential equations and the second-order Runge-Kutta numerical integration technique, and the second part, which represents an onboard guidance system, has the difference equations for the second-order digital fading memory filter. We solve the differential equations every H seconds, and the difference equations are solved every T_s seconds. It is important to note that the ratio T_s/H must be a large integer so that effects in between sampling instants are treated properly and accurately. Note that the Gaussian noise generator consists of the sum of six uniform noise generators as was discussed in Chapter 4.

The engagement simulation was exercised and a nominal case was run in which β of the fading memory filter was set to 0.8. Figure 7.2 compares the actual line-of-sight rate to the filter estimate of the derivative of the measurement. We can see that the filter estimate of the line-of-sight rate is smooth but lags the actual line-of-sight rate, indicating that the filter is sluggish.

Figure 7.3 indicates that we can effectively increase the bandwidth of the fading memory filter by decreasing β . Here we can see that the line-of-sight rate estimate no longer lags the actual signal when β is reduced from 0.8 to 0.3. However, we can see from the figure that the noisiness of the line-of-sight rate estimate is the price paid for reducing β . In other words, decreasing β increases the fading memory filter's noise transmission.

The results presented thus far are for a single flight with a particular noise stream. Answers will change for another flight with a different noise stream. To get accurate performance projection in terms of miss distance, we must run the program in the Monte Carlo mode. That is, repeated simulation trials must be conducted

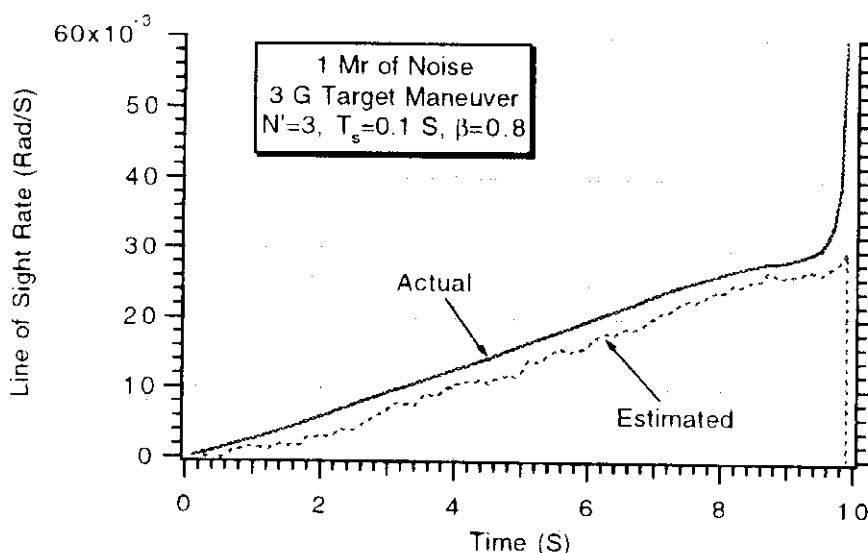


Fig. 7.2 Filter is sluggish and lags signal when $\beta = 0.8$.

Listing 7.1 Engagement simulation with second-order fading memory filter

```

GLOBAL DEFINE
    INCLUDE 'quickdraw.inc'
END
INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
VC=4000.
XNT=96.6
YIC=0.
VM=3000.
HEDEG=0.
BETA=.8
XNP=3.
SIGNOISE=.001
TF=10.
TS=.1
NOISE=1
Y=YIC
YD=-VM*HEDEG/57.3
YDIC=YD
T=0.
H=.01
S=0.
GFILTER=1.-BETA**2
HFILTER=(1.-BETA)**2
XLAMH=0.
XLAMDH=0.
XNC=0.
10 IF(T>(TF-.0001))GOTO 999
    YOLD=Y
    YDOLD=YD
    STEP=1
    GOTO 200
66 STEP=2
    Y=Y+H*YD
    YD=YD+H*YDD
    T=T+H
    GOTO 200
55 CONTINUE
    Y=.5*(YOLD+Y+H*YD)
    YD=.5*(YDOLD+YD+H*YDD)
    S=S+H
    IF(S<(TS-.0001))GOTO 10
    S=0.
    IF(NOISE.EQ.1)THEN
        CALL GAUSS(XLAMNOISE,SIGNOISE)
    ELSE
        XLAMNOISE=0.
    ENDIF
    RES=XLAM-(XLAMH+TS*XLAMDH)+XLAMNOISE

```

(Contd.)

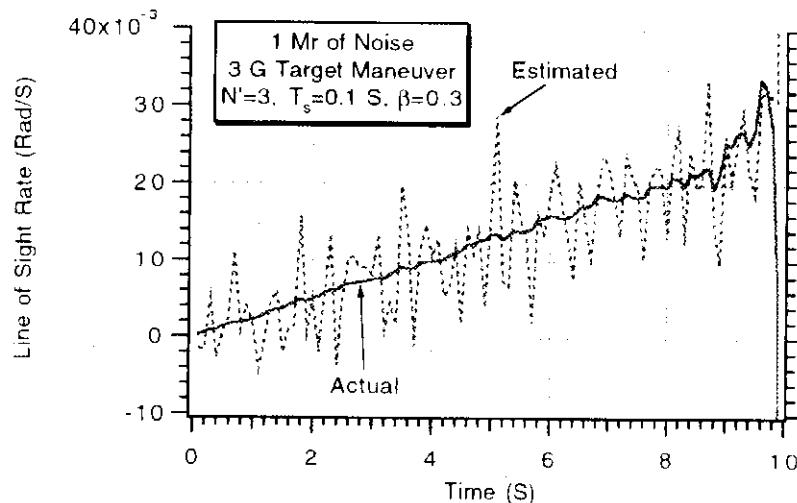
Listing 7.1 (Continued)

```

XLAMH=GFILTER*RES+XLAMH+TS*XLAMDH
XLAMDH=HFILTER*RES/TS+XLAMDH
XNC=XNP*VC*XLAMDH
WRITE(*,97)T,Y,XNC,XLAMD,XLAMDH
WRITE(1,97)T,Y,XNC,XLAMD,XLAMDH
GOTO 10
97  FORMAT(5F12.5)
200  CONTINUE
      TGO=TF-T+.00001
      RTM=VC*TGO
      XLAM=Y/(VC*TGO)
      XLAMD=(RTM*YD+Y*VC)/(RTM**2)
      YDD=XNT-XNC
      IF(STEP-1)66,66,55
999  CONTINUE
      WRITE(*,*)T,Y
      PAUSE
      CLOSE(1)
      END

SUBROUTINE GAUSS(X,SIG)
INTEGER SUM
SUM=0
DO 14 J=1,6
IRAN=Random()
SUM=SUM+IRAN
14  CONTINUE
X=SUM/65536.
X=1.414*X*SIG
RETURN
END

```

**Fig. 7.3 Decreasing β increases noise transmission of fading memory filter.**

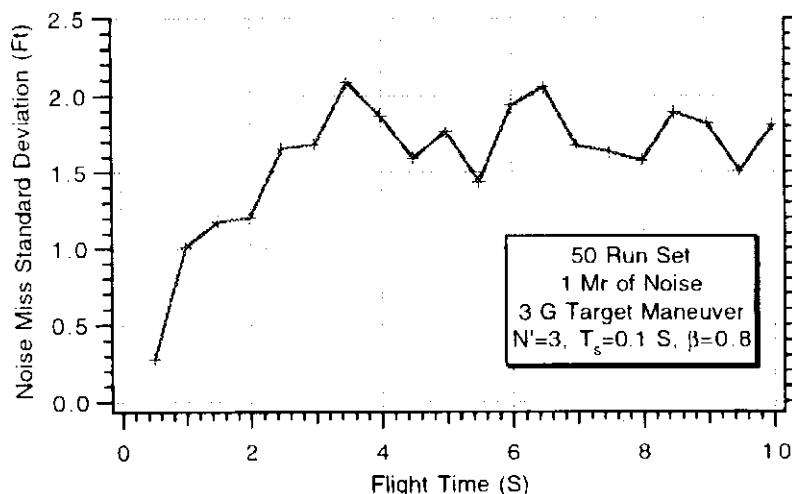


Fig. 7.4 Standard deviation of miss for various flight times.

for each flight time of interest. The resultant miss distance data must be postprocessed, as was done in Chapter 4 when dealing with the random target maneuver, to calculate the mean and standard deviation of the resultant miss distances. Listing 7.2 presents a modification to the engagement simulation of Listing 7.1. Here two loops are added to the program. One loop executes 50 simulation trials (RUN = 50) for each flight time of interest, and the other loop selects different flight times (TF ranges from 0.5 to 10 s in increments of 0.5 s). In other words, the simulation of Listing 7.2 runs Monte Carlo sets for engagements in which the flight time is a parameter. Postprocessing of the resultant data is conducted after statement label 999 in accordance with the formulas and routines developed in Chapter 4.

A nominal case was considered in which there was a constant 3-g target maneuver and 1 milliradian (mr) of measurement noise. A filter fading memory factor of 0.8 and a sampling time of 0.1 s were selected for the nominal case. A 50-run Monte Carlo set was run for 20 different values of flight time with the program of Listing 7.2 for a total of 1000 runs! The standard deviation and mean miss were computed for each of the 50-run Monte Carlo sets, and the results are displayed in Figs. 7.4 and 7.5. In this experiment there are only two disturbances. The target maneuver is deterministic (always 3 g), and the noise is a zero-mean random

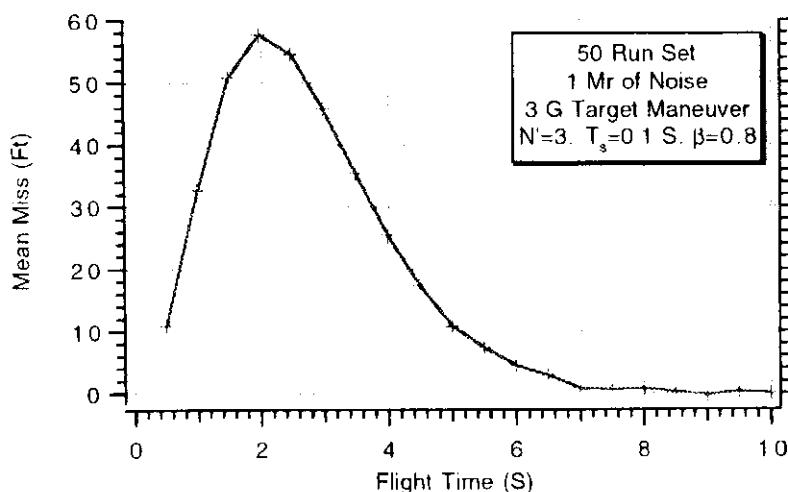


Fig. 7.5 Mean of miss for various flight times.

**Listing 7.2 Monte Carlo version of fading memory
filter in homing loop**

```

GLOBAL DEFINE
    INCLUDE 'quickdraw.inc'
END
DIMENSION Z(1000)
INTEGER RUN,STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
VC=4000.
XNT=96.6
YIC=0.
VM=3000.
HEDEG=0.
BETA=.8
XNP=3.
SIGNOISE=.001
TS=.1
RUN=50
NOISE=1
DO 60 TF=.5,10.,.5
Z1=0.
DO 20 I=1,RUN
    Y=YIC
    YD=-VM*HEDEG/57.3
    YDIC=YD
    T=0.
    H=.01
    S=0.
    GFILTER=1.-BETA**2
    HFILTER=(1.-BETA)**2
    XLAMH=0.
    XLAMDH=0.
    XNC=0.
10   IF(T>(TF-.0001))GOTO 999
        YOLD=Y
        YDOLD=YD
        STEP=1
        GOTO 200
66   STEP=2
        Y=Y+H*YD
        YD=YD+H*YDD
        T=T+H
        GOTO 200
55   CONTINUE
        Y=.5*(YOLD+Y+H*YD)
        YD=.5*(YDOLD+YD+H*YDD)
        S=S+H
C   COLLECT DATA PERIODICALLY
        IF(S<(TS-.0001))GOTO 10
        S=0.
        IF(NOISE.EQ.1)THEN

```

(Contd.)

Listing 7.2 (Continued)

```

        CALL GAUSS(XLAMNOISE,SIGNOISE)
ELSE
        XLAMNOISE=0.
ENDIF
RES=XLAM-(XLAMH+TS*XLAMDH)+XLAMNOISE
XLAMH=GFILTER*RES+XLAMH+TS*XLAMDH
XLAMDH=HFILTER*RES/TS+XLAMDH
XNC=XNP*VC*XLAMDH
GOTO 10
200 CONTINUE
TGO=TF-T+.00001
RTM=VC*TGO
XLAM=Y/(VC*TGO)
XLAMD=(RTM*YD+Y*VC)/(RTM**2)
YDD=XNT-XNC
IF(STEP-1)66,66,55
999 CONTINUE
Z(I)=Y
Z1=Z(I)+Z1
XMEAN=Z1/I
20 CONTINUE
SIGMA=0.
Z1=0.
DO 50 I=1,RUN
Z1=(Z(I)-XMEAN)**2+Z1
IF(I.EQ.1)THEN
        SIGMA=0.
ELSE
        SIGMA=SQRT(Z1/(I-1))
ENDIF
50 CONTINUE
WRITE(*,97)TF,SIGMA,XMEAN
WRITE(1,97)TF,SIGMA,XMEAN
60 CONTINUE
97 FORMAT(3F10.3)
PAUSE
END

SUBROUTINE GAUSS(X,SIG)
INTEGER SUM
SUM=0
DO 14 J=1,6
IRAN=Random()
SUM=SUM+IRAN
14 CONTINUE
X=SUM/65536.
X=1.414*X*SIG
RETURN
END

```

process. Therefore, we can assume that the standard deviation of the miss must be due to the noise, and the mean of the miss must be due to the target maneuver. Figures 7.4 and 7.5 show that for the value of β selected the noise-induced miss is small compared to the target-maneuver-induced miss for most flight times. This is not surprising because we know that the fading memory filter with $\beta = 0.8$ is sluggish.

To generate the data of Figs. 7.4 and 7.5, 1000 runs had to be made! One thousand run sets will have to be made each time a parameter of interest is changed. In addition, we were able to separate the contributions to the miss from the measurement noise and the target maneuver in the 1000 run set *only because* the noise was random and the target maneuver was deterministic. In a system with many deterministic and stochastic inputs, one would have to run 1000-run sets for one disturbance at a time in order to generate a miss distance error budget. Fortunately the adjoint technique allows us to get error budget information of this type in only *one* run! Since the system under consideration is a mixed continuous discrete system, we have to extend the rules for adjoints which we have already covered in Chapters 3 and 4.

Mixed Continuous Discrete Adjoint Theory^{4,5}

The rules for constructing an adjoint of a mixed continuous discrete system are simple and are similar to the adjoint rules for continuous systems. Given a linear time-varying discrete system with impulse response H_D in which the ratio of the time of flight to the sampling time is an integer given by

$$N = t_F / T_s$$

there exists an adjoint system with impulse response H_D^* . One can construct a mixed continuous discrete adjoint from the original system using the rules of Chapters 3 and 4 and the additional rule given in the next subsection.

Replace n by $N - n$ in the Arguments of All Variable Coefficients

Therefore, the rules for continuous and mixed continuous discrete adjoints are identical in that the signal flow of the original system is reversed and the time-varying gains in the original system are generated backwards for the adjoint system. In mixed continuous discrete systems the adjoint rules get slightly more complicated because additional elements are required for interfacing the continuous and discrete parts of the system. For example, a sampler or analog-to-digital converter is required as shown in Fig. 7.6, to make the connection from a continuous system to a discrete system.

The input/output characteristics of a sampler can easily be illustrated. For example, Fig. 7.7 shows that, if the input signal to the sampler is continuous, the output signal has the same shape but is defined only at each sampling instant by a number. These numbers, or sample points, are spaced T_s seconds apart.

Applying adjoint theory to mixed continuous discrete systems requires taking the adjoint of a sampler. The adjoint of a sampler is depicted in Fig. 7.8. Here the



Fig. 7.6 Connecting continuous and discrete systems.

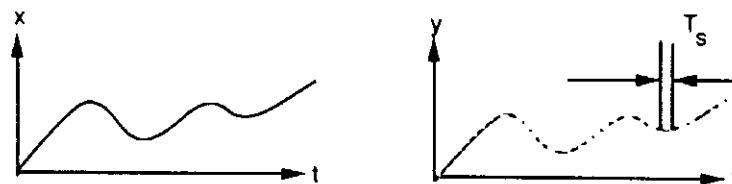


Fig. 7.7 Effect of sampling on continuous signal.

s block represents a pure derivative and the "hold" block will soon be defined. The z^{-1} block is Z transform notation and represents a pure delay of T_s seconds.

A hold network or digital-to-analog converter is required to connect signal flow from a discrete network to a continuous network as shown in Fig. 7.9.

If the input to the hold is a set of numbers, Fig. 7.10 illustrates the proper input/output characteristics of the hold network. Here we can see that, after a discrete signal has been "held," it becomes continuous.

The adjoint of a hold is shown in Fig. 7.11. Here the $1/s$ term is the Laplace transform representation of an integrator. Again, the z^{-1} term represents a pure delay of T_s seconds.

We now have enough rules to enable us to take the adjoint of a mixed continuous discrete system. Consider the model of Fig. 7.12. In this model there are three continuous linear time-varying networks with impulse responses H_{C1} , H_{C2} , and H_{C3} , respectively. White noise u_C with spectral density Φ_C enters the continuous portion of the system through the shaping network H_{C3} . In addition, a step disturbance of magnitude a also enters the system through the shaping network H_{C1} . The step input has been represented as an impulse through an integrator so that adjoint theory can be applied to this error source.

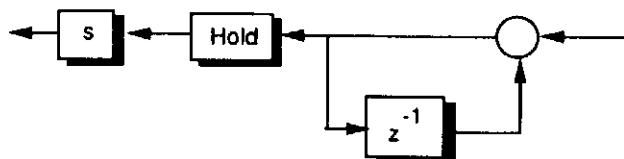


Fig. 7.8 Adjoint of a sampler.



Fig. 7.9 Connecting discrete and continuous systems.

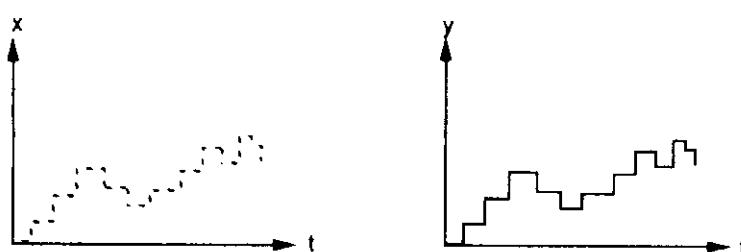


Fig. 7.10 Effects of holding a discrete signal.

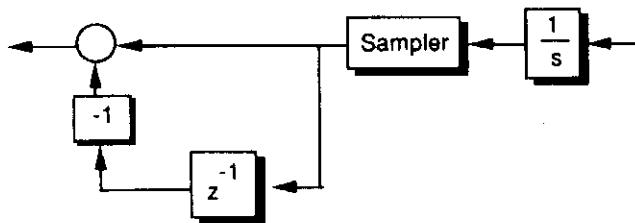


Fig. 7.11 Adjoint of hold.

In this example we are interested in observing the continuous quantity y at the final time t_F . The output of the network H_{C2} is sampled and sent through a discrete network with impulse response H_{D1} . Zero-mean Gaussian noise with variance σ_D^2 enters the discrete portion of the system through shaping network H_{D2} . After the resultant signal goes through the discrete network H_{D1} , the output is held and fed back to the continuous network H_{C1} , thus completing the loop. In this example we seek to find $y(t_F)$ due to each of the disturbances. Adjoint theory can readily be applied to this example.

Following the rules of adjoints we can obtain the adjoint system of Fig. 7.13. Although this adjoint model is driven by an impulse, we have seen in previous chapters that it is not necessary to simulate an impulse. The impulse becomes initial conditions on integrators in its forward path. The impulse is applied at time zero in the adjoint system because the output of interest in the original system is at the final time. The adjoint model shows a differentiator appearing before H_{C2}^* . Again, one need not simulate the differentiator but just use block diagram manipulation to eliminate it (i.e., feed it through H_{C2}^*).

The outputs of the adjoint model represent output sensitivities of the system. They are referred to as sensitivities because a change in their levels does not necessitate a rerunning of the adjoint simulation. As can be seen from Fig. 7.13, the new outputs can be calculated by inspection. Note that, in order to find $y(t_F)$ due to a continuous random disturbance, we square and integrate a continuous signal (i.e., output of H_{C3}^*). To find $y(t_F)$ due to a discrete random disturbance, we square and sum a discrete signal (i.e., output of H_{D2}^*).

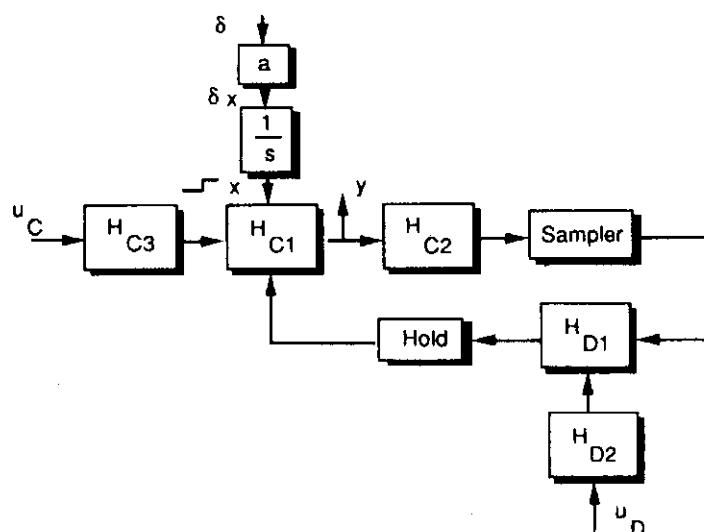


Fig. 7.12 Model of mixed continuous discrete system.

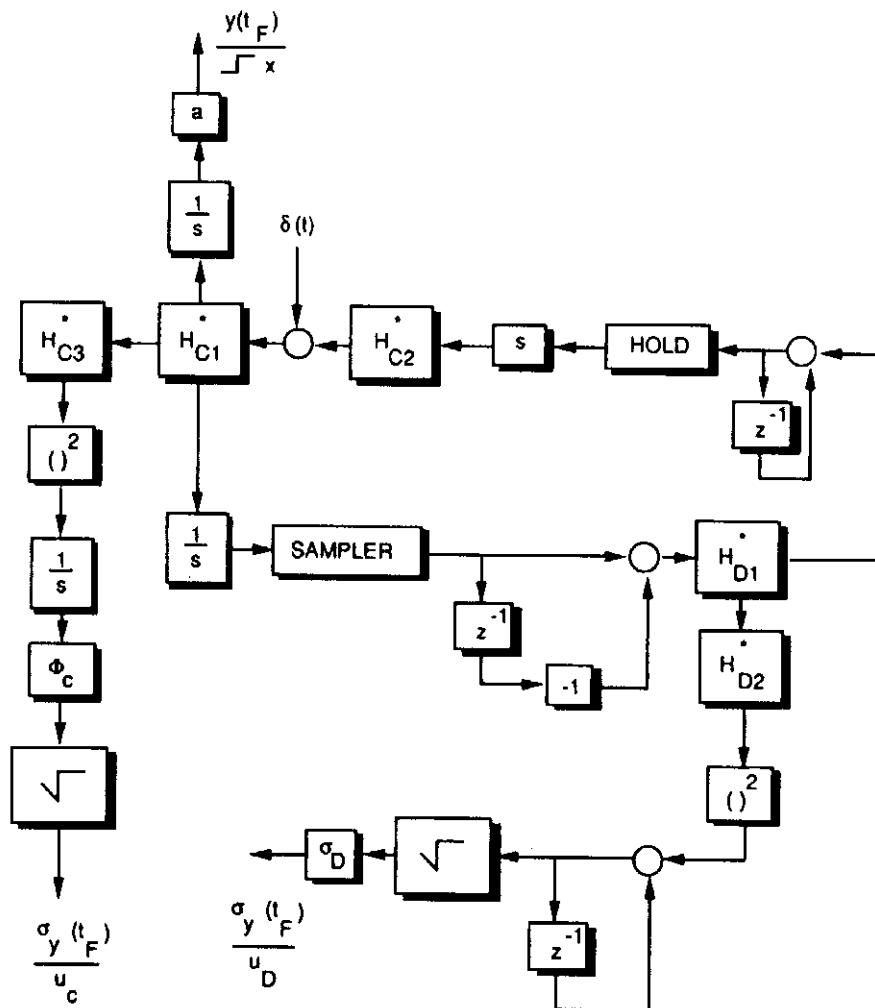


Fig. 7.13 Adjoint of mixed continuous discrete system.

Using Adjoints to Evaluate Filter Performance

Mixed continuous discrete adjoint theory can be applied to the engagement model of Fig. 7.1, a sample data homing loop containing a two-state fading memory filter. Recall that in this example there are two disturbances: a deterministic target maneuver and measurement noise on the line-of-sight angle. However, before we take the complete adjoint, let us realize that when the adjoint of a "sampler" is taken we will have a pure differentiator in the homing loop. It is desirable, for simulation reasons, to eliminate the differentiator by block diagram manipulation. This can easily be done by modifying the original system to have an extra integrator before the sampler. This can be accomplished by first generating the line-of-sight rate and then integrating it to get line-of-sight angle. First we must realize that the line-of-sight angle can be expressed as

$$\lambda = \frac{y}{R_{TM}} = \frac{y}{V_c t_{go}}$$

Taking the derivative of the preceding expression, using the quotient rule, and expressing the result in block diagram form we obtain Fig. 7.14.

The resultant adjoint block diagram for the entire homing loop, following the mixed continuous discrete adjoint rules of the previous section, appears in Fig. 7.15.

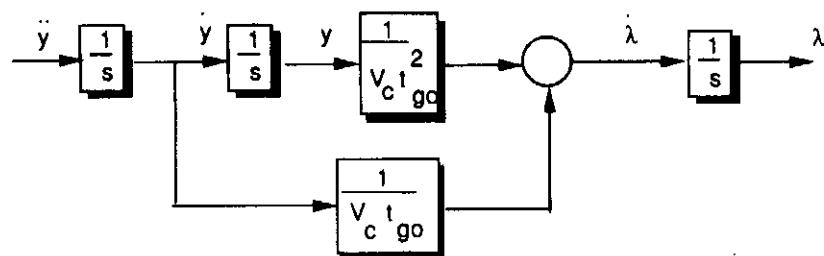


Fig. 7.14 Block diagram for line-of-sight rate.

The two disturbances of the original system become adjoint outputs, whereas the miss distance output of the original system becomes an impulsive input (or initial condition on integrator x_3) in the adjoint system. Note that, since the noise is digital, the adjoint noise miss distance sensitivity is obtained by squaring and summing the appropriate signal.

Listing 7.3 presents the FORTRAN adjoint program for the engagement model of Fig. 7.15 in which the homing loop contains a second-order fading memory filter. As with the original engagement simulation presented in this chapter, the adjoint program also consists of two sections: one for the differential equations and the other part for the difference equations. Care must also be taken in the adjoint program to ensure that the ratio of the sampling interval to the integration interval be a large integer.

A single adjoint run was made for the nominal case considered at the beginning of the chapter ($\beta = 0.8$). The target maneuver miss and noise miss outputs are plotted separately vs adjoint or flight time in Figs. 7.16 and 7.17. Superimposed on

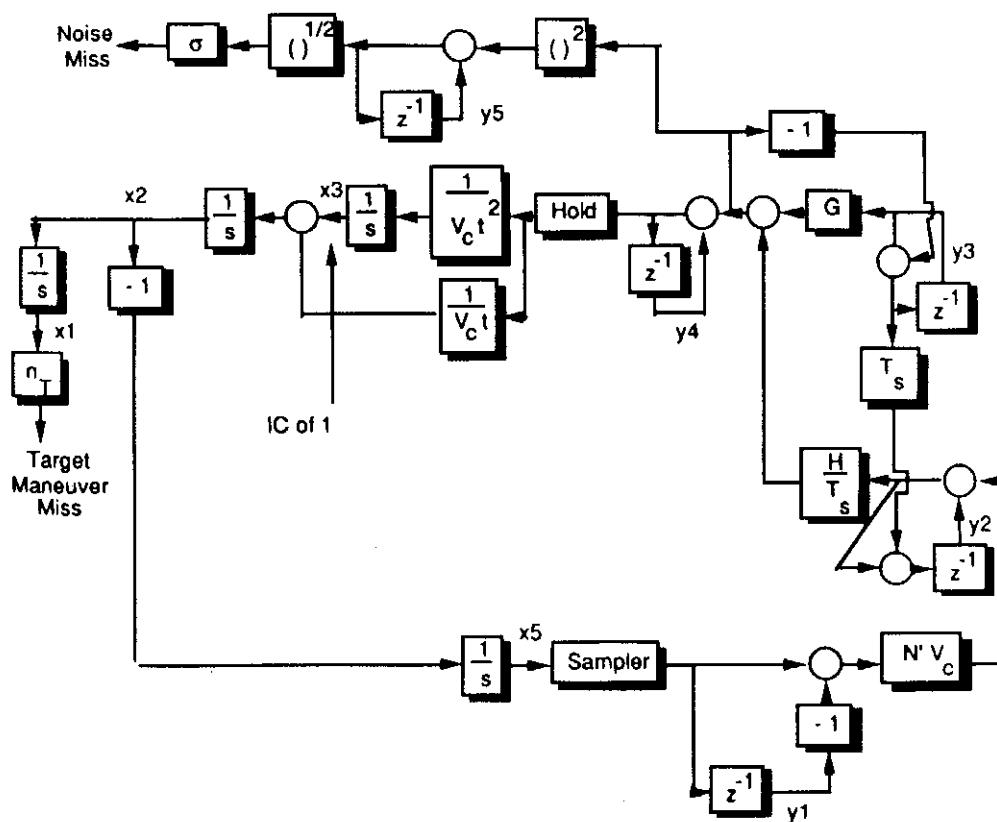


Fig. 7.15 Adjoint of second-order fading memory filter in homing loop.

**Listing 7.3 Adjoint engagement simulation with
two-state fading memory filter**

```

INTEGER STEP
DATA XNT,XNP,TF,TS/ 96.6,3.,10.,.1/
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
BETA=.8
SIGNOISE=.001
VC=4000.
T=0.
S=0.
TP=T+.00001
X1=0
X2=0
X3=1
X5=0.
Y1OLD=0.
Y2OLD=0.
Y3OLD=0.
Y4OLD=0.
Y5OLD=0.
H=.01
GFILTER=1.-BETA**2
HFILTER=(1.-BETA)**2
10 IF(TP>(TF-.00001))GOTO 999
S=S+H
X1OLD=X1
X2OLD=X2
X3OLD=X3
X5OLD=X5
STEP=1
GOTO 200
66 STEP=2
X1=X1+H*X1D
X2=X2+H*X2D
X3=X3+H*X3D
X5=X5+H*X5D
TP=TP+H
GOTO 200
55 CONTINUE
X1=(X1OLD+X1)/2+.5*H*X1D
X2=(X2OLD+X2)/2+.5*H*X2D
X3=(X3OLD+X3)/2+.5*H*X3D
X5=(X5OLD+X5)/2+.5*H*X5D
IF(S<(TS-.0001))GOTO 10
S=0.
TEMP1=(X5-Y1OLD)*XNP*VC
TEMP2=HFILTER*(Y2OLD+TEMP1)/TS+GFILTER*Y3OLD
Y1NEW=X5
Y2NEW=TEMP1+Y2OLD+TS*(Y3OLD-TEMP2)
Y3NEW=Y3OLD-TEMP2

```

(Contd.)

Listing 7.3 (Continued)

```

Y4NEW=Y4OLD+TEMP2
Y5NEW=Y5OLD+TEMP2*TEMP2
Y1OLD=Y1NEW
Y2OLD=Y2NEW
Y3OLD=Y3NEW
Y4OLD=Y4NEW
Y5OLD=Y5NEW
XMNOISE=SIGNOISE*SQRT(Y5NEW)
XMNT=XNT*X1
WRITE(*,97)TP,XMNT,XMNOISE
WRITE(1,97)TP,XMNT,XMNOISE
GOTO 10
97  FORMAT(3F10.5)
200  CONTINUE
TGO=TP+.00001
X1D=X2
X2D=X3+Y4OLD/(VC*TGO)
X3D=(Y4OLD)/(VC*TGO*TGO)
X5D=-X2
IF(STEP-1)66,66,55
999  CONTINUE
PAUSE
CLOSE(1)
END

```

these *single run* adjoint results are the standard deviation and mean of the Monte Carlo miss distance results, obtained with *1000 runs* (50 run sets for 20 flight times). We can see that both methods yield approximately the same answers. If there were more error sources, the miss distance performance projections could still have been obtained from the same adjoint run by monitoring additional outputs. Thus, we can see that the adjoint technique is a very powerful method for efficiently generating miss distance error budgets.

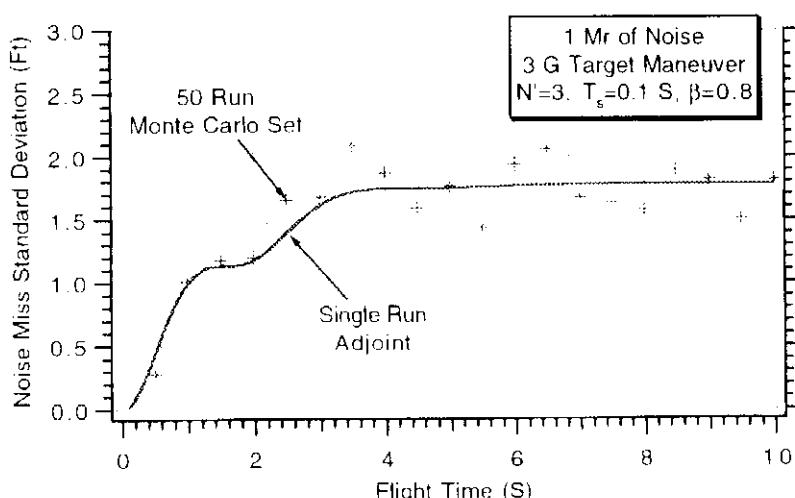


Fig. 7.16 Adjoint noise miss projections are in agreement with Monte Carlo results.

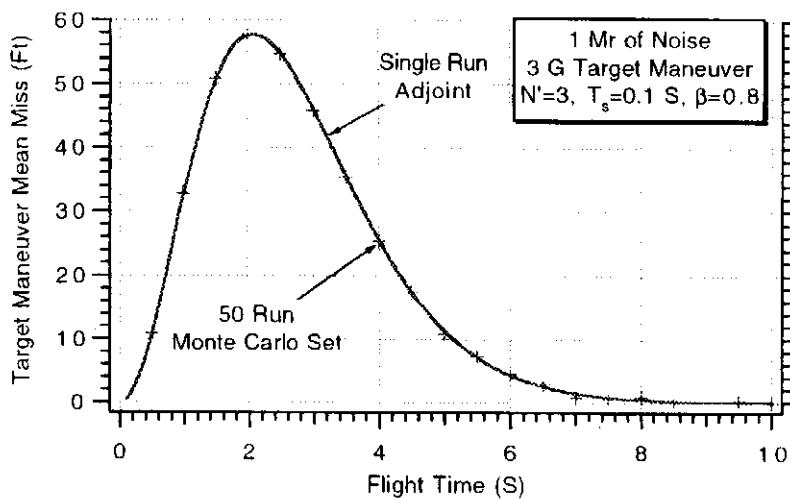


Fig. 7.17 Adjoint target maneuver miss projections are in agreement with Monte Carlo results.

Some Properties of Fading Memory Filters

The filter parameter β determines how much the filter will remember about past measurements, which in turn will determine the filter bandwidth or speed of response. Higher values of β yield a filter that remembers a great deal about the past. This type of filter will have low bandwidth and slow speed of response. Low values of β result in a high bandwidth fast filter. Figures 7.18 and 7.19 show, based on adjoint simulation results, how miss distance varies with the fading memory filter parameter β . It is not surprising that Fig. 7.18 shows dramatically improved miss distance results for the wider bandwidth filter (faster guidance systems yield smaller miss distances due to target maneuver). However, it is surprising that Fig. 7.19 shows that there is slightly less miss distance due to line-of-sight angle noise for the faster filter, even though we know that the filter has more noise transmission. In general, reducing the guidance system time constant will reduce the miss for most disturbances in a proportional navigation guidance system in the absence of parasitic and saturation effects. Ultimately excessive noise transmission will lead to saturation and increased miss distance.

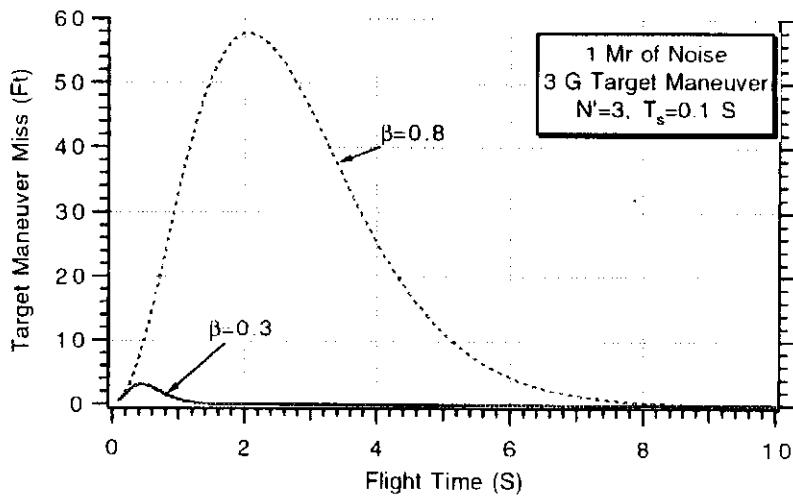


Fig. 7.18 Faster fading memory filter yields less miss due to target maneuver.

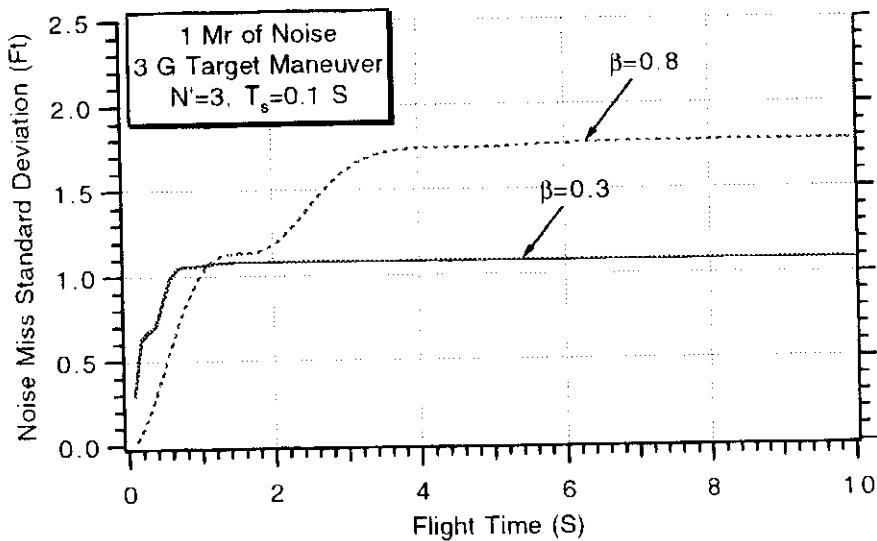


Fig. 7.19 Faster noise filter yields less miss due to line-of-sight angle noise.

Changing the sampling time also effects filter and system performance. Increasing the sampling rate (lower values for T_s) means that the filter has more information per unit time. Therefore, increasing the sampling rate should be beneficial. Figure 7.20 shows, from single flight results, that increasing the sampling rate (reduce T_s from 0.1 s to 0.05 s) removes the previously noted lag in the line-of-sight rate estimate when β is 0.8 (see Fig. 7.2 for comparison). The noise transmission appears to be about the same, or slightly reduced, from the case when the sampling time was 0.1 s.

From a system performance point of view, increasing the sampling rate should also be beneficial. In essence, we are speeding up the guidance system, which means for the inputs previously considered, miss should decrease. Adjoint results, which are displayed in Fig. 7.21, confirm that doubling the sampling rate (reducing T_s from 0.1 s to 0.05 s) dramatically reduces the miss sensitivity due to target maneuver.

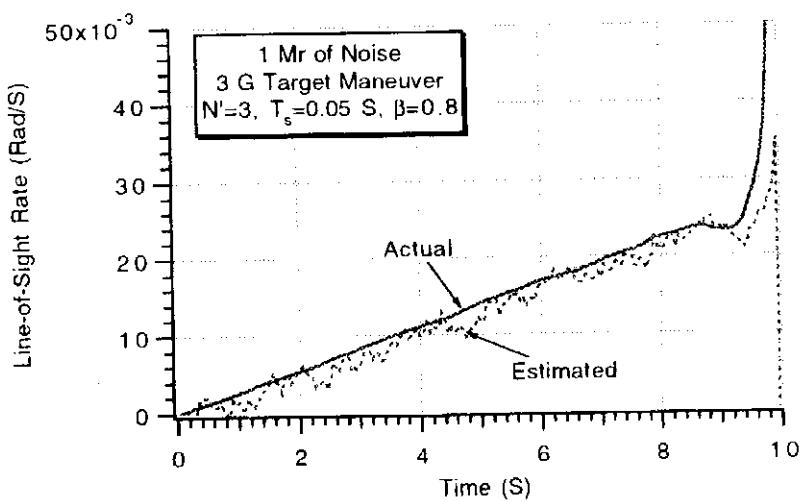


Fig. 7.20 Increasing sampling rate makes fading memory filter faster.

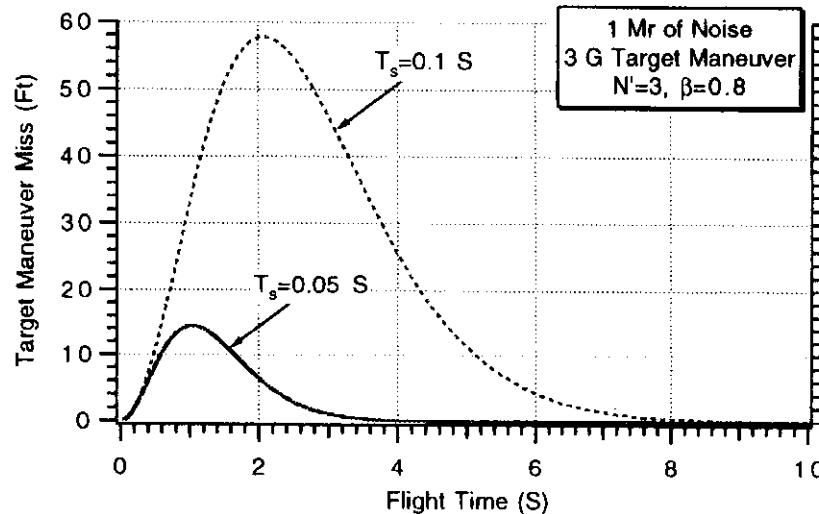


Fig. 7.21 Increasing sampling rate reduces miss due to target maneuver.

Figure 7.22 also confirms that the miss due to noise decreases with increasing sampling rate. Usually, increased system costs are associated with higher sampling rates. Therefore, financial considerations usually place an upper limit on practical achievable sampling rates.

It is important to note that in the preceding experiment the noise standard deviation remained constant as the data rate changed. In many systems the noise spectral density remains constant and so the noise standard deviation changes with changing data rate. The interested reader is referred to Appendix A for a more complete discussion of this topic.

Estimating Target Maneuver

In Chapter 8 we will investigate more advanced guidance laws. To implement more advanced guidance laws, we must have knowledge of all of the target states.

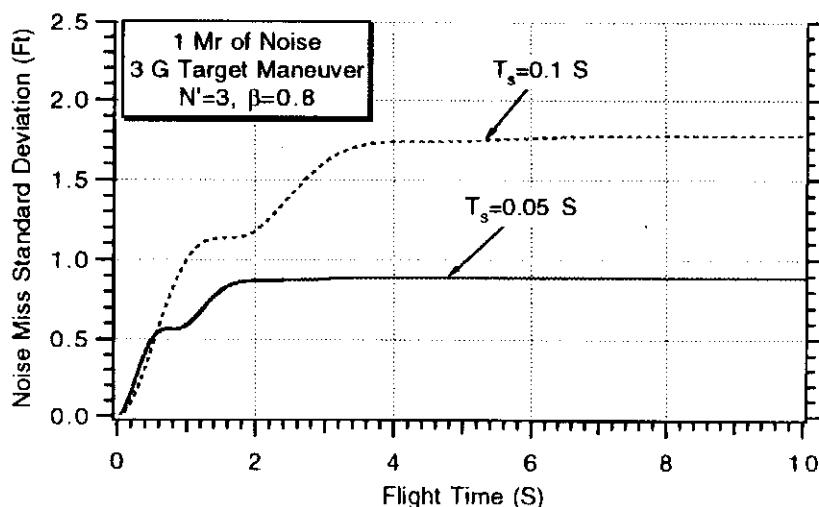


Fig. 7.22 Increasing sampling rate reduces miss due to noise.

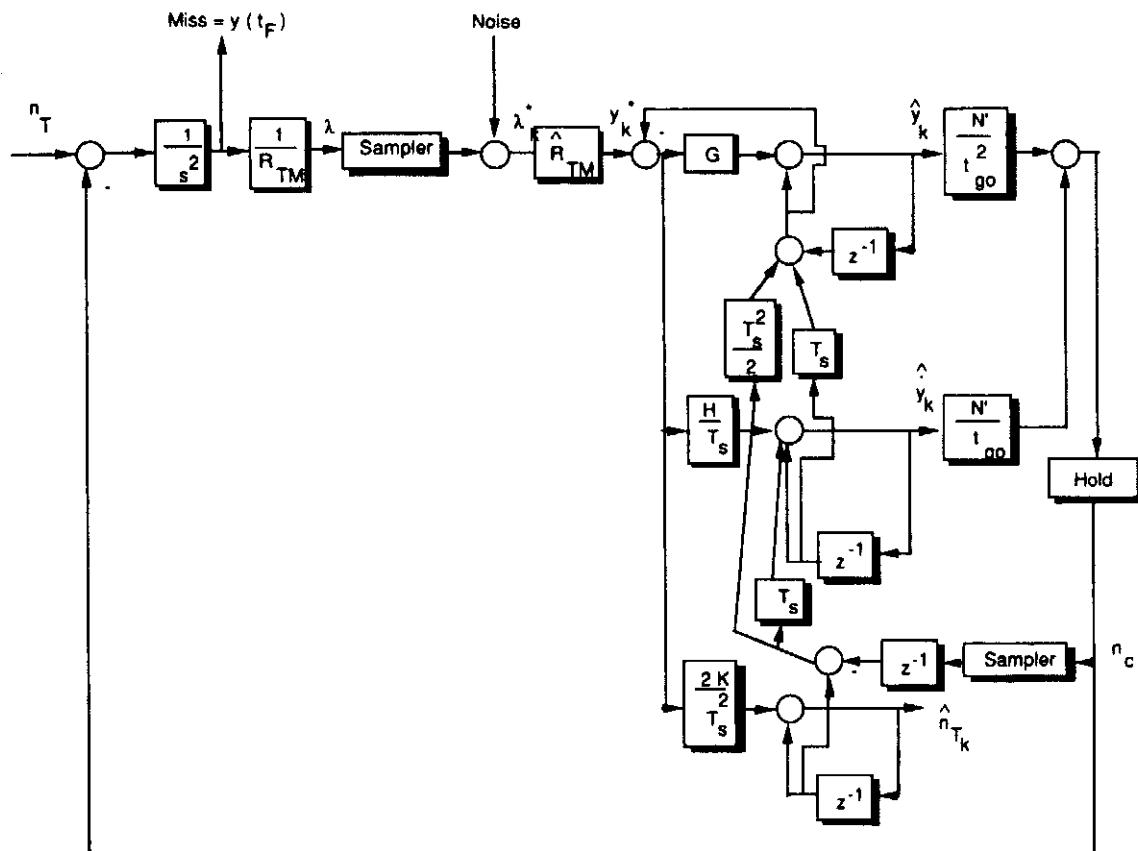


Fig. 7.23 Estimating target maneuver with three-state fading memory filter.

In other words, we must know what the target is doing. Mathematically stated, we would like to be able to estimate the current maneuver level of the target based on a noisy measurement of the line-of-sight angle. Theoretically it is impossible, without additional measured or a priori information, to estimate the maneuver level of the target based on angle-only measurements from a single sensor. However, many tactical radar homing missiles also measure range and range rate in addition to the the line-of-sight angle, which makes target acceleration estimation possible.

Figure 7.23 presents a guidance system that uses a three-state fading memory filter to estimate target acceleration from a measurement of the line-of-sight angle, range, and closing velocity. The noisy measurement of the line-of-sight angle is multiplied by a range measurement to get a pseudomeasurement of relative position y_k^* . The filter then estimates the derivatives of the measurement. Using knowledge of the missile acceleration, which is assumed to be known precisely, it is then possible to estimate target acceleration from a relative acceleration as shown in Fig. 7.23. With this type of guidance system we also need time to go information, which can be obtained from the range and range rate measurements, to implement either the proportional or augmented proportional navigation guidance law.

Listing 7.4 presents a FORTRAN engagement simulation with the three-state fading memory filter as shown in Fig. 7.23. Note that the three-state filter gains are different from the two-state filter gains.

Listing 7.4 Engagement simulation with three-state fading memory filter

```

GLOBAL DEFINE
    INCLUDE 'quickdraw.inc'
END
REAL KFILTER
INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
VC=4000.
XNT=96.6
YIC=0.
VM=3000.
HEDEG=0.
BETA=.8
XNP=3.
SIGNOISE=.001
TF=10.
TS=.1
NOISE=1
Y=YIC
YD=-VM*HEDEG/57.3
YDIC=YD
T=0.
H=.01
S=0.
GFILTER=1.-BETA**3
HFILTER=1.5*((1.-BETA)**2)*(1.+BETA)
KFILTER=.5*((1.-BETA)**3)
YH=0.
YDH=0.
XNTH=0.
XNC=0.
10 IF(T>(TF-.0001))GOTO 999
    YOLD=Y
    YDOLD=YD
    STEP=1
    GOTO 200
66 STEP=2
    Y=Y+H*YD
    YD=YD+H*YDD
    T=T+H
    GOTO 200
55 CONTINUE
    Y=.5*(YOLD+Y+H*YD)
    YD=.5*(YDOLD+YD+H*YDD)
    S=S+H
    IF(S<(TS-.0001))GOTO 10
    S=0.
    IF(NOISE.EQ.1)THEN
        CALL GAUSS(XLAMNOISE,SIGNOISE)

```

(Contd.)

Listing 7.4 (Continued)

```

ELSE
  XLAMNOISE=0.
ENDIF
YSTAR=RTM*(XLAM+XLAMNOISE)
RES=YSTAR-YH-TS*YDH-.5*TS*TS*(XNTH-XNC)
YH=GFILTER*RES+YH+TS*YDH+.5*TS*TS*(XNTH-XNC)
YDH=HFILTER*RES/TS+YDH+TS*(XNTH-XNC)
XNTH=2.*KFILTER*RES/(TS*TS)+XNTH
XLAMDH=(YH+YDH*TGO)/(VC*TGO*TGO)
XNC=XNP*VC*XLAMDH
WRITE(*,97)T,Y,XNC/32.2,XLAMD,XLAMDH,XNT/32.2,XNTH/32.2
WRITE(1,97)T,Y,XNC/32.2,XLAMD,XLAMDH,XNT/32.2,XNTH/32.2
GOTO 10
97 FORMAT(7F10.4)
200 CONTINUE
TGO=TF-T+.00001
RTM=VC*TGO
XLAM=Y/(VC*TGO)
XLAMD=(RTM*YD+Y*VC)/(RTM**2)
YDD=XNT-XNC
IF(STEP-1)66,66,55
999 CONTINUE
PAUSE
END

SUBROUTINE GAUSS(X,SIG)
INTEGER SUM
SUM=0
DO 14 J=1,6
IRAN=Random()
SUM=SUM+IRAN
14 CONTINUE
X=SUM/65536.
X=1.414*X*SIG
RETURN
END

```

A nominal case was run with the simulation of Listing 7.4 in which the fading memory factor of the filter was 0.8 and the sampling time was 0.1 s. Figure 7.24 compares the line-of-sight rate estimate of the filter with the actual line-of-sight rate for the nominal case. We can see that the filter estimate follows the geometric line-of-sight rate without excessive noise transmission.

Figure 7.25 shows, for the same case, the filter estimate of the target maneuver. Superimposed on the figure is the actual maneuver. We can see that for this case it takes the filter about 5 s to get a reasonable estimate of the maneuver level. A faster filter would have a smaller transient period but much more noise transmission. Estimates of the quality shown in Fig. 7.25 are sufficient for improving guidance system performance.

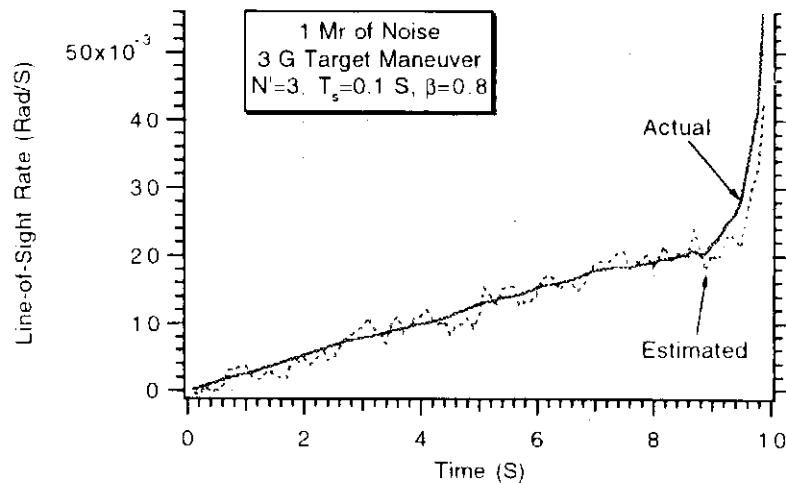


Fig. 7.24 Three-state filter yields excellent estimate of line-of-sight rate.

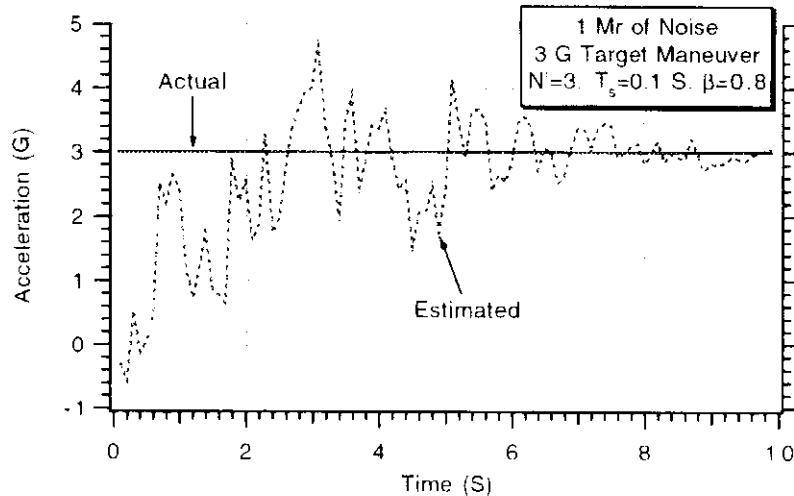


Fig. 7.25 Three-state fading memory filter is able to estimate target maneuver.

Summary

In this chapter it was shown how a simple constant-gain, digital noise filter, known as a fading memory filter, could be implemented in a missile guidance system. It was shown that both filter bandwidth and sampling rate are important parameters in determining overall system performance. The method of adjoints was extended so that it could be used to yield performance projections of a missile guidance system with a digital noise filter. Experiments confirmed that Monte Carlo simulation results were in complete agreement with single-run adjoint performance projections. Finally, it was shown how a fading memory filter could be utilized to provide target acceleration estimates.

References

¹Morrison, N., *Introduction to Sequential Smoothing and Prediction*, McGraw-Hill, New York, 1969.

²Bar-Shalom, Y., and Li, X. R., *Estimation and Tracking Principles, Techniques, and Software*, Artech House, Boston, 1993.

³Kalata, P. R., "The Tracking Index: A Generalized Parameter for Alpha-Beta-Gamma Target Trackers," *IEEE Transactions Aerospace and Electronic Systems*, AES-20, pp. 174-182, March 1994.

⁴Moroney, R., "The Adjoint of Mixed Continuous/Discrete Systems," Raytheon Memo RM-69-1, Jan. 1969.

⁵Zarchan, P., and Warren, R. S., "Discrete Adjoint Simulation," Raytheon Rept., BR-5440-1, Oct. 1969.



Advanced Guidance Laws

Introduction

THUS far we have used proportional navigation as an interceptor guidance law because it is easy to implement and is very effective. In fact, proportional navigation is used extensively in the tactical missile world. However, there are other more advanced guidance laws. These advanced guidance laws relax the interceptor acceleration requirements and also yield smaller miss distances. The price paid for these more advanced guidance laws is that more information, such as time to go and missile-target range, is required for their successful implementation. The concept of zero effort miss, originally introduced in Chapter 2, will be used to develop and understand new guidance laws. The zero effort miss concept will also be important when we move to the strategic world and encounter predictive guidance. The Schwartz inequality will be used to analytically derive optimal guidance laws.

Review of Proportional Navigation

The basic homing loop for a zero-lag proportional navigation guidance system, which first appeared in Chapter 2, is repeated for convenience in Fig. 8.1. In this zero-lag loop, the seeker, noise filter, and flight-control system dynamics have been neglected. As can be seen from the figure, the proportional navigation guidance law can be expressed as

$$n_c = N' V_c \dot{\lambda}$$

where N' is a gain known as the effective navigation ratio, V_c the closing velocity, and $\dot{\lambda}$ the line-of-sight angle.

We have already shown in Chapter 2 that closed-form solutions for the required missile acceleration exist for this zero-lag guidance system. The resultant formula for the missile acceleration n_c due to a step target maneuver was derived from the first-order time-varying proportional navigation homing loop differential equation originally presented in Chapter 2. The solution, which is repeated here for convenience, is given by

$$n_c = \frac{N'}{N' - 2} \left[1 - \left(1 - \frac{t}{t_F} \right)^{N' - 2} \right] n_T$$

where t_F is the time of flight, t the time, and n_T the target maneuver level. We can see from the closed-form solution that the required missile acceleration is

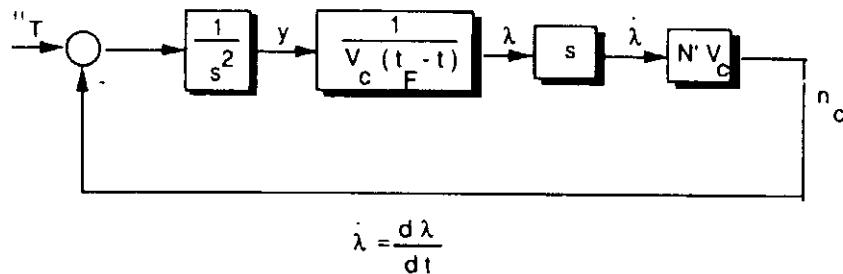


Fig. 8.1 Zero-lag proportional navigation homing loop.

directly proportional to the target maneuver acceleration level. Doubling the target acceleration level also doubles the missile acceleration requirements.

To convey the maximum amount of information concisely, the closed-form solution for the missile acceleration induced by target maneuver is normalized and displayed in Fig. 8.2 for different values of the effective navigation ratio. We can see that, regardless of the effective navigation ratio, the required missile acceleration induced by a target maneuver is largest at the end of the flight. Increasing the effective navigation ratio tends to reduce the maximum missile acceleration requirement. Of course we have already seen that, due to parasitic effects and possibly noise considerations, there is a practical upper limit on maximum allowable values for the effective navigation ratio.

The missile's maximum required acceleration, which occurs at the end of the flight ($t = t_F$), can be obtained from the closed-form solution as

$$n_{c \max} |_{PN} = \frac{N' n_T}{N' - 2}$$

Therefore, for an effective navigation ratio of 3, the missile requires three times the acceleration capability of the target for a successful intercept. However, increasing the effective navigation ratio from 3 to 5 reduces the required missile acceleration advantage from 3 to 1.67. Of course, other disturbances plus system dynamics will work in the direction of increasing the required missile acceleration advantage.

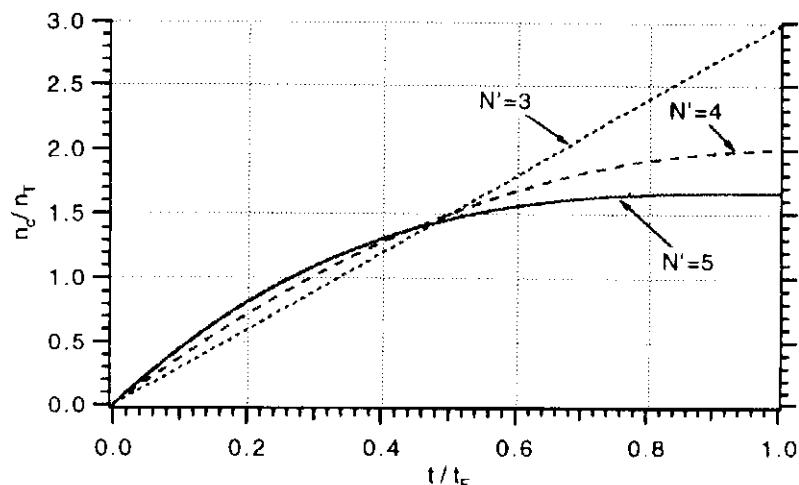


Fig. 8.2 Normalized missile acceleration due to target maneuver for proportional navigation guidance.

Augmented Proportional Navigation

More advanced guidance laws can be developed from the zero-lag homing loop model of Fig. 8.1. First we note that the line-of-sight angle can also be expressed as

$$\lambda = \frac{y}{R_{TM}} = \frac{y}{V_c(t_F - t)}$$

where y is the relative missile-target separation and R_{TM} the range from the missile to the target. We can find the line-of-sight rate by taking the derivative of the preceding expression, using the quotient rule, obtaining

$$\dot{\lambda} = \frac{y + \dot{y}t_{go}}{V_c t_{go}^2}$$

where t_{go} is the time to go until intercept and can be defined as

$$t_{go} = t_F - t$$

Thus, we can also express the proportional navigation guidance law as the mathematically equivalent expression

$$n_c = N' V_c \dot{\lambda} = \frac{N'(y + \dot{y}t_{go})}{t_{go}^2}$$

The expression in the parentheses of the preceding equation represents the future separation between missile and target. More simply, the expression in parentheses is the miss distance that would result if the missile made no further corrective acceleration and the target did not maneuver. This expression is referred to as the zero effort miss *ZEM*. Therefore, we can also think of proportional navigation as a guidance law in which acceleration commands are issued inversely proportional to the square of time to go and directly proportional to the zero effort miss.

If the target maneuvers, the zero effort miss must be augmented by an additional term. The new equation for the zero effort miss, in the presence of a constant target maneuver, is simply

$$ZEM_{TGT\ MVR} = y + \dot{y}t_{go} + 0.5n_T t_{go}^2$$

where n_T is the target maneuver acceleration level. Therefore, a perfectly plausible guidance law, in the presence of target maneuver, would be

$$n_c|_{APN} = \frac{N' ZEM_{TGT\ MVR}}{t_{go}^2} = N' V_c \dot{\lambda} + \frac{N' n_T}{2}$$

This new guidance law, known as augmented proportional navigation, is proportional navigation with an extra term to account for the maneuvering target.¹

A zero-lag augmented proportional navigation homing loop is shown in block diagram form in Fig. 8.3. The additional target maneuver term, required by the guidance law, appears as a feedforward term in the homing loop block diagram. As with the proportional navigation guidance law, we can also obtain closed-form solutions for the required missile acceleration due to a constant target maneuver

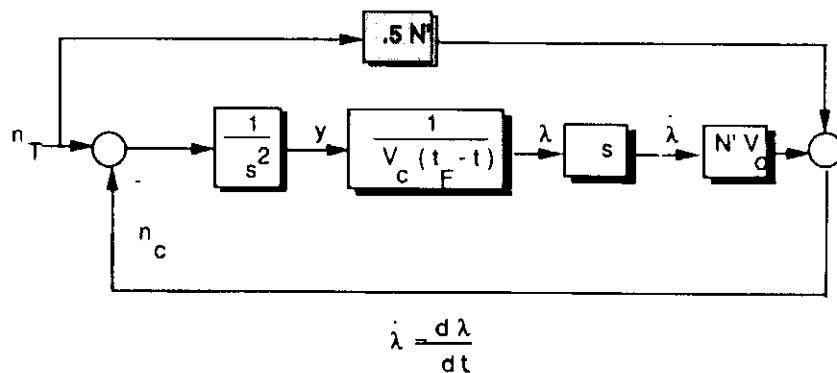


Fig. 8.3 Augmented proportional navigation homing loop.

for the zero-lag homing loop depicted in Fig. 8.3. The resultant solution for the required missile acceleration is

$$n_c|_{APN} = 0.5n_T N' \left(1 - \frac{t}{t_F}\right)^{N'-2}$$

The closed-form solution for the missile acceleration required to hit a maneuvering target with augmented proportional navigation is displayed in normalized form in Fig. 8.4. Here we can see that the required missile acceleration decreases monotonically with time, regardless of the effective navigation ratio, rather than increasing monotonically with time as was the case with proportional navigation. Increasing the effective navigation ratio increases the maximum acceleration at the beginning of the flight but also reduces the time at which the acceleration decays to negligible levels.

The maximum acceleration required by augmented proportional navigation to hit a maneuvering target is

$$n_{c \max}|_{APN} = 0.5N'n_T$$

This means that, for a navigation ratio of 3, augmented proportional navigation requires half the acceleration of the missile than with proportional navigation

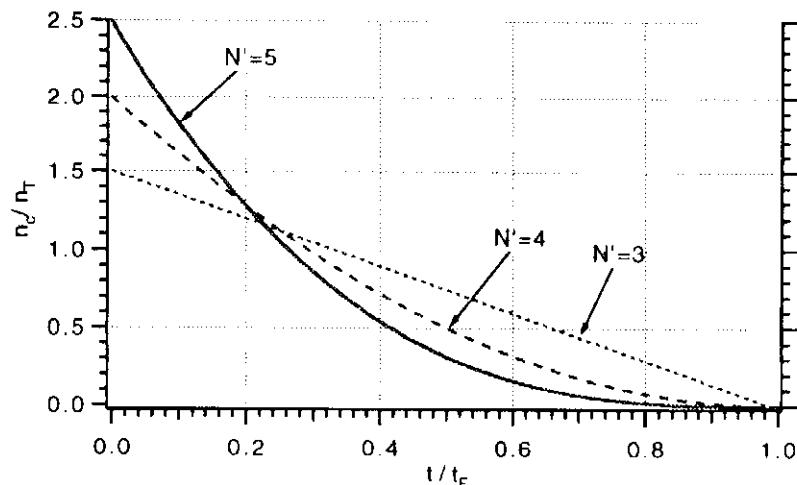


Fig. 8.4 Normalized acceleration for augmented proportional navigation to hit a maneuvering target.

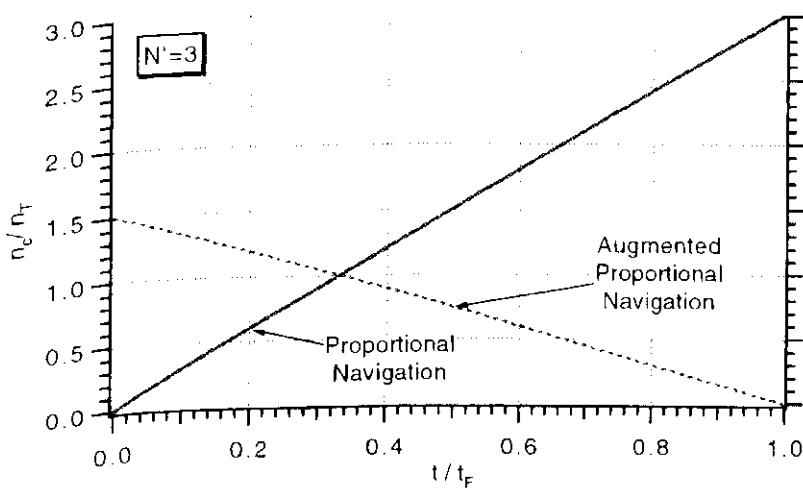


Fig. 8.5 Augmented proportional navigation requires less acceleration capability of missile for $\{N' = 3\}$.

guidance. However, for an effective navigation ratio of 5, augmented proportional navigation requires a larger maximum acceleration compared with proportional navigation guidance.

Comparative plots of proportional and augmented proportional navigation missile acceleration profiles for different values of effective navigation ratio due to a target maneuver appear in Figs. 8.5–8.7. Figure 8.5 shows that, with an effective navigation ratio of 3, augmented proportional navigation requires less acceleration capability of the missile than proportional navigation. This figure also indicates that augmented proportional navigation requires much less total acceleration than proportional navigation. This is not surprising because augmented proportional navigation is making use of extra information, namely, knowledge of the target maneuver. It is reasonable that this knowledge should enable the missile to maneuver in a more efficient manner.

Figure 8.6 shows that for an effective navigation ratio of 4 the maximum acceleration required by both guidance laws is the same. The total acceleration begins to be less with augmented proportional navigation at a normalized time of 0.3.

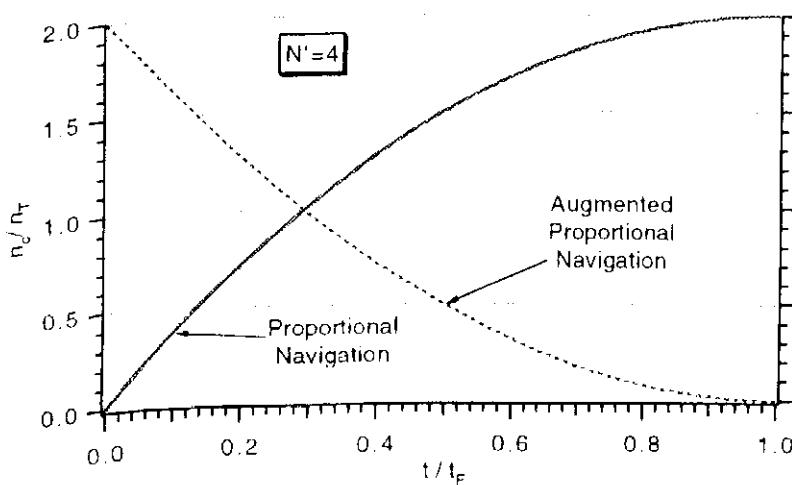


Fig. 8.6 Augmented proportional navigation requires the same acceleration capability of missile for $N' = 4$.

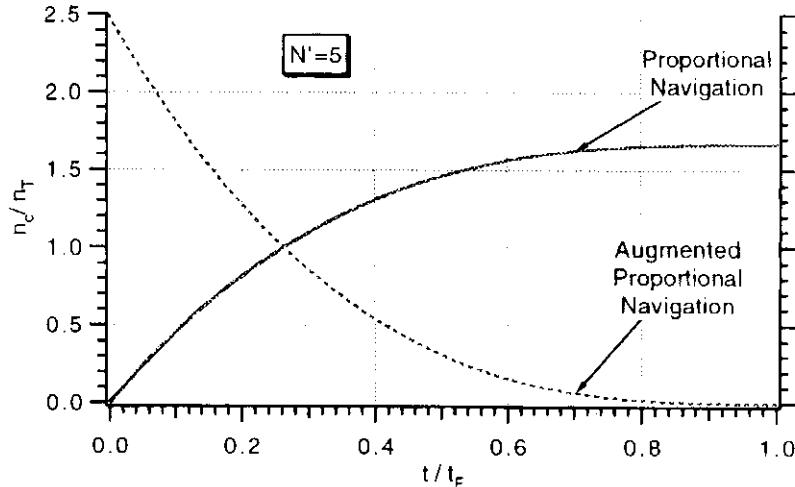


Fig. 8.7 Augmented proportional navigation requires more acceleration capability of missile for $N' = 5$.

This means that 70% of the time-augmented proportional navigation requires less acceleration than proportional navigation to hit a maneuvering target.

Figure 8.7 shows that, when the effective navigation ratio is 5, augmented proportional navigation requires a larger maximum acceleration capability of the missile than does proportional navigation. However, about 75% of the time-augmented proportional navigation requires less missile acceleration than proportional navigation.

It appears from Figs. 8.6 and 8.7 that augmented proportional navigation does not relax the maximum missile acceleration requirements imposed by proportional navigation guidance when the effective navigation ratio is greater than or equal to 4. However, in these cases, augmented proportional navigation appears to require less total acceleration than proportional navigation for most of the flight.

To quantify this observation more precisely, we need a performance index other than maximum acceleration. One possibility is to consider the total acceleration required or to find the area under the acceleration curve. We shall see in later chapters that strategic missiles require fuel to maneuver since they operate outside the atmosphere (i.e., they cannot generate lift by moving control surfaces). In these cases the missile maneuverability is referred to as a lateral divert capability. Lateral divert is directly related to the amount of fuel required by the interceptor to implement the guidance law and effect an intercept outside the atmosphere. The lateral divert is in fact the total area under the absolute value of the acceleration curve. Since missile acceleration is always positive, we can find the lateral divert requirements for proportional navigation by integrating the closed-form solution for the required missile acceleration, or

$$\Delta V_{PN} = \int_0^{t_F} n_c|_{PN} dt = \int_0^{t_F} \frac{N'}{N' - 2} \left[1 - \left(1 - \frac{t}{t_F} \right)^{N'-2} \right] n_T dt$$

After some algebra we obtain

$$\Delta V_{PN} = \frac{N' n_T t_F}{N' - 1}$$

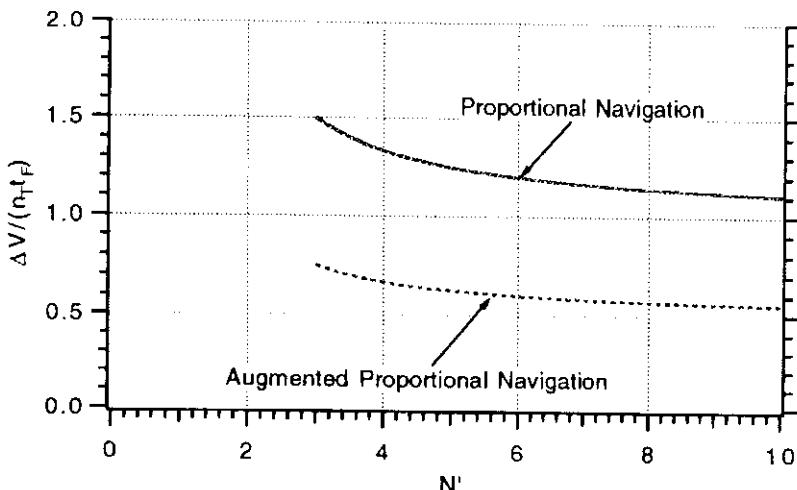


Fig. 8.8 Augmented proportional navigation has reduced divert requirement.

Thus, we can see that increasing the effective navigation ratio makes the lateral divert requirements smaller. Following the same procedure we can express the lateral divert required for augmented proportional navigation. First we must set up the integral as

$$\Delta V_{APN} = \int_0^{t_F} n_c|_{APN} dt = \int_0^{t_F} 0.5n_T N' \left(1 - \frac{t}{t_F}\right)^{N'-2} dt$$

Integration and simplification yields

$$\Delta V_{APN} = \frac{0.5N'n_T t_F}{N' - 1}$$

Figure 8.8 presents a comparative plot of the total energy or lateral divert required by the interceptor as a function of the effective navigation ratio for both guidance laws. The figure shows that the lateral divert requirements decrease with increasing effective navigation ratio for both guidance laws. We can also see from the formulas and figure that augmented proportional navigation always has one-half the lateral divert requirements of proportional navigation, regardless of the effective navigation ratio. Therefore, for strategic applications, augmented proportional navigation is a more fuel-efficient guidance law than proportional navigation.

Derivation of Augmented Proportional Navigation⁴

Thus far we have given a heuristic argument for the augmented proportional navigation guidance law. This is a good approach if the desire is to understand a guidance law concept, but it is not quite adequate for developing more advanced and complex laws.

Our model of the guidance system, for guidance law development, is shown in Fig. 8.9. In this zero-lag model we are saying that relative acceleration is the difference between target acceleration n_T and missile acceleration n_c .

We seek to find a guidance law that is a function of the system states. There are an infinite number of possible guidance laws; thus, it is necessary to state in mathematical terms what the guidance law should do. Certainly we would like to hit the target! Therefore, one feature of the guidance law should be a zero miss

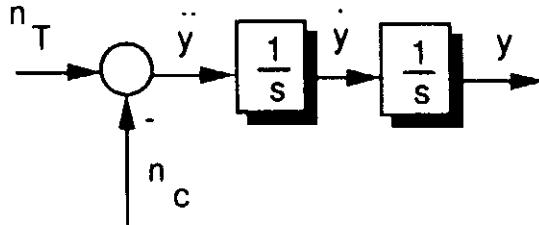


Fig. 8.9 Model for guidance law derivation.

distance requirement. In addition, we would like to hit the target in an efficient manner. In other words, we desire to use minimal total acceleration. A popular and mathematically convenient way of stating the guidance problem to be solved is that we desire to achieve zero miss subject to minimizing the integral of the square of the acceleration command, or

$$y(t_F) = 0 \quad \text{subject to minimizing} \int_0^{t_F} n_c^2(t) dt$$

Unfortunately, if we minimized a more meaningful performance index such as the integral of the absolute value of n_c , the solution would be mathematically intractable. Typically this type of problem with a quadratic performance index is solved using techniques from optimal control theory.^{2,3} However, this class of problem can be solved more easily using the Schwartz inequality.⁴ Before we begin, let us review a few fundamentals. A system of linear differential equations can always be represented in the following state space form:

$$\dot{x} = Fx + Gu$$

The system of Fig. 8.9 can be expressed in state space form as

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{n}_T \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_F \begin{bmatrix} y \\ \dot{y} \\ n_T \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}}_G n_c$$

where F is the 3×3 system dynamics matrix and G is the 3×1 vector. The solution to the state space vector differential equation is given at the final time t_F by the vector relationship.⁵

$$x(t_F) = \Phi(t_F - t)x(t) + \int_t^{t_F} \Phi(t_F - \lambda)G(\lambda)u(\lambda) d\lambda$$

where Φ is the fundamental matrix and is related to F according to

$$\Phi(t) = \mathcal{L}^{-1}[(sI - F)^{-1}]$$

where \mathcal{L}^{-1} is the inverse Laplace transform. This means that in order to find the fundamental matrix we must first invert the matrix $sI - F$ and then find the inverse Laplace transform of the resultant matrix expression.

For the model of Fig. 8.9 the fundamental matrix is found to be

$$\Phi(t) = \begin{bmatrix} 1 & t & 0.5t^2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}$$

Using the preceding fundamental matrix in the solution for the state space vector differential equation and only looking at the first state, we get

$$y(t_F) = y(t) + \dot{y}(t)(t_F - t) + 0.5n_T(t_F - t)^2 - \int_t^{t_F} (t_F - \lambda)n_c(\lambda) d\lambda$$

For convenience let us define the terms

$$f_1(t_F - t) = y(t) + \dot{y}(t)(t_F - t) + 0.5n_T(t_F - t)^2$$

and

$$h_1(t_F - \lambda) = t_F - \lambda$$

Then we can say that

$$y(t_F) = f_1(t_F - t) - \int_t^{t_F} h_1(t_F - \lambda)n_c(\lambda) d\lambda$$

For the condition in which we have zero miss distance [$y(t_F) = 0$], we can rewrite the preceding equation as

$$f_1(t_F - t) = \int_t^{t_F} h_1(t_F - \lambda)n_c(\lambda) d\lambda$$

If we apply the Schwartz inequality to the preceding expression, we get the relationship

$$f_1^2(t_F - t) \leq \int_t^{t_F} h_1^2(t_F - \lambda) d\lambda \int_t^{t_F} n_c^2(\lambda) d\lambda$$

Expressing the preceding inequality in terms of the desired acceleration command, we get

$$\int_t^{t_F} n_c^2(\lambda) d\lambda \geq \frac{f_1^2(t_F - t)}{\int_t^{t_F} h_1^2(t_F - \lambda) d\lambda}$$

The integral of the square of the commanded acceleration will be minimized when the equality sign holds in the preceding inequality. According to the Schwartz inequality, the equality sign holds when

$$n_c(\lambda) = kh_1(t_F - \lambda)$$

This means that the integral of the squared acceleration is minimized when

$$k = \frac{f_1(t_F - t)}{\int_t^{t_F} h_1^2(t_F - \lambda) d\lambda}$$

Therefore, the commanded acceleration is given by

$$n_c = \left[\frac{f_1(t_F - t)}{\int_t^{t_F} h_1^2(t_F - \lambda) d\lambda} \right] h_1(t_F - t)$$

Substitution yields the feedback control guidance law

$$n_c = \frac{3(y + \dot{y}t_{go} + 0.5n_T t_{go}^2)}{t_{go}^2}$$

where

$$t_{go} = t_F - t$$

We can see that the “optimal” guidance law is simply augmented proportional navigation with an effective navigation ratio of 3. The effective navigation ratio turns out to be 3 because we are minimizing the integral of the square of the acceleration. If we were to minimize another function of acceleration, we would get a different answer for the optimal effective navigation ratio. It is still important to note that the optimal guidance law is proportional to the zero effort miss and inversely proportional to the square of time to go.

Influence of Time Constants

Thus far we have seen that augmented proportional navigation may offer considerable advantages, in terms of required missile acceleration needed to effect an intercept, over the proportional navigation guidance law. It has been demonstrated that, under certain circumstances, augmented proportional navigation may be considered to be an optimal guidance law for a zero-lag guidance system. Let us see how augmented proportional navigation performs when there is a guidance system lag.

Consider a case where the flight time is 10 s, the missile has an effective navigation ratio of 4, and there is a 3-g target maneuver. Figure 8.10 displays the resultant commanded acceleration profile for both proportional and augmented proportional navigation. Since the effective navigation ratio is 4, we can compare these results directly with the normalized zero-lag guidance system results of Fig. 8.6. Figure 8.10 indicates that the lag does not change the value of the maximum value of acceleration for both guidance laws. In addition, the lag does not change the fact that augmented proportional navigation requires less acceleration

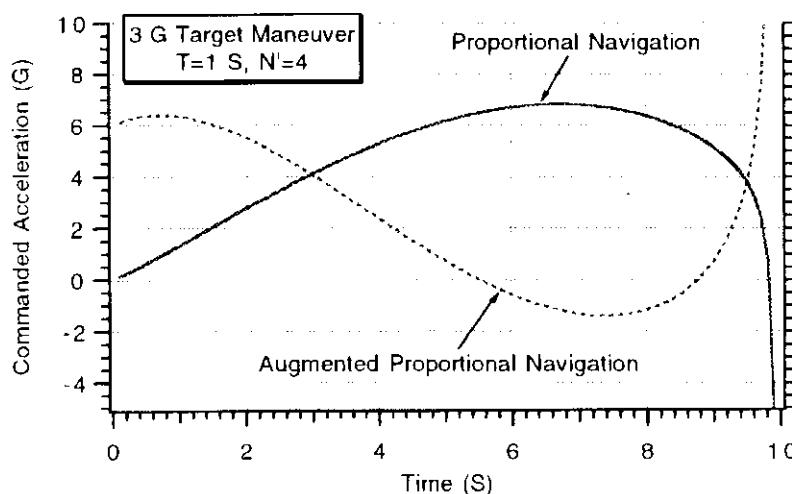


Fig. 8.10 Guidance law acceleration requirements in presence of single-lag guidance system.

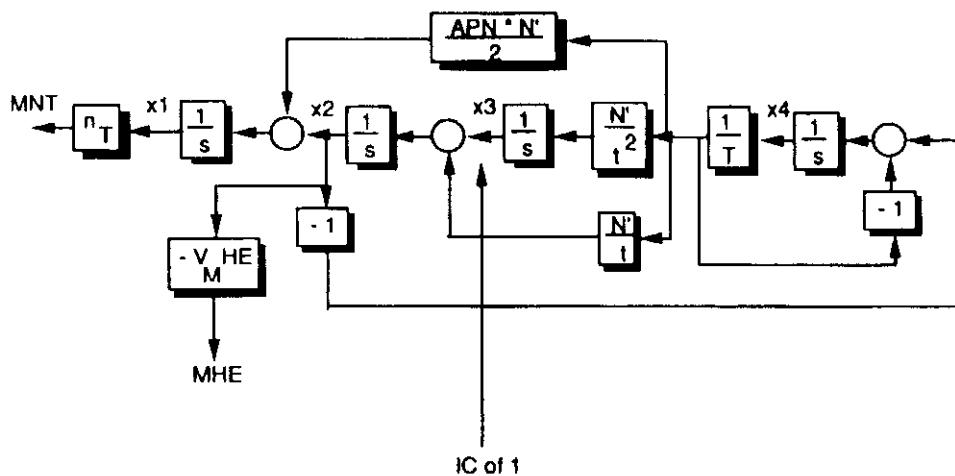


Fig. 8.11 Adjoint for investigating guidance laws in single-lag guidance system.

than proportional navigation about 70% of the time. The lag does slightly alter the shape of both acceleration profiles in the sense that the curves are not completely monotonically decreasing (*APN*) or monotonically increasing (*PN*).

Having seen that the lag does not change trends in acceleration, let us use the method of adjoints to perform a miss distance sensitivity analysis for both guidance laws in the presence of the lag. The adjoint block diagram of a single time constant system appears in Fig. 8.11. In this diagram we have proportional navigation if *APN* = 0 and augmented proportional navigation if *APN* = 1.

The FORTRAN listing for the adjoint simulation appears in Listing 8.1. The guidance system time constant is represented in FORTRAN by *TAU*. The listing shows that an initial condition of unity is applied to the *x*3 integrator to make a miss distance adjoint. The four adjoint differential equations can be found after statement label 200.

Two adjoint runs were made in which proportional navigation (*APN* = 0) and augmented proportional navigation (*APN* = 1) were used. The error source was a 3-g target maneuver in the presence of a 1-s flight-control system time constant. The value of the effective navigation ratio was 4. We can see from Fig. 8.12 that neither guidance law is superior from a miss distance point of

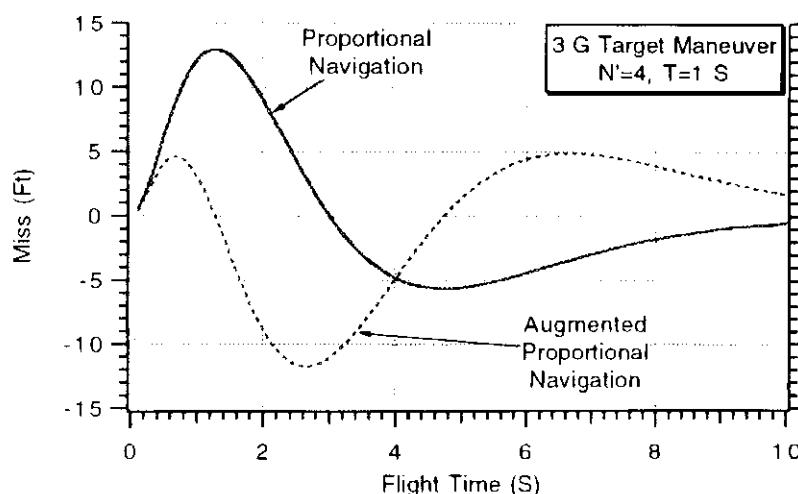


Fig. 8.12 Both guidance laws are comparable in terms of miss distance.

Listing 8.1 Adjoint simulation for inverse

```

INTEGER STEP
DATA XNT,XNP,TAU,TF,VM,HEDEG/ 96.6,4,1,10,3000.,-20./
DATA APN/1./
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
T=0.
S=0.
TP=T+.00001
X1=0
X2=0
X3=1
X4=0
H=.01
HE=HEDEG/57.3
10 IF(TP>(TF-.00001))GOTO 999
S=S+H
X1OLD=X1
X2OLD=X2
X3OLD=X3
X4OLD=X4
STEP=1
GOTO 200
66 STEP=2
X1=X1+H*X1D
X2=X2+H*X2D
X3=X3+H*X3D
X4=X4+H*X4D
TP=TP+H
GOTO 200
55 CONTINUE
X1=(X1OLD+X1)/2+.5*H*X1D
X2=(X2OLD+X2)/2+.5*H*X2D
X3=(X3OLD+X3)/2+.5*H*X3D
X4=(X4OLD+X4)/2+.5*H*X4D
IF(S<.09999)GOTO 10
S=0.
XMNT=XNT*X1
XMHE=-VM*HE*X2
WRITE(*,97)TP,XMNT,XMHE
WRITE(1,97)TP,XMNT,XMHE
GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
X1D=X2+X4*XNP*APN/(2.*TAU)
X2D=X3+XNP*X4/(TAU*TP)
X3D=XNP*X4/(TAU*TP*TP)
X4D=-X4/TAU-X2
IF(STEP-1)66,66,55
999 CONTINUE
PAUSE
CLOSE(1)
END

```

view (assuming the system is linear and we do not have acceleration saturation). Augmented proportional navigation yields smaller miss distances for shorter flight times, whereas proportional navigation yields smaller miss distances for longer flight times.

Of course in a zero-lag system both guidance laws would always yield zero miss distance. So we can see that, although the lag does not significantly alter the acceleration signature, it does cause miss distance!

Optimal Guidance^{1,3,4}

We have observed that by making use of target acceleration information we could derive a guidance law to reduce the missile acceleration requirements. It has been demonstrated in the previous example that guidance system lags cause miss distance. Generally, larger guidance system time constants yield larger miss distances (except for parasitic effects and some types of noise disturbances). If we had knowledge of the guidance system dynamics, could we derive a guidance law to eliminate miss distance? Mathematically speaking, the answer is yes!

A single-lag guidance system model for guidance law development is presented in Fig. 8.13. This model is identical to the one of Fig. 8.9, except that the guidance system dynamics has been represented by a single lag, or

$$\frac{n_L}{n_c} = \frac{1}{1 + sT}$$

where n_c is the commanded acceleration, n_L the achieved acceleration, and T the guidance system or flight-control system time constant.

Figure 8.13 can be expressed in state space form as

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{n}_T \\ \dot{n}_L \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{T} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ n_T \\ n_L \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{T} \end{bmatrix} n_c$$

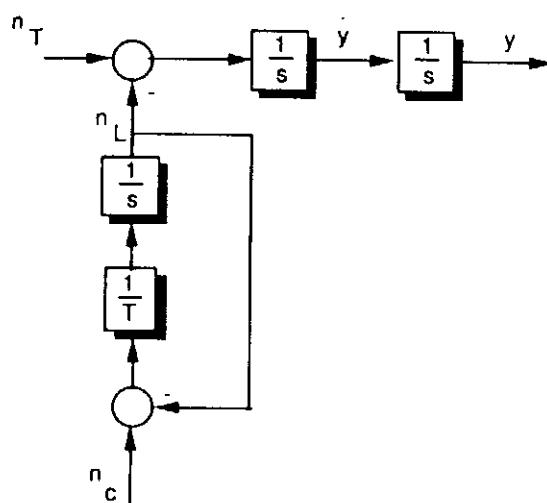


Fig. 8.13 Single-lag model for guidance law development.

Therefore, the fundamental matrix can be found to be

$$\Phi(t) = \begin{bmatrix} 1 & t & 0.5t^2 & -tT + T^2(1 - e^{-t/T}) \\ 0 & 1 & t & -T(1 - e^{-t/T}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{-t/T} \end{bmatrix}$$

Recall that we still seek a guidance law that yields zero miss subject to minimizing the integral of the square of the commanded acceleration, or

$$y(t_F) = 0 \quad \text{subject to minimizing } \int_0^{t_F} n_c^2(t) dt$$

Using a procedure similar to that used in the previous section, we can derive the important quantities

$$\begin{aligned} f_1(t_F - t) &= y + \dot{y}(t_F - t) + 0.5n_T(t_F - t)^2 \\ &\quad - T^2 n_L \left[e^{-(t_F-t)/T} + \frac{(t_F - t)}{T} - 1 \right] \\ h_1(t_F - \lambda) &= t_F - \lambda - T[1 - e^{-(t_F-\lambda)/T}] \end{aligned}$$

Calculating

$$\int_t^{t_F} h_1^2(t_F - \lambda) d\lambda = T^3 \left(0.5 - 0.5 e^{-2t_{go}/T} - \frac{2t_{go}e^{-t_{go}/T}}{T} - \frac{t_{go}^2}{T^2} + \frac{t_{go}}{T} + \frac{t_{go}^3}{3T^3} \right)$$

and defining

$$x = \frac{t_{go}}{T}$$

we obtain the optimal guidance law

$$n_c = \frac{N'}{t_{go}^2} [y + \dot{y}t_{go} + 0.5n_T t_{go}^2 - n_L T^2(e^{-x} + x - 1)]$$

where the bracketed quantity is the zero effort miss, and the effective navigation ratio is no longer a constant but is related to the guidance system time constant and time to go by the relationship

$$N' = \frac{6x^2(e^{-x} - 1 + x)}{2x^3 + 3 + 6x - 6x^2 - 12xe^{-x} - 3e^{-2x}}$$

The effective navigation ratio for the optimal guidance law is displayed in normalized form in Fig. 8.14. We can see that at the beginning of the flight (long time to go) the effective navigation ratio is approximately constant and is approaching 3. As we get closer to intercept (small time to go), the effective navigation ratio grows considerably.

In Fig. 8.15 a theoretically optimal guidance law is implemented in a single-lag guidance system. It is assumed that precise knowledge of the target and missile acceleration is available. The only error disturbance shown in this guidance system is target maneuver n_T .

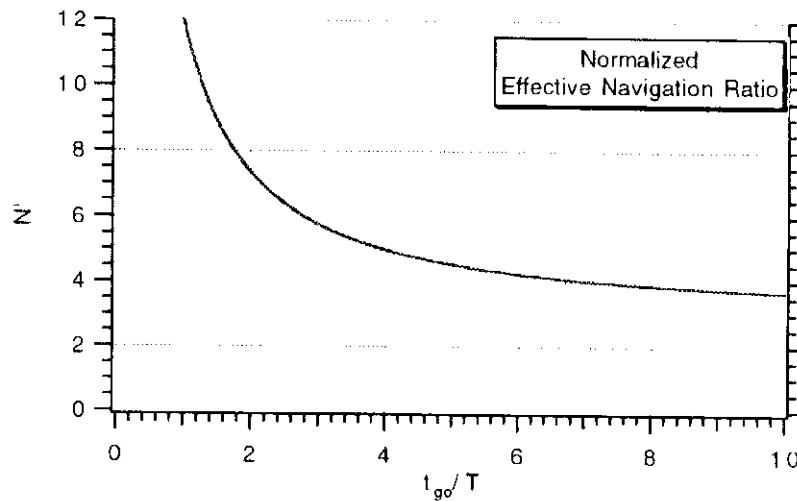


Fig. 8.14 Normalized effective navigation ratio for optimal guidance law.

The guidance law has been represented with control gains C_1-C_4 . These gains are functions of the time to go to intercept and the guidance system time constant. They are defined as

$$C_1 = \frac{N'}{t_{go}^2}$$

$$C_2 = \frac{N'}{t_{go}}$$

$$C_3 = 0.5N'$$

$$C_4 = \frac{-N'(e^{-x} + x - 1)}{x^2}$$

where N' is the optimal effective navigation ratio, which has been defined previously.

A case was run for a 1-s guidance system time constant in which the flight time was 10 s and the error disturbance was a 3-g target maneuver. Figure 8.16 shows the acceleration profile for proportional navigation ($N' = 4$), augmented

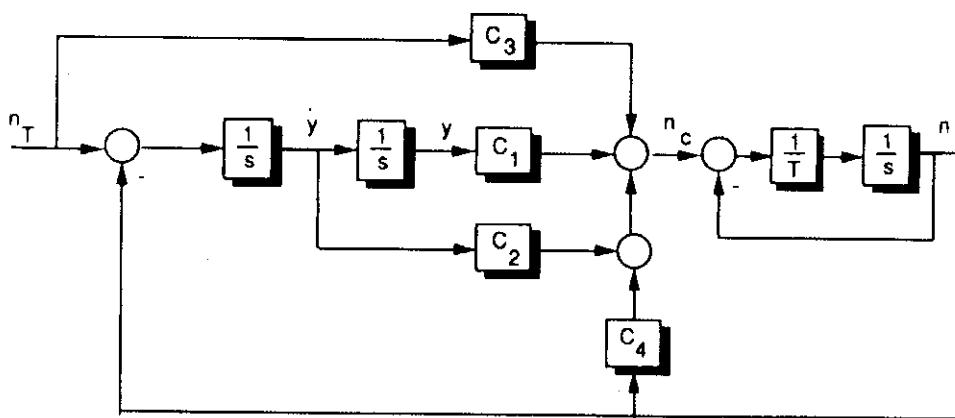


Fig. 8.15 Theoretical optimal single time constant guidance system.

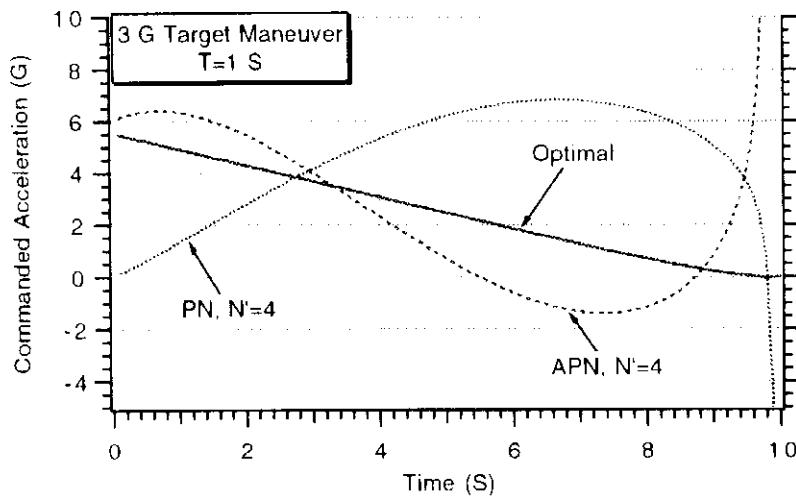


Fig. 8.16 Acceleration comparison for various guidance laws.

proportional navigation ($N' = 4$), and optimal guidance. If we compare this figure with Fig. 8.5 we can see that the optimal guidance acceleration profile appears to be identical to the augmented proportional navigation acceleration profile *for a zero-lag guidance system and an effective navigation ratio of 3!* This means that the guidance law is dynamically canceling out the guidance system dynamics.

If the optimal guidance law were attempting to make the single-lag guidance system appear to be a zero-lag augmented proportional navigation guidance system, then the miss distance should be zero—just as it is in a zero-lag system. To test this theory, an adjoint block diagram of a single-lag optimal guidance system was constructed from Fig. 8.15 and appears in Fig. 8.17. The control gains become C^* because they must be reversed in time according to adjoint theory. The miss sensitivities due to target maneuver MNT and heading error MHE are indicated in the figure.

The FORTRAN listing of the adjoint simulation of the optimal single time constant guidance system with various guidance law options appears in Listing 8.2.

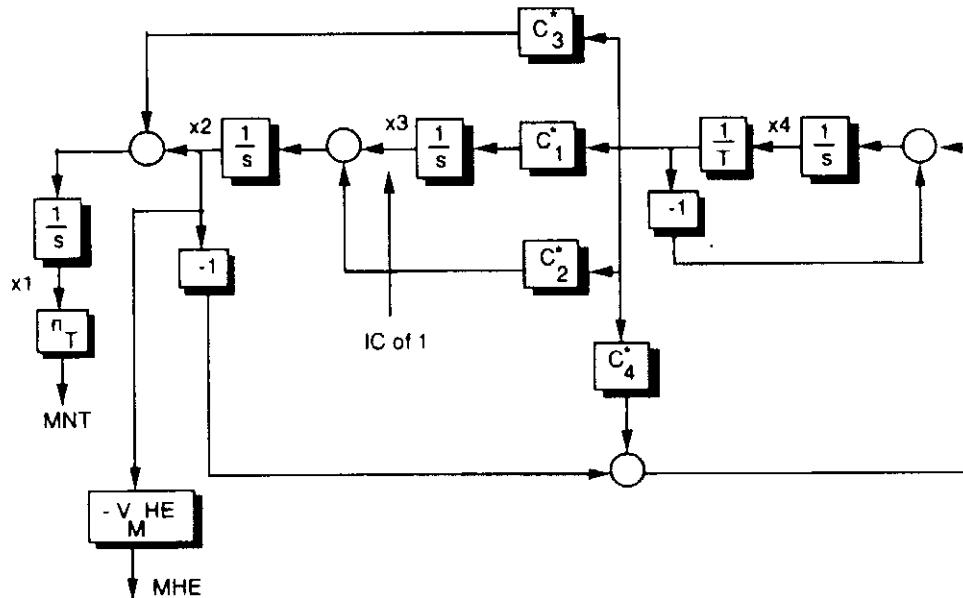


Fig. 8.17 Adjoint of theoretical optimal single time constant guidance system.

Listing 8.2 Adjoint simulation of optimal guidance system

```

INTEGER STEP,APN
DATA XNT,XNP,TAU,TF,VM,HEDEG/ 96.6,4,1,10,3000.,-20./
DATA APN/0/
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
T=0.
S=0.
TP=TP+.00001
X1=0.
X2=0.
X3=1.
X4=0.
XNPP=0.
H=.01
HE=HEDEG/57.3
10 IF(TP>(TF-.00001))GOTO 999
S=S+H
X1OLD=X1
X2OLD=X2
X3OLD=X3
X4OLD=X4
STEP=1
GOTO 200
66 STEP=2
X1=X1+H*X1D
X2=X2+H*X2D
X3=X3+H*X3D
X4=X4+H*X4D
TP=TP+H
GOTO 200
55 CONTINUE
X1=(X1OLD+X1)/2+.5*H*X1D
X2=(X2OLD+X2)/2+.5*H*X2D
X3=(X3OLD+X3)/2+.5*H*X3D
X4=(X4OLD+X4)/2+.5*H*X4D
IF(S<.09999)GOTO 10
S=0.
XMNT=XNT*X1
XMHE=-VM*HE*X2
WRITE(*,97)TP,XMNT,XMHE
WRITE(1,97)TP,XMNT,XMHE
GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
TGO=TP+.00001
IF(APN.EQ.0)THEN
    C1=XNP/(TGO*TGO)
    C2=XNP/TGO
    C3=0.
    C4=0.

```

(Contd.)

Listing 8.2 (Continued)

```

ELSEIF(APN.EQ.1)THEN
  C1=XNP/(TGO*TGO)
  C2=XNP/TGO
  C3=.5*XNP
  C4=0.
ELSE
  X=TGO/TAU
  TOP=6.*X*X*(EXP(-X)-1.+X)
  BOT1=2*X*X*X+3.+6.*X-6.*X*X
  BOT2=-12.*X*EXP(-X)-3.*EXP(-2.*X)
  XNPP=TOP/(.0001+BOT1+BOT2)
  C1=XNPP/(TGO*TGO)
  C2=XNPP/TGO
  C3=.5*XNPP
  C4=-XNPP*(EXP(-X)+X-1.)/(X*X)
ENDIF
X1D=X2+C3*X4/TAU
X2D=X3+C2*X4/TAU
X3D=C1*X4/TAU
X4D=-X4/TAU-X2+C4*X4/TAU
IF(STEP-1)66,66,55
999  CONTINUE
PAUSE
CLOSE(1)
END

```

We can see from the listing that the parameter *APN* determines the guidance law used. *APN* = 0 denotes proportional navigation, *APN* = 1 represents augmented proportional navigation, and *APN* = 2 defines optimal guidance.

Figure 8.18 shows the miss sensitivity of all the guidance laws to a 3-g step target maneuver in the presence of a single-lag guidance system with a time constant of 1 s. We can see that the optimal guidance law always yields zero miss distance. Therefore, as predicted, the optimal guidance system is attempting to

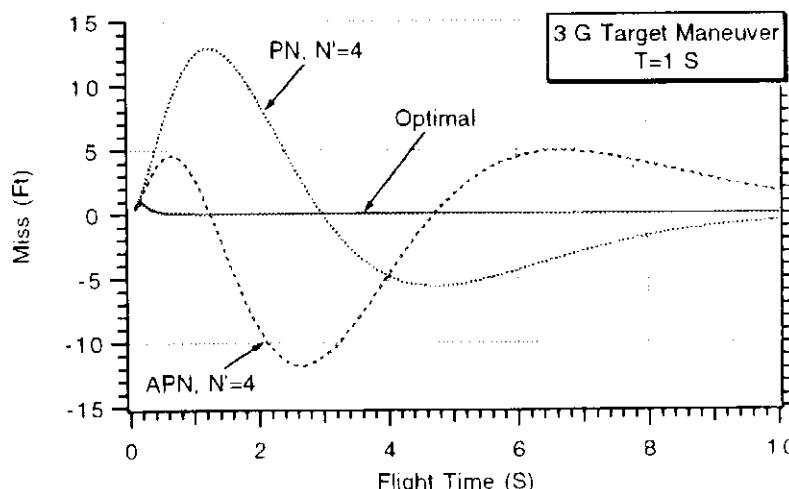


Fig. 8.18 Optimal guidance system does not have miss distance.

make the single time constant guidance system appear to be a zero-lag augmented proportional navigation guidance system with an effective navigation ratio of 3. The interested reader is referred to Appendix A to see how the optimal guidance system performs when time to go information is degraded.

Summary

In this chapter we have shown how some advanced guidance laws can be derived both heuristically and mathematically. The method of adjoints was used to show the performance advantages of the more advanced guidance laws. In practice, one must also test the advanced guidance concepts in the presence of parasitic effects to ensure that performance is still better than proportional navigation.¹

References

¹Nesline, F. W., and Zarchan, P., "A New Look at Classical Versus Modern Homing Guidance," *Journal of Guidance and Control*, Vol. 4, No. 1, Jan.-Feb. 1981, pp. 78-85.

²Bryson, A. E., and Ho, Y. C., *Applied Optimal Control*, Blaisdell, Waltham, MA, 1969.

³Cottrell, R. G., "Optimal Intercept Guidance for Short-Range Tactical Missiles," *AIAA Journal*, Vol. 9, July 1971, pp. 1414-1415.

⁴Kliger, I., "A Simple Derivation of Certain Optimal Control Laws," Raytheon, Memo SAD-1230, Bedford, MA, Nov. 1970.

⁵Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.



Kalman Filters and the Homing Loop

Introduction

KALMAN filtering concepts can be used in the homing loop to produce an optimal digital noise filter. The filter is considered optimal because the errors in the estimates of the system states are minimized in the least-squares sense. We shall see that, for the filter to be truly optimal, the statistics of the measurement and process noise must be known. If this information is lacking or inaccurate, the filter performance will degrade. However, we shall also see that in homing loop applications the Kalman filter cannot only perform very well with significant errors in the knowledge of the statistics, but it may even be desirable to lie to the filter to achieve a desired filter bandwidth. Finally, we shall demonstrate that when Kalman filtering concepts are used it is possible to apply advanced guidance techniques and substantially improve system performance.

Theoretical Equations¹

For linear systems Kalman filters can provide optimal estimators in the least-squares sense. To apply Kalman filtering theory, our model of the real world must be described by a matrix differential equation of the form

$$\dot{x} = Fx + Gu + w$$

where x is a column vector describing the states of the system, F the system dynamics matrix, u a known control vector, and w a white noise process. There is a process noise matrix Q that is related to the process noise vector according to

$$Q = E[w w^T]$$

In other words, Q is the expectation of the white process noise times its transpose. The filter will be optimal if the measurements available are linearly related to the states according to

$$z = Hx + v$$

where z is the measurement vector, H the measurement matrix, and v the white noise measurement. The measurement noise matrix R is related to the measurement noise vector v according to

$$R = E[v v^T]$$

The preceding relationships are valid for continuous systems. Since we are not taking measurements continuously but plan to receive information every T_s ,

seconds, we need to discretize our system model. The fundamental matrix Φ is related to the system dynamics matrix according to

$$\Phi(t) = \mathcal{L}^{-1}\{[sI - F]^{-1}\}$$

where I is the identity matrix and \mathcal{L}^{-1} the inverse Laplace transform. For discrete systems we can say that the discrete transition matrix is given by

$$\Phi_k = \Phi(T_s)$$

where T_s is the sampling time. In other words, the discrete fundamental matrix is simply the continuous fundamental matrix evaluated at the sampling time. The discrete form of the measurement equation is now

$$z_k = Hx_k + v_k$$

and

$$R_k = \sigma_n^2$$

where σ_n^2 is the variance of the measurement noise. The resultant form of the discrete Kalman filter is given by the recursive relationship in matrix form

$$\hat{x}_k = \Phi_k \hat{x}_{k-1} + G_k u_{k-1} + K_k (z_k - H\Phi_k \hat{x}_{k-1} - HG_k u_{k-1})$$

where G_k is obtained from

$$G_k = \int_0^{T_s} \Phi(\tau) G \, d\tau$$

and K_k represents the Kalman gain matrix. The Kalman gains are computed, while the filter is operating, from the matrix Riccati equations. The Riccati equations are a set of recursive matrix equations given by

$$\begin{aligned} M_k &= \Phi_k P_{k-1} \Phi_k^T + Q_k \\ K_k &= M_k H^T [H M_k H^T + R_k]^{-1} \\ P_k &= (I - K_k H) M_k \end{aligned}$$

where P_k is a covariance matrix representing errors in the state estimates before an update, and M_k is the covariance matrix representing errors in the state estimates after an update. The discrete process noise matrix Q_k can be found from the continuous process noise matrix Q and the fundamental matrix according to

$$Q_k = \int_0^{T_s} \Phi(\tau) Q \Phi^T(\tau) \, d\tau$$

To start the Riccati equations, we need an initial covariance matrix P_0 . This matrix represents the initial uncertainty in the error in the estimate. Choosing appropriate values for this initial matrix is in itself an art.¹

Application to Homing Loop

To demonstrate the utility of Kalman filtering, let us consider the zero-lag homing loop model of Fig. 9.1. In this guidance system we measure noisy relative

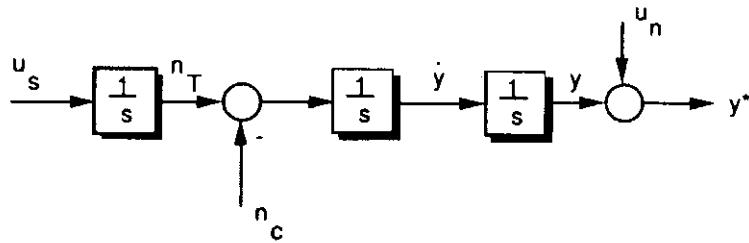


Fig. 9.1 Homing loop model for Kalman filter development.

position y^* and are attempting to estimate relative position, relative velocity, and target acceleration. In our model the missile acceleration n_c is assumed to be known, and the target acceleration is considered to be modeled as a white noise through an integrator. We have shown in Chapter 4 mathematically that the shaping filter equivalent of a target maneuver with constant amplitude but random starting time (where the starting time is uniformly distributed over the flight time) is white noise through an integrator. According to the results of Chapter 4, the spectral density of this white noise process is given by

$$\Phi_s = n_T^2 / t_F$$

where n_T is the maneuver level and t_F the flight time. In Chapter 4 we also showed via a numerical experiment that this model is statistically equivalent to a maneuver of constant amplitude whose starting time is equally likely to occur anywhere during the flight.

We can express the model of Fig. 9.1 in state space form as

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{n}_T \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ n_T \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} n_c + \begin{bmatrix} 0 \\ 0 \\ u_s \end{bmatrix}$$

F G w

In the previous section we showed that the fundamental matrix could be found from the system dynamics matrix. After some computation the fundamental matrix for the model of Fig. 9.1 turns out to be

$$\Phi(t) = \begin{bmatrix} 1 & t & 0.5t^2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}$$

or in discrete form

$$\Phi_k = \begin{bmatrix} 1 & T_s & 0.5T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}$$

The measurement equation can also be expressed in discrete form as

$$y_k^* = [1 \ 0 \ 0] \begin{bmatrix} y_k \\ \dot{y}_k \\ n_{T_k} \end{bmatrix} + u_k$$

H

where the variance of u_k , known as R_k , is given by σ_u^2 . The discrete form of G can

be found from

$$\mathbf{G}_k = \int_0^{T_s} \Phi(\tau) \mathbf{G}(\tau) d\tau = \begin{bmatrix} -0.5T_s^2 \\ -T_s \\ 0 \end{bmatrix}$$

The Kalman filter for the model of Fig. 9.1 can now be expressed in matrix form as

$$\begin{bmatrix} \hat{y}_k \\ \hat{\dot{y}}_k \\ \hat{n}_{T_k} \end{bmatrix} = \begin{bmatrix} 1 & T_s & 0.5T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{y}_{k-1} \\ \hat{\dot{y}}_{k-1} \\ \hat{n}_{T_{k-1}} \end{bmatrix} + \begin{bmatrix} -0.5T_s^2 \\ -T_s \\ 0 \end{bmatrix} n_{c_{k-1}}$$

$$+ \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix} \left[y_k^* - [100] \begin{bmatrix} 1 & T_s & 0.5T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{y}_{k-1} \\ \hat{\dot{y}}_{k-1} \\ \hat{n}_{T_{k-1}} \end{bmatrix} - [100] \begin{bmatrix} -0.5T_s^2 \\ -T_s \\ 0 \end{bmatrix} n_{c_{k-1}} \right]$$

This Kalman filter is shown in block diagram form as part of the homing loop in Fig. 9.2. In this diagram z^{-1} represents a pure delay so that $z^{-1}y_k$ means y_{k-1} . In our model the measurement of the line-of-sight angle λ_k^* is corrupted by noise. We create a pseudomeasurement of relative position y_k^* by a multiplication of the line-of-sight angle measurement by our estimate or measurement of the range from missile to target. The Kalman filter then provides optimal estimates of relative position, relative velocity, and target acceleration. In this model we are using proportional navigation guidance where the guidance command is related to the

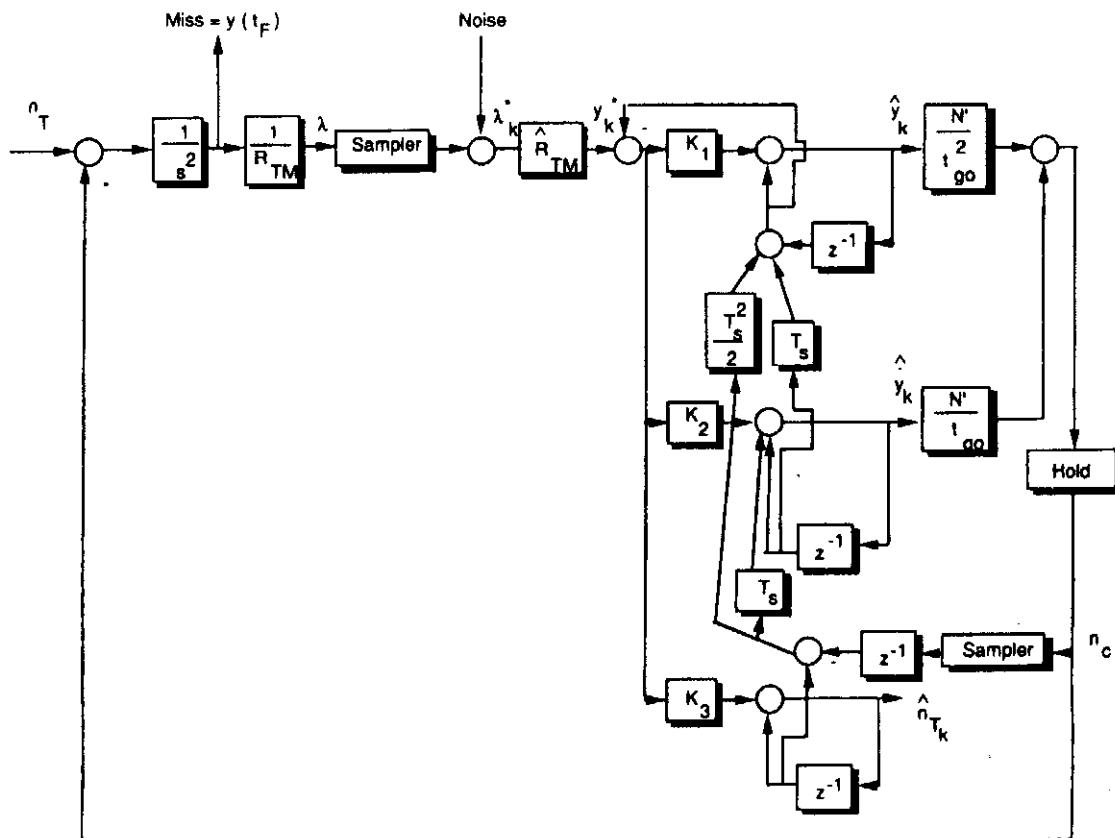


Fig. 9.2 Kalman filter as part of homing loop.

state estimates according to

$$n_{ck}|_{\text{PN}} = \frac{N'}{t_{go}^2} \hat{y}_k + \frac{N'}{t_{go}} \dot{\hat{y}}_k$$

It is easy to show that this command is mathematically equivalent to the more recognizable form of proportional navigation, or

$$n_{ck}|_{\text{PN}} = N' V_c \hat{\lambda}$$

Kalman Gains

In order for the Kalman filter to operate, we need to first compute the filter gains K_k . These gains are obtained from a set of recursive equations known as the matrix Riccati equations, which were stated in the first section. The first of the Riccati equations is

$$M_k = \Phi_k P_{k-1} \Phi_k^T + Q_k$$

where M_k represents the covariance matrix of errors in the estimates after updates. For the three-state system of Fig. 9.2, this matrix can be expanded in scalar form by multiplying out the matrices and by recognizing that M_k is symmetric. Substitution of the necessary matrices yields

$$M = \begin{bmatrix} 1 & T_s & 0.5T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{12} & P_{22} & P_{23} \\ P_{13} & P_{23} & P_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ T_s & 1 & 0 \\ 0.5T_s^2 & T_s & 1 \end{bmatrix}$$

$$+ \Phi_s \begin{bmatrix} \frac{T_s^5}{20} & \frac{T_s^4}{8} & \frac{T_s^3}{6} \\ \frac{T_s^4}{8} & \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ \frac{T_s^3}{6} & \frac{T_s^2}{2} & T_s \end{bmatrix}$$

The second Riccati equation is used to obtain the Kalman gains. It appears from the second Riccati equation,

$$K_k = M_k H^T [H M_k H^T + R_k]^{-1}$$

that it is necessary to take a matrix inverse. However, for the model of Fig. 9.1, the R_k matrix is 1×1 ; hence, we can take the scalar inverse by inspection and obtain

$$K_1 = \frac{M_{11}}{M_{11} + \sigma_N^2}$$

$$K_2 = \frac{M_{12}}{M_{11} + \sigma_N^2}$$

$$K_3 = \frac{M_{13}}{M_{11} + \sigma_N^2}$$

Finally, the third Riccati equation is used to obtain the covariance matrix of the errors in the estimates before an update. The third equation,

$$P_k = (I - K_k H)M_k$$

can easily be expanded to

$$P = \begin{bmatrix} (1 - K_1)M_{11} & (1 - K_1)M_{12} & (1 - K_1)M_{13} \\ -K_2 M_{11} + M_{12} & -K_2 M_{12} + M_{22} & -K_2 M_{13} + M_{23} \\ -K_3 M_{11} + M_{13} & -K_3 M_{12} + M_{23} & -K_3 M_{13} + M_{33} \end{bmatrix}$$

Numerical Examples

To start the Riccati equations, we need an initial covariance matrix P_0 . A particularly useful form for the homing loop model considered is

$$P_0 = \begin{bmatrix} \sigma_{\text{noise}}^2 & 0 & 0 \\ 0 & \left[\frac{V_M H E}{57.3} \right]^2 & 0 \\ 0 & 0 & n_T^2 \end{bmatrix}$$

where only diagonal elements are used. The initial value of the error in the estimate of position is simply the variance of the measurement noise. The initial guess in the velocity error estimate is related to missile velocity and the expected heading error. Finally, our initial value in the uncertainty in target acceleration is represented by the magnitude of the maximum possible acceleration. This is by no means the only way to initialize the covariance matrix, but it is not bad.

Listing 9.1 presents a FORTRAN listing of a program used to solve the Riccati equations recursively for the Kalman gains. In this program it is assumed that the angular measurement noise is 1 milliradian (mr). This noise must be converted to a positional noise by the multiplication of range. The process noise model is considered to be a target maneuver of amplitude 3 g, with starting time that is uniformly distributed over the 10-s flight time. We can see from the listing that the Riccati equations have been expanded to scalar form and that the symmetry property of the Riccati equations has been exploited.

Figure 9.3 displays the three Kalman gain profiles resulting from solving the Riccati equations with initial conditions as shown in Listing 9.1. We can see that, unlike the constant-gain digital fading memory filter, the Kalman filter has time-varying gains. After an initial transient period, the gains appear to be monotonically increasing. This means that, after awhile, the filter bandwidth is continually increasing.

To see how the filter is performing, we must not only look at the filter gains but must also investigate the accuracy of the various state estimates. The covariance matrix has information on the accuracy of the state estimates if the filter's model of the real world is accurate. If the filter model is not matched to the real world, then the performance projections offered by the covariance matrix are not particularly useful. One way of getting more meaningful performance projections is by placing the three-state Kalman filter in the homing loop.

Listing 9.2 presents an engagement simulation, based upon the model of Fig. 9.2, with the three-state Kalman filter included. In the nominal case we are not using the

Listing 9.1 Listing of FORTRAN program to solve Riccati equations

```

REAL M11,M12,M13,M22,M23,M33,K1,K2,K3
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
VC=4000.
XNT=96.6
VM=3000.
HEDEG=20.
SIGRIN=.001
TS=.1
TF=10.
TS2=TS*TS
TS3=TS2*TS
TS4=TS3*TS
TS5=TS4*TS
PHIN=XNT*XNT/TF
RTM=VC*TF
SIGNOISE=SIGRIN
SIGPOS=RTM*SIGNOISE
SIGN2=SIGPOS**2
P11=SIGN2
P12=0.
P13=0.
P22=(VM*HEDEG/57.3)**2
P23=0.
P33=XNT*XNT
T=0.
H=.01
S=0.
10 IF(T>(TF-.0001))GOTO 999
TGO=TF-T+.000001
RTM=VC*TGO
SIGNOISE=SIGRIN
SIGPOS=RTM*SIGNOISE
SIGN2=SIGPOS**2
M11=P11+TS*P12+.5*TS2*P13+TS*(P12+TS*P22+.5*TS2*P23)
M11=M11+.5*TS2*(P13+TS*P23+.5*TS2*P33)+TS5*PHIN/20.
M12=P12+TS*P22+.5*TS2*P23+TS*(P13+TS*P23+.5*TS2*P33)+TS4*PHIN/8.
M13=P13+TS*P23+.5*TS2*P33+PHIN*TS3/6.
M22=P22+TS*P23+TS*(P23+TS*P33)+PHIN*TS3/3.
M23=P23+TS*P33+.5*TS2*PHIN
M33=P33+PHIN*TS
K1=M11/(M11+SIGN2)
K2=M12/(M11+SIGN2)
K3=M13/(M11+SIGN2)
P11=(1.-K1)*M11
P12=(1.-K1)*M12
P13=(1.-K1)*M13
P22=-K2*M12+M22
P23=-K2*M13+M23
P33=-K3*M13+M33

```

(Contd.)

Listing 9.1 (Continued)

```

      WRITE(*,97)T,K1,K2,K3
      WRITE(1,97)T,K1,K2,K3
      T=T+TS
      GOTO 10
97   FORMAT(4F10.3)
999  CONTINUE
      PAUSE
      CLOSE(1)
      END

```

estimate of the target acceleration for guidance purposes. A careful examination of the listing shows that the simulation is divided into continuous and discrete parts. In the continuous section we are integrating the differential equations for the relative velocity and acceleration using the second-order Runge-Kutta numerical integration technique. In the discrete section we are solving the Riccati equations for the Kalman gains and using the recursive Kalman filter to generate state estimates. We go to the continuous section every integration interval H , and we go to the discrete section every sampling interval T_s . For the simulation to work properly, T_s/H must be an integer.

Figure 9.4 shows that in the nominal case the Kalman filter accurately estimates the 3-g target maneuver after about 3 s. This is consistent with Fig. 9.3, which also shows that it takes about 3 s for the Kalman gains to go through their initial transient period. Note that, after about 5 s, the error in the estimate of target acceleration has been stabilized and is quite small, as shown in Fig. 9.4. The filter's internal prediction of how well it is estimating target acceleration can be found by taking the square root of the third diagonal element in the covariance matrix. Figure 9.5 shows that the single flight error in the estimate of target acceleration agrees with the covariance matrix predictions in the sense that it is within the theoretical bounds approximately 68% of the time.

Thus far the filter knows the truth about the real world in the sense that it knows the measurement and process noise statistics exactly. In practice, since these

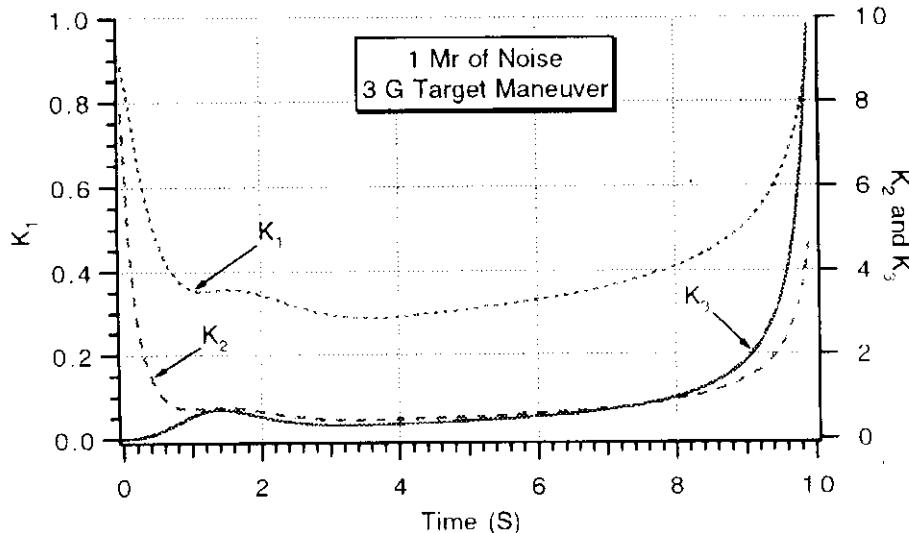


Fig. 9.3 Kalman gain profiles for nominal case.

Listing 9.2 Listing of Kalman filter in homing loop

```

GLOBAL DEFINE
INCLUDE 'quickdraw.inc'
END
REAL M11,M12,M13,M22,M23,M33,K1,K2,K3
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
VC=4000.
XNT=96.6
YIC=0.
VM=3000.
HEDEG=0.
HEDEGFIL=20.
XNP=3.
SIGRIN=.001
TS=.1
APN=0.
TF=10.
Y=YIC
YD=-VM*HEDEG/57.3
YDIC=YD
TS2=TS*TS
TS3=TS2*TS
TS4=TS3*TS
TS5=TS4*TS
PHIN=XNT*XNT/TF
RTM=VC*TF
SIGNOISE=SIGRIN
SIGPOS=RTM*SIGNOISE
SIGN2=SIGPOS**2
P11=SIGN2
P12=0.
P13=0.
P22=(VM*HEDEGFIL/57.3)**2
P23=0.
P33=XNT*XNT
T=0.
H=.01
S=0.
YH=0.
YDH=0.
XNTH=0.
XNC=0.
10 IF(T>(TF-.0001))GOTO 999
YOLD=Y
YDOLD=YD
STEP=1
GOTO 200
66 STEP=2
Y=Y+H*YD
YD=YD+H*YDD

```

(Contd.)

Listing 9.2 (Continued)

```

T=T+H
GOTO 200
55 CONTINUE
Y=.5*(YOLD+Y+H*YD)
YD=.5*(YDOLD+YD+H*YDD)
S=S+H
IF(S<(TS-.0001))GOTO 10
S=0.
TGO=TF-T+.000001
RTM=VC*TGO
SIGNOISE=SIGRIN
SIGPOS=RTM*SIGNOISE
SIGN2=SIGPOS**2
M11=P11+TS*P12+.5*TS2*P13+TS*(P12+TS*P22+.5*TS2*P23)
M11=M11+.5*TS2*(P13+TS*P23+.5*TS2*P33)+TS5*PHIN/20.
M12=P12+TS*P22+.5*TS2*P23+TS*(P13+TS*P23+.5*TS2*P33)+TS4*PHIN/8.
M13=P13+TS*P23+.5*TS2*P33+PHIN*TS3/6.
M22=P22+TS*P23+TS*(P23+TS*P33)+PHIN*TS3/3.
M23=P23+TS*P33+.5*TS2*PHIN
M33=P33+PHIN*TS
K1=M11/(M11+SIGN2)
K2=M12/(M11+SIGN2)
K3=M13/(M11+SIGN2)
P11=(1.-K1)*M11
P12=(1.-K1)*M12
P13=(1.-K1)*M13
P22=-K2*M12+M22
P23=-K2*M13+M23
P33=-K3*M13+M33
CALL GAUSS(XLAMNOISE,SIGNOISE)
YSTAR=RTM*(XLAM+XLAMNOISE)
RES=YSTAR-YH- TS*YDH-.5*TS*TS*(XNTH-XNC)
YH=K1*RES+YH+TS*YDH+.5*TS*TS*(XNTH-XNC)
YDH=K2*RES+YDH+TS*(XNTH-XNC)
XNTH=K3*RES+XNTH
XLAMDH=(YH+YDH*TGO)/(VC*TGO*TGO)
XNC=XNP*VC*XLAMDH+APN*.5*XNP*XNTH
ERRNT=XNT-XNTH
SP33=SQRT(P33)
SP33P=-SP33
WRITE(*,97)T,Y,XNC/32.2,XNT/32.2,XNTH/32.2,ERRNT/32.2,SP33/32.2,
1      SP33P/32.2
WRITE(1,97)T,Y,XNC/32.2,XNT/32.2,XNTH/32.2,ERRNT/32.2,SP33/32.2,
1      SP33P/32.2
GOTO 10
97 FORMAT(8F10.3)
200 CONTINUE
TGO=TF-T+.00001
RTM=VC*TGO

```

(Contd.)

Listing 9.2 (Continued)

```

XLAM=Y/(VC*TGO)
XLAMD=(RTM*YD+Y*VC)/(RTM**2)
YDD=XNT-XNC
IF(STEP-1)66,66,55
999 CONTINUE
PAUSE
CLOSE(1)
END

SUBROUTINE GAUSS(X,SIG)
INTEGER SUM
SUM=0
DO 14 J=1,6
IRAN=Random()
SUM=SUM+IRAN
14 CONTINUE
X=SUM/65536.
X=1.414*X*SIG
RETURN
END

```

statistics are never known a priori, one adjusts the bandwidth of the filter to a desirable level based on other considerations. For example, if we tell the filter that there is 10 mr of angle noise rather than 1 mr, the first Kalman gain value is approximately halved, as shown in Fig. 9.6. Since the filter gain is decreasing, the filter bandwidth must also be decreasing. This means that, when the filter thinks there is more measurement noise, it does more filtering or slows down (lower bandwidth).

An experiment was conducted to illustrate the impact that filter bandwidth has on the resultant estimate. In this experiment the actual noise level was kept at 1 mr, whereas the filter estimate of the measurement noise statistics was changed from

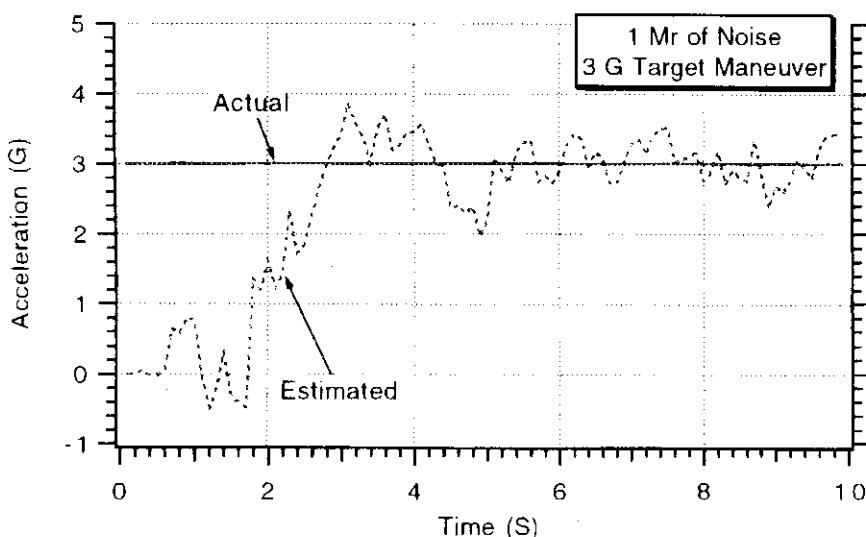


Fig. 9.4 Kalman filter estimate of target maneuver for nominal case.

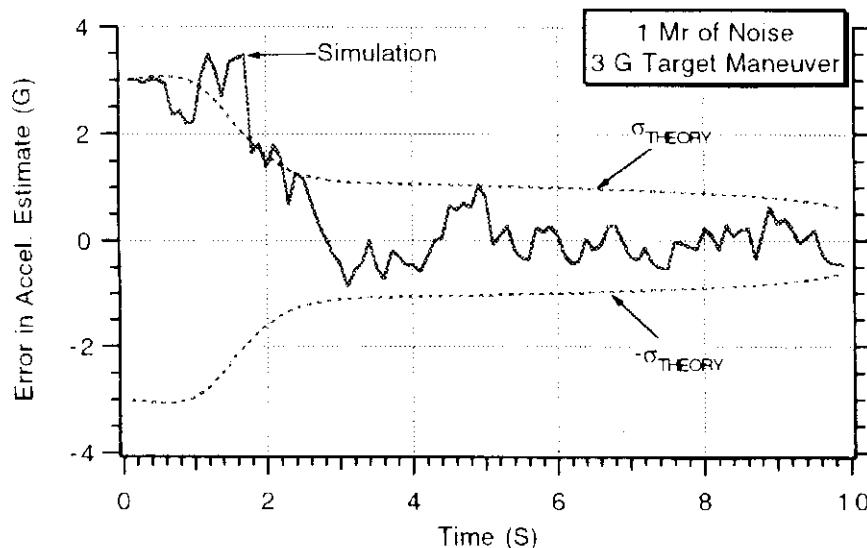


Fig. 9.5 Kalman filter prediction of performance is excellent.

1 mr (matched case, filter assumption correct) to 10 mr (mismatched case, filter assumption wrong). Figure 9.7 shows that when the filter is matched its estimate of target maneuver becomes very good at about 3 s. If the filter thinks there is 10 mr of measurement noise, it takes about 6 s for the filter estimates to become very good. Thus, telling the filter that there is more measurement noise (even if there is not) is a practical way of making the filter more sluggish or decreasing its bandwidth.

If we tell the filter that there is either 1 mr or 10 mr of measurement noise but actually turn the real measurement noise off, we have a deterministic case. We can then make flight time a parameter for the case in which there is a 3-g target maneuver and evaluate system miss distance caused by the target maneuver. Figure 9.8 shows that the faster system ($\sigma = 1$ mr) has less miss distance induced by target maneuver than the slower system ($\sigma = 10$ mr). However, in both cases we can see that the guidance system time constant must be very small since the miss distance sensitivity to target maneuver is small and rapidly approaches zero.

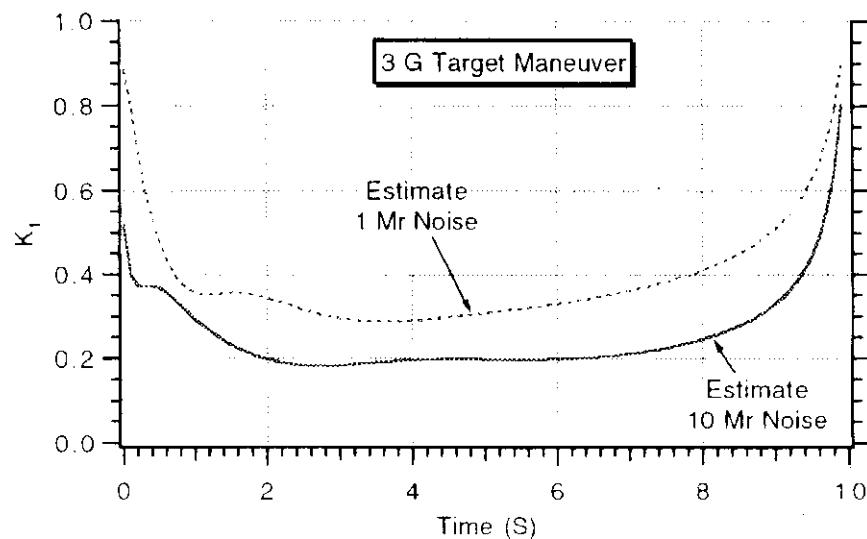


Fig. 9.6 Increasing measurement noise estimate decreases Kalman gain.

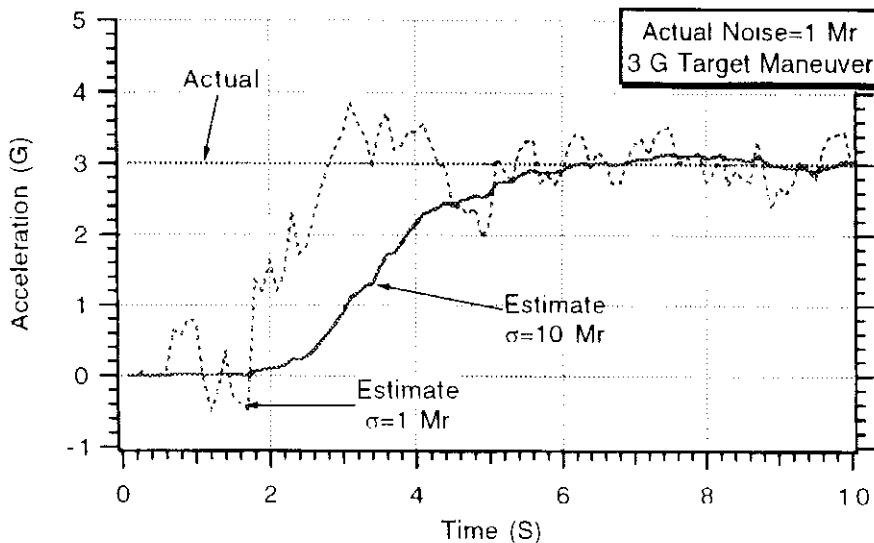


Fig. 9.7 Filter becomes sluggish when it thinks there is more noise.

If we run our simulation with measurement noise only ($\sigma = 1 \text{ mr}$ and no target maneuver), we must operate in the Monte Carlo mode. Fifty-run Monte Carlo sets were made for 20 different values of flight time for a total of 1000 runs. Figure 9.9 shows how the standard deviation of the noise-induced miss varies with flight time for a case in which the filter is optimal ($\sigma = 1 \text{ mr}$) and one in which the filter bandwidth has been intentionally decreased ($\sigma = 10 \text{ mr}$). We can see from Fig. 9.9 that decreasing the filter bandwidth (telling the filter that there is more measurement noise) decreases the system miss distance due to the actual measurement noise (1 mr). This behavior is opposite to that of the two-state digital fading memory filter (see Fig 7.19) in which decreasing the filter bandwidth always appeared to increase miss distance! Of course, the constant gain digital fading memory filter bandwidth was fixed, whereas the Kalman filter bandwidth is time-varying. By comparing Figs. 9.8 and 9.9 we can see that the guidance system designer has a juggling act. Increasing the filter bandwidth reduces the miss due to target maneuver while increasing the miss due to noise. The optimal practical filter bandwidth is dependent on the levels of the input disturbances.

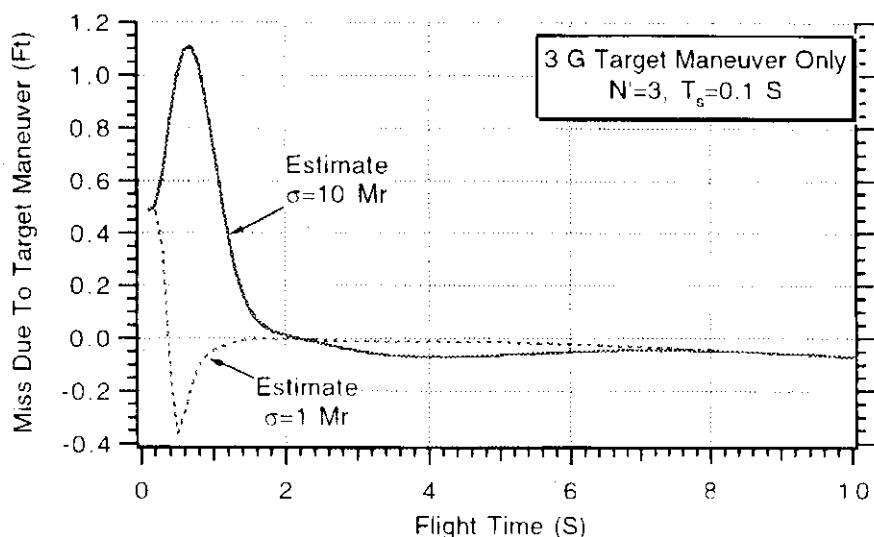


Fig. 9.8 Kalman filter guidance system has small sensitivity to target maneuver.

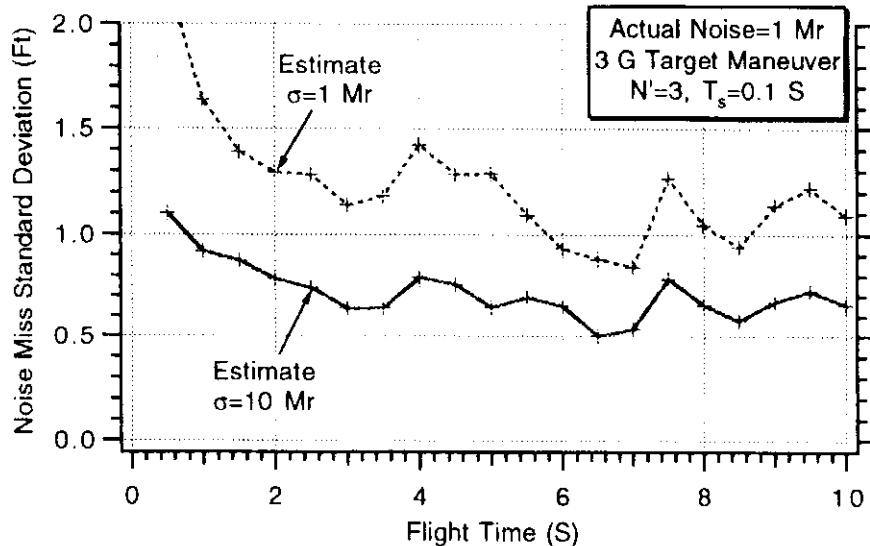


Fig. 9.9 Decreasing filter bandwidth decreases noise-induced miss.

The sampling time can also have a profound effect on filtering properties and system performance. Figure 9.10 shows that increasing the sampling time T_s from 0.1 to 0.5 s (or decreasing sampling rate from 10 to 2 Hz) increases the Kalman gain. We saw from Chapter 7 on digital fading memory filters that decreasing the sampling rate tends to decrease the total system bandwidth. Thus, the Kalman filter is attempting to increase its bandwidth to compensate for the decrease in system bandwidth due to sampling at a lower rate.

Figure 9.11 shows that the filter estimate of target acceleration for both sampling times is about the same. This means that the filter has successfully compensated for the effective decrease in system bandwidth due to a decrease in the sampling rate.

Although Kalman filter performance appears to be approximately independent of sampling rate, system performance is not! If we remove the actual measurement noise from the simulation and run with target maneuver only for different flight times, we can generate miss distance curves. Figure 9.12 shows how the target-

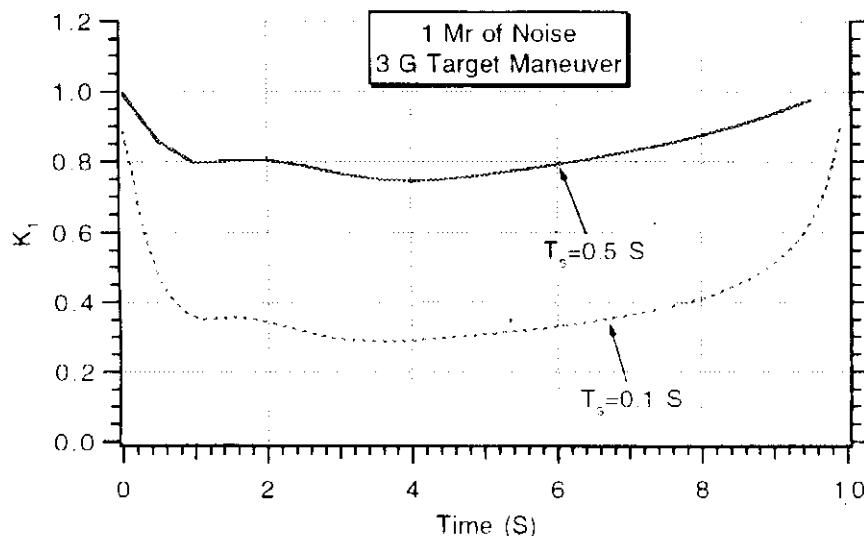


Fig. 9.10 Kalman gain increases with decreasing sampling rate.

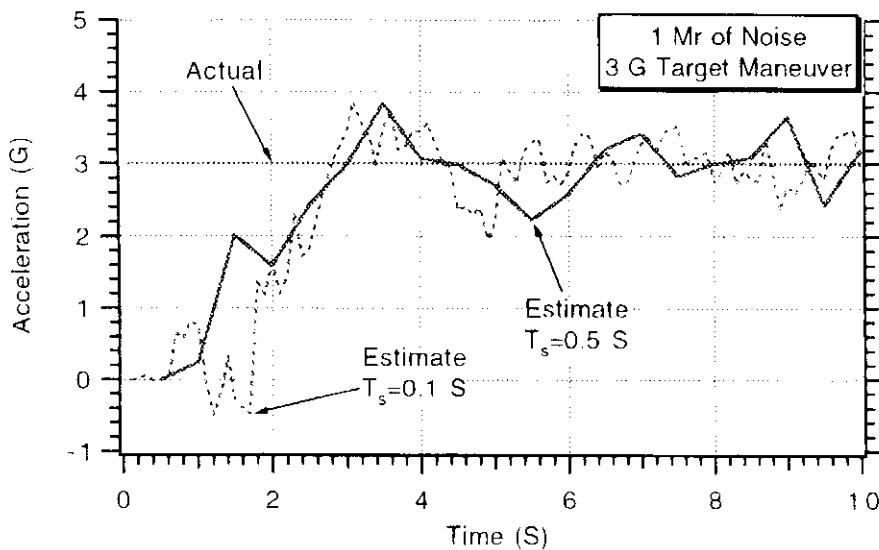


Fig. 9.11 Kalman filter bandwidth appears to be independent of sampling rate.

maneuver-induced miss varies with the sampling rate. We can see that the miss for $T_s = 0.5 \text{ s}$ is much greater than the miss for $T_s = 0.1 \text{ s}$ for flight times less than 2 s.

The simulation was also run with measurement noise only in the Monte Carlo mode. Figure 9.13 shows that decreasing the sampling rate also increases the measurement-noise-induced miss. Generally, hardware costs increase with higher sampling rates. Therefore, an important job of the guidance system designer is to set a limit on the sampling rate to get both acceptable cost and adequate performance.

It is important to note that in the preceding experiment the noise standard deviation remained constant when the data rate changed. In many systems the noise spectral density remains constant and so the noise standard deviation must change as the data rate changes. The interested reader is referred to Appendix A for a more complete discussion of this topic.

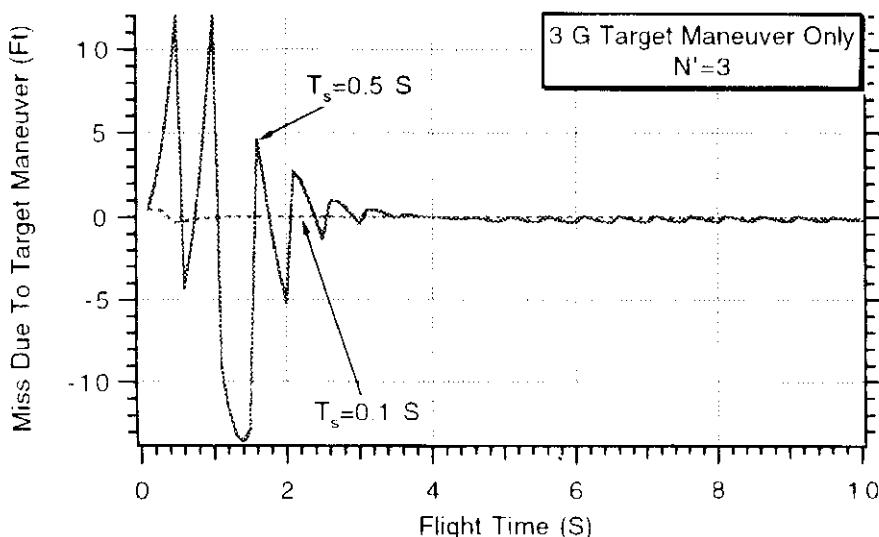


Fig. 9.12 Target maneuver miss increases with decreasing sampling rate.

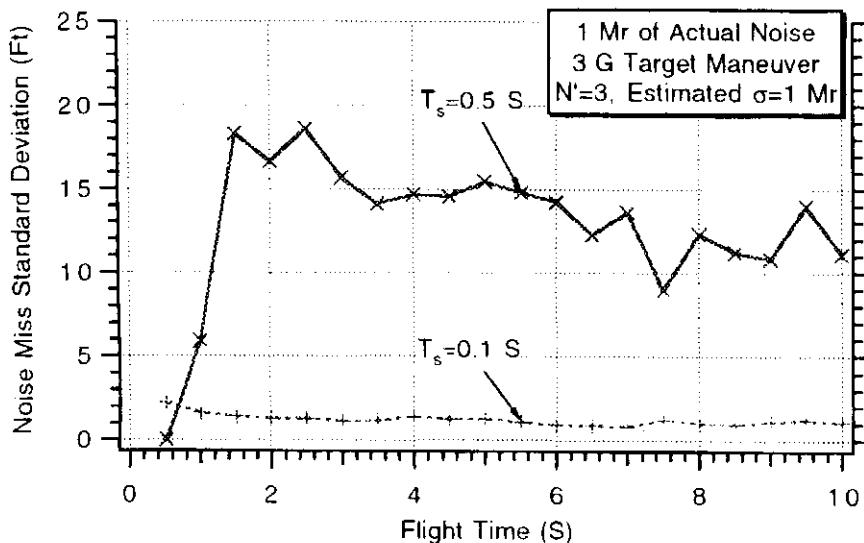


Fig. 9.13 Measurement noise miss increases with decreasing sampling rate.

Experiments With Optimal Guidance²

In Chapter 8 we derived an optimal guidance law that attempted to cancel out the guidance system dynamics and, in addition, we relaxed the missile acceleration requirements. In this section we will show how an optimal guidance system might be implemented and provide a numerical example to illustrate how such a system might perform in the presence of measurement noise.

Listing 9.3 presents a FORTRAN Monte Carlo simulation of an optimal guidance system with a three-state digital Kalman filter and a single-lag representation of the flight-control system. The filter structure is identical to the one shown in Fig. 9.2, except that the achieved missile acceleration n_L rather than the commanded acceleration n_c is fed back into the filter. The filter estimates relative position and velocity, which can be converted into a line-of-sight rate estimate as shown in the listing. In addition, the filter estimates the target maneuver level. The achieved missile acceleration is assumed to be known perfectly. This quantity is fed into the filter and, in addition, is used as part of an optimal guidance law as was discussed in Chapter 8 and can be seen in Listing 9.3.

A 50-run Monte Carlo set was made with the engagement model of Listing 9.3. In the nominal case the flight-control system time constant was set to 0.5 s, the effective navigation ratio was 3, and the sampling time was 0.1 s. The nominal error disturbances, as can be seen from Listing 9.3, consist of 1 mr of measurement noise and a constant 3-g target maneuver occurring at the beginning of flight. Figures 9.14 and 9.15 show 50-run Monte Carlo results for the standard deviation and mean miss distances for this case as a function of the flight time. Both figures show results for proportional navigation ($APN = 0$) and an optimal guidance law ($APN = 2$). Since there is one random disturbance and one deterministic disturbance, we can interpret the standard deviation of the miss to be the noise-induced miss and the mean of the miss to be the target-maneuver-induced miss. Both figures clearly show that, for the case in which the guidance time constant is 0.5 s, optimal guidance yields smaller miss distances, even in the presence of measurement noise errors. The differences between the guidance laws is greatest for the smaller flight times. If the ratio of the flight time to the guidance system time constant is large, proportional navigation is known to be an effective

Listing 9.3 Monte Carlo engagement simulation to test optimal guidance

```

GLOBAL DEFINE
    INCLUDE 'quickdraw.inc'
END
DIMENSION Z(1000)
INTEGER RUN,APN
REAL M11,M12,M13,M22,M23,M33,K1,K2,K3
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
VC=4000.
XNT=96.6
YIC=0.
VM=3000.
HEDEG=20.
XNP=3.
SIGNOISE=.001
TS=.1
TAU=.5
NOISE=1
RUN=50
APN=0
XLIM=999999.
106 CONTINUE
DO 60 TF=.5,10,.5
Z1=0.
DO 20 I=1,RUN
Y=YIC
YD=0.
TS2=TS*TS
TS3=TS2*TS
TS4=TS3*TS
TS5=TS4*TS
PHIN=XNT*XNT/TF
RTM=VC*TF
SIGPOS=RTM*SIGNOISE
SIGN2=SIGPOS**2
P11=SIGN2
P12=0.
P13=0.
P22=(VM*HEDEG/57.3)**2
P23=0.
P33=XNT*XNT
T=0.
H=.01
S=0.
YH=0.
YDH=0.
XNTH=0.
XNC=0.
XNL=0.
10 IF(T>(TF-.0001))GOTO 999

```

(Contd.)

Listing 9.3 (Continued)

```

YOLD=Y
YDOLD=YD
XNLOLD=XNL
STEP=1
GOTO 200
66 STEP=2
Y=Y+H*YD
YD=YD+H*YDD
XNL=XNL+H*XNLD
T=T+H
GOTO 200
55 CONTINUE
Y=.5*(YOLD+Y+H*YD)
YD=.5*(YDOLD+YD+H*YDD)
XNL=.5*(XNLOLD+XNL+H*XNLD)
S=S+H
IF(S<(TS-.0001))GOTO 10
S=0.
TGO=TF-T+.000001
RTM=VC*TGO
SIGPOS=RTM*SINOISE
SIGN2=SINOISE**2
M11=P11+TS*P12+.5*TS2*P13+TS*(P12+TS*P22+.5*TS2*P23)
M11=M11+.5*TS2*(P13+TS*P23+.5*TS2*P33)+TS5*PHIN/20.
M12=P12+TS*P22+.5*TS2*P23+TS*(P13+TS*P23+.5*TS2*P33)+TS4*PHIN/8.
M13=P13+TS*P23+.5*TS2*P33+PHIN*TS3/6.
M22=P22+TS*P23+TS*(P23+TS*P33)+PHIN*TS3/3.
M23=P23+TS*P33+.5*TS2*PHIN
M33=P33+PHIN*TS
K1=M11/(M11+SIGN2)
K2=M12/(M11+SIGN2)
K3=M13/(M11+SIGN2)
P11=(1.-K1)*M11
P12=(1.-K1)*M12
P13=(1.-K1)*M13
P22=-K2*M12+M22
P23=-K2*M13+M23
P33=-K3*M13+M33
IF(NOISE.EQ.1)THEN
    CALL GAUSS(XLAMNOISE,SINOISE)
ELSE
    XLAMNOISE=0.
ENDIF
YSTAR=RTM*(XLAM+XLAMNOISE)
RES=YSTAR-YH-TS*YDH-.5*TS*TS*(XNTH-XNL)
YH=K1*RES+YH+TS*YDH+.5*TS*TS*(XNTH-XNL)
YDH=K2*RES+YDH+TS*(XNTH-XNL)
XNTH=K3*RES+XNTH
XLAMDH=(YH+YDH*TGO)/(VC*TGO*TGO)

```

(Contd.)

Listing 9.3 (Continued)

```

IF(APN.EQ.0)THEN
  XNC=XNP*VC*XLAMDH
ELSEIF(APN.EQ.1)THEN
  XNC=XNP*VC*XLAMDH+APN*.5*XNP*XNTH
ELSE
  X=TGO/TAU
  TOP=6.*X*X*(EXP(-X)-1.+X)
  BOT1=2*X*X*X+3.+6.*X-6.*X*X
  BOT2=-12.*X*EXP(-X)-3.*EXP(-2.*X)
  XNPP=TOP/(.0001+BOT1+BOT2)
  XNEW=XNPP*XNL*(EXP(-X)+X-1.)/(X*X)
  XNC=XNPP*VC*XLAMDH+.5*XNPP*XNTH-XNEW
ENDIF
IF(XNC>XLIM)XNC=XLIM
IF(XNC<-XLIM)XNC=-XLIM
GOTO 10
200 CONTINUE
TGO=TF-T+.00001
RTM=VC*TGO
XLAM=Y/(VC*TGO)
XLAMD=(RTM*YD+Y*VC)/(RTM**2)
XNLD=(XNC-XNL)/TAU
YDD=XNT-XNL
IF(STEP-1)66,66,55
999 CONTINUE
Z(I)=Y
Z1=Z(I)+Z1
XMEAN=Z1/I
20 CONTINUE
SIGMA=0.
Z1=0.
DO 50 I=1,RUN
Z1=(Z(I)-XMEAN)**2+Z1
IF(I.EQ.1)THEN
  SIGMA=0.
ELSE
  SIGMA=SQRT(Z1/(I-1))
ENDIF
50 CONTINUE
WRITE(*,97)TF,SIGMA,XMEAN
WRITE(1,97)TF,SIGMA,XMEAN
60 CONTINUE
97 FORMAT(3F10.3)
PAUSE
CLOSE(1)
END

SUBROUTINE GAUSS(X,SIG)
INTEGER SUM

```

(Contd.)

Listing 9.3 (Continued)

```

SUM=0
DO 14 J=1,6
IRAN=Random()
SUM=SUM+IRAN
14  CONTINUE
X=SUM/65536.
X=1.414*X*SIG
RETURN
END

```

guidance law. Thus, the optimal guidance law, discussed in a deterministic setting in Chapter 8, can be implemented and made to work successfully in a more realistic setting. Optimal guidance is yielding superior performance to proportional navigation because it is attempting to cancel out dynamically the flight-control system time constant. We can see from both figures that optimal guidance performance, unlike that of proportional navigation, is approximately independent of flight time.

Another performance advantage of optimal guidance is that it is supposed to relax the missile acceleration requirements. The previous case was rerun for a 10-s flight. This flight time was chosen because the performance of both proportional navigation and optimal guidance is about the same, from a miss distance point of view, as can be seen from Figs. 9.14 and 9.15. The reason for this is that the 10-s flight time is large compared to the 0.5-s flight-control system time constant. However, in the new case to be run the engagement simulation was made nonlinear in the sense that missile acceleration saturation effects were included. The missile commanded acceleration limit was made a parameter in the study. Figure 9.16 displays the mean miss distance vs the acceleration limit for a case in which there was a 3-g target maneuver and 1 mr of measurement noise. We can see from the figure that the acceleration requirements for optimal guidance are clearly relaxed.

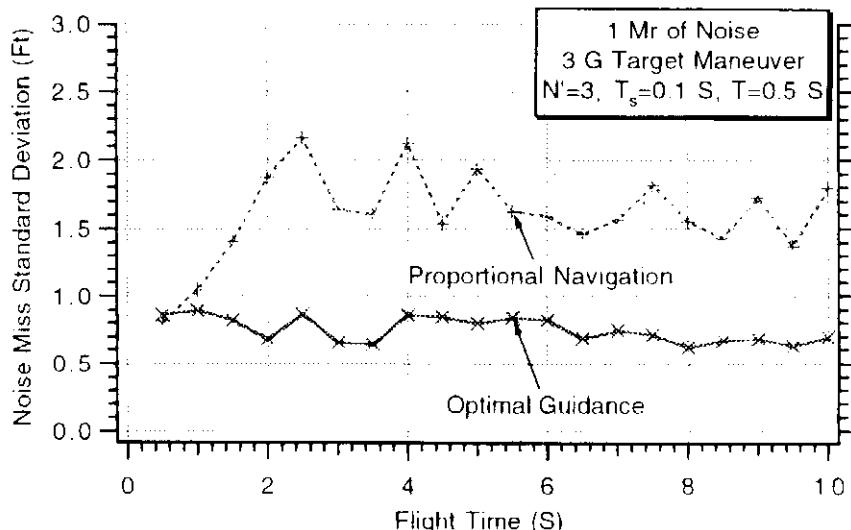


Fig. 9.14 Optimal guidance yields smaller noise-induced miss in presence of large guidance system time constant.

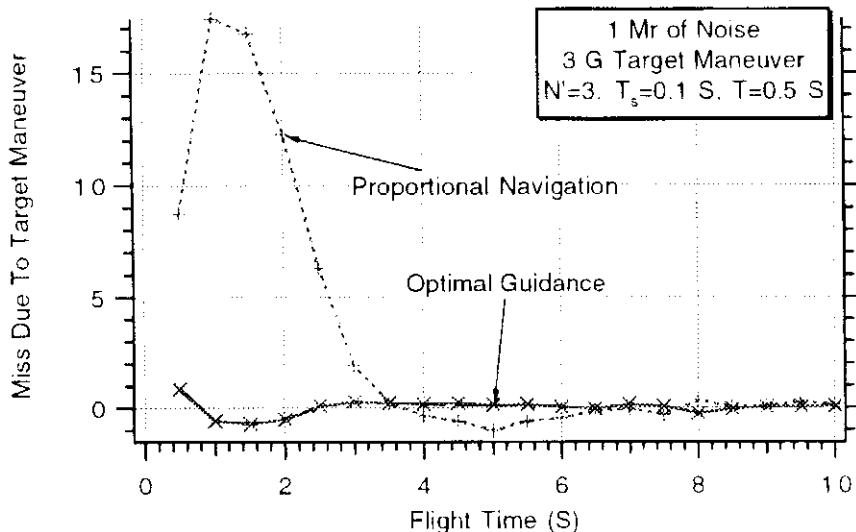


Fig. 9.15 Optimal guidance yields smaller target-maneuver-induced miss in presence of large guidance system time constant.

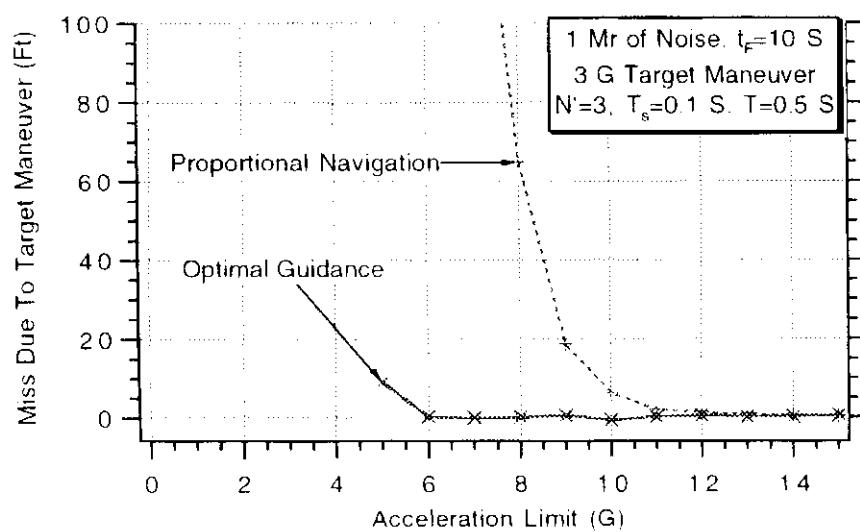


Fig. 9.16 Optimal guidance reduces missle acceleration requirements.

Summary

In this chapter we have shown how both Kalman filtering and optimal guidance concepts could be applied to a missile guidance system. It was shown, via a numerical example, that when these concepts were applied there were substantial performance benefits and a relaxing of missile acceleration requirements. However, range and time to go information must be available for Kalman filtering and optimal guidance to work. If the required information is lacking or inaccurate, the performance of this type of guidance system may degrade to the point where its performance is worse than that of a conventional proportional navigation guidance system.²

References

¹Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.

²Nesline, F. W., and Zarchan, P., "A New Look at Classical Versus Modern Homing Guidance," *Journal of Guidance and Control*, Vol. 4, Jan.-Feb. 1981, pp. 78-85.



Other Forms of Tactical Guidance

Introduction

THUS far we have studied proportional navigation type homing guidance systems. In this type of guidance system the missile seeker provides the information required for guidance by receiving energy reflected from the target (i.e., radar signal). The virtue of homing guidance is that measurement accuracy is continually improving because the missile (and its seeker) is getting closer to the target as the flight progresses.

With command guidance a missile seeker is not required. A source that is external to the missile both transmits and receives the energy (i.e., radar signal). One limitation of command guidance is that, as intercepts take place further away from the energy source, measurement accuracy and hence guidance accuracy degrade. Another limitation of command guidance is that the external energy source must illuminate the target often enough (i.e., high data rate) to make guidance effective. This means that the energy source can only service a few targets simultaneously in a command guidance implementation.

In this chapter we shall discuss three types of command guidance systems: proportional navigation command guidance, beam rider guidance, and command to line-of-sight guidance. The reasons for implementing these systems have more to do with cost, ease of implementation, and lack of susceptibility to countermeasures than with performance benefits. However, the guidance features of these systems will be compared via numerical examples.

Proportional Navigation Command Guidance¹

Figure 10.1 shows the basic geometry involved in a proportional navigation command guidance system. The energy source and receiver (a radar application in this example), which are collocated, track both the missile and target. For radar applications we can assume that measurements of both angle and range, with respect to the radar, of both the missile and target are available. In the notation of Fig. 10.1, we have missile measurements of R_M and θ_M and target measurements of R_T and θ_T .

To implement proportional navigation guidance in the command guidance system of Fig. 10.1, we must first derive line-of-sight angle λ information from the measurements available. To do this we must first recognize that

$$\tan \theta_T = R_{T2}/R_{T1}$$

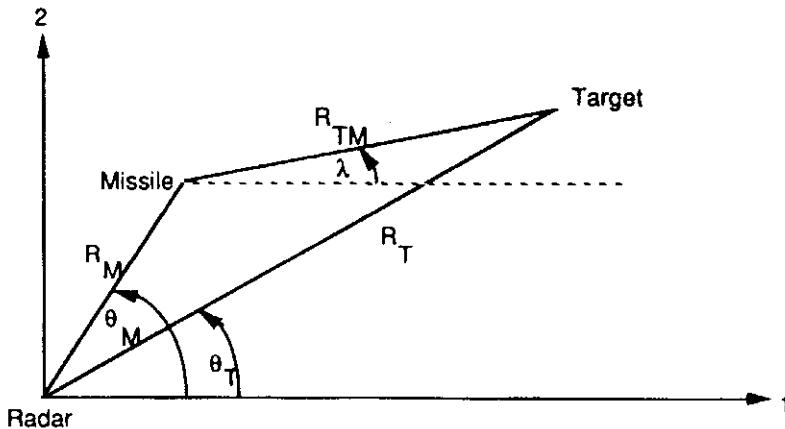


Fig. 10.1 Fundamentals of proportional navigation command guidance.

Since R_T and θ_T are measured, we can express the inertial components of the distance from the radar to the target as

$$R_{T1} = R_T \cos \theta_T$$

$$R_{T2} = R_T \sin \theta_T$$

In a similar way we can express the components of the range from the radar to the missile by first recognizing that

$$\tan \theta_M = R_{M2}/R_{M1}$$

and then expressing the inertial components of the range from the energy source to the missile in terms of the measurements as

$$R_{M1} = R_M \cos \theta_M$$

$$R_{M2} = R_M \sin \theta_M$$

We can now find the relative missile-target range inertial components from

$$R_{TM1} = R_{T1} - R_{M1}$$

$$R_{TM2} = R_{T2} - R_{M2}$$

Finally, the line-of-sight angle can be expressed in terms of the relative range components as

$$\lambda = \tan^{-1}(R_{TM2}/R_{TM1})$$

To develop the line-of-sight rate signal required for proportional navigation, we must take the derivative of the line-of-sight angle via a digital filter. To illustrate a possible implementation of proportional navigation command guidance, via simulation, a model is postulated. In this hypothetical model the missile states are known perfectly (i.e., retroreflector on missile), and the range from the energy source to the target is also known perfectly. In this example the measurement of the target angle is corrupted by 1 milliradian (mr) of noise.

An effective line-of-sight rate is reconstructed from the angular measurement according to the previously developed relationships in this section in order to derive a pseudomeasurement of the line-of-sight angle. A two-state digital fading

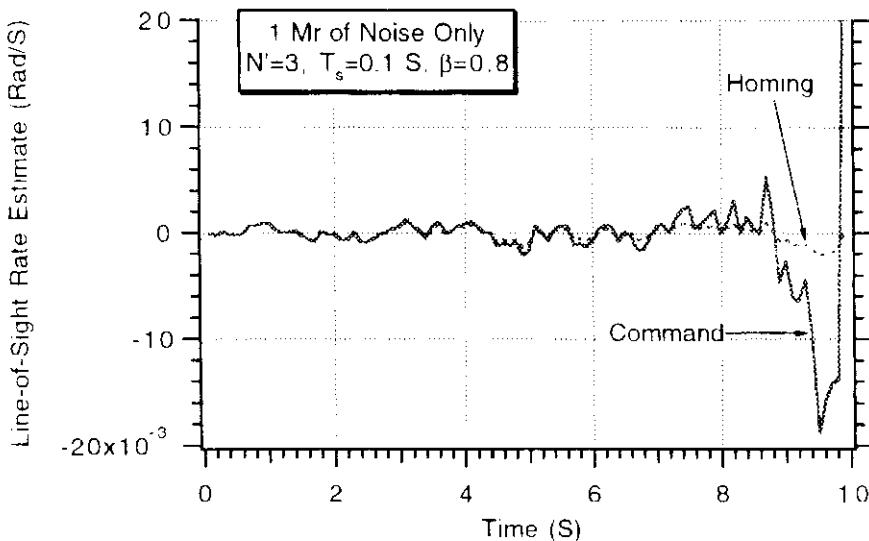


Fig. 10.2 Command guidance system has more noise transmission.

memory filter is used to get an estimate of the line-of-sight rate from the pseudomeasurement. The line-of-sight rate estimate is used to implement a proportional navigation guidance law. A FORTRAN command guidance simulation, using the previously discussed concepts, appears in Listing 10.1.

The command guidance simulation was run with 1 mr of noise on the measurement of the target angle. Figure 10.2 presents the line-of-sight rate estimate generated by the digital fading memory filter ($\beta = 0.8$). Superimposed on the figure are homing guidance results for the same example. We can see from Fig. 10.2 that the noise transmission appears to be approximately the same for both command and homing guidance for most of the flight. However, toward the end of the flight there is a dramatic increase in the noise transmission of the command guidance system.

To explain why command guidance appears to have excessive noise transmission when compared to homing guidance, it is necessary to develop an analytical model. For analysis and understanding a linearized model of a command guidance system will be developed. The linearization will be accomplished by using small-angle approximations when possible. For convenience let us define new variables:

$$y_T = R_{T2}$$

$$y_M = R_{M2}$$

$$y = R_{T2} - R_{M2}$$

where y_T and y_M are the perpendicular distances from the target and missile to the ground. The relative perpendicular missile-target separation is termed y . Using small-angle approximations leads to the simplified expressions for the target and missile angles

$$\theta_T \approx y_T/R_T$$

$$\theta_M \approx y_M/R_M$$

**Listing 10.1 Proportional navigation command
guidance simulation**

```

GLOBAL DEFINE
    INCLUDE 'quickdraw.inc'
END
INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
VM=3000.
VT=1000.
XNT=0.
RM1IC=0.
RM2IC=10000.
RT1IC=40000.
RT2IC=10000.
HEDEG=0.
XNP=3.
BETA=.8
TS=.1
SIGNOISE=.001
NOISE=1
RT1=RT1IC
RT2=RT2IC
RM1=RM1IC
RM2=RM2IC
BETAT=0.
VT1=-VT*COS(BETAT)
VT2=VT*SIN(BETAT)
HE=HEDEG/57.3
GFILTER=1.-BETA**2
HFILTER=(1.-BETA)**2
XLAMH=0.
XLAMDH=0.
XNC=0.
T=0.
S=0.
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
XLAM=ATAN2(RTM2,RTM1)
XLEAD=ASIN(VT*SIN(BETAT+XLAM)/VM)
THET=XLAM+XLEAD
VM1=VM*COS(THET+HE)
VM2=VM*SIN(THET+HE)
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
10 IF(VC<0.)GOTO 999
IF(RTM<1000.)THEN
    H=.0002
ELSE
    H=.01

```

(Contd.)

Listing 10.1 (Continued)

```

ENDIF
BETATOLD=BETAT
RT1OLD=RT1
RT2OLD=RT2
RM1OLD=RM1
RM2OLD=RM2
VM1OLD=VM1
VM2OLD=VM2
STEP=1
GOTO 200
66 STEP=2
BETAT=BETAT+H*BETATD
RT1=RT1+H*VT1
RT2=RT2+H*VT2
RM1=RM1+H*VM1
RM2=RM2+H*VM2
VM1=VM1+H*AM1
VM2=VM2+H*AM2
T=T+H
GOTO 200
55 CONTINUE
BETAT=.5*(BETATOLD+BETAT+H*BETATD)
RT1=.5*(RT1OLD+RT1+H*VT1)
RT2=.5*(RT2OLD+RT2+H*VT2)
RM1=.5*(RM1OLD+RM1+H*VM1)
RM2=.5*(RM2OLD+RM2+H*VM2)
VM1=.5*(VM1OLD+VM1+H*AM1)
VM2=.5*(VM2OLD+VM2+H*AM2)
S=S+H
IF(S<(TS-.0001))GOTO 10
S=0.
IF(NOISE.EQ.1)THEN
    CALL GAUSS(THETTNOISE,SIGNOISE)
ELSE
    THETTNOISE=0.
ENDIF
THETTM=THETT+THETTNOISE
THETMM=THETM
RT1M=RT*COS(THETTM)
RT2M=RT*SIN(THETTM)
RM1M=RM*COS(THETMM)
RM2M=RM*SIN(THETMM)
XLAMM=ATAN2(RT2M-RM2M,RT1M-RM1M)
RES=XLAMM-(XLAMH+TS*XLAMDH)
XLAMH=GFILTER*RES+XLAMH+TS*XLAMDH
XLAMDH=HFILTER*RES/TS+XLAMDH
XNC=XNP*VC*XLAMDH
RT1KM=RT1/3280.
RT2KM=RT2/3280.
RM1KM=RM1/3280.

```

(Contd.)

Listing 10.1 (Continued)

```

RM2KM=RM2/3280.
WRITE(*,97)T,XLAMD,XLAMDH,XNC,RTM
WRITE(1,97)T,XLAMD,XLAMDH,XNC,RTM
GOTO 10
97  FORMAT(5F12.5)
200  CONTINUE
     THETT=ATAN2(RT2,RT1)
     THETM=ATAN2(RM2,RM1)
     RT=SQRT(RT1**2+RT2**2)
     RM=SQRT(RM1**2+RM2**2)
     RTM1=RT1-RM1
     RTM2=RT2-RM2
     RTM=SQRT(RTM1**2+RTM2**2)
     VTM1=VT1-VM1
     VTM2=VT2-VM2
     VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
     XLAM=ATAN2(RTM2,RTM1)
     XLAMD=(RTM1*VTM2-RTM2*VTM1)/(RTM*RTM)
     XNC=XNP*VC* XLAMDH
     AM1=-XNC*SIN(XLAM)
     AM2=XNC*COS(XLAM)
     VT1=-VT*COS(BETAT)
     VT2=VT*SIN(BETAT)
     BETATD=XNT/VT
     IF(STEP-1)66,66,55
999  CONTINUE
     WRITE(*,*)T,RTM
     PAUSE
     CLOSE(1)
     END

SUBROUTINE GAUSS(X,SIG)
INTEGER SUM
SUM=0
DO 14 J=1,6
IRAN=Random()
SUM=SUM+IRAN
14  CONTINUE
X=SUM/65536.
X=1.414*X*SIG
RETURN
END

```

The line-of-sight angle, as in the homing case, can also be derived using the small-angle approximation, or

$$\lambda \approx y/R_{\text{TM}}$$

where R_{TM} is the separation between missile and target. As in the homing case, the miss distance for command guidance is defined as

$$\text{Miss} = y(t_F)$$

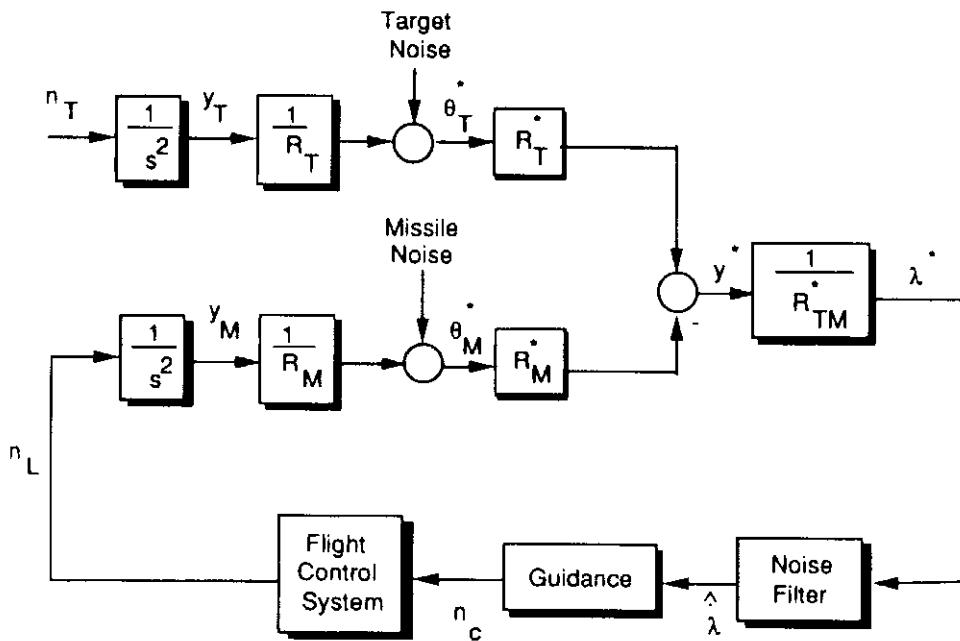


Fig. 10.3 Conceptual linearized command guidance block diagram.

From the linearized equations presented thus far we can draw a block diagram representing a command guidance system. Figure 10.3 presents the command guidance block diagram. Here we can see that target acceleration is integrated twice to yield target position and a division by range produces the target angle θ_T . Thus far this process represents the physics of the command guidance engagement. The target angle measurement is corrupted by noise. An expression for the missile angle θ_M is derived in a similar manner. A measurement of the missile angle is also contaminated by noise. To generate guidance commands, the measured missile and target angles are multiplied by their respective range measurements and subtracted in order to get the measured relative missle-target position y^* . Another division by the measured range from missle to target is required to generate a pseudomeasurement of the line-of-sight angle λ^* required for guidance.

Note that in the command guidance process, unlike the homing process, various range measurements were required (R_T^* , R_M^* , and R_{TM}^*) just to get the pseudomeasurement of the line-of-sight angle. In homing, line-of-sight angle information is available from the seeker, without the need for range measurements. In proportional navigation command guidance, a noise filter, with derivative action, is required to remove the noise and get an estimate of the line-of-sight rate so that a guidance command can be generated. The guidance command is up-linked to the missle, which responds to the command via its flight control system. The resultant missle acceleration n_L completes the command guidance loop of Fig. 10.3.

To develop a command guidance model that resembles the homing loop, let us make the assumption that all ranges are known perfectly (we can always change this assumption without making the model nonlinear) and that the missle noise is insignificant compared to the target noise. Since ranges are known perfectly, we can cancel actual and measured ranges and use block diagram manipulation to get an even simpler model . The command guidance system of Fig. 10.3 can now be drawn in a form that resembles the homing loop. The linearized proportional navigation command guidance loop is shown in Fig. 10.4.

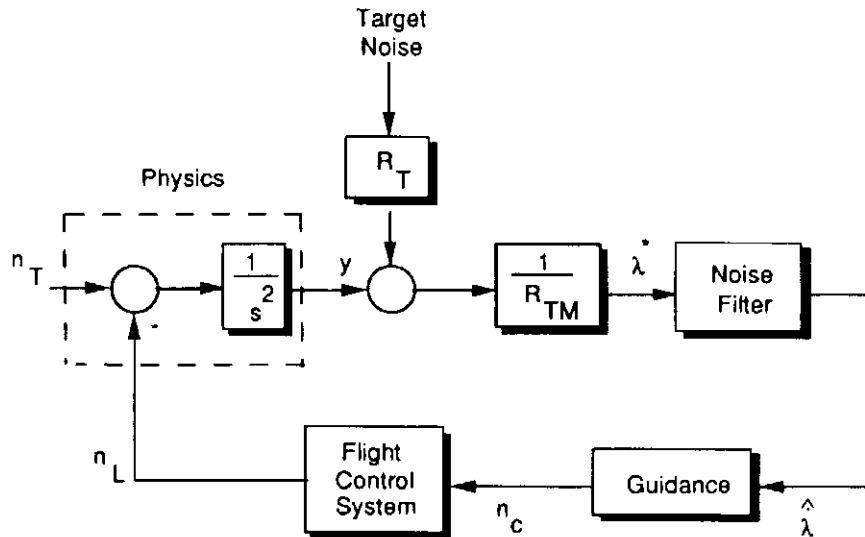


Fig. 10.4 Linearized proportional navigation command guidance loop.

At first glance, the command guidance loop looks identical to the standard homing loop diagram, shown in Fig. 10.5, but there are major differences.

In the homing loop diagram of Fig. 10.5, range is not required, and the geometric line-of-sight angle comes from physics. Angular noise corrupts the measurement, by the seeker, of the line-of-sight angle. In our examples so far, we have considered this noise to be range-independent (constant standard deviation). In the basic command guidance loop shown in Fig. 10.4, only relative position comes from physics, and computation of the line-of-sight angle requires range information. We can see from Fig. 10.4 that the effective noise on the line-of-sight angle is the actual angular measurement noise entering the system multiplied by the range from the energy source to the target R_T and divided by the range from missile to target R_{TM} . Since the range from the missile to target tends towards zero as the flight progresses, the effective noise on the line-of-sight angle will get very large near the end of the flight. This means that command guidance systems will generally have to contend with more noise than homing systems near the end of the flight, thus explaining the experimental results of Fig. 10.2.

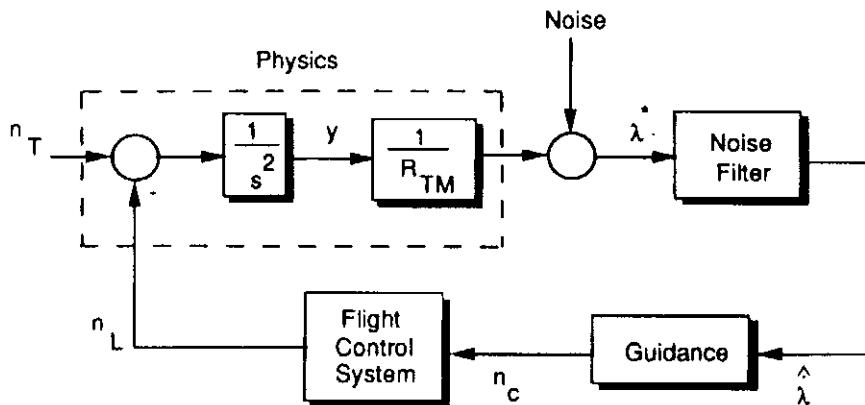


Fig. 10.5 Linearized proportional navigation homing guidance loop.

Beam Rider Guidance²

Beam riding is another form of command guidance. The object of beam riding is to fly the missile along a beam (i.e., radar or laser) that is continuously pointed at the target. Since the missile is attempting to fly along a moving beam, the missile guidance commands must be a function of the angular deviation of the missile from the beam. If the beam is always on the target and the missile is always on the beam, an intercept will result. The beam riding guidance principle is probably one of the first methods used because of its simplicity and ease of implementation.

The beam riding missile-target engagement equations are nearly identical to those of homing guidance. Since the proportional navigation guidance law is not used in beam riding, line-of-sight rate information is not required. However, as with proportional navigation command guidance, the missile and target angles with respect to the beam generator θ_M and θ_T are important. A typical beam riding geometry is shown in Fig. 10.6, with all important quantities marked. For simplicity we have assumed the beam to be of zero width. In practice we must keep the missile inside the beam for the beam riding principle to work.

From Fig. 10.6 the missile and target angles can be obtained by trigonometry, or

$$\theta_M = \tan^{-1}(R_{M2}/R_{M1})$$

$$\theta_T = \tan^{-1}(R_{T2}/R_{T1})$$

The distance formula can be used to obtain the missile and target ranges from the energy source as

$$R_M = (R_{M1}^2 + R_{M2}^2)^{\frac{1}{2}}$$

$$R_T = (R_{T1}^2 + R_{T2}^2)^{\frac{1}{2}}$$

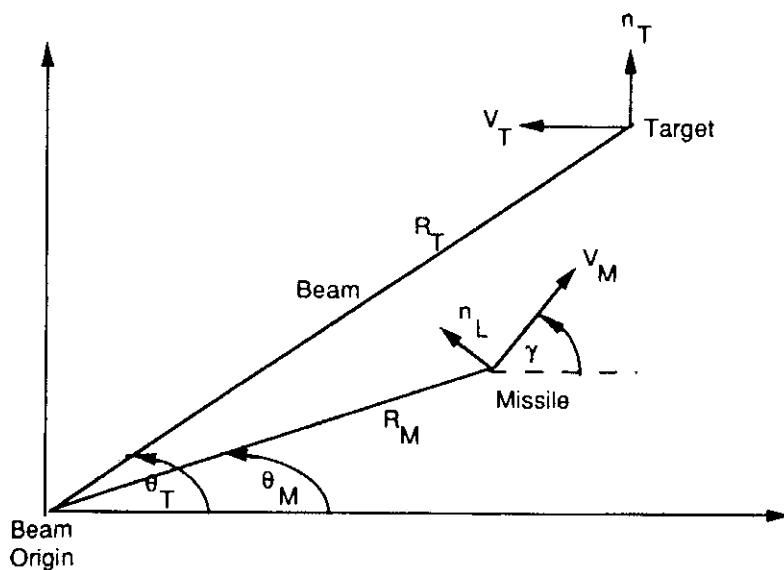


Fig. 10.6 Beam rider geometry.

Again, from Fig. 10.6 we can see that the distance the missile is from the beam y is simply given by

$$y = R_M(\theta_T - \theta_M)$$

If the missile is always on the beam ($y = 0$), then the missile will surely hit the target. Therefore, as in the homing and command guidance cases, it is desired to minimize y at the end of flight. Mathematically we are again attempting to null $y(t_F)$, which means we are trying to drive the miss distance to zero. The simplest possible implementation of a guidance law for a beam rider system is to make the missile acceleration command n_c proportional to y , or

$$n_c = Ky = KR_M(\theta_T - \theta_M)$$

Thus, we can see that the guidance command is proportional to the angular displacement off the beam.

A two-dimensional FORTRAN engagement simulation listing for the beam rider loop, utilizing guidance commands that are proportional to the off-axis beam displacement, appears in Listing 10.2. We can see from the listing that the guidance gain, K (XNP in the listing) has a nominal value of 10.

In beam rider guidance the missile is initially fired at the target (i.e., along the line-of-sight that is along the beam), whereas in proportional navigation the missile is fired in a direction to lead the target (i.e., aimed at expected intercept point). Running the engagement simulation of Listing 10.2 yields the trajectory for the nominal beam rider case that appears in Fig. 10.7. The curvature in the beam riding missile trajectory is due solely to the movement of the beam. In the benign case considered there is no target maneuver and the missile is traveling at three times the speed of the target. A proportional navigation trajectory, for this case, would result in a straight-line trajectory since the missile would initially be on a collision course.

Although there does not appear to be anything unusual with the beam rider trajectory, we can see from Fig. 10.8 that the missile acceleration required by beam rider guidance is oscillatory. In addition, the resultant miss distance for this case is 7 ft even though the target is not maneuvering and there are no guidance system lags!

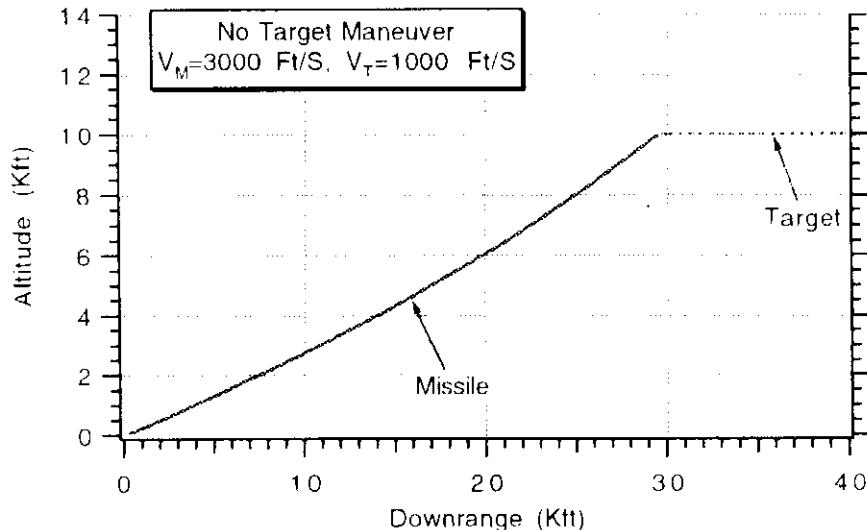


Fig. 10.7 Nominal beam rider trajectory.

Listing 10.2 FORTRAN beam rider engagement simulation

```

INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
VM=3000.
VT=1000.
XNT=0.
RM1IC=0.
RM2IC=1.
RT1IC=40000.
RT2IC=10000.
HEDEG=0.
XNP=10.
TS=.1
RT1=RT1IC
RT2=RT2IC
RM1=RM1IC
RM2=RM2IC
BETAT=0.
VT1=-VT*COS(BETAT)
VT2=VT*SIN(BETAT)
HE=HEDEG/57.3
XNC=0.
T=0.
S=0.
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
THETT=ATAN2(RT2,RT1)
VM1=VM*COS(THETT+HE)
VM2=VM*SIN(THETT+HE)
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
10 IF(VC<0.)GOTO 999
      IF(RTM<1000.)THEN
          H=.0002
      ELSE
          H=.01
      ENDIF
      BETATOLD=BETAT
      RT1OLD=RT1
      RT2OLD=RT2
      RM1OLD=RM1
      RM2OLD=RM2
      VM1OLD=VM1
      VM2OLD=VM2
      STEP=1
      GOTO 200
66   STEP=2
      BETAT=BETAT+H*BETATD

```

(Contd.)

Listing 10.2 (Continued)

```

RT1=RT1+H*VT1
RT2=RT2+H*VT2
RM1=RM1+H*VM1
RM2=RM2+H*VM2
VM1=VM1+H*AM1
VM2=VM2+H*AM2
T=T+H
GOTO 200
55  CONTINUE
BETAT=.5*(BETATOLD+BETAT+H*BETATD)
RT1=.5*(RT1OLD+RT1+H*VT1)
RT2=.5*(RT2OLD+RT2+H*VT2)
RM1=.5*(RM1OLD+RM1+H*VM1)
RM2=.5*(RM2OLD+RM2+H*VM2)
VM1=.5*(VM1OLD+VM1+H*AM1)
VM2=.5*(VM2OLD+VM2+H*AM2)
S=S+H
IF(S<(TS-.0001))GOTO 10
S=0.
RT1K=RT1/1000.
RT2K=RT2/1000.
RM1K=RM1/1000.
RM2K=RM2/1000.
WRITE(*,97)T,RT1K,RT2K,RM1K,RM2K,XNC/32.2
WRITE(1,97)T,RT1K,RT2K,RM1K,RM2K,XNC/32.2
GOTO 10
97  FORMAT(6F10.3)
200  CONTINUE
THETT=ATAN2(RT2,RT1)
THETM=ATAN2(RM2,RM1)
RT=SQRT(RT1**2+RT2**2)
RM=SQRT(RM1**2+RM2**2)
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
XLAM=ATAN2(RTM2,RTM1)
XLAMD=(RTM1*VTM2-RTM2*VTM1)/(RTM*RTM)
XNC=XNP*RM*(THETT-THETM)
AM1=-XNC*SIN(XLAM)
AM2=XNC*COS(XLAM)
VT1=-VT*COS(BETAT)
VT2=VT*SIN(BETAT)
BETATD=XNT/VT
IF(STEP-1)66,66,55
999  CONTINUE
WRITE(*,*)T,RTM
PAUSE
CLOSE(1)
END

```

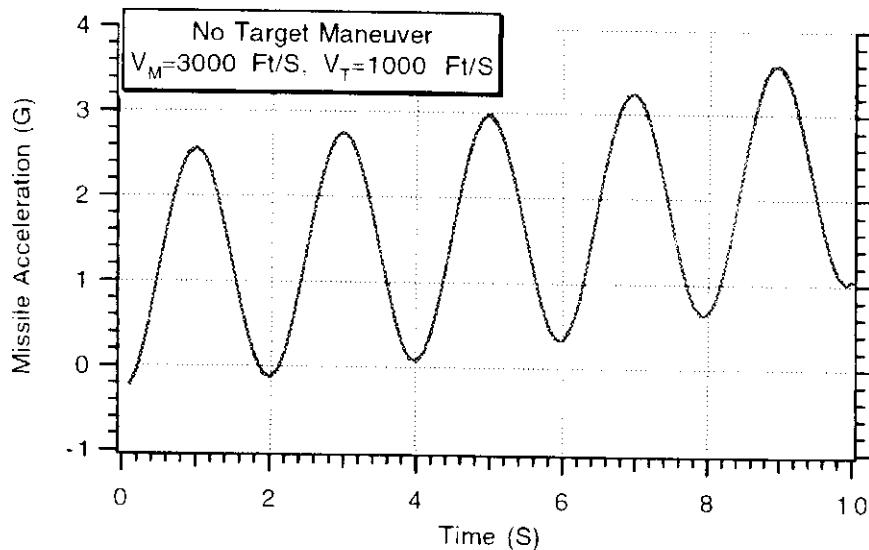


Fig. 10.8 Simple beam rider guidance results in oscillatory missile acceleration profile.

If we reduce the guidance gain K from 10 to 1, the missile acceleration oscillations decrease in frequency, but the miss distance increases from 7 ft to 102 ft, as shown in Fig. 10.9. Clearly something is wrong!

To understand why the beam rider guidance system does not appear to work satisfactorily, it is necessary to develop an analytical model. For this purpose we will linearize the beam rider loop using our standard small-angle approximations. The resultant linearized beam rider loop with a perfect flight control system is shown in Fig. 10.10.

The transfer function $G(s)$ represents the beam rider guidance law in Fig. 10.10. We have already attempted the simplest possible implementation of a guidance law for a beam rider missile, which is to make the missile acceleration command n_c proportional to the angular displacement off the beam. Since we have already implemented

$$n_c = K R_M (\theta_T - \theta_M) = Ky$$

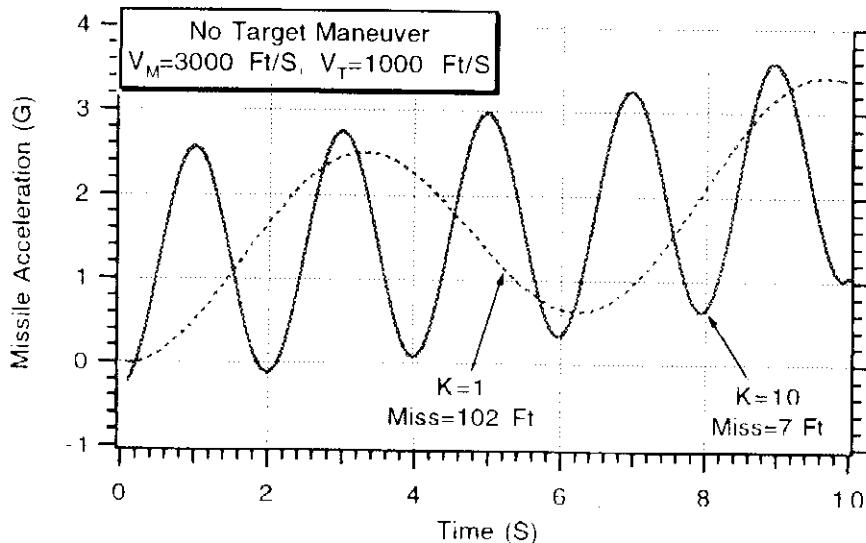


Fig. 10.9 Reducing beam rider gain does not improve system performance.

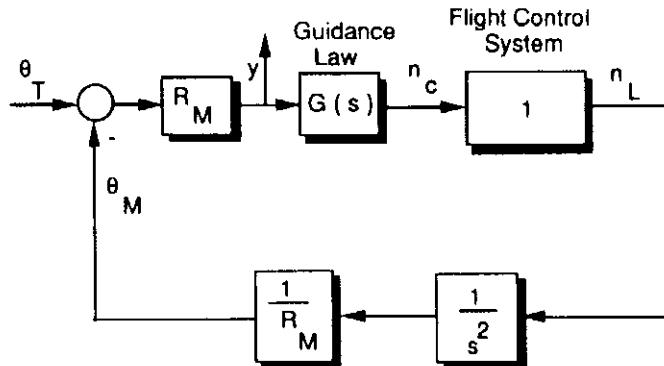


Fig. 10.10 Conceptual model of beam rider guidance loop.

the guidance law transfer function is simply

$$G(s) = K$$

If we treat the range from the energy source to the missile R_M as a constant, then we can express the transfer function from missile acceleration to target angle as

$$\frac{n_L}{\theta_T}(s) = \frac{KR_M s^2}{s^2 + K}$$

As was shown in the Laplace transform Table 1.1 of Chapter 1, this transfer function is that of a sinusoid with natural frequency K . Thus, decreasing K will decrease the frequency of oscillation as we saw in the simulation results of Fig. 10.9. One way of stabilizing (eliminating the oscillatory behavior of) the beam rider loop is to add a lead-lag compensation network. For example, one possible compensation network is

$$G(s) = 10 \frac{1 + s/2}{1 + s/20}$$

Figure 10.11 shows that the beam rider acceleration commands are completely stabilized when the lead-lag network is added to the beam rider loop. The guidance gain K is still 10; hence, the miss distance remains the same, namely, 7 ft. However, since the target is not maneuvering, the beam-rider-induced miss distance is still

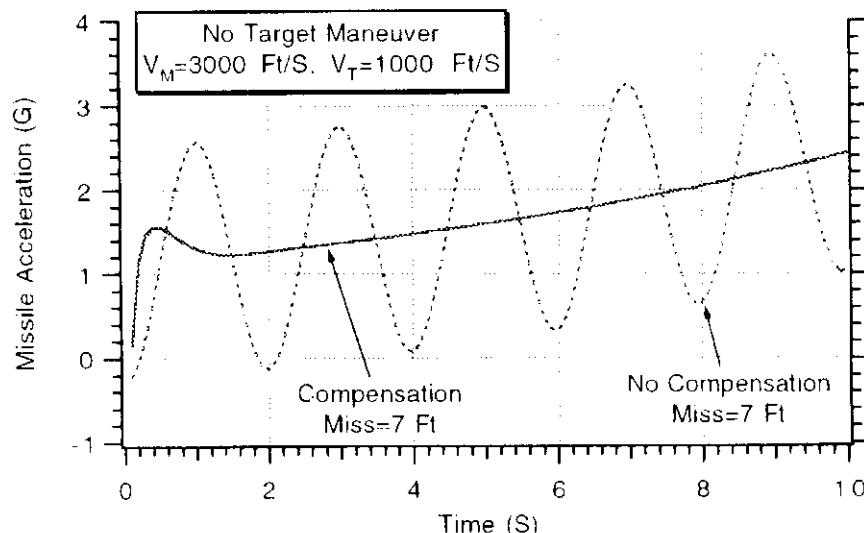


Fig. 10.11 Lead-lag network stabilizes beam rider guidance system.

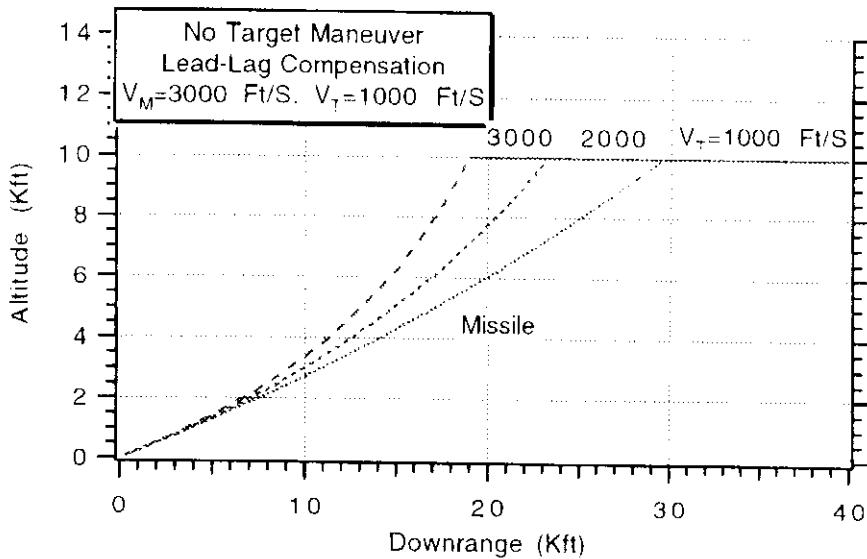


Fig. 10.12 Beam rider trajectories for different target speeds.

quite large compared to proportional navigation guidance. In this case, for a zero-lag guidance system, proportional navigation would yield zero miss distance.

Theoretically, one launches a missile, using proportional navigation guidance, on a collision triangle. The missile leads the target so that an intercept will occur if both the target and missile continue to fly in the same direction. As long as the missile can close on the target, proportional navigation performance is essentially independent of geometry. A beam rider missile, on the other hand, must always be near the beam. If the beam generator and missile are initially collocated, one launches a beam rider missile along the line of sight. Since a beam rider missile does not lead the target, it must change the trajectory continuously to catch the target. Figure 10.12 shows beam rider trajectories for cases in which the target speed is made a parameter. The curvature in the missile trajectory becomes more apparent as the target speed increases.

Figure 10.13 shows the corresponding beam rider commanded missile acceleration profiles, along with the resultant miss distances. In each case the beam rider

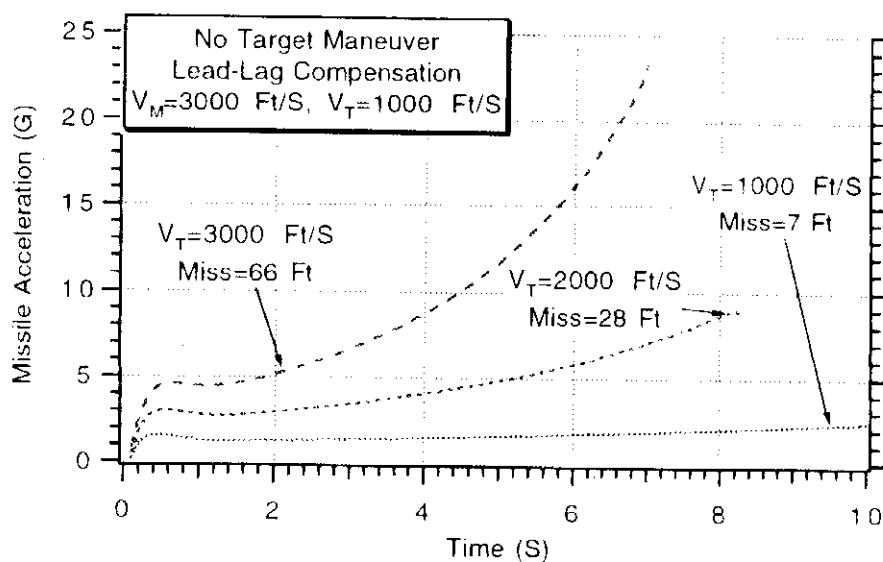


Fig. 10.13 Beam rider system performance is highly dependent on target speed.

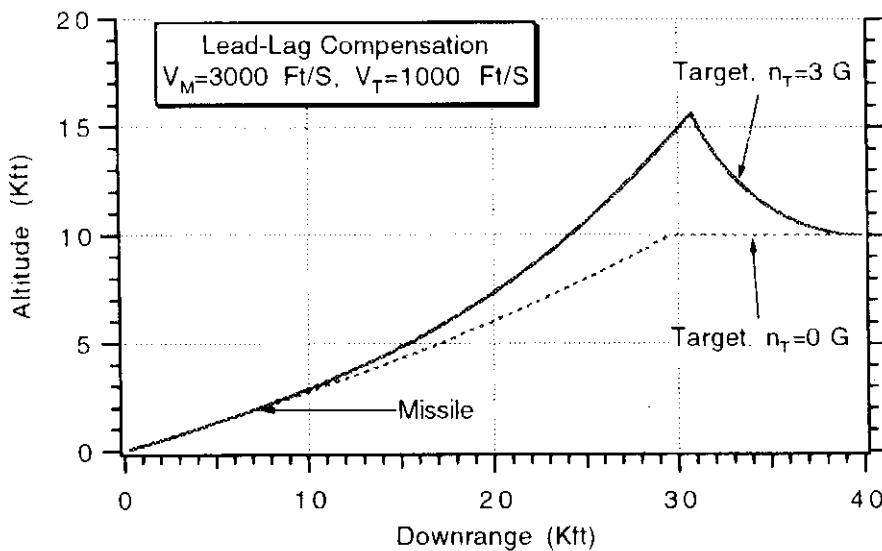


Fig. 10.14 Beam rider trajectories for maneuvering and nonmaneuvering targets.

missile had the previously discussed lead-lag compensation network with gain 10. We can see that both miss distance and acceleration increase with increasing target speed. Thus, beam rider performance, unlike proportional navigation performance, is highly dependent on the target speed. For the cases studied, a missile launched on a collision triangle and employing proportional navigation guidance would achieve zero miss distance with zero acceleration requirements!

Figure 10.14 compares the trajectories for beam rider guidance for a 3-g maneuvering target and a target that does not maneuver. In both cases the beam rider guidance system has a lead-lag network with gain 10 and a 3 to 1 missile to target speed advantage.

Although beam rider guidance appears to be successful against a maneuvering target, the miss distance increases from 7 ft (nonmaneuvering case) to about 19 ft, as shown in Fig. 10.15. In addition, the missile acceleration requirements increase significantly when the target maneuvers.

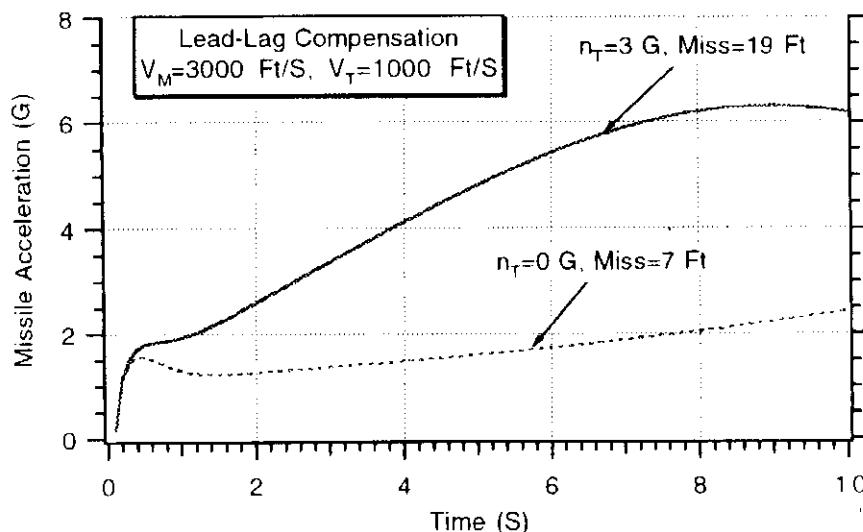


Fig. 10.15 Maneuvering target increases miss distance and acceleration requirements for beam rider guidance.

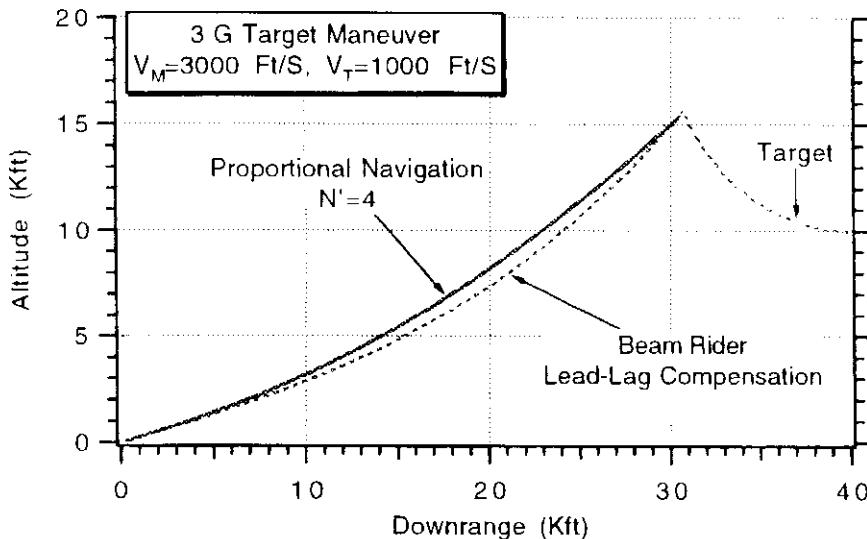


Fig. 10.16 Trajectory comparison of beam rider guidance and proportional navigation.

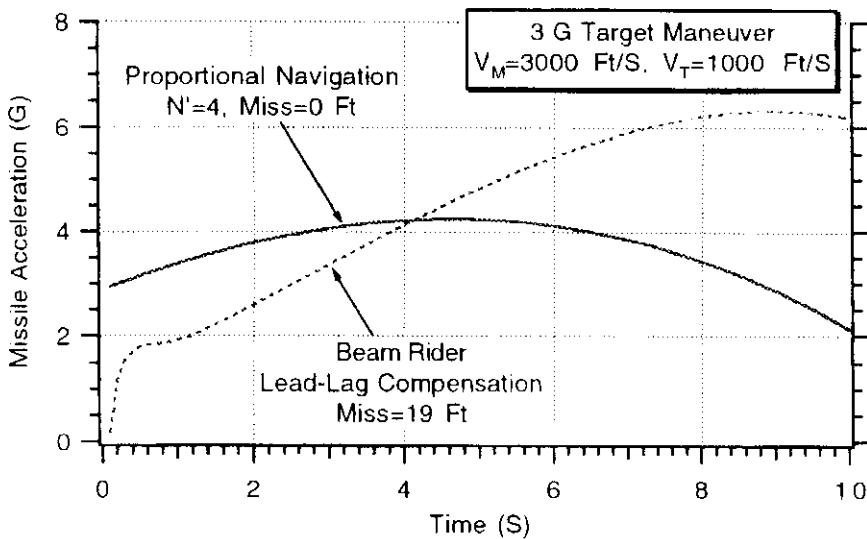


Fig. 10.17 Proportional navigation yields smaller miss distance and lower acceleration requirements than beam rider guidance.

Recall that for a zero-lag guidance system proportional navigation achieved zero miss distance against a maneuvering target. Figure 10.16 compares the trajectories of both beam rider guidance and proportional navigation for a case in which the target executes a 3-g target maneuver. We can see that there is less curvature in the proportional navigation trajectory because the missile is initially leading the target.

Figure 10.17 shows what we have already suspected, namely, that proportional navigation yields zero miss distance and much smaller acceleration requirements than a beam rider guidance system when the target is maneuvering.

Command to Line-of-Sight Guidance^{2,3}

Beam riding performance can be significantly improved by taking the beam motion into account. This is analogous to homing guidance in which proportional

navigation performance is improved by taking target maneuver information into account (augmented proportional navigation). If we recall that the angle of the beam with respect to the target (see Fig. 10.6) is given by

$$\theta_T = \tan^{-1}(R_{T2}/R_{T1})$$

then the angular velocity and acceleration of the beam can be found by taking successive derivatives of the preceding expression, yielding

$$\dot{\theta}_T = \frac{R_{T1}V_{T2} - R_{T2}V_{T1}}{R_T^2}$$

$$\ddot{\theta}_T = \frac{a_{T2}\cos\theta_T - a_{T1}\sin\theta_T - 2\dot{\theta}_T\dot{R}_T}{R_T}$$

where the derivative of the range from the radar to the target can be found from

$$\dot{R}_T = \frac{R_{T1}V_{T1} + R_{T2}V_{T2}}{R_T}$$

The acceleration perpendicular to the beam a_{TP} can be expressed in terms of the inertial coordinates of target acceleration, or

$$a_{TP} = -a_{T1}\sin\theta_T + a_{T2}\cos\theta_T$$

Therefore, combining terms yields an equivalent expression for the beam acceleration as

$$a_{TP} = R_T\ddot{\theta}_T + 2\dot{R}_T\dot{\theta}_T$$

By analogy, if we want the missile to stay on the beam, we are striving to ensure that

$$\ddot{\theta}_M = \ddot{\theta}_T$$

and

$$\dot{\theta}_M = \dot{\theta}_T$$

If these conditions are met, then the missile acceleration perpendicular to the beam can be found from

$$a_{MP} = R_M\ddot{\theta}_M + 2\dot{R}_M\dot{\theta}_M$$

Substitution yields the final expression for the beam acceleration:

$$a_{MP} = R_M\ddot{\theta}_T + 2\dot{R}_M\dot{\theta}_T$$

Adding the beam acceleration term a_{MP} to the nominal acceleration generated by the beam rider equations yields command to line-of-sight (CLOS) guidance. A command to line-of-sight guidance loop is shown in Fig. 10.18. Here we can see that the beam rider loop remains unchanged and an extra feedforward path, representing the acceleration of the beam, has been added. The beam rider acceleration command has thus been modified to include an extra term.

To see how the extra acceleration term changes system requirements and performance, a case was run for a nonmaneuvering target. Figure 10.19 compares

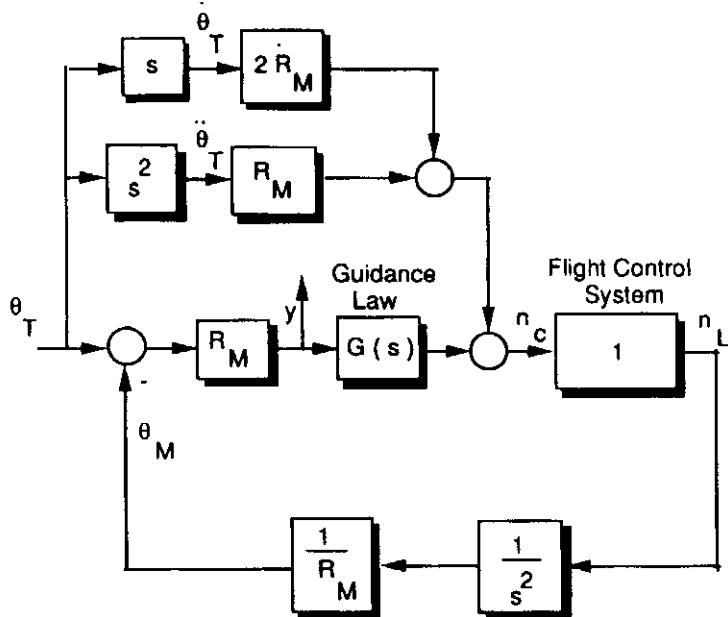


Fig. 10.18 Command to line-of-sight guidance system.

the acceleration profiles of both beam rider and CLOS guidance. First, the figure shows that the acceleration requirements at the beginning of the flight are somewhat less for command to line-of-sight guidance than beam rider guidance. The reason for this is that command to line-of-sight guidance is taking into account beam motion. Next, the figure also shows that, although for most of the flight the acceleration requirements for both guidance systems are about the same, the miss distance performance of command to line-of-sight guidance is better.

We saw in the previous section that beam rider performance was highly dependent, in terms of miss distance and acceleration requirements, on target speed. Figure 10.20 shows that, although the acceleration requirements of command to line-of-sight guidance are still dependent on target speed, near zero miss distance can be achieved against a nonmaneuvering target, independent of target speed!

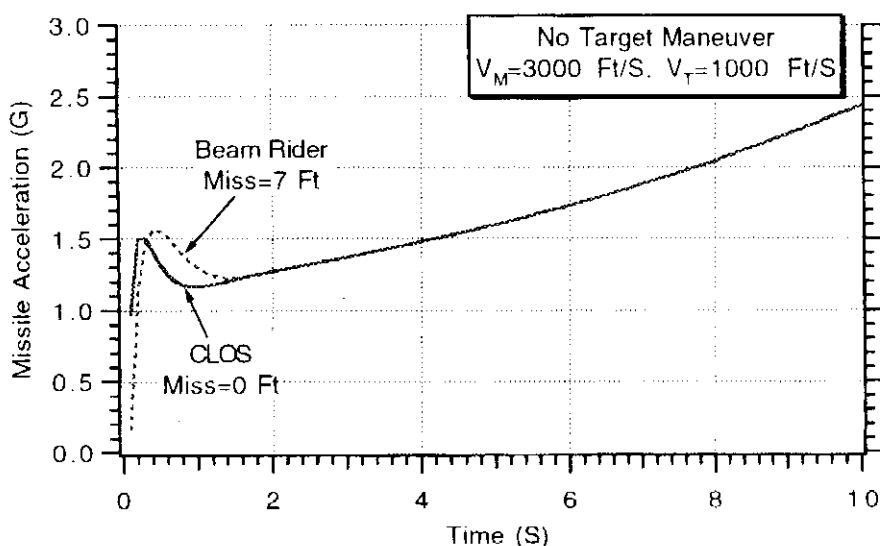


Fig. 10.19 Command to line-of-sight guidance yields smaller miss distance than beam rider guidance.

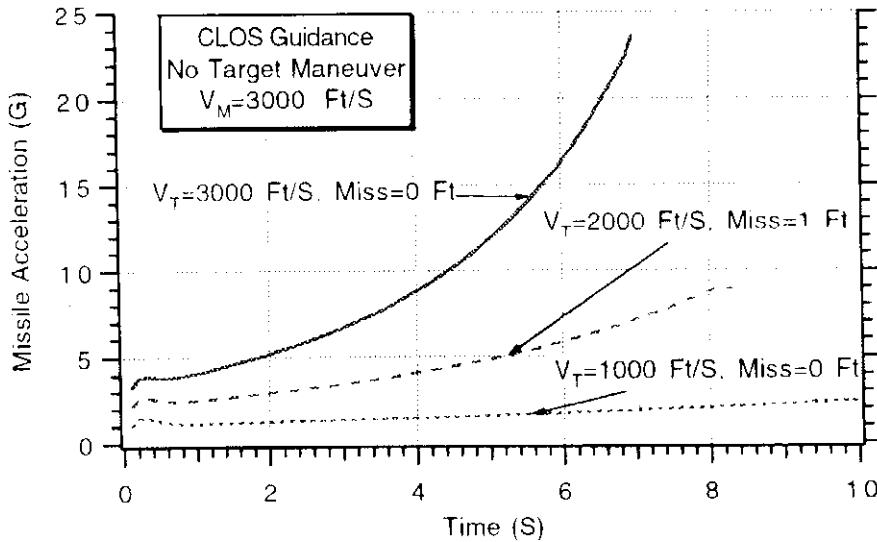


Fig. 10.20 CLOS acceleration requirements are dependent on target speed.

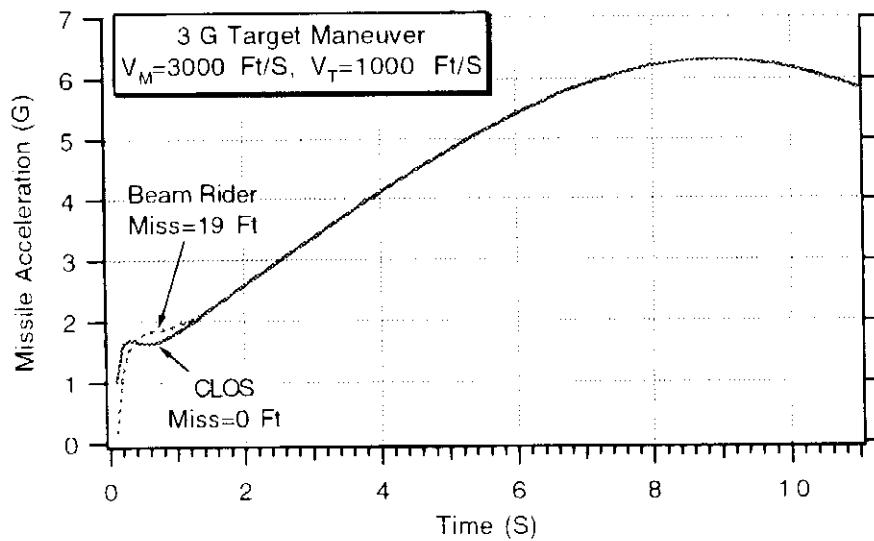


Fig. 10.21 CLOS guidance has smaller miss distance against maneuvering target.

Finally, Fig. 10.21 shows that the acceleration requirements for hitting a maneuvering target with command to line-of-sight guidance and beam rider guidance are about the same. However, command to line-of-sight guidance, in the case examined, had zero miss distance, whereas beam rider guidance yielded a miss distance of 19 ft.

Thus, we can see that taking the beam motion into account improves the performance, if not the acceleration requirements, of a beam rider system.

Summary

In this chapter we studied three different types of command guidance systems: proportional navigation, beam rider, and command to line of sight. We showed that a command proportional navigation guidance system will have to contend with more effective noise on the line of sight than its homing counterpart. It was

demonstrated that beam rider guidance required lead-lag compensation in order to guide effectively on the target. Beam rider performance, unlike that of proportional navigation, was shown to be very dependent on target speed and geometry. Finally, it was shown that beam rider performance could be substantially improved by taking beam motion into account.

References

- ¹Alpert, J., "Miss Distance Analysis for Command Guided Missiles," *Journal of Guidance, Control, and Dynamics*, Vol. 11, Nov.-Dec. 1988, pp. 481-487.
- ²Garnell, P., and East, D. J., *Guided Weapon Control System*, Pergamon, Oxford, 1977.
- ³Heap, E., "Methodology of Research into Command-Line-of-Sight and Homing Guidance," AGARD Lecture Series No. 52 on Guidance of Tactical Missiles, May 1952.



Tactical Zones

Introduction

In the material presented so far it has been assumed that the missile can reach the target and generate sufficient lift to maneuver. For purposes of understanding fundamental guidance issues, we have neglected the fact that the ability of a tactical aerodynamic missile to maneuver is dependent upon its speed, physical characteristics (i.e., wing size), and altitude. In addition, we have also assumed impulsive constant velocity missiles and have not taken into consideration that a missile must burn propellant in order to get up to speed. The resultant missile total weight is directly related to its payload weight and design speed (i.e., propellant weight). Actually, it will soon become obvious that in some tactical missiles most of the weight is the propellant weight. Finally, we have also neglected the fact that while an aerodynamic missile is coasting the missile speed diminishes due to atmospheric drag. In this chapter we shall briefly address previously neglected issues and show how to modify previous computations to account for these important effects.

Velocity Computation

A tactical missile gets up to speed by burning propellant. If the missile is launched from the air, it already has a large initial speed. However, if the missile is launched from the ground, it needs more propellant to reach the same speed since it is starting from rest. In addition, a ground-based missile needs additional propellant since it must travel through more of the denser atmosphere. Figure 11.1 shows a typical weight and thrust profile for a boost-coast missile. The initial or total weight of the missile is denoted W_T , and its final weight, after the propellant is expended, is the glide weight W_G . The glide weight consists of the missile structure, electronics, and warhead. While propellant is being consumed, the thrust is assumed to be constant, with magnitude T .

We can find the magnitude of the missile velocity after all of the propellant is consumed from basic physics. Applying Newton's second law yields^{1,2}

$$F = ma = m \frac{dV}{dt} = T$$

The change in velocity with respect to time can be expressed in terms of thrust and weight as

$$\frac{dV}{dt} = \frac{T}{m} = \frac{Tg}{W}$$

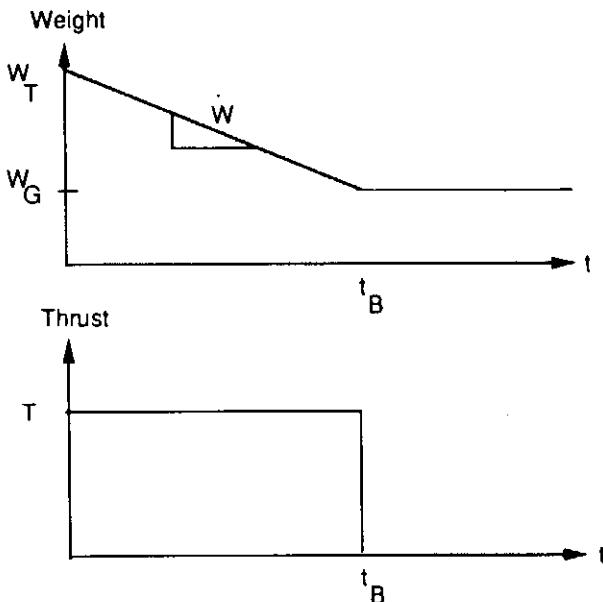


Fig. 11.1 Boost-coast thrust-weight profile.

Recognizing that as long as the missile is burning propellant ($0 < t < t_B$), we can express the instantaneous missile weight as

$$W = \dot{W}t + W_T$$

where the derivative of the weight is negative (weight is decreasing). Thus, we can find an expression for the change in velocity due to the burning of propellant by direct integration, or

$$\int_{V_0}^{V_1} dV = Tg \int_0^{t_B} \frac{dt}{W}$$

Substitution of the expression for the missile weight into the integral yields

$$V_1 - V_0 = \Delta V = Tg \int_0^{t_B} \frac{dt}{\dot{W}t + W_T}$$

Evaluation of the integral yields

$$\Delta V = \frac{-Tg}{\dot{W}} \ln \frac{W_T}{\dot{W}t_B + W_T}$$

Thus, the change in velocity depends only on the missile total weight, glide weight, thrust magnitude, and rate at which the propellant is burning. The preceding velocity formula, also known as the rocket equation, represents the maximum change in velocity we can impart. Practical effects such as gravity and atmospheric drag will usually work in the direction of decreasing ΔV . The preceding expression can be made more concise and useful by specifying fuel effectiveness in terms of a parameter known as the specific impulse I_{sp} . It is a positive number in units of seconds and is related to the thrust and change in missile weight according to

$$I_{sp} = \frac{-T}{\dot{W}}$$

More fuel-efficient missiles have higher values of specific impulse. Typically, for tactical missiles, the specific impulse has values ranging from 200 s to 300 s. By substituting the specific impulse definition into the velocity change formula, we get

$$\Delta V = I_{sp} g \ln \frac{W_T}{W_G}$$

Now the change in missile velocity during a burn depends only on the total weight, glide weight, and specific impulse. However, the total missile weight is the sum of the glide weight and the propellant weight, or

$$W_T = W_P + W_G$$

The fuel mass fraction mf is defined as the ratio of the missile propellant weight to the total missile weight, or

$$mf = \frac{W_P}{W_T}$$

Because a ground-to-air missile requires more fuel than an air-to-air missile to reach the same speed, it would have a larger fuel mass fraction value. We can now express the change in missile velocity in terms of the specific impulse and fuel mass fraction, or

$$\Delta V = I_{sp} g \ln \frac{1}{1 - mf}$$

Using the preceding equation, Fig. 11.2 shows how the change in missile velocity varies with fuel mass fraction and specific impulse. We can see from the figure that, if the fuel mass fraction is 0.3 and the specific impulse is 200 s, the change in velocity is about 2300 ft/s. This means that, if the missile were launched from the ground, its final speed, in the absence of drag and gravitational effects, would be 2300 ft/s. If the missile with the same fuel mass fraction were launched from an aircraft traveling at 1000 ft/s, its final speed would be 3300 ft/s.

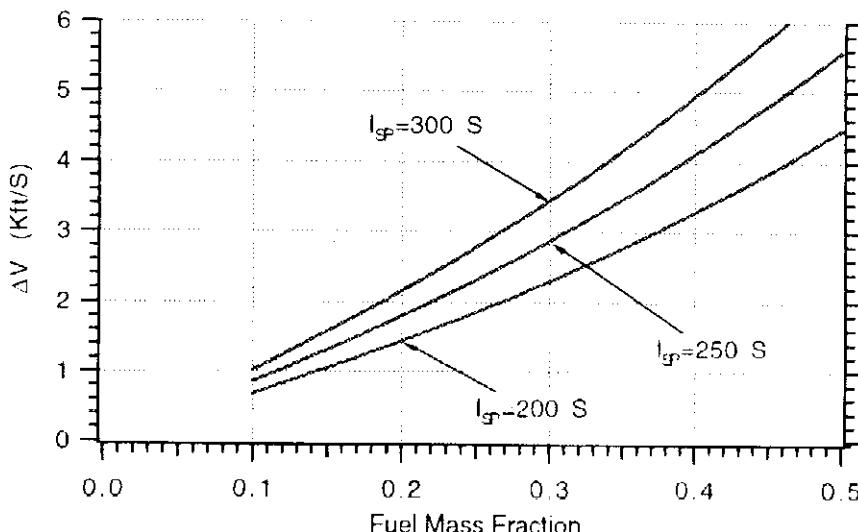


Fig. 11.2 Increasing fuel mass fraction or specific impulse increases missile speed.

Drag³

Tactical missiles work within the atmosphere. Aerodynamic drag causes the missile to slow down and have less maneuver capability. The drag F_{drag} can be expressed as

$$F_{\text{drag}} = Q S_{\text{ref}} C_{D0}$$

where Q is the dynamic pressure, S_{ref} a reference area, and C_{D0} the zero-lift drag. The dynamic pressure is a function of the air density ρ and velocity V and is given by

$$Q = \frac{\rho V^2}{2}$$

In the English system of units used throughout the text, air density is measured in slug per cubic foot (slug/ft³). The reference area is the cross-sectional area of the missile body and is therefore related to the physical characteristics of the missile. The zero-lift drag is a function of the missile speed and aerodynamic shape. Since the air density decreases with altitude, the influence of drag is greatest at the lower altitudes. For analytical reasons it is convenient to use an exponential approximation to the atmosphere. One such approximation below 30,000 ft altitude is given by

$$\rho = 0.002378 e^{-h/30,000} \quad (h < 30,000 \text{ ft})$$

whereas above 30,000 ft the exponential approximation becomes

$$\rho = 0.0034 e^{-h/22,000} \quad (h \geq 30,000 \text{ ft})$$

where h is measured in feet.

To check the validity of the exponential approximations, the 1962 U.S. Standard Atmosphere is displayed as a function of altitude in Fig. 11.3. Superimposed on the figure are the exponential approximations. The solid curve represents the

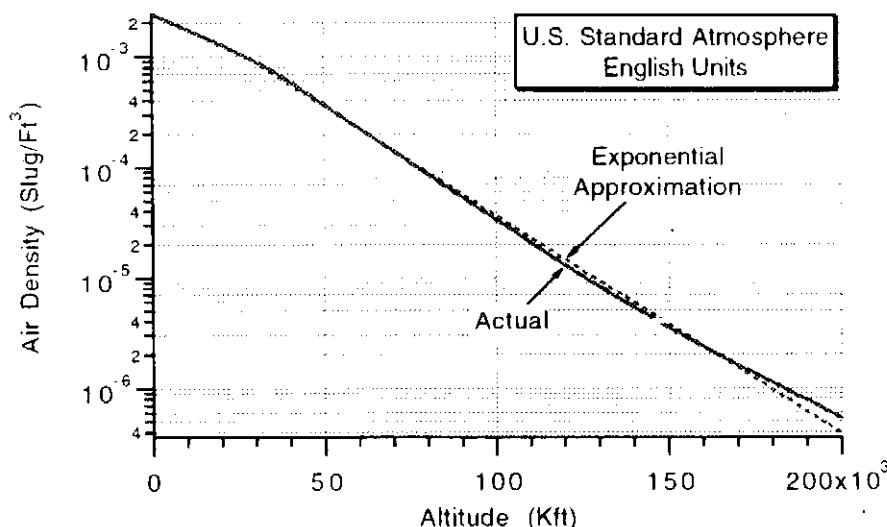


Fig. 11.3 Exponential approximation for air density is very accurate.

actual data points for the standard U.S. atmosphere, whereas the dashed curve represents the preceding exponential approximation. We can see that the exponential approximation is quite accurate.

In the absence of induced drag effects the drag F_{drag} can be expressed in terms of Newton's second law as

$$F = ma = -F_{\text{drag}} = m \frac{dV}{dt}$$

Therefore, the rate of change of velocity can be found from

$$\frac{dV}{dt} = \frac{-F_{\text{drag}}}{m} = \frac{-QS_{\text{ref}}C_{D0}}{m} = \frac{-0.5\rho V^2 S_{\text{ref}} C_{D0}}{m}$$

Setting up the integral in a useful form for integration, we get

$$\int_{V_0}^{V_1} \frac{dV}{V^2} = \int_{t_0}^{t_1} \frac{-0.5\rho S_{\text{ref}} C_{D0}}{m} dt$$

If we assume that the air density does not change (constant altitude) and that the zero-lift drag is constant, we can get a closed-form expression for the new velocity due to drag, or

$$\frac{1}{V_1} = \frac{\rho S_{\text{ref}} C_{D0} \Delta t}{2m} + \frac{1}{V_0}$$

where V_0 is the initial velocity, V_1 is the velocity Δt seconds later, and Δt is defined as

$$\Delta t = t_1 - t_0$$

If we define the initial drag deceleration D_0 to be

$$D_0 = \frac{\rho S_{\text{ref}} C_{D0} V_0^2}{2m}$$

and a characteristic time T_0 as

$$T_0 = V_0/D_0$$

then we can express the velocity Δt seconds later in the simpler form

$$V_1 = \frac{V_0 T_0}{T_0 + \Delta t}$$

Integrating again yields the total distance R covered in Δt seconds as

$$R = V_0 T_0 \ln \left(1 + \frac{\Delta t}{T_0} \right)$$

We now have sufficient information to perform some preliminary calculation concerning the effects of drag on velocity loss and range covered. However, it is first important to see how the magnitude of the drag deceleration is influenced by the zero-lift drag and altitude. Often it is convenient to combine the weight, reference

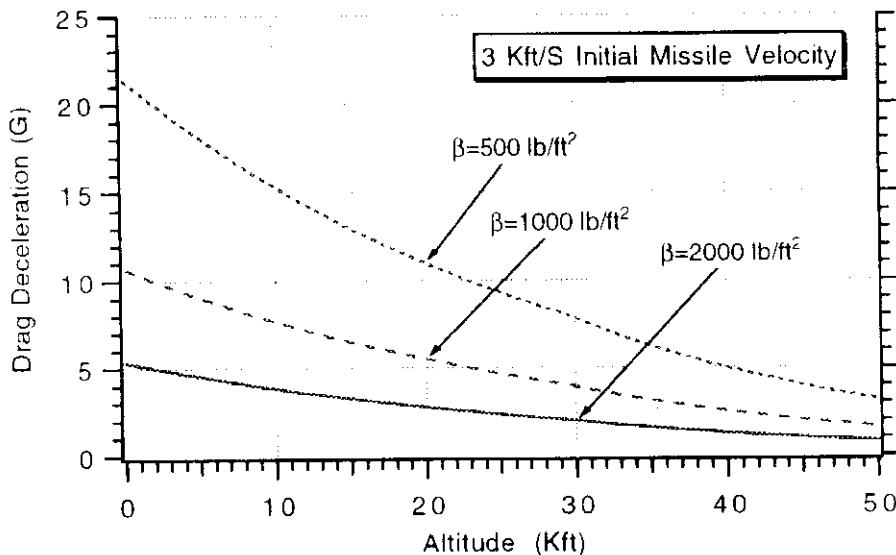


Fig. 11.4 Drag deceleration decreases with increasing altitude and increasing ballistic coefficient.

area, and zero-lift drag into an expression known as the ballistic coefficient β , which is defined as

$$\beta = \frac{W}{C_{D0} S_{\text{ref}}}$$

where β is in units of pounds per square feet. Figure 11.4 shows how the drag deceleration varies with altitude for different values of ballistic coefficient. We can see that increasing the ballistic coefficient (reducing zero-lift drag) or increasing the altitude reduces the drag deceleration. It is also apparent that, for a velocity of 3000 ft/s, a ballistic coefficient of 500 lb/ft² yields a drag deceleration of about 22 g at sea level, 10 g at 25 kft, and 3.5 g at 50 kft. In addition, Fig. 11.4 shows that, as altitude increases, the drag deceleration becomes less dependent on the ballistic coefficient. Eventually, at the higher altitudes, there is no deceleration due to drag.

If we still assume a missile at sea level with an initial velocity of 3000 ft/s, missile speed decreases due to drag as a function of time for different ballistic coefficients, as shown in Fig. 11.5. The velocity drops to about half of its original value in only 4.5 s for $\beta = 500 \text{ lb/ft}^2$, in about 9 s for $\beta = 1000 \text{ lb/ft}^2$, and in about 17.5 s for $\beta = 2000 \text{ lb/ft}^2$.

Figure 11.6 shows the ranges covered for the cases given earlier. At sea level the range covered until the missile velocity drops to half of its value is about 9 kft for $\beta = 500 \text{ lb/ft}^2$, about 20 kft for $\beta = 1000 \text{ lb/ft}^2$, and about 35 kft for $\beta = 2000 \text{ lb/ft}^2$. If the missile is not considered to be effective after it has dropped more than half of its velocity, then we can consider these values to be kinematic zones of effectiveness at sea level. The missile will have longer kinematic reach if its ballistic coefficient is higher.

Figure 11.7 shows the velocity loss at 50 kft altitude. We can see that increasing the altitude reduces the missile's velocity loss due to drag and increases the effective range of the missile. We can see from Fig. 11.7 that at 50 kft it takes much longer for the velocity to drop to half of its original value. At 50 kft the velocity drops

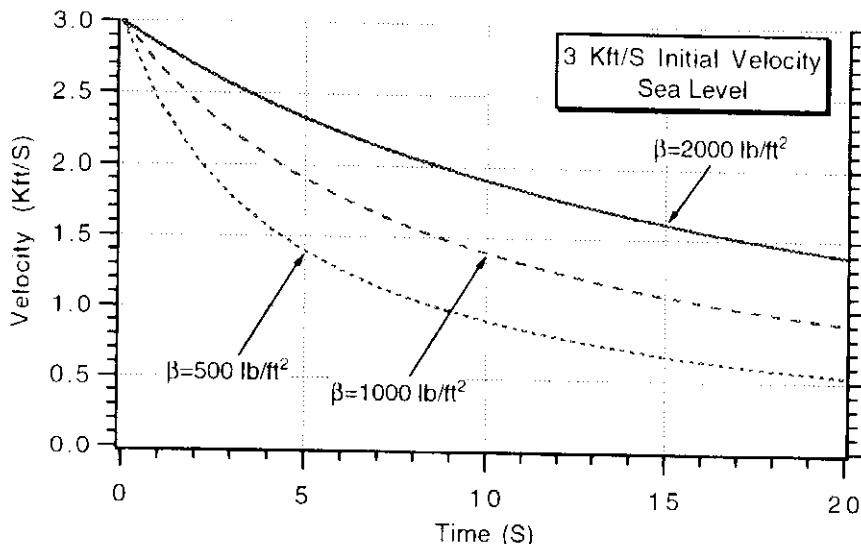


Fig. 11.5 Velocity drops faster with smaller ballistic coefficient.

to about half of its original value in about 30 s (vs 4.5 s at sea level) for $\beta = 500 \text{ lb/ft}^2$, in about 60 s (vs 9 s at sea level) for $\beta = 1000 \text{ lb/ft}^2$, and in more than 100 s (vs 17.5 s at sea level) for $\beta = 2000 \text{ lb/ft}^2$.

Figure 11.8 shows the range covered as a function of time for missiles with varying ballistic coefficients at an altitude of 50 kft. We can see from the figure that the range covered until the velocity drops to half of its value is about 70 kft (vs 9 kft at sea level) for $\beta = 500 \text{ lb/ft}^2$, about 125 kft (vs 20 kft at sea level) for $\beta = 1000 \text{ lb/ft}^2$, and more than 200 kft (vs 35 kft at sea level) for $\beta = 2000 \text{ lb/ft}^2$.

In summary, increasing the ballistic coefficient (or reducing the zero-lift drag) can have a big payoff in terms of increased zone of effectiveness for aerodynamic missiles that must fly through the more dense atmosphere at low altitudes. Tactical radar homing missiles tend to have a nose with a high fineness ratio in order to make them more aerodynamically efficient. The high fineness-ratio nose also tends to exacerbate parasitic radome effects.⁴

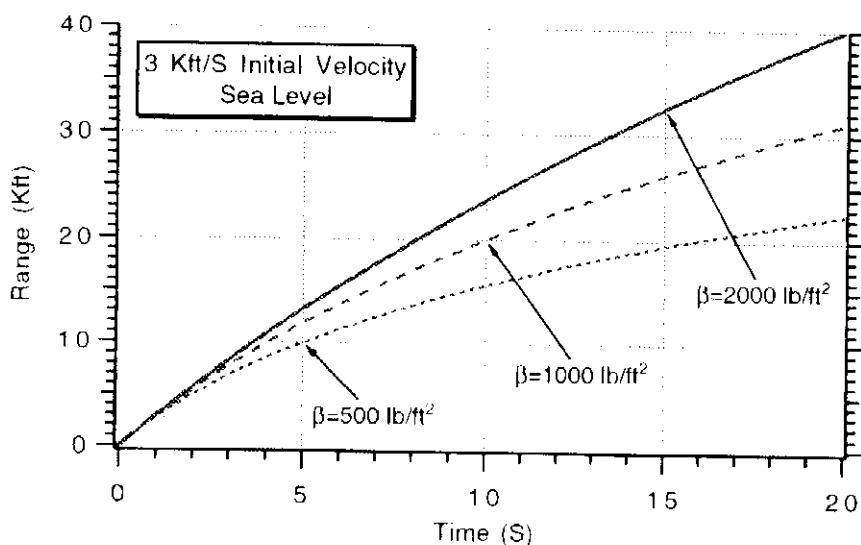


Fig. 11.6 Zone of effectiveness at sea level is not large.

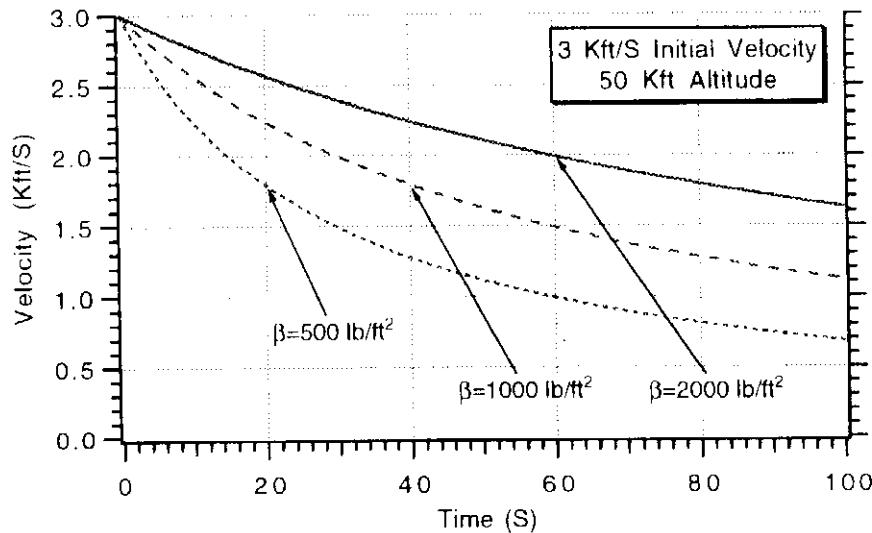


Fig. 11.7 Velocity drops much slower at 50 kft altitude.

Acceleration^{1,2}

Just as there was a drag coefficient to determine slowdown, there is a lift coefficient C_L to determine missile maneuverability. From Newton's second law we can say that

$$F = ma = mn_L = Q S_{\text{ref}} C_L$$

where n_L represents the missile's acceleration capability. Therefore, the missile acceleration capability, expressed in units of gravity, is given by

$$\frac{n_L}{g} = \frac{0.5\rho V^2 S_{\text{ref}} C_L}{W}$$

where W is missile weight in units of pounds. The lift coefficient is a function of the missile aerodynamic shape, speed, angle of attack, and wing size. Larger wings and increasing angle of attack both work in the direction of increasing the lift coefficient.

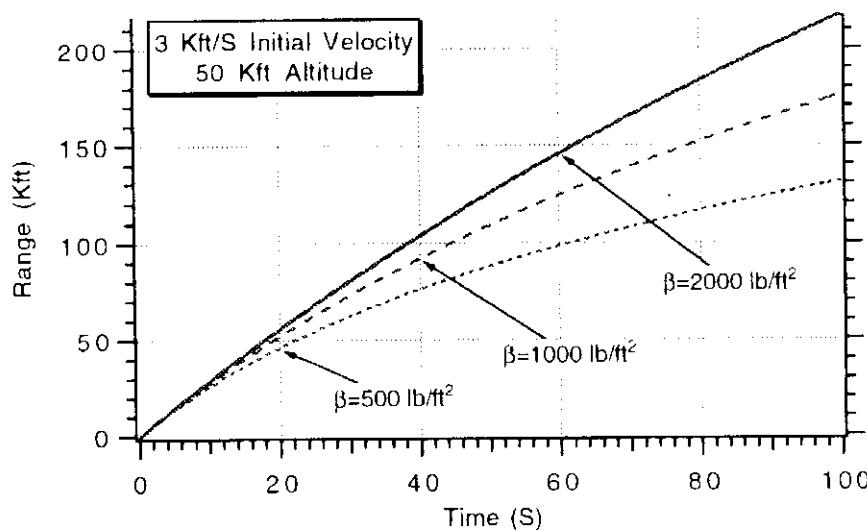


Fig. 11.8 Zone of effectiveness is much greater at 50 kft altitude.

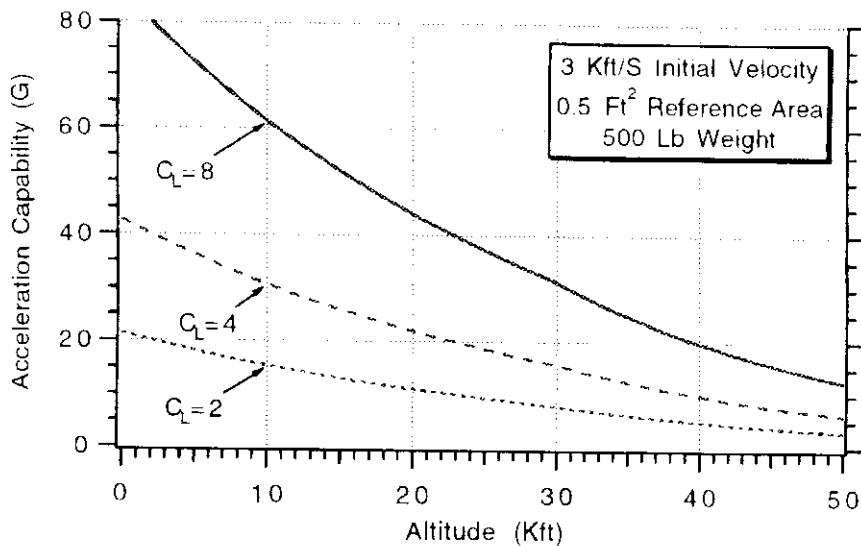


Fig. 11.9 Missile maneuverability decreases dramatically with increasing altitude.

To demonstrate the sensitivity of the missile acceleration capability to the lift coefficient and altitude, it is best to consider a numerical example. Consider a missile weighing 500 lb with a 0.5-ft² reference area and traveling at 3000 ft/s. Figure 11.9 shows how the missile acceleration capability decreases with increasing altitude and decreasing lift coefficient. It is important to note that a missile may have an aerodynamic acceleration capability, at a given altitude, which is far in excess of its structural capability. A loading analysis is required to set practical limits on the maximum allowable commanded missile acceleration. Figure 11.9 shows an example of a missile ($C_L = 4$) which has a 40-g capability at sea level that diminishes to about a 10-g capability at 50 kft altitude. Reducing missile weight or increasing the missile reference area (but keeping weight constant) works in the direction of increasing the missile aerodynamic maneuverability.

Speed also plays an important role in determining missile aerodynamic maneuverability. Figure 11.10 shows that decreasing the missile speed significantly decreases the missile maneuverability. A missile that travels at 3000 ft/s at 20 kft

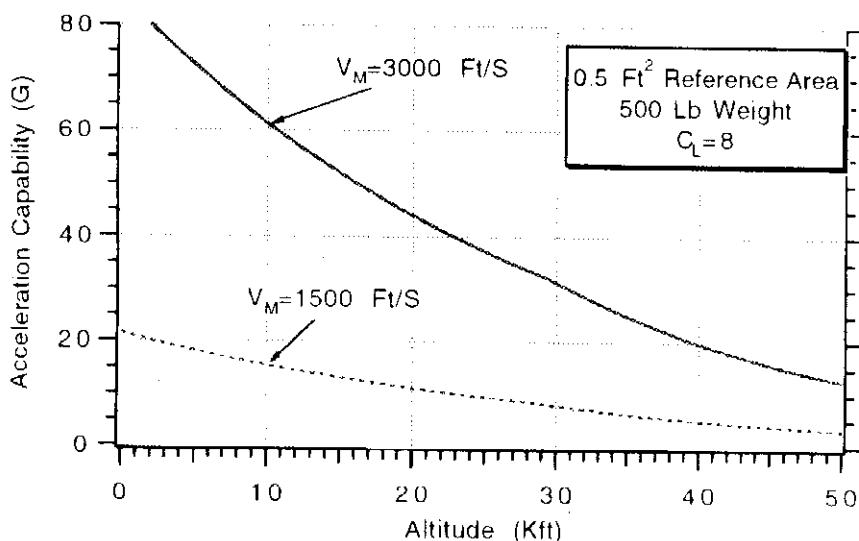


Fig. 11.10 Decreasing speed decreases maneuverability.

altitude has a maneuverability in excess of 40 g. Halving the missile speed more than halves its maneuverability. We have seen in previous chapters that a missile requires a certain acceleration advantage to effectively engage maneuvering targets. Therefore, for a given altitude and missile configuration, there is a minimum speed at which the missile can effectively engage a responsive threat.

Gravity

Thus far in our analysis we have neglected gravity and assumed a constant-altitude missile. Actually, gravity will eventually cause a coasting missile to crash to the ground. If we neglect the atmosphere and launch an impulsive 3000-ft/s missile at various flight-path angles γ , we will get different range capabilities due to gravity alone, as shown in Fig. 11.11. As expected, the 45-deg launch results in maximum range.

Atmospheric drag will of course prevent the missile from achieving the range capabilities indicated in Fig. 11.11. Listing 11.1 presents a FORTRAN simulation of an impulsive missile launched at a flight-path angle *GAMDEG* missile in the presence of gravity and an atmosphere (zero-lift-drag). As in the previous section, the ballistic coefficient is used rather than the zero-lift drag coefficient to account for zero-lift-drag-induced slowdown effects. Lift-induced drag is neglected in this simplified analysis.

Consider a 45-deg sea-level launch of a missile that attains a velocity of 3000 ft/s instantaneously. Cases were run in which the ballistic coefficient varied from 500 lb/ft² to infinity (no drag). Figure 11.12 shows that drag dramatically changes the maximum range capability of the interceptor. The maximum range for a 45-deg launch angle decreases from about 300 kft (no-drag case) to about 100 kft for a ballistic coefficient of 2000 lb/ft², to about 55 kft for a ballistic coefficient of 1000 lb/ft², and to 30 kft for a ballistic coefficient of 500 lb/ft².

Drag becomes less important at the higher altitudes. The previous case was repeated, but the initial launch altitude was increased from sea level to 50 kft.

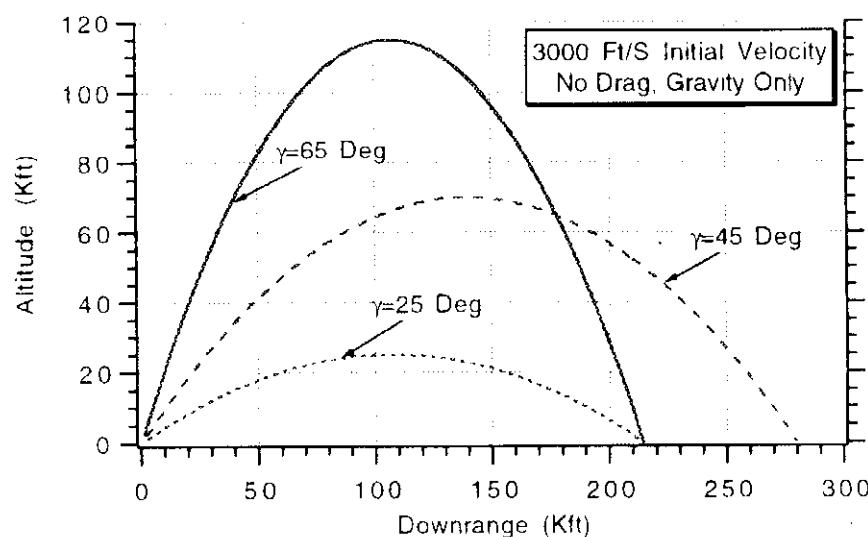


Fig. 11.11 Trajectory profiles for various launch angles.

Listing 11.1 Trajectory simulation

```

INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
H=.01
VM=3000.
BETA=1000.
T=0.
S=0.
GAMDEG=45.
VM1=VM*COS(GAMDEG/57.3)
VM2=VM*SIN(GAMDEG/57.3)
RM1=0.
RM2=0.
10 IF(T>0..AND.RM2<=0.)GOTO 999
RM1OLD=RM1
RM2OLD=RM2
VM1OLD=VM1
VM2OLD=VM2
STEP=1
GOTO 200
66 STEP=2
RM1=RM1+H*VM1
RM2=RM2+H*VM2
VM1=VM1+H*AM1
VM2=VM2+H*AM2
T=T+H
GOTO 200
55 CONTINUE
RM1=.5*(RM1OLD+RM1+H*VM1)
RM2=.5*(RM2OLD+RM2+H*VM2)
VM1=.5*(VM1OLD+VM1+H*AM1)
VM2=.5*(VM2OLD+VM2+H*AM2)
S=S+H
IF(S<.99999)GOTO 10
S=0.
RM1K=RM1/1000.
RM2K=RM2/1000.
WRITE(*,97)T, RM1K, RM2K
WRITE(1,97)T, RM1K, RM2K
GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
IF(RM2<30000.)THEN
    RHO=.002378*EXP(-RM2/30000)
ELSE
    RHO=.0034*EXP(-RM2/22000)
ENDIF
VM=SQRT(VM1**2+VM2**2)

```

(Contd.)

Listing 11.1 (Continued)

```

Q=.5*RHO*VM*VM
GAM=ATAN2(VM2,VM1)
DRAG=Q*32.2/BETA
AM1=-DRAG*COS(GAM)
AM2=-32.2-DRAG*SIN(GAM)
IF(STEP-1)66,66,55
999 CONTINUE
VM=SQRT(VM1**2+VM2**2)
RM1K=RM1/1000.
RM2K=RM2/1000.
WRITE(*,97)T,VM1,VM2
WRITE(1,97)T,VM1,VM2
PAUSE
CLOSE(1)
END

```

Figure 11.13 shows that the differences between the drag free trajectory and the one in which the ballistic coefficient is 2000 lb/ft² is much smaller than before. The maximum range without drag in this case is about 350 kft, whereas the maximum range for a ballistic coefficient of 2000 lb/ft² is about 275 kft. For a ballistic coefficient of 500 lb/ft² the maximum range reduces to 175 kft.

Thus, we can see that drag not only plays a role in reducing missile speed so that it has less acceleration capability but it also plays a significant role in determining the kinematic reach of the missile. For long-range ground-launched missiles, trajectory shaping is often used to get the missile to higher altitudes as quickly as possible so that range and velocity losses due to drag can be minimized.

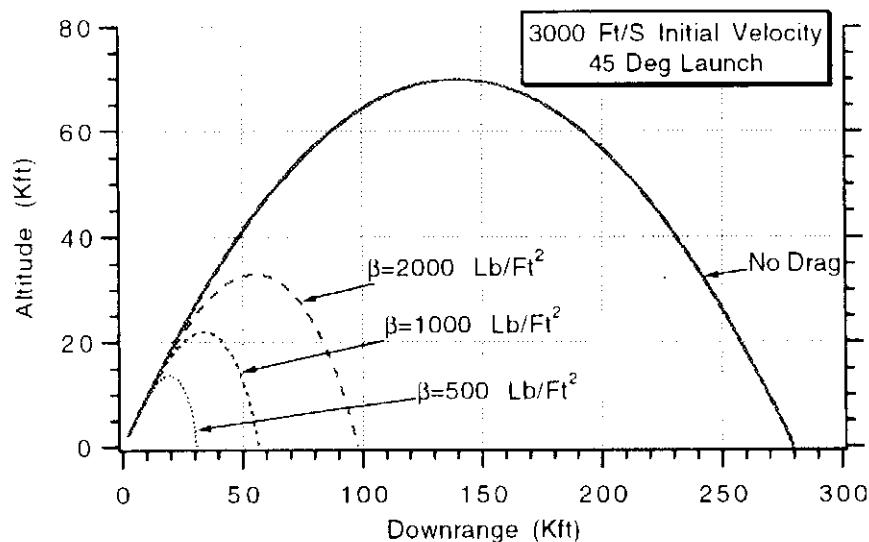


Fig. 11.12 Drag dramatically reduces range capability of missile.

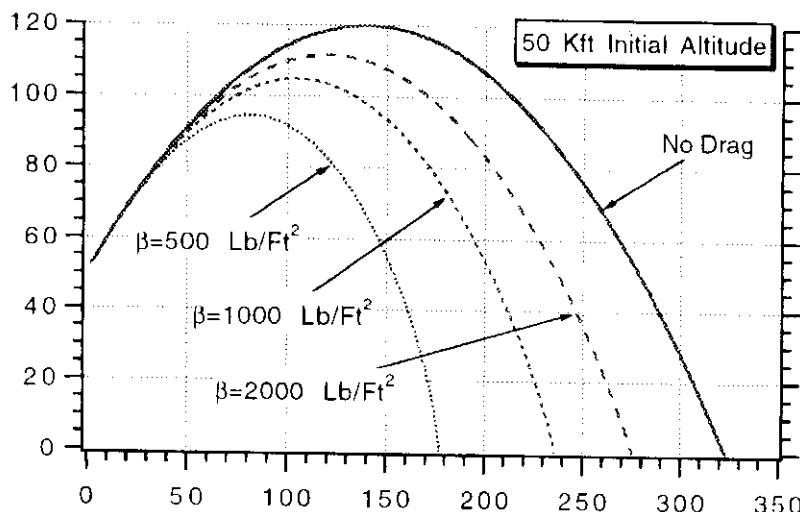


Fig. 11.13 Drag effects reduce considerably if launch altitude is high.

Summary

In this chapter we have considered and shown how to model previously neglected effects. A simple design formula, known as the rocket equation, was derived in order to show the influence of propellant weight on missile speed capability. In addition, we investigated how drag reduces the kinematic reach of the missile and how the atmosphere helps in providing the missile with lift to maneuver. Finally, a numerical example was presented showing how to generate flyout zones.

References

- ¹Locke, A. S., *Guidance*, Van Nostrand, Toronto, 1955.
- ²Jerger, J. J., *System Preliminary Design*, Van Nostrand, Princeton, NJ, 1960.
- ³Travers, P., "Interceptor Dynamics," Unpublished Lecture Notes, Raytheon, circa 1971.
- ⁴Eichblatt, E. (ed.), *Test and Evaluation of the Tactical Missile*, Vol. 119, Progress in Astronautics and Aeronautics, AIAA, Washington, DC, 1989.



Strategic Considerations

Introduction

In all of the work presented so far we have based our models on a flat-Earth, constant-gravity model. For tactical interceptor missions, where speeds are less than 5000 ft/s, altitudes under 100 kft, and ranges covered under 100 n.mi., these assumptions are reasonable. In the strategic world where speeds are near-orbital and the distances covered are intercontinental, the flat-Earth constant-gravity assumption is not only inaccurate but can also give misleading results in terms of the size of the zone of effectiveness. However, we shall also see in subsequent chapters that the guidance lessons learned in the tactical world are still valid and give valuable insight into the requirements and effectiveness of strategic interceptors. Before proceeding with the development of models, it is worthwhile to review some of the historical background of strategic ballistic missiles.

Background¹

Germany's V-2 was the world's first long-range ballistic missile. When a loophole in the Treaty of Versailles was found, the Wehrmacht's Ordnance Department authorized the development of this large long-range rocket and selected Artillery Captain Walter Dornberger to supervise the project. After 14 years of testing, the V-2 was ready for field use and was finally deployed in the fall of 1944, and it was launched from mobile field battery positions in France and Holland. Each single-stage missile weighed nearly 30,000 lb, reached a burnout velocity of about 6000 ft/s, and had a range of approximately 230 miles. Between September 1944 and March 1945 German field units launched more than 3000 V-2 missiles. Approximately 1900 missiles were launched against Allied targets on the European continent, primarily Antwerp, Belgium. The rest fell on London and southern England.

After the war, the U.S. Army brought German V-2 engineers and enough pieces for about 80 missiles into this country. As part of Project Hermes, more than 70 V-2s were launched by the U.S. during the late 1940s and early 1950s. These rockets formed the basis for U.S. strategic ballistic missile technology and were also essential for subsequent advances in the exploration of space.

Gravitational Model

In the tactical world, in the absence of thrust, drag, and lift, the flat-Earth constant-gravity assumption is easy to understand. In this mathematical model

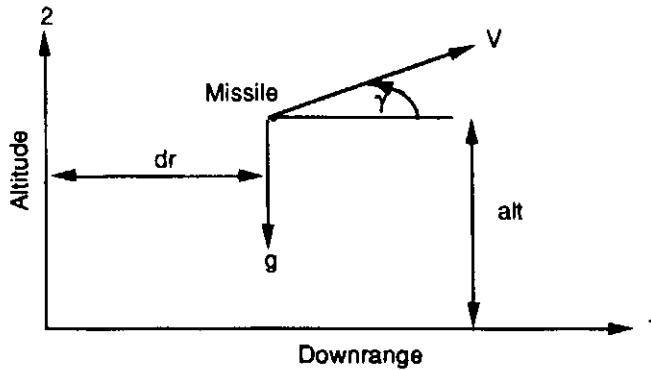


Fig. 12.1 Missile in gravity field using flat-Earth model.

the gravitational acceleration is independent of altitude with value 32.2 ft/s^2 , always in a downward direction. The tactical missile inertial coordinate system is fixed to the surface of a flat Earth and is depicted in Fig. 12.1. Here the missile has velocity V and is at a flight-path angle γ with respect to the surface of the Earth. In addition the missile is in a constant-gravity field with acceleration level g . The missile is at an initial location that is distance dr downrange from the origin of the coordinate system and at an altitude alt from the surface of the Earth.

The differential equations acting on the missile are

$$\dot{V}_1 = 0$$

$$\dot{V}_2 = -g$$

$$\dot{R}_1 = V_1$$

$$\dot{R}_2 = V_2$$

where V is velocity and R is range. The down-range component is denoted by 1, and the altitude component is denoted by 2. The initial conditions for velocity and position are given by

$$V_1(0) = V \cos \gamma$$

$$V_2(0) = V \sin \gamma$$

$$R_1(0) = dr$$

$$R_2(0) = alt$$

Since the coordinate system is inertial, we can integrate directly in the down-range and altitude directions to get velocity from acceleration and position from velocity. In this model, the gravitational acceleration is always 32.2 ft/s^2 in the negative altitude direction, regardless of altitude. Therefore, we know that this model can only be valid at the lower altitudes, since in actuality the gravitational acceleration decreases with increasing altitudes.

In general, a body in a gravitational field can be depicted in an Earth-centered coordinate system shown in Fig. 12.2. In this system the Earth is nonrotating and the gravitational acceleration acting on the missile is toward the center of the Earth. The missile has velocity V with respect to a reference that is tangent to the Earth

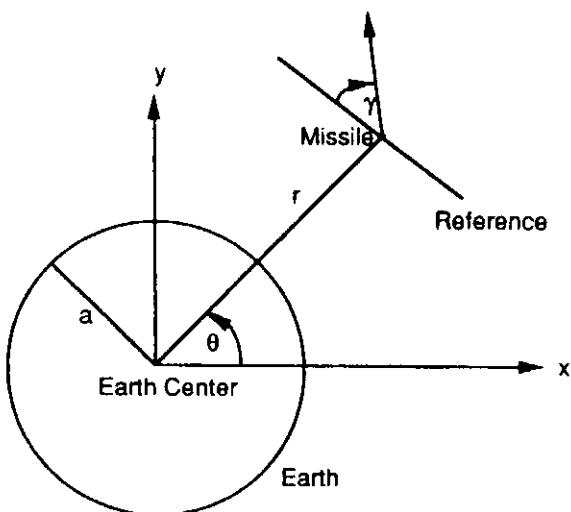


Fig. 12.2 Missile in gravity field.

and perpendicular to r (line from center of Earth to missile). The radius of the Earth is denoted by a in this figure.

According to Newton's law of universal gravitation, two bodies attract each other with a force that acts along a line connecting the two bodies. The force is proportional to the product of the masses of the two bodies and inversely proportional to the square of the distance between the two bodies. If one of the bodies is the Earth and the mass of the second body is negligible compared to the Earth, Newton's law of universal gravitation can be expressed in vector form as^{2,3}

$$\ddot{r} = \frac{-gm r}{r^3}$$

where r is a vector from the center of the Earth to the second body, and gm is known as the gravitational parameter with the value

$$gm = 1.4077 * 10^{16} \text{ ft}^3/\text{s}^2$$

For simulation purposes and to be consistent with the work we have already done with tactical interceptors, it is natural to desire to express Newton's law in Cartesian coordinates. We shall soon see that for analytical purposes it will be more convenient to work in polar coordinates. By substituting

$$r = xi + yj$$

we can express Newton's law of universal gravitation in Earth-centered inertial coordinates (x, y) as

$$\ddot{x} = \frac{-gm x}{(x^2 + y^2)^{1.5}}$$

$$\ddot{y} = \frac{-gm y}{(x^2 + y^2)^{1.5}}$$

where x and y are component distances from the center of the Earth to the body or missile. From Fig. 12.2 we can see that the initial conditions for the preceding

differential equations are

$$x(0) = (a + alt_0) \cos \theta_0$$

$$y(0) = (a + alt_0) \sin \theta_0$$

$$\dot{x}(0) = V \cos \left(\frac{\pi}{2} - \gamma + \theta_0 \right)$$

$$\dot{y}(0) = V \sin \left(\frac{\pi}{2} - \gamma + \theta_0 \right)$$

where V is the initial missile velocity, alt_0 the initial missile altitude with respect to the surface of the Earth, γ the angle the velocity vector makes with respect to the reference, and θ_0 the initial angular location of the missile with respect to the x axis. Velocity and position components, with respect to the center of the Earth, can be found from repeated integration of the preceding differential equations. Once we have found the location of the missile with respect to the center of the Earth, it is useful to express the missile location with respect to the surface of the Earth. The instantaneous altitude of the missile can simply be found by finding the distance from the center of the Earth to the missile and then subtracting the Earth's radius, or

$$alt = (x^2 + y^2)^{0.5} - a$$

We can find the distance traveled along the surface of the Earth by referring to Fig. 12.3.

In general, the initial location of the missile can be expressed in vector notation as

$$\mathbf{r}_0 = x_0 \mathbf{i} + y_0 \mathbf{j}$$

and the future location of the missile at any arbitrary time can be expressed as

$$\mathbf{r} = x \mathbf{i} + y \mathbf{j}$$

The angle between the two vectors \mathbf{r}_0 and \mathbf{r} can be found from the definition of the vector dot product, or

$$\theta = \cos^{-1} \frac{\mathbf{r}_0 \cdot \mathbf{r}}{|\mathbf{r}_0||\mathbf{r}|}$$

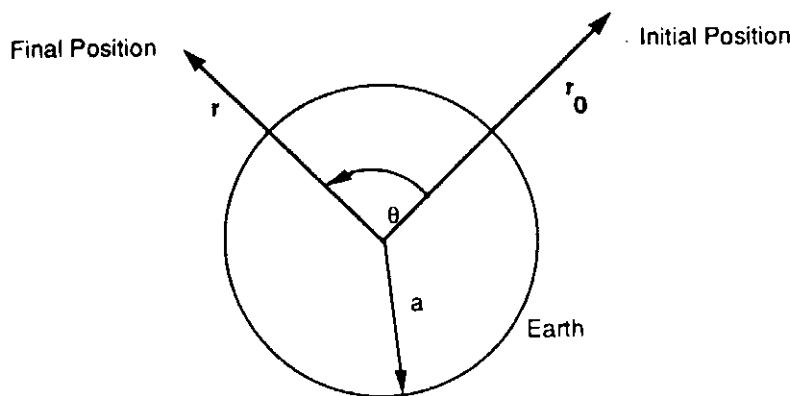


Fig. 12.3 Projecting distance missile travels on surface of Earth.

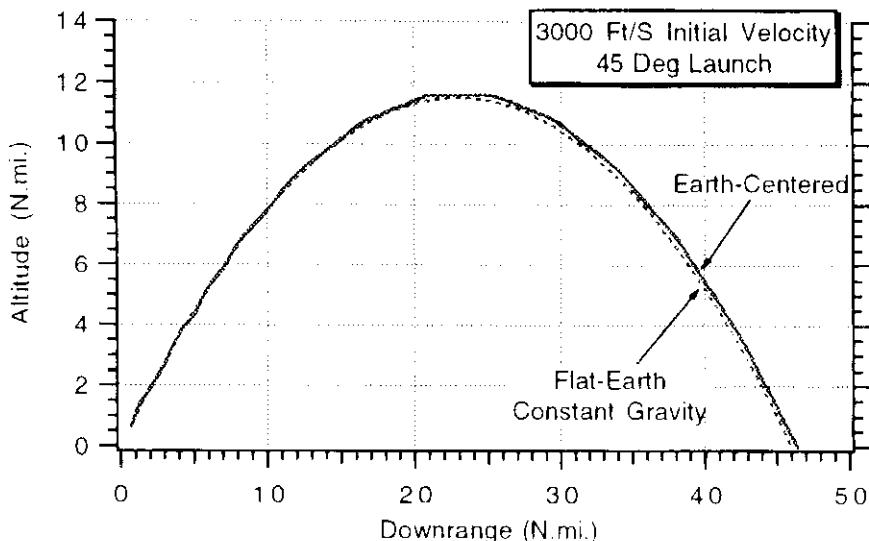


Fig. 12.4 Both models yield same answers when missile speed is small.

Therefore, the distance traveled, which is projected on the surface of a circular Earth, is given by

$$dist = a\theta$$

For comparative purposes, the equations of motion for a missile in a gravity field were programmed using the flat-Earth constant-gravity model and the Earth-centered coordinate system using Newton's law of universal gravitation. The FORTRAN gravity field simulation appears in Listing 12.1. We can see from the listing that the position and velocity components in the flat-Earth model are denoted $RT1$, $RT2$, $VT1$, and $VT2$, respectively. In the Earth-centered system, the position and velocity components are denoted X , Y , $X1$, and $Y1$, respectively. The differential equations describing the missile in a gravity field for both coordinate systems can be found after statement label 200. In the Earth-centered system, the missile position (x , y) is converted to a downrange and altitude so that a trajectory comparison can be made with answers obtained from the flat-Earth model.

A case was run in which an impulsive missile was launched from the surface of the Earth at a 45-deg angle. The initial missile velocity was 3000 ft/s. Figure 12.4 shows that the flat-Earth model (valid for a tactical missile) and the Earth-centered coordinate system model (valid for a strategic missile) yield the same missile trajectories. The total range traveled in both cases is about 47 n.mi., and the maximum altitude is about 12 n.mi.

Figure 12.5 shows that, when the initial speed of the impulsive missile is doubled to 6000 ft/s, we start to see some differences in the resultant missile trajectories. In this case the missile travels about 180 n.mi. and the maximum altitude reached is about 50 n.mi. Remember that the correct answers are the ones given by the Earth-centered coordinate system differential equations. However, even in this case, the flat-Earth approximation (i.e., constant-gravity model) is fairly accurate.

Figure 12.6 shows, that when the impulsive missile speed is again doubled to 12,000 ft/s, the flat-Earth model yields large discrepancies in the resultant missile trajectory. The missile actually travels much farther than the flat-Earth model indicates, since the gravitational acceleration is reduced at the higher altitudes according to Newton's law of universal gravitation. In this case the distance traveled is more than 800 n.mi., and the peak altitude is about 220 n.mi.

Listing 12.1 Gravity field simulation

```

INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
H=.01
A=2.0926E7
GM=1.4077E16
GAM=45.
ALTNM=0.
V=3000.
ALT=ALTNM/6076.
ANG=0.
VRX=V*COS(1.5708-GAM/57.3+ANG)
VRY=V*SIN(1.5708-GAM/57.3+ANG)
G=32.2
S=0.
SCOUNT=0.
RT1=ALT*COS(ANG)
RT2=ALT*SIN(ANG)
VT1=VRX
VT2=VRY
X=(A+ALT)*COS(ANG)
Y=(A+ALT)*SIN(ANG)
XFIRST=X
YFIRST=Y
X1=VRX
Y1=VRY
T=0.
10 IF(ALTNM<0.)GOTO 999
RT1OLD=RT1
RT2OLD=RT2
VT1OLD=VT1
VT2OLD=VT2
XOLD=X
YOLD=Y
X1OLD=X1
Y1OLD=Y1
STEP=1
GOTO 200
66 STEP=2
RT1=RT1+H*RT1D
RT2=RT2+H*RT2D
VT1=VT1+H*VT1D
VT2=VT2+H*VT2D
X=X+H*XD
Y=Y+H*YD
X1=X1+H*X1D
Y1=Y1+H*Y1D
T=T+H
GOTO 200
55 RT1=(RT1OLD+RT1)/2.+.5*H*RT1D

```

(Contd.)

Listing 12.1 (Continued)

```

RT2=(RT2OLD+RT2)/2.+.5*H*RT2D
VT1=(VT1OLD+VT1)/2.+.5*H*VT1D
VT2=(VT2OLD+VT2)/2.+.5*H*VT2D
X=(XOLD+X)/2+.5*H*XD
Y=(YOLD+Y)/2+.5*H*YD
X1=(X1OLD+X1)/2+.5*H*X1D
Y1=(Y1OLD+Y1)/2+.5*H*Y1D
S=S+H
SCOUNT=SCOUNT+H
IF(SCOUNT.LT.1.99999)GOTO 10
SCOUNT=0.
RT1NM=RT1/6076.
RT2NM=RT2/6076.
ALTNM=(SQRT(X**2+Y**2)-A)/6076.
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTNM)
WRITE(*,97)T,RT1NM,RT2NM,DISTNM,ALTNM
WRITE(1,97)T,RT1NM,RT2NM,DISTNM,ALTNM
GOTO 10
97 FORMAT(5F10.3)
200 CONTINUE
AT1=0.
AT2=-G
RT1D=VT1
RT2D=VT2
VT1D=AT1
VT2D=AT2
TEMBOT=(X**2+Y**2)**1.5
X1D=-GM*X/TEMBOT
Y1D=-GM*Y/TEMBOT
XD=X1
YD=Y1
IF(STEP-1)66,66,55
999 CONTINUE
PAUSE
END

SUBROUTINE DISTANCE(XT,YT,XF,YF,DISTNM)
SAVE
R=SQRT(XT**2+YT**2)
RF=SQRT(XF**2+YF**2)
A=2.0926E7
CBETA=(XT*XF+YT*YF)/(R*RF)
IF(CBETA<1.)THEN
    BETA=ACOS(CBETA)
    DISTNM=A*BETA/6076.
ELSE
    DISTNM=(XT-XF)/6076.
ENDIF
RETURN
END

```

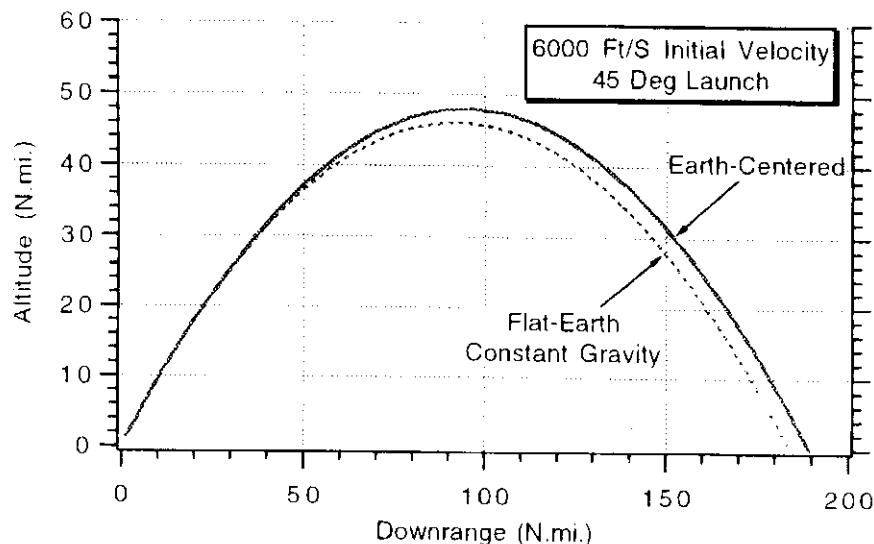


Fig. 12.5 Flat-Earth model is still fairly accurate when missile speed is doubled.

Polar Coordinate System^{4,5}

In the previous section we have shown how to accurately simulate a missile in a gravity field. The differential equations representing Newton's law of universal gravitation were first presented in vector form and then converted for simulation purposes to an Earth-centered Cartesian coordinate system. The Earth-centered coordinate system is extremely useful for simulation work because all integration can be done directly in the inertial frame. However, in order to get insight into the nature of trajectories in a gravity field and to get closed-form solutions, it is more convenient to work analytically in a polar coordinate system whose origin is also at the center of the Earth. Figure 12.7 displays the polar coordinate system from which we proceed with our analysis.

In Fig. 12.7 we have defined a moving coordinate system that has the missile at the origin. The new coordinate system has an i' component along the distance vector and a j' component perpendicular to r . The relationship between the inertial Earth-centered coordinate system and the moving coordinate system is depicted in Fig. 12.8.

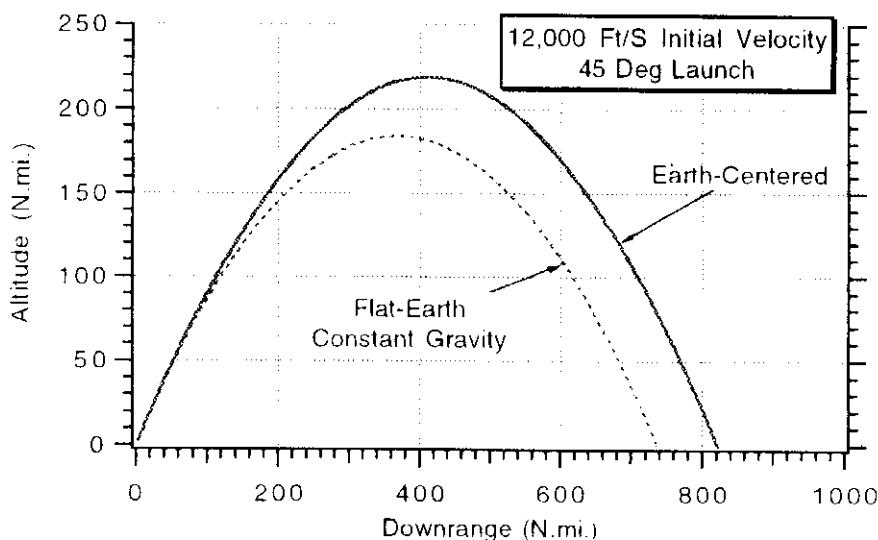


Fig. 12.6 Flat-Earth model is inaccurate when missile speed is again doubled.

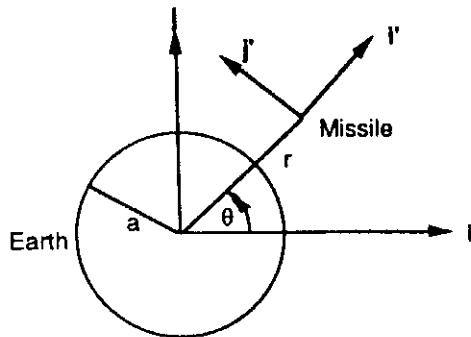


Fig. 12.7 Polar coordinate system with missile in gravity field.

The relationship between the fixed and moving coordinate frames can be expressed mathematically as

$$\mathbf{i}' = \cos \theta \mathbf{i} + \sin \theta \mathbf{j}$$

$$\mathbf{j}' = -\sin \theta \mathbf{i} + \cos \theta \mathbf{j}$$

Since the new coordinate system is moving, we can express its rate of change with respect to the polar angle θ . Differentiating the preceding set of expressions with respect to the polar angle yields

$$\frac{d\mathbf{i}'}{d\theta} = -\sin \theta \mathbf{i} + \cos \theta \mathbf{j} = \dot{\theta} \mathbf{j}'$$

$$\frac{d\mathbf{j}'}{d\theta} = -\cos \theta \mathbf{i} - \sin \theta \mathbf{j} = -\dot{\theta} \mathbf{i}'$$

We can now find the rate of change of the new coordinate system as a function of time according to the chain rule, or

$$\frac{d\mathbf{i}'}{dt} = \frac{d\theta}{dt} \frac{d\mathbf{i}'}{d\theta} = \dot{\theta} \mathbf{j}'$$

$$\frac{d\mathbf{j}'}{dt} = \frac{d\theta}{dt} \frac{d\mathbf{j}'}{d\theta} = -\dot{\theta} \mathbf{i}'$$

We now have sufficient information so that we can take derivatives of vectors. The distance vector \mathbf{r} can be expressed in the moving coordinate system as

$$\mathbf{r} = r \mathbf{i}'$$

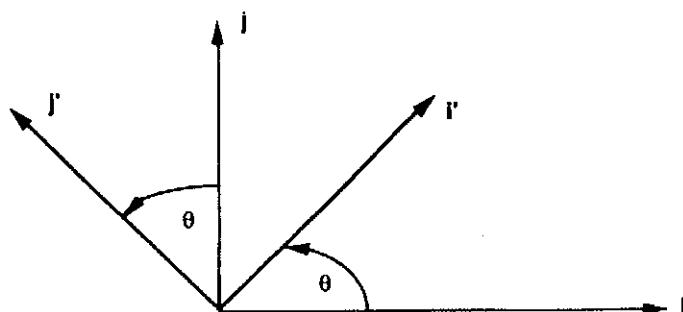


Fig. 12.8 Relationship between fixed and moving coordinate frames.

Taking the derivative of the preceding expression yields

$$\dot{\mathbf{r}} = r\dot{\mathbf{i}}' + \dot{r}\mathbf{i}'$$

However, we have just shown that

$$\dot{\mathbf{i}}' = -\dot{\theta}\mathbf{j}'$$

$$\dot{\mathbf{j}}' = \dot{\theta}\mathbf{i}'$$

Therefore, substitution yields the radial velocity expression

$$\dot{\mathbf{r}} = r\dot{\theta}\mathbf{j}' + \dot{r}\mathbf{i}'$$

Taking the derivative once more yields

$$\ddot{\mathbf{r}} = (\ddot{r} - r\dot{\theta}^2)\mathbf{i}' + (r\ddot{\theta} + 2\dot{r}\dot{\theta})\mathbf{j}'$$

We know that gravitational acceleration is along \mathbf{i}' and there is no acceleration along \mathbf{j}' . Therefore, the preceding vector differential equation can be expressed as the following two scalar differential equations:

$$\frac{-gm}{r^2} = \ddot{r} - r\dot{\theta}^2$$

$$0 = r\ddot{\theta} + 2\dot{r}\dot{\theta}$$

Since

$$\frac{d}{dt}(r^2\dot{\theta}) = 2r\dot{r}\dot{\theta} + r^2\ddot{\theta}$$

we can say that

$$\frac{d}{dt}(r^2\dot{\theta}) = 0$$

Integration yields a constant of integration that must be a moment arm times a tangential velocity, or

$$r^2\dot{\theta} = (a + alt)V \cos \gamma$$

In summary, the differential equations describing a missile in a gravity field can also be expressed in polar coordinates as

$$\ddot{r} - r\dot{\theta}^2 + \frac{gm}{r^2} = 0$$

$$r^2\dot{\theta} = (a + alt)V \cos \gamma$$

where the initial conditions are

$$r(0) = a + alt$$

$$\theta(0) = 0$$

$$\dot{r}(0) = V \sin \gamma$$

A FORTRAN simulation was set up to demonstrate that the polar and Cartesian Earth-centered differential equations are equivalent. Listing 12.2 presents the

Listing 12.2 Gravity field simulation with different coordinate systems

```

INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
H=.01
A=2.0926E7
GM=1.4077E16
GAM=45.
ALTNM=0.
V=24000.
ANGDEG=0.
ANG=ANGDEG/57.3
VRX=V*COS(1.5708-GAM/57.3+ANG)
VRY=V*SIN(1.5708-GAM/57.3+ANG)
ALT=ALTNM/6076.
S=0.
SCOUNT=0.
R0=A+ALT
R1=V*SIN(GAM/57.3)
PSI=0.
X=(A+ALT)*COS(ANG)
Y=(A+ALT)*SIN(ANG)
XFIRST=X
YFIRST=Y
X1=VRX
Y1=VRY
T=0.
10 IF(ALTNM<0.)GOTO 999
R0OLD=R0
R1OLD=R1
PSIOLD=PSI
XOLD=X
YOLD=Y
X1OLD=X1
Y1OLD=Y1
STEP=1
GOTO 200
66 STEP=2
R0=R0+H*R0D
R1=R1+H*R1D
PSI=PSI+H*PSID
X=X+H*XD
Y=Y+H*YD
X1=X1+H*X1D
Y1=Y1+H*Y1D
T=T+H
GOTO 200
55 R0=(R0OLD+R0)/2+.5*H*R0D
R1=(R1OLD+R1)/2+.5*H*R1D
PSI=(PSIOLD+PSI)/2+.5*H*PSID
X=(XOLD+X)/2+.5*H*XD
Y=(YOLD+Y)/2+.5*H*YD

```

(Contd.)

Listing 12.2 (Continued)

```

X1=(X1OLD+X1)/2+.5*H*X1D
Y1=(Y1OLD+Y1)/2+.5*H*Y1D
S=S+H
SCOUNT=SCOUNT+H
IF(SCOUNT.LT.9.99999)GOTO 10
SCOUNT=0.
SPOLARNM=A*PSI/6076.
ALTPOLARNM=(R0-A)/6076.
ALTNM=(SQRT(X**2+Y**2)-A)/6076.
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTNM)
WRITE(*,97)T,SPOLARNM,ALTPOLARNM,DISTNM,ALTNM
WRITE(1,97)T,SPOLARNM,ALTPOLARNM,DISTNM,ALTNM
GOTO 10
97  FORMAT(5F10.3)
200  CONTINUE
      PSID=(A+ALT)*V*COS(GAM/57.3)/(R0*R0)
      R1D=-GM/(R0*R0)+R0*PSID*PSID
      R0D=R1
      TEMBOT=(X**2+Y**2)**1.5
      X1D=-GM*X/TEMBOT
      Y1D=-GM*Y/TEMBOT
      XD=X1
      YD=Y1
      IF(STEP-1)66,66,55
999  CONTINUE
      PAUSE
      CLOSE(1)
      END

      SUBROUTINE DISTANCE(XT,YT,XF,YF,DISTNM)
      SAVE
      R=SQRT(XT**2+YT**2)
      RF=SQRT(XF**2+YF**2)
      A=2.0926E7
      CBETA=(XT*XF+YT*YF)/(R*RF)
      IF(CBETA<1.)THEN
          BETA=ACOS(CBETA)
          DISTNM=A*BETA/6076.
      ELSE
          DISTNM=(XT-XF)/6076.
      ENDIF
      RETURN
      END

```

gravity field simulation for both coordinate systems. The position and velocity components in the Cartesian system appear in the listing as X , Y , $X1$, and $Y1$, respectively. The range, its derivative, and the polar angle in the polar coordinate system are denoted by $R0$, $R1$, and PSI in the listing. The differential equations for both the Cartesian and polar coordinate systems appear after label 200 in the listing.

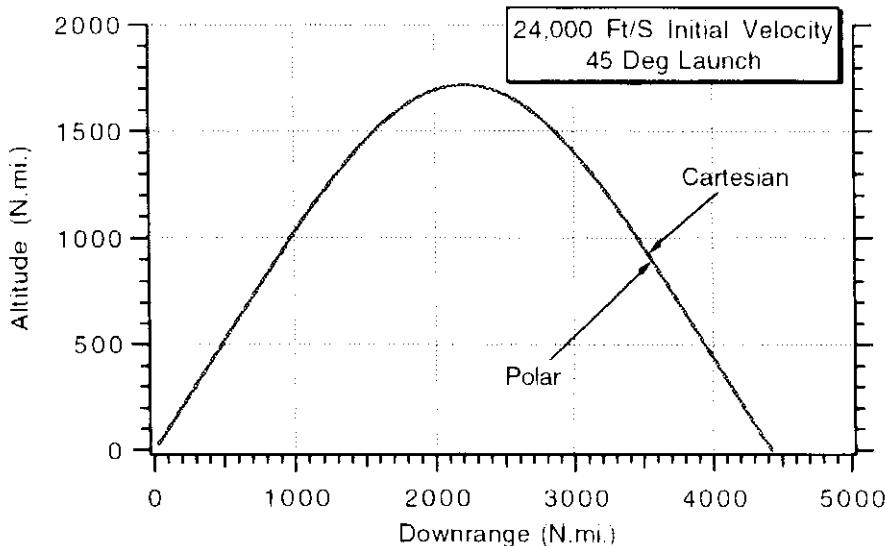


Fig. 12.9 Polar and Earth-centered gravity field equations yield identical trajectories.

An experiment was run in which a missile with an initial velocity 24,000 ft/s was launched from the surface of the Earth at an angle of 45 deg with respect to the reference. The resultant trajectories for both sets of differential equations appear in Fig. 12.9. We can see that the resultant trajectories are identical for all practical purposes. It is interesting to note that with an initial speed of 24,000 ft/s the impulsive missile traveled nearly 4500 n.mi. and reached an altitude of 1700 n.mi.

Closed-Form Solutions^{4,5}

For those readers familiar with the literature on astrodynamics, apologies are offered in advance for the text's unconventional notation. Many other authors use r_e or R_e rather than a for the radius of the Earth and use a rather than a_1 for the semimajor axis of an ellipse. The choice of a for the radius of the Earth was solely historical (i.e., it was used in Ref. 4), whereas the use of a_1 for the semimajor axis of an ellipse was to avoid further confusion. In this section we will solve the previously derived differential equations of Newton's law of universal gravitation expressed in polar coordinates. In other words, we seek to find closed-form solutions of the polar differential equations

$$\ddot{r} - r\dot{\theta}^2 + \frac{gm}{r^2} = 0$$

$$r^2\dot{\theta} = (a + alt)V \cos \gamma$$

For convenience let us define constants r_0 and p such that

$$r_0 = a + alt$$

$$p = (a + alt)V \cos \gamma = r_0 V \cos \gamma$$

In addition, we will define an inverse range to be

$$u = 1/r$$

The goal is to convert both polar differential equations to one second-order differential equation in terms of u . First we know from the chain rule that u varies with time according to

$$\frac{du}{dt} = \frac{d\theta}{dt} \frac{du}{d\theta} = \dot{\theta} \frac{du}{d\theta} = \frac{p}{r^2} \frac{du}{d\theta}$$

An alternate way of seeing how u changes with respect to time is

$$\frac{du}{dt} = \frac{du}{dr} \frac{dr}{dt} = \frac{-1}{r^2} \frac{dr}{dt}$$

Equating both expressions yields

$$\frac{dr}{dt} = -p \frac{du}{d\theta}$$

Next we define z to be

$$z = \frac{dr}{dt}$$

Using the chain rule to see how z changes with respect to time yields

$$\frac{dz}{dt} = \frac{d\theta}{dt} \frac{dz}{d\theta} = \frac{p}{r^2} \frac{d}{d\theta} \left[\frac{dr}{dt} \right] = \frac{p}{r^2} \frac{d}{d\theta} \left[-p \frac{du}{d\theta} \right]$$

Therefore, we can say that

$$\frac{dz}{dt} = \frac{d^2r}{dt^2} = \frac{-p^2}{r^2} \frac{d^2u}{d\theta^2}$$

Substitution allows us to rewrite the second-order differential equation in range as

$$\ddot{r} - r\dot{\theta}^2 + \frac{gm}{r^2} = 0 = \frac{-p^2}{r^2} \frac{d^2u}{d\theta^2} - r \frac{p^2}{r^4} + gm u^2$$

Simplification yields

$$\frac{d^2u}{d\theta^2} + u = \frac{gm}{p^2}$$

For purposes that will be obvious later we can define a new constant to be

$$\lambda = \frac{r_0 V^2}{gm}$$

We can now summarize the transformed range polar differential equation to be

$$\frac{d^2u}{d\theta^2} + u = \frac{1}{\lambda r_0 \cos^2 \gamma}$$

The original initial conditions on the polar differential equations were

$$r(0) = r_0$$

$$\dot{r}(0) = V \sin \gamma$$

Since we already know that

$$\frac{du}{d\theta} = -\frac{1}{p} \frac{dr}{dt} = \frac{-1}{r_0 V \cos \gamma} \frac{dr}{dt}$$

we can say that

$$\frac{du}{d\theta}(0) = \frac{-\tan \gamma}{r_0}$$

The other initial condition is simply

$$u(0) = \frac{1}{r(0)} = \frac{1}{r_0}$$

The solution to the preceding second-order differential equation is

$$u = A \sin \theta + B \cos \theta + \frac{1}{\lambda r_0 \cos^2 \gamma}$$

where A and B can be found from the initial conditions. After some algebra we obtain the complete solution in terms of u as

$$u = \frac{1 - \cos \theta}{\lambda r_0 \cos^2 \gamma} + \frac{1}{r_0} \frac{\cos(\theta + \gamma)}{\cos \gamma}$$

However, since

$$u = 1/r$$

the solution in terms of r becomes

$$\frac{r_0}{r} = \frac{1 - \cos \theta}{\lambda \cos^2 \gamma} + \frac{\cos(\theta + \gamma)}{\cos \gamma}$$

or, more conveniently,

$$\begin{aligned} r &= \frac{r_0 \lambda \cos^2 \gamma}{1 - \cos \theta + \lambda \cos \gamma \cos(\theta + \gamma)} \\ &= \frac{r_0 \lambda \cos^2 \gamma}{1 - \lambda \sin \theta \cos \gamma \sin \gamma - \cos \theta (1 - \lambda \cos^2 \gamma)} \end{aligned}$$

Thus, given missile altitude (r_0), velocity (λ), and flight-path angle γ , we find the missile location r as a function of the central angle θ . The preceding closed-form solution is also the equation of an ellipse in a polar coordinate system. To prove this interesting fact we must first recognize that the equation for an ellipse in polar coordinates is

$$r = \frac{a_1(1 - e^2)}{1 - e \cos(\theta - \omega)} = \frac{a_1(1 - e^2)}{1 - e \sin \theta \sin \omega - e \cos \theta \cos \omega}$$

where a_1 is the semimajor axis, e the eccentricity, and ω the argument of the

apogee. The trajectory equation and the equation for an ellipse are equivalent if

$$e \sin \omega = \lambda \cos \gamma \sin \gamma$$

$$e \cos \omega = 1 - \lambda \cos^2 \gamma$$

Squaring and adding the preceding equations yields an expression for the eccentricity in terms of λ , or

$$e = [1 + \lambda(\lambda - 2) \cos^2 \gamma]^{0.5}$$

The trajectory equation yields a circle if $e = 0$, an ellipse if $0 < e < 1$, a parabola if $e = 1$, and a hyperbola for $e > 1$. If we set the flight-path angle γ to zero, we can see that we get circular motion if $\lambda = 1$, elliptical motion for $0 < \lambda < 2$, parabolic motion for $\lambda = 2$, and hyperbolic motion for $\lambda > 2$. Since we can express the initial velocity in terms of λ as

$$V = \sqrt{\frac{\lambda gm}{r_0}}$$

we can determine the trajectory shape from the magnitude of the velocity!

The Earth-centered trajectory generator was modified so that the initial flight-path angle was zero and the initial velocity expressed in terms of λ according to the preceding velocity equation. In addition, the outputs, rather than being downrange and altitude, were expressed in the natural x, y units (i.e., distance from the center of the Earth converted to nautical miles). Listing 12.3 presents the resultant FORTRAN orbit generator program. We can see from the listing that the missile is initially at 1000 n.mi. altitude.

A case was run in which λ was set to 1. Figure 12.10 shows that the simulation indicates that the missile trajectory is indeed circular—as theory predicted!

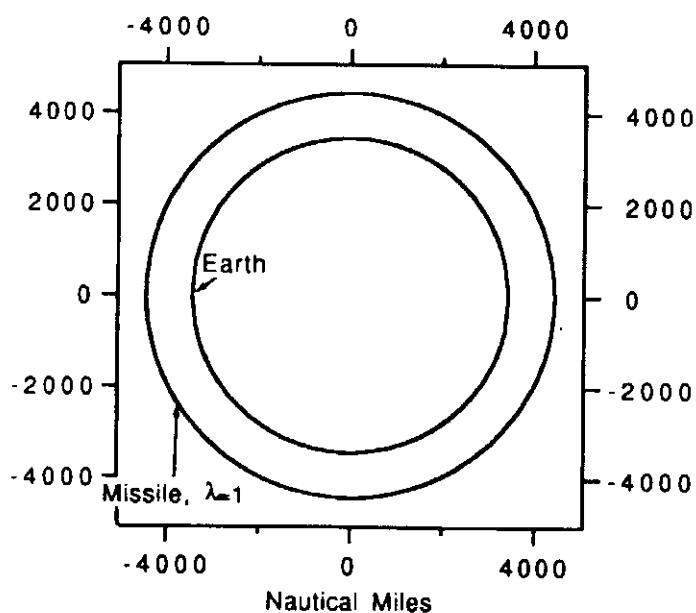


Fig. 12.10 Simulation yields circular orbit when λ is unity.

Listing 12.3 FORTRAN orbit generator

```

INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
H=.01
A=2.0926E7
GM=1.4077E16
GAM=0.
ALTNM=1000.
ALT=ALTNM*6076.
XLAM=1.
V=SQRT(GM*XLAM/(A+ALT))
ANGDEG=90.
ANG=ANGDEG/57.3
VRX=V*COS(1.5708-GAM/57.3+ANG)
VRY=V*SIN(1.5708-GAM/57.3+ANG)
S=0.
SCOUNT=0.
X=(A+ALT)*COS(ANG)
Y=(A+ALT)*SIN(ANG)
XFIRST=X
YFIRST=Y
X1=VRX
Y1=VRY
T=0.
TF=30000.
10 IF(T>TF)GOTO 999
XOLD=X
YOLD=Y
X1OLD=X1
Y1OLD=Y1
STEP=1
GOTO 200
66 STEP=2
X=X+H*XD
Y=Y+H*YD
X1=X1+H*X1D
Y1=Y1+H*Y1D
T=T+H
GOTO 200
55 X=(XOLD+X)/2+.5*H*XD
Y=(YOLD+Y)/2+.5*H*YD
X1=(X1OLD+X1)/2+.5*H*X1D
Y1=(Y1OLD+Y1)/2+.5*H*Y1D
S=S+H
SCOUNT=SCOUNT+H
IF(SCOUNT.LT.49.99999)GOTO 10
SCOUNT=0.
XNM=X/6076.
YNM=Y/6076.

```

(Contd.)

Listing 12.3 (Continued)

```

      WRITE(*,97)T,XNM,YNM
      WRITE(1,97)T,XNM,YNM
      GOTO 10
97   FORMAT(3F12.3)
200   CONTINUE
      TEMBOT=(X**2+Y**2)**1.5
      X1D=-GM*X/TEMBOT
      Y1D=-GM*Y/TEMBOT
      XD=X1
      YD=Y1
      IF(STEP-1)66,66,55
999   CONTINUE
      PAUSE
      CLOSE(1)
      END

```

Figure 12.11 shows that when λ was set to 1.5 the simulation got an elliptical orbit for the missile—as theory predicted! Values of λ between 0 and 2 should yield elliptical orbital motion, with 1 being circular.

Theory says that if we set $\lambda = 0.5$ we should also get an elliptical orbit. However, Fig. 12.12 shows that, although the simulation indicates an elliptical orbit, it is one that intersects the earth! Therefore, values of λ between 0 and 1 yield suborbital motion. Although this type of trajectory is not appropriate for a satellite, it is appropriate for a ballistic missile! Finally Fig. 12.13 shows that when $\lambda = 2$ we have achieved escape velocity and the missile motion is parabolic. This type of orbit does not intersect the Earth.

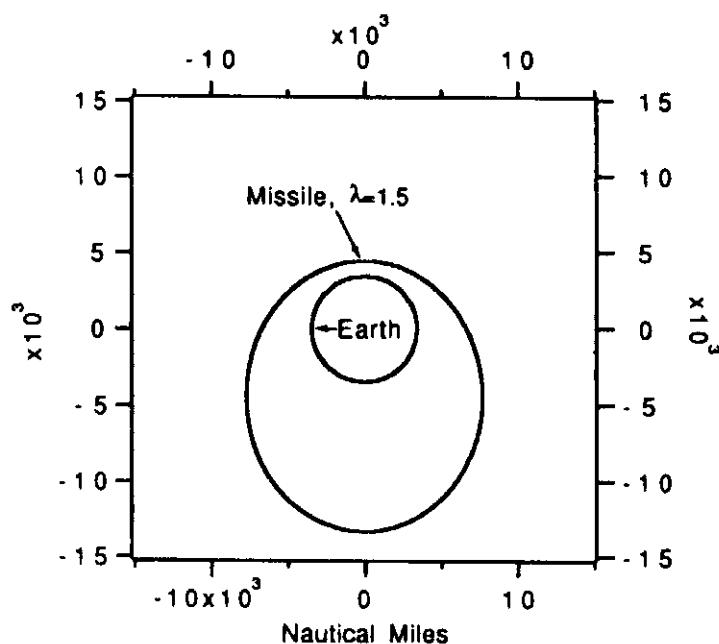


Fig. 12.11 Setting $\lambda = 1.5$ yields elliptical orbit.

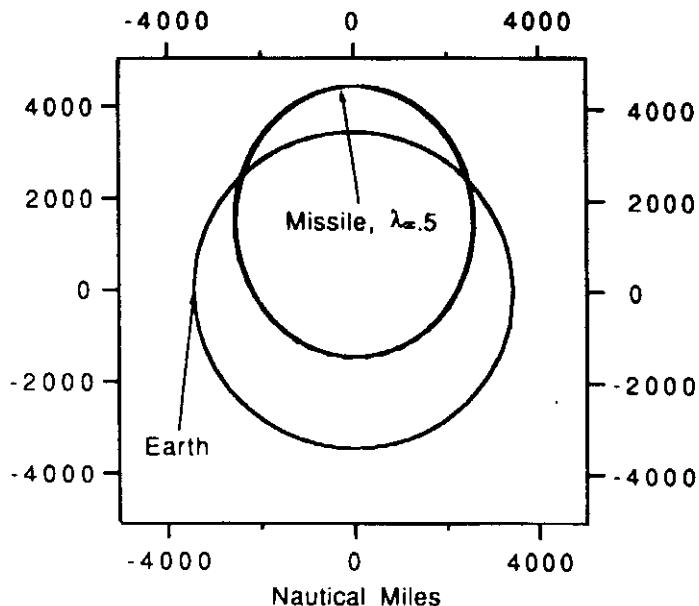


Fig. 12.12 Setting λ too small results in orbit that intersects Earth.

Hit Equation^{4,5}

We have seen that the previously derived trajectory equation is useful in obtaining closed-form solutions for satellite orbits and ballistic missile trajectories. If we specialize in the ballistic missile case, we can also get closed-form solutions from the trajectory equation, which, given an initial missile flight-path angle, altitude, and distance to be traveled (missile hits the Earth at that distance), will define the magnitude of the missile velocity required.

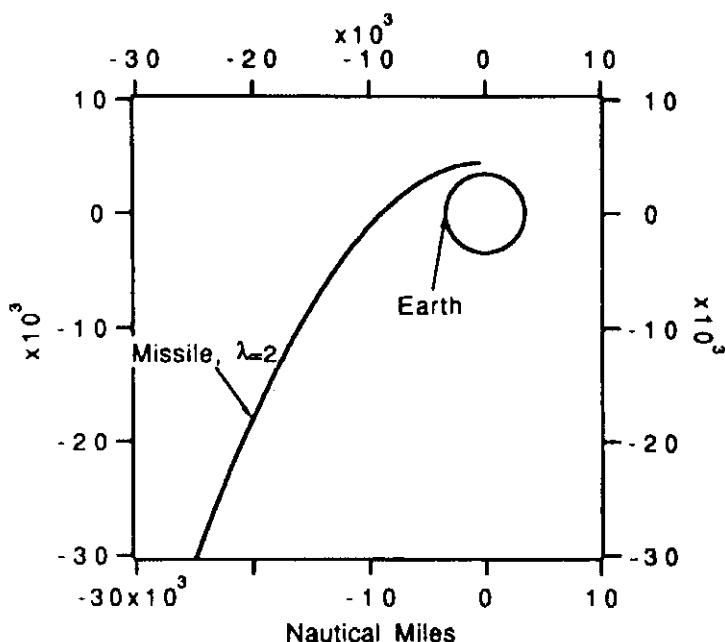


Fig. 12.13 Setting $\lambda = 2$ results in parabolic trajectory for missile.

Listing 12.4 Simulation to demonstrate validity of velocity formula

```

INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
H=.01
A=2.0926E7
GM=1.4077E16
GAMDEG=23.
GAM=GAMDEG/57.3
DISTNM=6000.
ANGDEG=0.
ANG=ANGDEG/57.3
PHI=DISTNM*6076./A
ALTNM=0.
ALT=ALTNM*6076.
R0=A+ALT
TOP=GM*(1.-COS(PHI))
TEMP=R0*COS(GAM)/A-COS(PHI+GAM)
BOT=R0*COS(GAM)*TEMP
V=SQRT(TOP/BOT)
VRX=V*COS(1.5708-GAM+ANG)
VRY=V*SIN(1.5708-GAM+ANG)
S=0.
SCOUNT=0.
X=(A+ALT)*COS(ANG)
Y=(A+ALT)*SIN(ANG)
XFIRST=X
YFIRST=Y
X1=VRX
Y1=VRY
T=0.
10 IF(ALT<0.)GOTO 999
XOLD=X
YOLD=Y
X1OLD=X1
Y1OLD=Y1
STEP=1
GOTO 200
66 STEP=2
X=X+H*XD
Y=Y+H*YD
X1=X1+H*X1D
Y1=Y1+H*Y1D
T=T+H
GOTO 200
55 X=(XOLD+X)/2+.5*H*XD
Y=(YOLD+Y)/2+.5*H*YD
X1=(X1OLD+X1)/2+.5*H*X1D
Y1=(Y1OLD+Y1)/2+.5*H*Y1D
ALT=SQRT(X**2+Y**2)-A
S=S+H
SCOUNT=SCOUNT+H
IF(SCOUNT.LT.9.99999)GOTO 10

```

(Contd.)

Listing 12.4 (Continued)

```

SCOUNT=0.
XNM=X/6076.
YNM=Y/6076.
ALTNM=ALT/6076.
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTNM)
WRITE(*,97)T,XNM,YNM,DISTNM,ALTNM
WRITE(1,97)T,XNM,YNM,DISTNM,ALTNM
GOTO 10
97 FORMAT(5F10.3)
200 CONTINUE
TEMBOT=(X**2+Y**2)**1.5
X1D=-GM*X/TEMBOT
Y1D=-GM*Y/TEMBOT
XD=X1
YD=Y1
IF(STEP-1)66,66,55
999 CONTINUE
XNM=X/6076.
YNM=Y/6076.
ALTNM=ALT/6076.
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTNM)
WRITE(*,97)T,XNM,YNM,DISTNM,ALTNM
WRITE(1,97)T,XNM,YNM,DISTNM,ALTNM
PAUSE
CLOSE(1)
END

SUBROUTINE DISTANCE(XT,YT,XF,YF,DISTNM)
SAVE
R=SQRT(XT**2+YT**2)
RF=SQRT(XF**2+YF**2)
A=2.0926E7
CBETA=(XT*XF+YT*YF)/(R*RF)
IF(CBETA<1.)THEN
    BETA=ACOS(CBETA)
    DISTNM=A*BETA/6076.
ELSE
    DISTNM=(XT-XF)/6076.
ENDIF
RETURN
END

```

If we desire for the missile to travel a distance $dist$ then the total central angle traveled ϕ is given by (see Fig. 12.7)

$$\phi = dist/a$$

where a is the radius of the Earth. The missile hits the Earth when $r = a$. Therefore, substituting $r = a$ and $\theta = \phi$ into the trajectory equation solution yields

$$\frac{r_0}{a} = \frac{1 - \cos \phi}{\lambda \cos^2 \gamma} + \frac{\cos(\phi + \gamma)}{\cos \gamma}$$

Recognizing that

$$\lambda = \frac{r_0 V^2}{gm}$$

we can solve for the velocity. After some algebra we obtain

$$V = \sqrt{\frac{gm(1 - \cos \phi)}{r_0 \cos \gamma [(r_0 \cos \gamma / a) - \cos(\phi + \gamma)]}}$$

This equation tells us the velocity required to hit a target a certain distance away from our launch point, given we want to launch with a certain flight-path angle γ . Figure 12.14 displays the velocity formula in graphic form. We can see that, as expected, longer distances require larger missile velocities. If the initial flight-path angle is too large, the ballistic missile will never hit the Earth because the resultant velocity will exceed the escape velocity ($\lambda = 2$) and the trajectory will not be elliptical.

Listing 12.4 presents a modified ballistic missile simulation using the preceding velocity formula to derive the desired initial velocity given the desired flight-path angle, initial missile altitude, and distance to be covered. From the listing we can see that the missile is launched from the surface of the Earth with an initial flight-path angle of 23 deg. The target is 6000 n.mi. downrange on the surface of the Earth. Although the velocity formula was derived from solutions in the polar coordinate system, the simulation is based in the Cartesian coordinate system.

Figure 12.15 presents simulation results, in the form of an altitude vs down-range plot for the nominal case of Listing 12.4. We can see that the missile indeed travels the desired distance of 6000 n.mi. before hitting the surface of the Earth. The peak altitude for the missile is in excess of 800 n.mi. Figure 12.16 presents the same trajectory information in a way in which the curvature of the Earth is apparent.

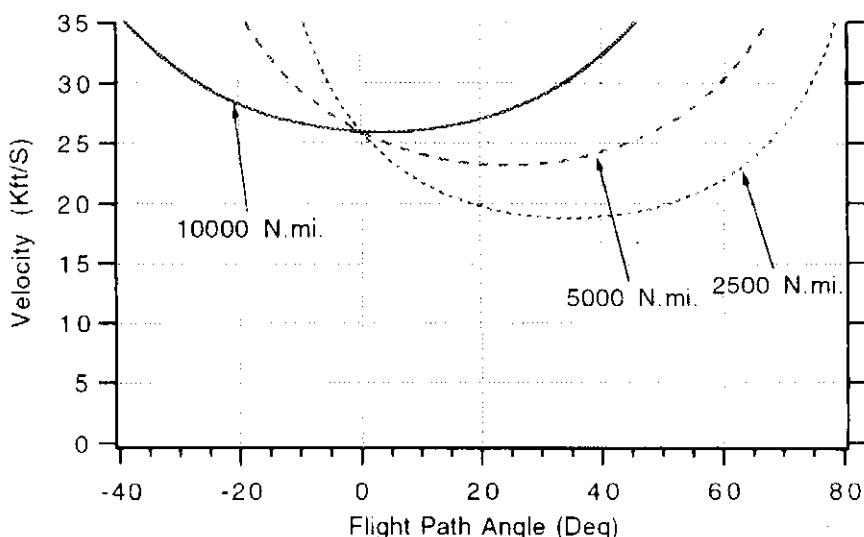


Fig. 12.14 Required velocity depends on range to be traveled and desired flight-path angle.

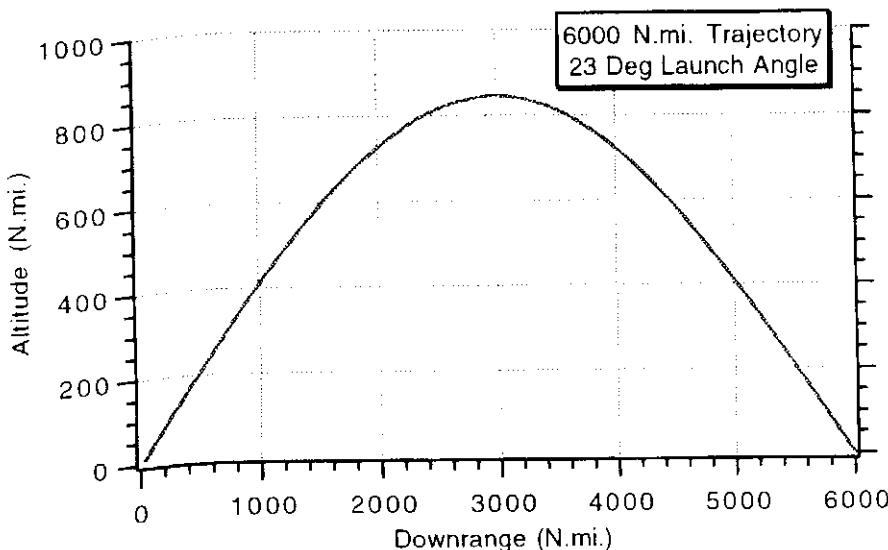


Fig. 12.15 Closed-form solution for velocity is accurate.

Flight Time^{4,5}

We have already seen that, given a distance to be covered and initial flight-path angle, it was possible to derive a formula for the required velocity. Also associated with this velocity is the time to reach the target or time of flight t_F . It is also possible, based on the trajectory equation solution for r , to derive a closed-form solution for the time of flight. From the original gravity field differential equation in polar coordinates, we know that

$$r^2 \frac{d\theta}{dt} = r_0 V \cos \gamma$$

We can cross multiply terms to set up the integrals

$$\int_0^\phi r^2 d\theta = \int_0^{t_F} r_0 V \cos \gamma dt$$

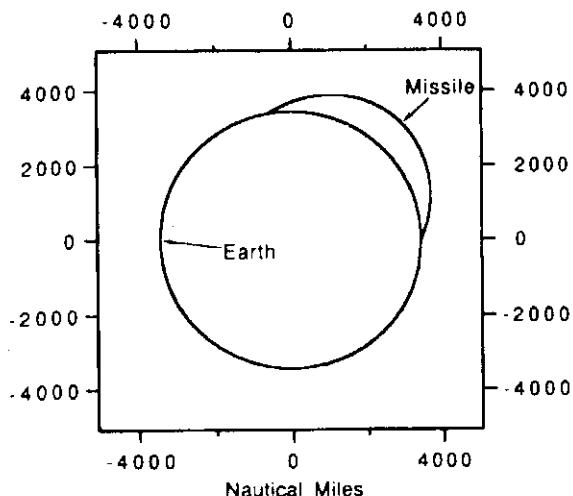


Fig. 12.16 Six thousand nautical miles, 23-deg trajectory.

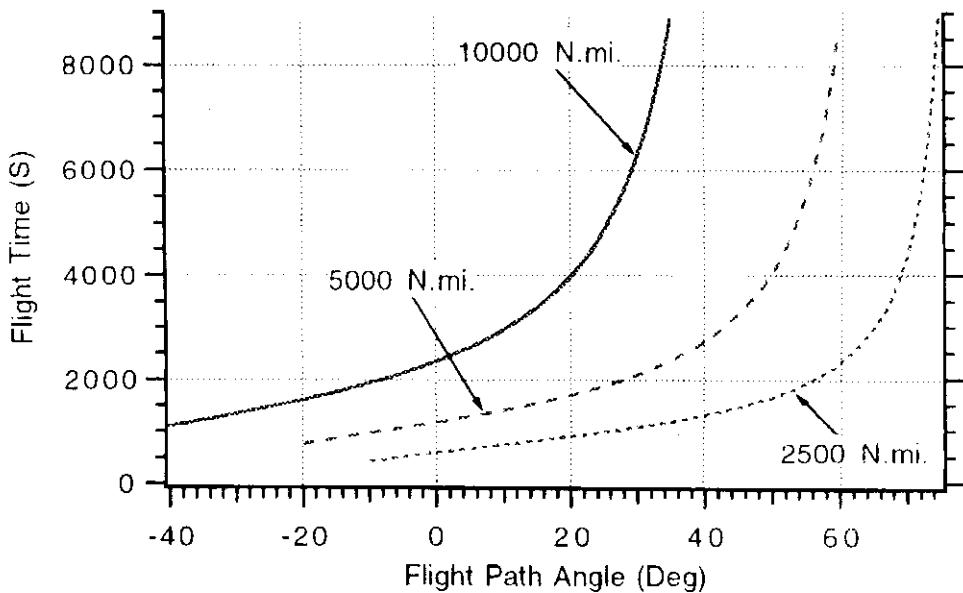


Fig. 12.17 Flight time increases with increasing flight-path angle and increasing distance to be traveled.

Integration of the right-hand side of the equation and substitution of the trajectory solution into the left-hand side yields the integral

$$t_F = \frac{1}{r_0 V \cos \gamma} \int_0^\phi \frac{r_0^2 \lambda^2 \cos^4 \gamma}{[1 - \cos \theta + \lambda \cos \gamma \cos(\theta + \gamma)]^2} d\theta$$

After integration and much algebra, the closed-form solution assuming $\lambda < 2$ for the flight time simplifies to

$$t_F = \frac{r_0}{V \cos \gamma} \left\{ \frac{\tan \gamma (1 - \cos \phi) + (1 - \lambda) \sin \phi}{(2 - \lambda) \left[\frac{1 - \cos \phi}{\lambda \cos^2 \gamma} + \frac{\cos(\gamma + \phi)}{\cos \gamma} \right]} \right. \\ \left. + \frac{2 \cos \gamma}{\lambda \left(\frac{2}{\lambda} - 1 \right)^{1.5}} \tan^{-1} \left(\frac{\sqrt{\frac{2}{\lambda} - 1}}{\cos \gamma \cot \frac{\phi}{2} - \sin \gamma} \right) \right\}$$

Figure 12.17 displays the flight time formula in graphic form. We can see that, as expected, it takes longer for a ballistic missile to travel greater distances. In addition, increasing the flight-path angle tends to increase the time of flight. For example, it takes about 1800 s for a ballistic missile to travel 5000 n.mi. when the flight-path angle is 20 deg. Increasing the flight-path angle to 40 deg increases the flight time to nearly 2800 s. We can also see that flight time increases monotonically and smoothly with increasing values of flight-path angle. We shall make use of this interesting observation later.

Summary

This chapter was our first introduction into the strategic world. We saw that the constant-gravity, flat-Earth model used for tactical interceptors was not correct for strategic interceptors. Simulation models based on Newton's law of universal

gravitation were derived from first principles. It was shown that an Earth-centered Cartesian system was useful for simulation and a polar model was more useful for analytical work. A closed-form solution was obtained for a ballistic missile's velocity in terms of flight-path angle and distance to be covered, and another expression was derived relating the flight time to the velocity. Simulation results confirmed the closed-form solutions. We shall make much use of these relationships later.

References

- ¹Kennedy, G. P., *Rockets, Missiles and Spacecraft of the National Air and Space Museum*, Smithsonian Institution Press, Washington DC, 1983.
- ²Bate, R. R., Mueller, D. D., and White, J. E., *Fundamentals of Astrodynamics*, Dover, New York, 1971.
- ³Battin, R. H., *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA Education Series, New York, 1987.
- ⁴Wheelon, A. D., "Free Flight of a Ballistic Missile," *ARS Journal*, Vol. 29, Dec. 1959, pp. 915-926.
- ⁵Regan, F., *Re-Entry Vehicle Dynamics*, AIAA Education Series, New York, 1984.



Boosters

Introduction

WE have seen in Chapter 12 that, in order for a ballistic interceptor to travel long distances or go into orbit, it must attain speeds in excess of 20 kft/s. From the rocket equation we saw in Chapter 11 that, with fuel mass fractions of less than 0.5 (i.e., tactical missiles), it was impossible to reach these speeds. In this chapter we shall investigate preliminary booster designs so that speeds required for strategic travel can be achieved.

Review

In Chapter 11 we saw that the change in velocity is related to specific impulse I_{sp} and fuel mass fraction mf according to the rocket equation, or

$$\Delta V = I_{sp} g \ln \frac{1}{1 - mf}$$

Figure 13.1 displays the rocket equation in graphic form. From this figure we can see that fuel mass fractions approaching 0.9 are required if we wish to attain speeds in excess of 20 kft/s for fuel specific impulses of less than 300 s. The figure clearly shows that fuel mass fractions of less than 0.5 (i.e., tactical missiles) lead to velocities that are not adequate for a strategic application.

We can think of a strategic interceptor as consisting of two sections: booster and payload. A single-stage booster (we will consider staging later in this chapter) consists of fuel and structure denoted by weights W_P and W_S , respectively, as shown in Fig. 13.2. Initially we will consider that the sole purpose of the single-stage booster is to get the payload up to speed. The payload, denoted by weight W_{pay} , consists of structure, electronics, a divert engine, and fuel. The purpose of the payload for strategic guided interceptors is to acquire the target and maneuver, using divert fuel, to hit the target.

If it is desired that the interceptor change its velocity by amount ΔV , then the weight of the structure, fuel, and payload must also follow the rocket equation as

$$W_S + W_P + W_{pay} = (W_S + W_{pay}) \exp \left(\frac{\Delta V}{g I_{sp}} \right)$$

where I_{sp} denotes the specific impulse of the booster fuel and is measured in seconds. The fuel mass fraction has been defined as the ratio of the propellant weight to the total weight. To simplify computations in this chapter, an approximate

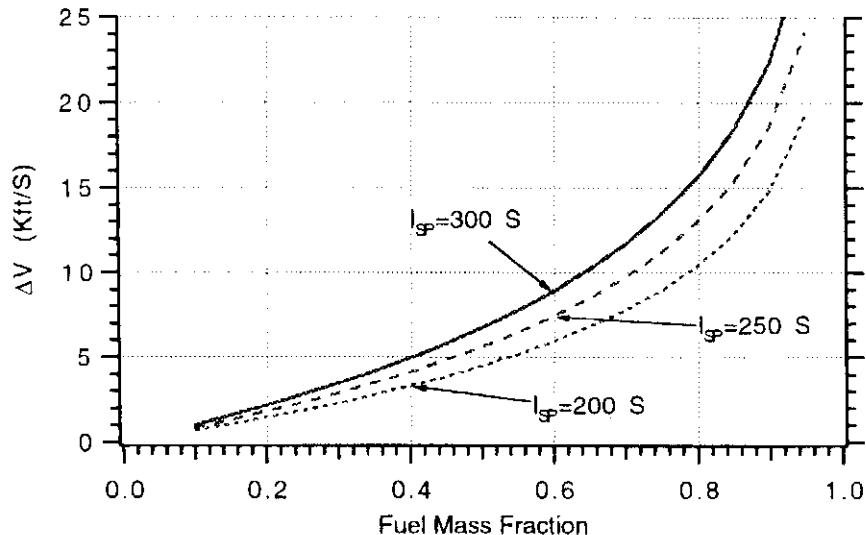


Fig. 13.1 Large fuel mass fractions are required for strategic applications.

fuel mass fraction mf^* is defined as the ratio of the propellant weight to the sum of the propellant weight plus structure or

$$mf^* = \frac{W_P}{W_P + W_S}$$

For small payloads the approximate and actual fuel mass fractions are equivalent. We can express the weight of the booster structure to the propellant weight and fuel approximate mass fraction according to

$$W_S = \frac{W_P(1 - mf^*)}{mf^*}$$

Substitution of the preceding relationship into the rocket equation yields, after some algebra, a formula for the propellant weight in terms of the payload weight, velocity desired, approximate fuel mass fraction, and specific impulse. This relationship can be expressed as

$$W_P = W_{\text{pay}} \left[\exp \left(\frac{\Delta V}{g I_{\text{sp}}} \right) - 1 \right] \left/ \left[\frac{1}{mf^*} - \frac{1 - mf^*}{mf^*} \exp \left(\frac{\Delta V}{g I_{\text{sp}}} \right) \right] \right.$$

The total interceptor weight W_{tot} consists of the booster fuel and structure plus the payload, or

$$W_{\text{tot}} = W_S + W_P + W_{\text{pay}}$$

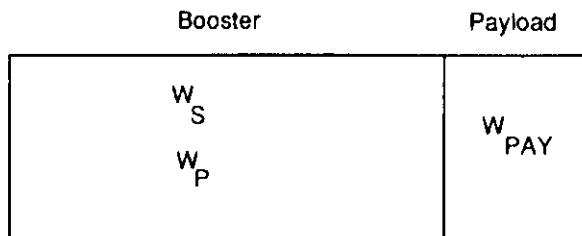


Fig. 13.2 Single-stage strategic interceptor model.

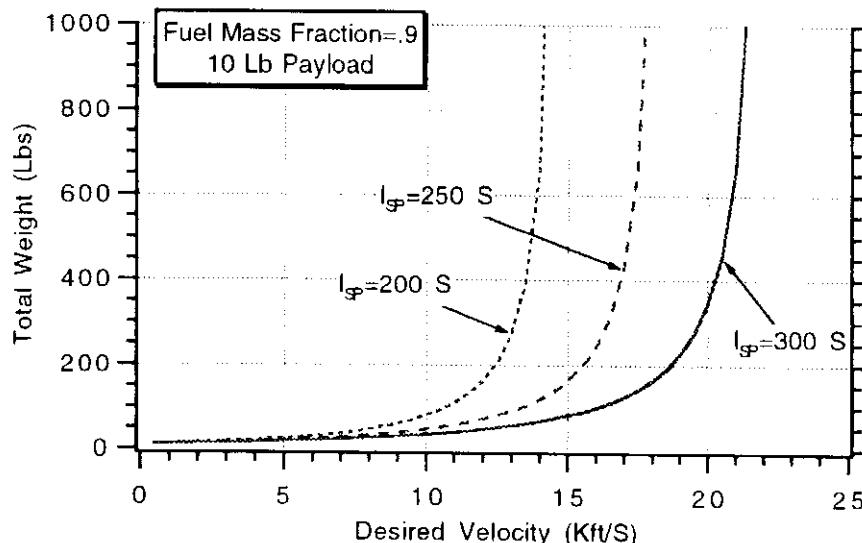


Fig. 13.3 A great deal of weight is required to bring small payloads to strategic speeds.

Based upon the preceding relationships, Fig. 13.3 displays the total weight vs the desired change in velocity for an approximate fuel mass fraction of 0.9 and payload weight of 10 lb. We can see that, for a booster to reach a desired velocity of 20 kft/s from rest (in the absence of atmospheric drag), with a specific impulse of 300 s, more than 150 lb of total weight is required—just for a 10-lb payload! Doubling the payload weight will double the total weight. Decreasing the specific impulse or decreasing the fuel mass fraction both work in the direction of increasing the total weight.

Staging

We have seen in the previous section that it can take a great deal of total weight to propel small payloads to near-orbital speeds. One way of reducing the total weight for a given approximate fuel mass fraction and specific impulse is to use staging. Figure 13.4 presents a two-stage booster. In this figure, the propellant and structural weights are indicated in each of the stages.

Therefore, the second-stage propellant weight can be expressed as

$$W_{P2} = W_{pay} \left[\exp \left(\frac{\Delta V_2}{g I_{sp2}} \right) - 1 \right] / \left[\frac{1}{mf2^*} - \frac{1 - mf2^*}{mf2^*} \exp \left(\frac{\Delta V_2}{g I_{sp2}} \right) \right]$$

where ΔV_2 is the desired velocity change attributed to the second stage, $mf2^*$ is the second-stage fuel mass fraction, and I_{sp2} is the second-stage specific impulse.

Booster 1	Booster 2	Payload
w_{S1} w_{P1}	w_{S2} w_{P2}	w_{PAY}

Fig. 13.4 Two-stage booster.

The structural weight of the second stage can then be expressed as

$$W_{S2} = \frac{W_{P2}(1 - mf2^*)}{mf2^*}$$

The weight of the second stage plus payload $W_{\text{tot}2}$ is simply

$$W_{\text{tot}2} = W_{P2} + W_{S2} + W_{\text{pay}}$$

We can now find the propellant weight of the first stage by treating $W_{\text{tot}2}$ as an effective payload. The resultant weight is

$$W_{P1} = W_{\text{tot}2} \left[\exp \left(\frac{\Delta V_1}{g I_{sp1}} \right) - 1 \right] / \left[\frac{1}{mf1^*} - \frac{1 - mf1^*}{mf1^*} \exp \left(\frac{\Delta V_1}{g I_{sp1}} \right) \right]$$

where ΔV_1 is the desired velocity change attributed to the first stage, $mf1^*$ is the first-stage approximate fuel mass fraction, and I_{sp1} is the first-stage specific impulse. The structural weight of the first stage can then be expressed as

$$W_{S1} = \frac{W_{P1}(1 - mf1^*)}{mf1^*}$$

Finally, the total interceptor weight (first stage plus rest) is given by

$$W_{\text{tot}} = W_{P1} + W_{S1} + W_{\text{tot}2}$$

Using the preceding relationships for a two-stage interceptor, the total weight was calculated as a function of desired velocity change for various values of specific impulse. It was assumed that each stage of the interceptor had equal specific impulses and equal approximate fuel mass fractions. In addition, it was also assumed that half of the desired velocity was obtained with the first stage and the second half of the desired velocity was obtained with the second stage. Figure 13.5 shows how the total weight varies. For a desired velocity of 20 kft/s, Fig. 13.5 shows that, for a 300-s specific impulse, approximately 100 lb of total weight are required for a 10-lb payload using a two-stage interceptor. Figure 13.3 shows that for the same case a one-stage interceptor requires more than 150 lb of total weight. Thus, staging appears to be beneficial.

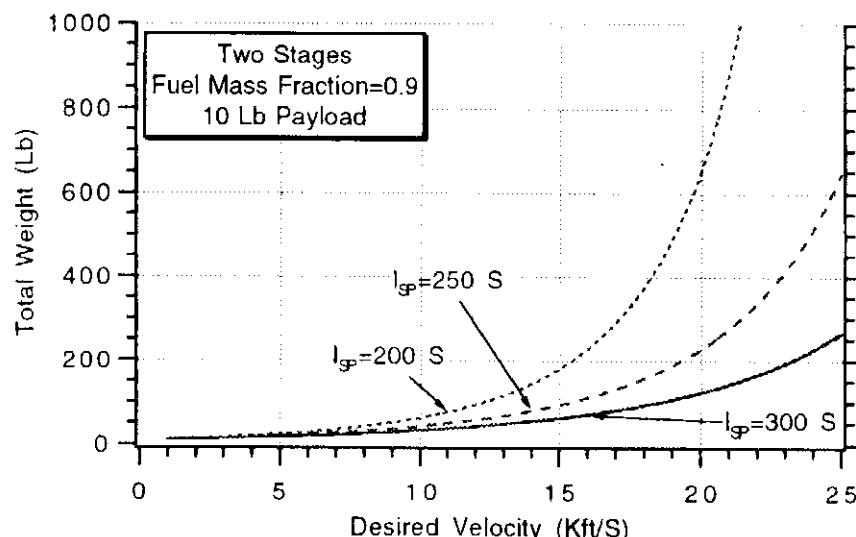


Fig. 13.5 Adding a stage reduces total weight requirements.

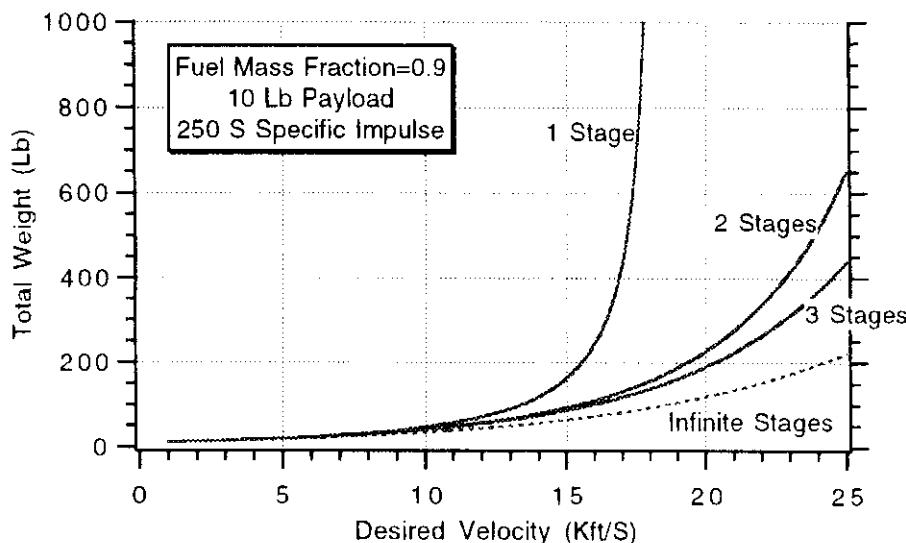


Fig. 13.6 Three stages yield near-minimal weight.

If the approximate fuel mass fraction were unity, the structural weight would be zero. In this case there would be no benefit to staging. In a sense, the unity fuel mass fraction case represents the minimum total weight that can propel a payload to a desired velocity for a given specific impulse. Figure 13.6 presents a comparison of weight requirements for different staging options. In the comparison an approximate fuel mass fraction of 0.9 and specific impulse 250 s were assumed for each of the stages. In addition, it was assumed that each stage contributed an equal fraction to the total desired velocity change. Superimposed on the figure is the infinite stage case (approximate fuel mass fraction equals unity) to represent minimal attainable weight. We can see that three stages get near-optimal answers for the case in which the approximate fuel mass fraction is 0.9 and specific impulse is 250 s.

Booster Numerical Example

We now have enough information so that we can begin, to first order, to model the boost phase of a strategic interceptor. In the previous section we derived formulas so that we could calculate weights based on desired velocity, approximate fuel mass fraction, and specific impulse. The maximum axial acceleration will occur right before staging, since that is where the interceptor weight is a minimum. If the maximum axial acceleration for each stage is given, then we have enough information to find the thrust levels for each of the stages. For example, in a two-stage strategic interceptor, the thrust level during stage 1, T_1 , is given by

$$T_1 = a_{\max 1}(W_{\text{tot}2} + W_{S1})$$

where $a_{\max 1}$ is the maximum axial acceleration of the first stage, in units of gravity, and $W_{\text{tot}2} + W_{S1}$ is the weight of the first stage right before staging. The thrust level of the second stage can be found in a similar way and is given by

$$T_2 = a_{\max 2}(W_{\text{pay}} + W_{S2})$$

where $a_{\max 2}$ is the maximum axial acceleration of the second stage, in units of gravity, and $W_{\text{pay}} + W_{S2}$ is the weight of the second stage right before staging. We can find the thrust burn times from specific impulse and thrust information. The

Table 13.1 Simulation outputs

Symbol	Definition	Value
W_{tot}	Total interceptor weight	6169 lb
W_{P1}	First-stage propellant weight	3474 lb
W_{S1}	First-stage structural weight	613 lb
W_{P2}	Second-stage propellant weight	1685 lb
W_{S2}	Second-stage structural weight	297 lb
T_1	Thrust level of first stage	26,950 lb
t_{B1}	Thrust burn time of first stage	32.2 s
T_2	Thrust level of second stage	3973 lb
t_{B2}	Thrust burn time of second stage	106 s

first- and second-stage burn times are given by

$$t_{B1} = \frac{I_{sp1} W_{P1}}{T_1}$$

$$t_{B2} = \frac{I_{sp2} W_{P2}}{T_2}$$

We now have enough information so that, given sufficient high-level information, we can compute a hypothetical booster's thrust-weight profiles. Listing 13.1 presents a FORTRAN program in which thrust-weight information is computed to yield a desired velocity change. The program assumes a two-stage booster with a 100-lb payload. The specific impulse for both stages is the same and is 250 s, and the approximate fuel mass fraction for both stages is also the same and is 0.85. The desired change in velocity is 20,000 ft/s with the first stage contributing one-third of the desired ΔV and the second stage contributing the rest. The maximum axial acceleration in both stages is specified to be 10 g. The program also integrates the computed acceleration to check if the desired velocity is reached.

The program was run with the nominal inputs, and the interceptor total weight was computed to be 6169 lb. Table 13.1 summarizes the program's computation of key parameters.

Figure 13.7 presents the information of Table 13.1 in graphic form (but not to scale) as a thrust-weight profile. The sharp weight drops at 32.2 s and 138.2 s represent staging events (structural weight dropped). After the interceptor is finished burning propellant at 138.2 s, the total weight is the payload weight of 100 lb, as can be seen from the figure.

The FORTRAN program of Listing 13.1 also had a capability to integrate the one-dimensional equation of motion

$$\dot{V} = \frac{gT}{W}$$

where g is the gravitational acceleration, T is the thrust level, and W the interceptor weight. Values for the instantaneous thrust and weight are obtained from Fig. 13.7. Figure 13.8 displays the resultant velocity and acceleration profiles for the nominal case. We can first see that the desired velocity goal of 20 kft/s has been reached by the end of the second-stage burn and that one-third of the velocity was attained at

Listing 13.1 FORTRAN thrust-weight computations

```

INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
XISP1=250.
XISP2=250.
XMF1=.85
XMF2=.85
WPAY=100.
DELV=20000.
DELV1=.3333*DELV
DELV2=.6667*DELV
AMAX1=10.
AMAX2=10.
TOP2=WPAY*(EXP(DELV2/(XISP2*32.2))-1.)
BOT2=1/XMF2-((1.-XMF2)/XMF2)*EXP(DELV2/(XISP2*32.2))
WP2=TOP2/BOT2
WS2=WP2*(1-XMF2)/XMF2
WTOT2=WP2+WS2+WPAY
TRST2=AMAX2*(WPAY+WS2)
TB2=XISP2*WP2/TRST2
TOP1=WTOT2*(EXP(DELV1/(XISP1*32.2))-1.)
BOT1=1/XMF1-((1.-XMF1)/XMF1)*EXP(DELV1/(XISP1*32.2))
WP1=TOP1/BOT1
WS1=WP1*(1-XMF1)/XMF1
WTOT=WP1+WS1+WTOT2
TRST1=AMAX1*(WTOT2+WS1)
TB1=XISP1*WP1/TRST1
DELVK=DELV/1000.
WRITE(*,*)'TOTAL WEIGHT= ',WTOT
WRITE(*,*)'WP2= ',WP2
WRITE(*,*)'WS2= ',WS2
WRITE(*,*)'WP1= ',WP1
WRITE(*,*)'WS1= ',WS1
WRITE(*,*)'TRST2= ',TRST2
WRITE(*,*)'TRST1= ',TRST1
WRITE(*,*)'TB2= ',TB2
WRITE(*,*)'TB1= ',TB1
PAUSE
H=.01
T=0.
S=0.
V=0.
10 IF(T>(TB1+TB2))GOTO 999
VOLD=V
STEP=1
GOTO 200
66 STEP=2
V=V+H*A
T=T+H
GOTO 200
55 V=(VOLD+V)/2+.5*H*A

```

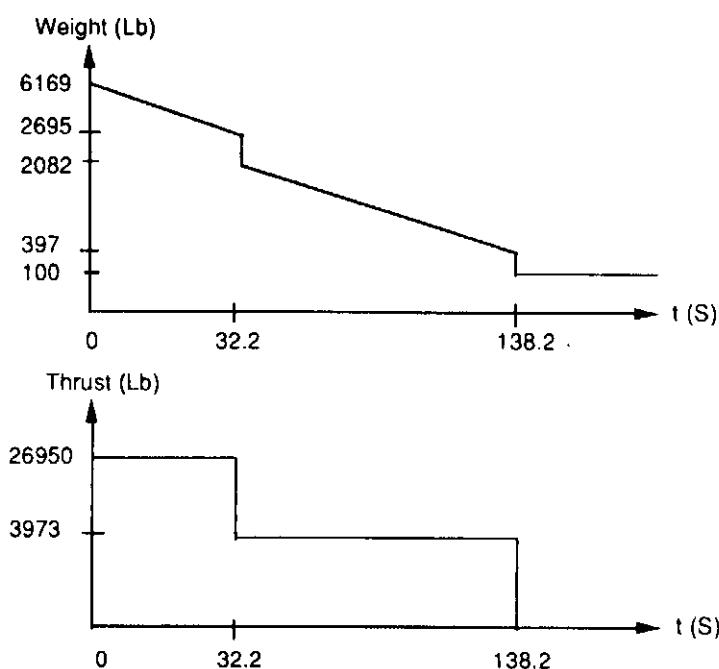
(Contd.)

Listing 13.1 (Continued)

```

S=S+H
IF(S<.99999)GOTO 10
S=0.
AG=A/32.2
VK=V/1000.
WRITE(*,97)T,VK,AG
WRITE(1,97)T,VK,AG
GOTO 10
97  FORMAT(3F10.3)
200  CONTINUE
IF(T<TB1)THEN
    WGT=-WP1*T/TB1+WTOT
    TRST=TRST1
ELSEIF(T<(TB1+TB2))THEN
    WGT=-WP2*T/TB2+WTOT2+WP2*TB1/TB2
    TRST=TRST2
ELSE
    WGT=WPAY
    TRST=0.
ENDIF
A=32.2*TRST/WGT
IF(STEP-1)66,66,55
999  CONTINUE
AG=A/32.2
VK=V/1000.
WRITE(*,97)T,VK,AG
WRITE(1,97)T,VK,AG
PAUSE
CLOSE(1)
END

```

**Fig. 13.7 Thrust-weight profiles for nominal case.**

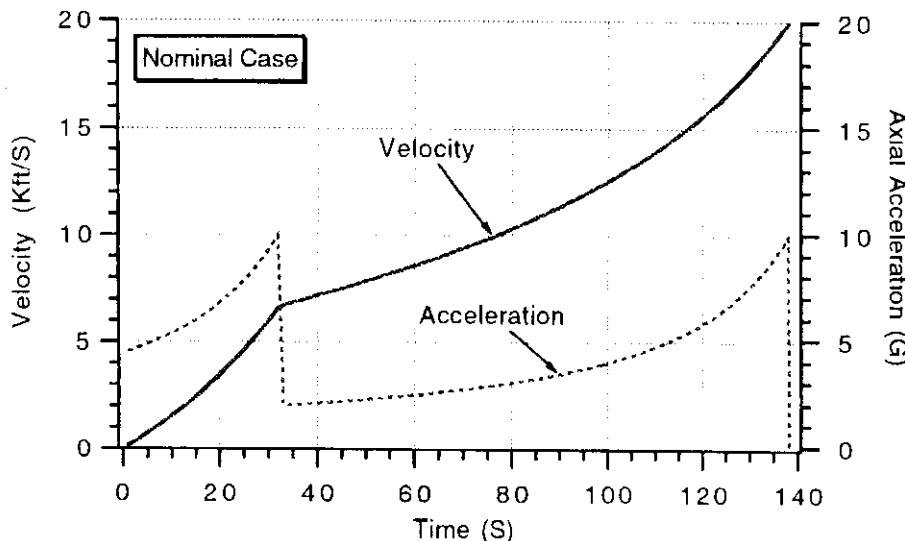


Fig. 13.8 Velocity and acceleration goals met with nominal design.

the end of the first-stage burn. We can also see from the acceleration profile that the desired maximum acceleration level of 10 g was also met. However, the axial booster acceleration is not constant and varied between 4 g and 10 g during the first-stage burn and varied between 2 g and 10 g during the second-stage burn.

Gravity Turn¹

Now that we have a nominal two-stage booster design, we would like to simulate its flight. Since booster steering is beyond the scope of this chapter, we will assume that the booster is launched at an initial flight-path angle γ with respect to the surface of the Earth. For counterclockwise travel Fig. 13.9 indicates the appropriate sign conventions and angle definitions, whereas for clockwise travel Fig. 13.10 is appropriate.

If we attempt to align the thrust vector with the booster velocity vector, we will obtain a gravity turn. The acceleration due to the booster thrusting a_T is given by

$$a_T = \frac{gT}{W}$$

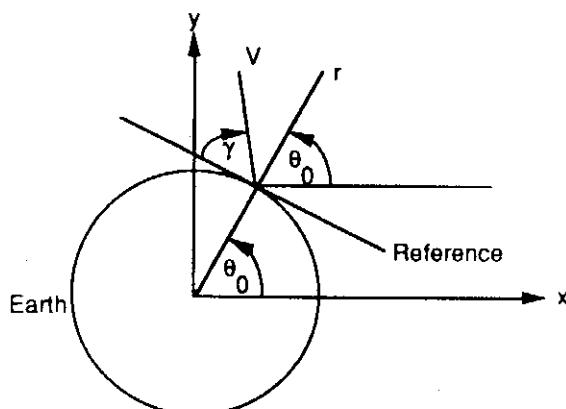


Fig. 13.9 Counterclockwise travel.

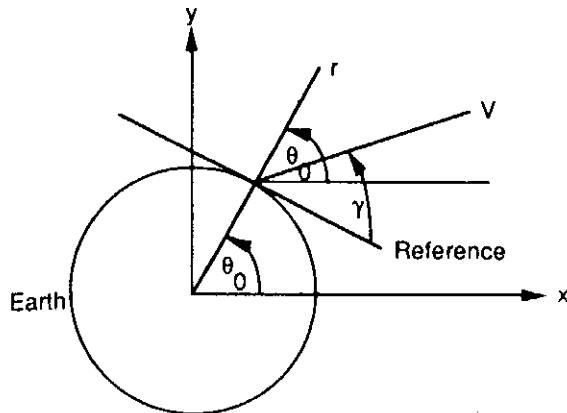


Fig. 13.10 Clockwise travel.

where T is the thrust magnitude in pounds, W the missile weight, and g is 32.2 ft/s^2 . The booster velocity V at any time could be found from the velocity components as

$$V = (\dot{x}^2 + \dot{y}^2)^{0.5}$$

Therefore, during a gravity turn at any time the components of acceleration acting on the booster in our Earth-centered coordinate system are given by

$$\begin{aligned}\ddot{x} &= \frac{-gm x}{(x^2 + y^2)^{1.5}} + \frac{a_T \dot{x}}{V} \\ \ddot{y} &= \frac{-gm y}{(x^2 + y^2)^{1.5}} + \frac{a_T \dot{y}}{V}\end{aligned}$$

where the initial conditions on velocity are related to the initial flight-path angle and location. For counterclockwise travel, the velocity initial conditions are

$$\begin{aligned}\dot{x}(0) &= V(0) \cos(\pi/2 - \gamma_0 + \theta_0) \\ \dot{y}(0) &= V(0) \sin(\pi/2 - \gamma_0 + \theta_0)\end{aligned}$$

whereas for clockwise travel the appropriate velocity initial conditions are

$$\begin{aligned}\dot{x}(0) &= V(0) \cos(-\pi/2 + \gamma_0 + \theta_0) \\ \dot{y}(0) &= V(0) \sin(-\pi/2 + \gamma_0 + \theta_0)\end{aligned}$$

The initial components of the booster location are given by

$$\begin{aligned}x(0) &= (a + alt) \cos \theta_0 \\ y(0) &= (a + alt) \sin \theta_0\end{aligned}$$

Listing 13.2 presents a FORTRAN program that, given some booster design parameters, finds the appropriate thrust-weight profiles and, in addition, flies the booster through a gravity turn. We can see from the listing that the nominal booster design is the default case and that the initial flight-path angle of the booster during the gravity turn is 85 deg. During the trajectory the flight-path angle will start from 85 deg and gradually reduce to smaller values.

Cases were run with the nominal design, and the initial flight-path angle was made a parameter. The resultant trajectories, shown in Fig. 13.11, indicate that

Listing 13.2 Gravity turn simulation

```

INTEGER STEP
LOGICAL LEFT
LEFT=.TRUE.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
XISP1=250.
XISP2=250.
XMF1=.85
XMF2=.85
WPAY=100.
DELV=20000.
DELV1=.3333*DELV
DELV2=.6667*DELV
AMAX1=10.
AMAX2=10.
GAMDEG=85.
TOP2=WPAY*(EXP(DELV2/(XISP2*32.2))-1.)
BOT2=1/XMF2-((1.-XMF2)/XMF2)*EXP(DELV2/(XISP2*32.2))
WP2=TOP2/BOT2
WS2=WP2*(1-XMF2)/XMF2
WTOT2=WP2+WS2+WPAY
TRST2=AMAX2*(WPAY+WS2)
TB2=XISP2*WP2/TRST2
TOP1=WTOT2*(EXP(DELV1/(XISP1*32.2))-1.)
BOT1=1/XMF1-((1.-XMF1)/XMF1)*EXP(DELV1/(XISP1*32.2))
WP1=TOP1/BOT1
WS1=WP1*(1-XMF1)/XMF1
WTOT=WP1+WS1+WTOT2
TRST1=AMAX1*(WTOT2+WS1)
TB1=XISP1*WP1/TRST1
DELVK=DELV/1000.
H=.01
T=0.
S=0.
A=2.0926E7
GM=1.4077E16
ALTNM=0.
ALT=ALTNM*6076.
ANGDEG=90.
ANG=ANGDEG/57.3
IF(LEFT)THEN
    VRX=COS(1.5708-GAMDEG/57.3+ANG)
    VRY=SIN(1.5708-GAMDEG/57.3+ANG)
ELSE
    VRX=COS(-1.5708+GAMDEG/57.3+ANG)
    VRY=SIN(-1.5708+GAMDEG/57.3+ANG)
ENDIF
X=(A+ALT)*COS(ANG)
Y=(A+ALT)*SIN(ANG)
ALT=SQRT(X**2+Y**2)-A

```

(Contd.)

Listing 13.2 (Continued)

```

XFIRST=X
YFIRST=Y
X1=VRX
Y1=VRY
10 IF(ALT<0..AND.T>10.)GOTO 999
XOLD=X
YOLD=Y
X1OLD=X1
Y1OLD=Y1
STEP=1
GOTO 200
66 STEP=2
X=X+H*XD
Y=Y+H*YD
X1=X1+H*X1D
Y1=Y1+H*Y1D
T=T+H
GOTO 200
55 CONTINUE
X=(XOLD+X)/2+.5*H*XD
Y=(YOLD+Y)/2+.5*H*YD
X1=(X1OLD+X1)/2+.5*H*X1D
Y1=(Y1OLD+Y1)/2+.5*H*Y1D
ALT=SQRT(X**2+Y**2)-A
S=S+H
IF(S<9.99999)GOTO 10
S=0.
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTNM)
ALTNM=(SQRT(X**2+Y**2)-A)/6076.
XNM=X/6076.
YNM=Y/6076.
WRITE(*,97)T,DISTNM,ALTNM,XNM,YNM
WRITE(1,97)T,DISTNM,ALTNM,XNM,YNM
GOTO 10
97 FORMAT(5F10.3)
200 CONTINUE
IF(T<TB1)THEN
    WGT=-WP1*T/TB1+WTOT
    TRST=TRST1
ELSEIF(T<(TB1+TB2))THEN
    WGT=-WP2*T/TB2+WTOT2+WP2*TB1/TB2
    TRST=TRST2
ELSE
    WGT=WPAY
    TRST=0.
ENDIF
AT=32.2*TRST/WGT
VEL=SQRT(X1**2+Y1**2)

```

(Contd.)

Listing 13.2 (Continued)

```

AXT=AT*X1/VEL
AYT=AT*Y1/VEL
TEMBOT=(X**2+Y**2)**1.5
X1D=-GM*X/TEMBOT+AXT
Y1D=-GM*Y/TEMBOT+AYT
XD=X1
YD=Y1
IF(STEP-1)66,66,55
999 CONTINUE
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTNM)
ALTNM=(SQRT(X**2+Y**2)-A)/6076.
XNM=X/6076.
YNM=Y/6076.
WRITE(*,97)T,DISTNM,ALTNM,XNM,YNM
WRITE(1,97)T,DISTNM,ALTNM,XNM,YNM
PAUSE
CLOSE(1)
END
SUBROUTINE DISTANCE(XT,YT,XF,YF,DISTNM)
SAVE
R=SQRT(XT**2+YT**2)
A=2.0926E7
CBETA=(XT*XF+YT*YF)/(R*A)
IF(CBETA<1.)THEN
    BETA=ACOS(CBETA)
    DISTNM=A*BETA/6076.
ELSE
    DISTNM=(XF-XT)/6076.
ENDIF
RETURN
END

```

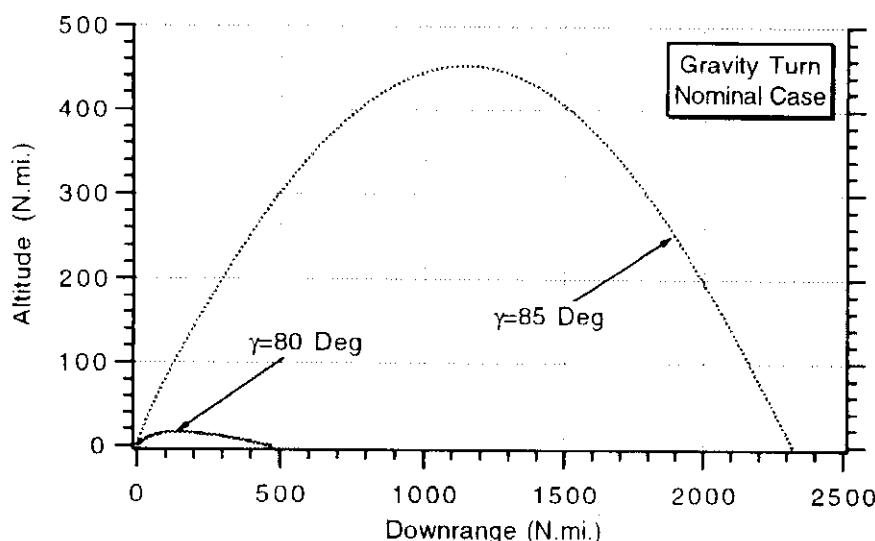


Fig. 13.11 Large flight-path angles are required for initial booster design.

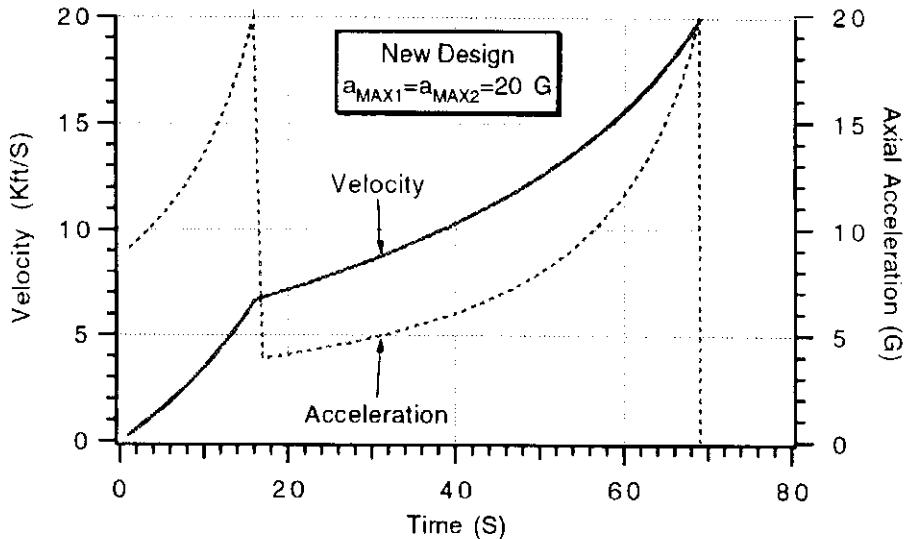


Fig. 13.12 Doubling booster axial acceleration halves burn time.

large flight-path angles are required just to get a trajectory for a gravity turn! If the flight-path angle is less than 80 deg, the booster will immediately crash into the Earth. As the booster thrusts, the flight-path angle rapidly decreases due to the small booster acceleration (about 4 g at the beginning). Eventually the flight-path angle decreases to the point where the component of the booster acceleration perpendicular to the surface of the Earth is not sufficient to overcome gravity.

To remedy the situation so that we could get smaller flight-path angles to yield longer range trajectories, the maximum axial booster acceleration during each stage was increased from 10 g to 20 g. The resultant velocity and acceleration profiles due to this change appear in Fig. 13.12. We can see that the booster still reaches a velocity of 20 kft/s, but in half the time of the nominal design.

Gravity turns were performed, via the simulation, for the new booster design, and the results for different flight-path angles appear in Fig. 13.13. We can see that the larger axial booster acceleration allowed the booster to experience lower flight-path angles (without crashing into the ground), which increased the booster

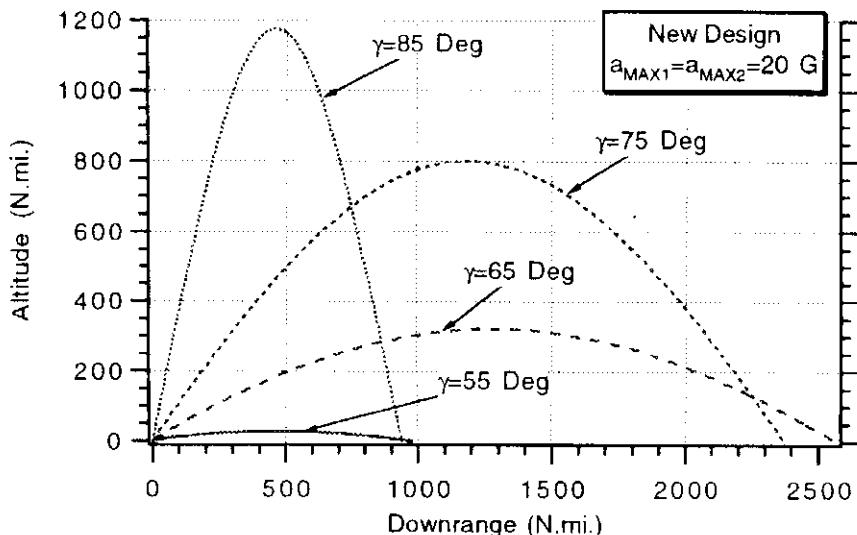


Fig. 13.13 New booster design yields longer flyout ranges.

range. With the nominal design, the maximum range achieved with a flight-path angle of 85 deg was about 2300 n.mi. The new design, which permitted a lower flight-path angle of 65 deg, increased the maximum range about 2600 n.mi.

Summary

In this chapter we have attempted to show that it takes a great deal of booster weight to bring a small payload to near-orbital speeds. High-level formulas were developed and presented so that booster parameters could be specified from fundamental rocket equation relationships. The impact of a key booster parameter on a simple gravity turn trajectory was demonstrated via a simple numerical example.

Reference

¹Regan, F., *Re-Entry Vehicle Dynamics*, AIAA Education Series, New York, 1984.



Lambert Guidance

Introduction

A particular problem, known as the problem of Lambert, has intrigued mathematicians for centuries. The solution to this problem is important for navigating spacecraft and for putting strategic missiles on a collision triangle. Elegant numerical solutions exist for the Lambert problem that are based on the known properties of a body in a gravity field.¹ The best of these solutions are numerically very efficient and accurate and, in fact, currently serve as fundamental algorithms in steering both spacecraft and ballistic missiles. Unfortunately, these elegant solutions are extremely difficult to understand because they involve subtle points in conic sections and a detailed understanding of hypergeometric series. In this chapter we shall use an easy to understand but numerically inefficient algorithm for solving Lambert's problem. It will then be shown how to speed up the algorithm by two orders of magnitude using a simple numerical technique. We shall then show how this solution can be used to steer a strategic boosting missile on a collision triangle with a threat.

Statement of Lambert's Problem

A body in a gravity field satisfies Newton's law of universal gravitation, or

$$\ddot{x} = \frac{-gm x}{(x^2 + y^2)^{1.5}}$$

$$\ddot{y} = \frac{-gm y}{(x^2 + y^2)^{1.5}}$$

Assume that the initial location of a body in the gravity field is given by

$$x(0) = x_0$$

$$y(0) = y_0$$

and it is desired that t_F seconds later the body be at location

$$x(t_F) = x_F$$

$$y(t_F) = y_F$$

Lambert's problem is to find the initial velocity orientation of the body in the gravity

field so that the preceding initial conditions and boundary values are satisfied, or

$$\dot{x}(0) = ?$$

$$\dot{y}(0) = ?$$

Solution to Lambert's Problem

We showed in Chapter 12 that, given an initial flight-path angle and distance to be traveled, the initial missile velocity required to hit an object on the surface of the Earth is given by

$$V = \sqrt{\frac{gm(1 - \cos \phi)}{r_0 \cos \gamma [(r_0 \cos \gamma/a) - \cos(\phi + \gamma)]}}$$

where ϕ is the central angle separating the initial location of the missile and its intended target, γ the initial flight-path angle of the missile, a the radius of the Earth, and r_0 the initial distance from the center of the Earth to the missile, which can be expressed as

$$r_0 = a + alt$$

where alt is the initial altitude of the missile with respect to the surface of the Earth. Although the velocity equation was derived for hitting an object on the surface of the Earth, it can be made more general. If we desire to hit a target at any location r_F the preceding velocity equation can be modified to

$$V = \sqrt{\frac{gm(1 - \cos \phi)}{r_0 \cos \gamma [(r_0 \cos \gamma/r_F) - \cos(\phi + \gamma)]}}$$

In this new formula r_F is defined as

$$r_F = a + alt_F$$

where alt_F is the altitude of the intended target.

If the velocity vector is oriented for counterclockwise travel as shown in Fig. 14.1, then, given the preceding solution for the total required velocity, we can find the

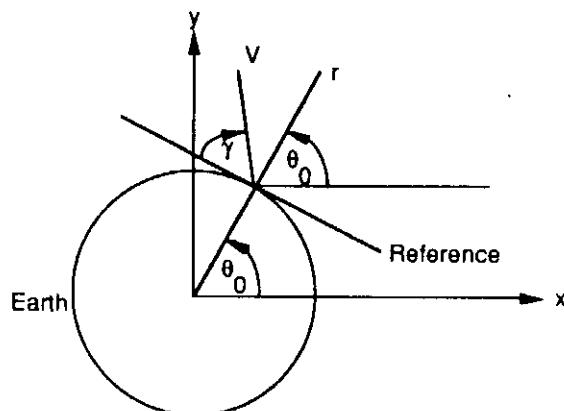


Fig. 14.1 Counterclockwise travel.

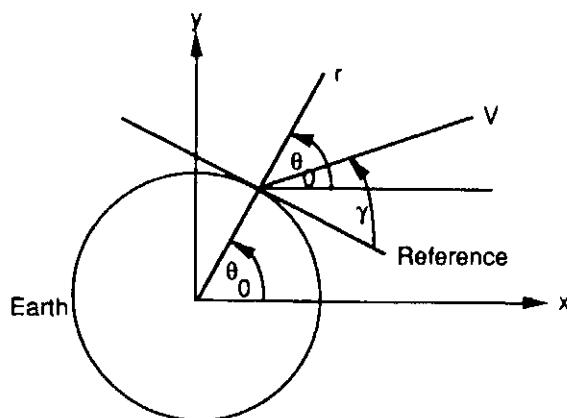


Fig. 14.2 Clockwise travel.

initial conditions on the velocity components in the Earth-centered system by trigonometry as

$$\dot{x}(0) = V \cos(\pi/2 - \gamma + \theta_0)$$

$$\dot{y}(0) = V \sin(\pi/2 - \gamma + \theta_0)$$

where γ is the orientation of the missile velocity with respect to a reference that is tangent to the Earth and perpendicular to the vector from the center of the Earth to the initial location of the missile. We can see from Fig. 14.1 that θ_0 is the initial angular location of the missile with respect to the x axis of the Earth-centered Cartesian coordinate system.

On the other hand, if the velocity vector is intended to travel clockwise as shown in Fig. 14.2, then the initial conditions on the velocity components in the Earth-centered system can easily be shown to be

$$\dot{x}(0) = V \cos(\gamma - \pi/2 + \theta_0)$$

$$\dot{y}(0) = V \sin(\gamma - \pi/2 + \theta_0)$$

In Chapter 12 we also derived a formula for the time required for the missile to reach its intended target (t_F). The formula, which is valid for elliptical travel ($\lambda < 2$), does not require the target to be on the surface of the Earth and is given by

$$t_F = \frac{r_0}{V \cos \gamma} \left\{ \frac{\tan \gamma (1 - \cos \phi) + (1 - \lambda) \sin \phi}{(2 - \lambda) \left[\frac{1 - \cos \phi}{\lambda \cos^2 \gamma} + \frac{\cos(\gamma + \phi)}{\cos \gamma} \right]} + \frac{2 \cos \gamma}{\lambda[(2/\lambda) - 1]^{1.5}} \tan^{-1} \left(\frac{\sqrt{(2/\lambda) - 1}}{\cos \gamma \cot(\phi/2) - \sin \gamma} \right) \right\}$$

where V is the required velocity to hit the object, and λ was defined in Chapter 12 as

$$\lambda = \frac{r_0 V^2}{gm}$$

and ϕ is the angular distance to be traveled.

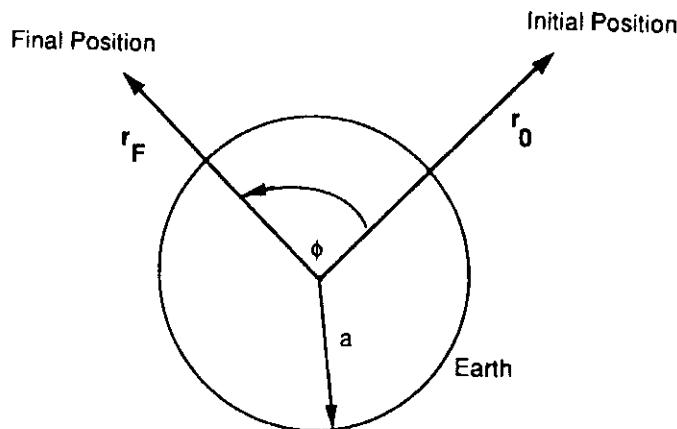


Fig. 14.3 Central angle between initial and final position.

To find the angular distance to be traveled, consider the geometry of Fig. 14.3, in which the initial and final position of an object in a gravity field are shown. In this figure \mathbf{r}_0 denotes a vector from the center of the Earth to the initial location of the object, and \mathbf{r}_F denotes a vector from the center of the Earth to the final location of the object. The angle between the vectors is the central angle ϕ .

The central angle can be found from the definition of the vector dot product, or

$$\phi = \cos^{-1} \frac{\mathbf{r}_0 \cdot \mathbf{r}_F}{|\mathbf{r}_0||\mathbf{r}_F|}$$

Believe it or not, we now have sufficient information to numerically solve Lambert's problem!

If we know the initial and final destination of the target, we have just shown that we can find the central angle ϕ . With a central angle, \mathbf{r}_0 , \mathbf{r}_F , and a flight-path angle γ , sufficient information is available to find the required velocity from our closed-form solution. The resultant velocity can then be used to solve for the flight time from our other closed-form solution. It is important to note that the flight time and velocity obtained are *exact* solutions for the flight-path angle used. Stated mathematically, we can say that given γ , \mathbf{r}_0 , and \mathbf{r}_F we can use the following relationships, which are based on exact closed-form solutions:

$$\phi = f(\mathbf{r}_0, \mathbf{r}_F)$$

$$V = f(\mathbf{r}_0, \mathbf{r}_F, \phi, \gamma)$$

$$t_F = f(V, \phi, \gamma)$$

Recall that in Lambert's problem we are given \mathbf{r}_0 , \mathbf{r}_F , and t_F and seek to find V and γ . If we use the preceding relationships, we do not know how to choose γ , nor are we guaranteed that a particular value of γ will yield the desired flight time t_F .

We can solve the problem by the method of brute force. That is, we work out all solutions until we find the one that satisfies the constraints of the problem. For example, we start with $\gamma = -90$ deg, solve for the velocity, and then solve for the time of flight. If the flight time is less than the desired flight time, we repeat the procedure with a slightly larger value of γ . We stop the loop when the computed flight time is greater than the desired flight time. If the flight-path angle that satisfies the preceding procedure is negative, we know that the solution must be rejected since it requires the missile to travel through the Earth. This numerical

method converges because we saw in Fig. 12.17 that flight time is smooth and monotonically increasing with increasing flight-path angle.

Numerical Example

Listing 14.1 presents sample FORTRAN code, using double-precision arithmetic, for finding the Lambert solution, based on the procedure developed in the previous section. In the notation of Listing 14.1 we can say that, given an initial angle and altitude for the missile (*XLONGMDEG*, *ALTNMM*), an initial angle and altitude for the target (*XLONGTDEG*, *ALTNMT*), and a desired flight time (*TF*), the program iterates on the flight-path angle (*GAMDEG*) until a solution is found. From the listing we can see that the program consists of two loops. The first loop iterates on the flight-path angle in units of 0.1 deg. When a flight time is found that exceeds the desired flight time, we exit the loop for another loop that increments the flight-path angle (after decreasing the last flight path angle by 0.15 deg) in very fine units of 0.0001 deg. This loop is required to get extremely precise answers. When the desired flight time is achieved, we exit the loop and the routine. For most cases the program takes about 1.5 s to execute on a 16 MHz, 32-bit microcomputer with a math coprocessor. The routine, as written, is about 100 times slower than more elegant Lambert routines.¹ We shall show in the next section that by performing a more intelligent search it is possible to find the correct solution to Lambert's problem in a few iterations, thus making this approach very competitive with more elegant Lambert routines. However, the goal in this section is to develop a routine that simply works and is easy to understand.

To demonstrate how the routine works, the nominal case, shown in the listing, was run. In this case the missile is on the surface of the Earth 45 deg away from the target. It is desired to find the velocity orientation of the missile (*VRX*, *VRY*) so that the missile will hit the target in exactly 1000 s. Figure 14.4 shows that the solution converges to the exact value in 1084 iterations. However, the solution appears to be approximately correct after 335 iterations.

To investigate the tradeoff between accuracy vs number of iterations required, a simple experiment was conducted. First the second loop of Listing 14.1 was

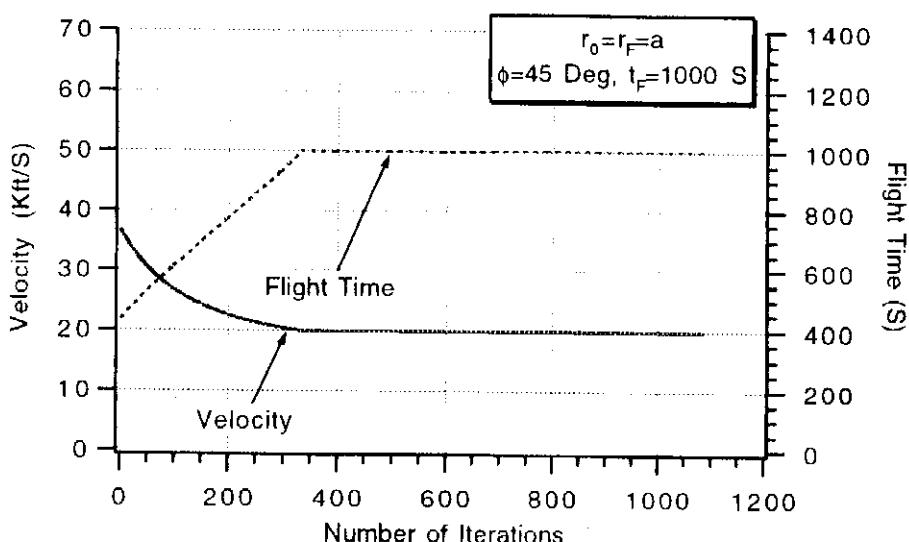


Fig. 14.4 It takes 1084 iterations to get exact solution.

Listing 14.1 Lambert routine using brute force approach

```

IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
XLONGMDEG=45.
XLONGTDEG=90.
ALTNMT=0.
ALTNMM=0.
TF=1000.
PI=3.14159
DEGRAD=360./(2.*PI)
A=2.0926E7
GM=1.4077E16
ALTT=ALTNMT*6076.
ALTM=ALTNMM*6076.
XLONGM=XLONGMDEG/DEGRAD
XLONGT=XLONGTDEG/DEGRAD
XM=(A+ALTM)*COS(XLONGM)
YM=(A+ALTM)*SIN(XLONGM)
XT=(A+ALTT)*COS(XLONGT)
YT=(A+ALTT)*SIN(XLONGT)
CALL LAMBERT(XM,YM,TF,XT,YT,VRXM,VRYM,XLONGM,XLONGT)
WRITE(*,97)TF,VRXM,VRYM
WRITE(1,97)TF,VRXM,VRYM
97 FORMAT(3F10.3)
PAUSE
CLOSE(1)
END

SUBROUTINE LAMBERT(XIC,YIC,TFDES,XF,YF,VRX,VRY,XLONGM
,XLONGT)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
A=2.0926E7
GM=1.4077E16
RIC=SQRT(XIC**2+YIC**2)
RF=SQRT(XF**2+YF**2)
CPHI=(XIC*XF+YIC*YF)/(RIC*RF)
PHI=ACOS(CPHI)
R0=RIC
PI=3.14159
DEGRAD=360./(2.*PI)
ICOUNT=0
DO 10 GAMDEG=-90.,90.,1
    GAM=GAMDEG/DEGRAD
    TOP=GM*(1.-COS(PHI))
    TEMP=R0*COS(GAM)/RF-COS(PHI+GAM)
    BOT=R0*COS(GAM)*TEMP
    IF(TOP<0..OR.BOT<0.)GOTO 10
    V=SQRT(TOP/BOT)

```

(Contd.)

Listing 14.1 (Continued)

```

IF (XLONGT>XLONGM) THEN
  VRX=V*COS(PI/2.-GAM+XLONGM)
  VRY=V*SIN(PI/2.-GAM+XLONGM)
ELSE
  VRX=V*COS(-PI/2.+GAM+XLONGM)
  VRY=V*SIN(-PI/2.+GAM+XLONGM)
END IF
XLAM=R0*V*V/GM
TOP1=TAN(GAM)*(1-COS(PHI))+(1-XLAM)*SIN(PHI)
BOT1P=(1-COS(PHI))/(XLAM*COS(GAM)*COS(GAM))
BOT1=(2-XLAM)*(BOT1P+COS(GAM+PHI)/COS(GAM))
TOP2=2*COS(GAM)
IF((2/XLAM-1)<0.)GOTO 10
BOT2=XLAM*((2/XLAM-1)**1.5)
TOP3=SQRT(2/XLAM-1)
BOT3=COS(GAM)/TAN(PHI/2)-SIN(GAM)
TEMP=(TOP2/BOT2)*ATAN2(TOP3,BOT3)
TF=R0*(TOP1/BOT1+TEMP)/(V*COS(GAM))
IF(TF>TFDES)THEN
  EXIT
ENDIF
10  CONTINUE
GAMDEGNEW=GAMDEG-.15
GAMDEGFIN=GAMDEG+1.
DO 20 GAMDEG=GAMDEGNEW,GAMDEGFIN,.0001
  GAM=GAMDEG/DEGRAD
  TOP=GM*(1.-COS(PHI))
  TEMP=R0*COS(GAM)/RF-COS(PHI+GAM)
  BOT=R0*COS(GAM)*TEMP
  IF(TOP<0..OR.BOT<0.)GOTO 20
  V=SQRT(TOP/BOT)
  IF (XLONGT>XLONGM) THEN
    VRX=V*COS(PI/2.-GAM+XLONGM)
    VRY=V*SIN(PI/2.-GAM+XLONGM)
  ELSE
    VRX=V*COS(-PI/2.+GAM+XLONGM)
    VRY=V*SIN(-PI/2.+GAM+XLONGM)
  END IF
  XLAM=R0*V*V/GM
  TOP1=TAN(GAM)*(1-COS(PHI))+(1-XLAM)*SIN(PHI)
  BOT1P=(1-COS(PHI))/(XLAM*COS(GAM)*COS(GAM))
  BOT1=(2-XLAM)*(BOT1P+COS(GAM+PHI)/COS(GAM))
  TOP2=2*COS(GAM)
  IF((2/XLAM-1)<0.)GOTO 20
  BOT2=XLAM*((2/XLAM-1)**1.5)
  TOP3=SQRT(2/XLAM-1)
  BOT3=COS(GAM)/TAN(PHI/2)-SIN(GAM)
  TEMP=(TOP2/BOT2)*ATAN2(TOP3,BOT3)
  TF=R0*(TOP1/BOT1+TEMP)/(V*COS(GAM))

```

(Contd.)

Listing 14.1 (Continued)

```

IF(TF>TFDES)THEN
  EXIT
ENDIF
20  CONTINUE
RETURN
END

```

Table 14.1 Accuracy experiments

Condition	V_{RX}	V_{RY}	t_F	Iterations
Nominal	-7696	18,329	1000	1084
Remove second loop	-7668	18,332	1001	335
One-degree increments	-7418	18,360	1014	34

removed from the Lambert subroutine so that the flight-path angle was only incremented in steps of 0.1 deg. Table 14.1 shows that the number of iterations required were reduced from 1084 to 335 and the resultant velocity accuracy (VRX , VRY) appears to be reduced slightly. Actually, the velocities are exact for a 1001-s flight but approximate for a Lambert solution requiring a 1000-s flight. Next, the first loop was modified so that the flight-path angle was incremented in steps of 1 deg (increased from 0.1 deg steps). Table 14.1 shows that the number of iterations was reduced to only 34, but the accuracy loss was more significant if the desired flight time is truly 1000 s. These answers are exact in the sense a hit will result in 1014 s but inaccurate for the Lambert solution requiring exactly 1000 s.

Speeding Up Lambert Routine

The routine for numerically solving Lambert's problem, presented in the previous section, can be speeded up by more than two orders of magnitude! We have already demonstrated that the brute force search on all possible flight-path angles results in many iterations. We can considerably restrict the brute force search and eliminate many iterations by recalling that the velocity formula was shown in Chapter 12 to be

$$V = \sqrt{\frac{gm(1 - \cos \phi)}{r_0 \cos \gamma [(r_0 \cos \gamma / r_F) - \cos(\phi + \gamma)]}}$$

Since in this text we are only interested trajectories for ballistic missiles, we can immediately rule out cases that lead to escape velocity ($\lambda = 2$) or

$$\lambda = 2 = \frac{V^2 r_0}{gm}$$

Substitution of the escape velocity condition into the velocity formula yields

$$2 = \frac{(1 - \cos \phi)}{\cos \gamma [(r_0 \cos \gamma / r_F) - \cos(\phi + \gamma)]}$$

We can solve the preceding equation for the flight-path angle γ . After much algebra we get two solutions corresponding to the minimum and maximum flight-path angles as

$$\gamma_{\min} = \tan^{-1} \left\{ \left[\sin \phi - \sqrt{\frac{2r_0}{r_F} (1 - \cos \phi)} \right] / (1 - \cos \phi) \right\}$$

$$\gamma_{\max} = \tan^{-1} \left\{ \left[\sin \phi + \sqrt{\frac{2r_0}{r_F} (1 - \cos \phi)} \right] / (1 - \cos \phi) \right\}$$

It should not be surprising that there are two solutions for the flight-path angle since we have already observed this phenomenon in 12.14. We also noticed in Figs. 12.14 and 12.17 that the solution for the velocity and time of flight were smooth, well-behaved functions of the flight-path angle. Based on the nonpathological nature of these solutions and the fact that the flight-path angle is well bounded, we do not have to evaluate each flight-path angle but can instead perform a more efficient search in finding the flight-path angle that corresponds to the desired flight time. For example, we can use an algorithm known as the secant method³ to perform the search or

$$\gamma_{n+1} = \gamma_n + \frac{(\gamma_n - \gamma_{n-1})(t_{\text{TDES}} - t_{F_n})}{t_{F_n} - t_{F_{n-1}}}$$

We can see from the preceding equation that the new flight-path angle γ_{n+1} is related to previous values γ_n , γ_{n-1} . At each iteration the new computed value of flight-path angle is limited to the minimum and maximum possible values of the flight-path angle derived from the escape velocity condition. The search is terminated when the computed flight time t_{F_n} is sufficiently close to the desired flight time t_{TDES} .

Listing 14.2 is identical to the test program of Listing 14.1, except this time the Lambert routine is more efficient. We can see from the new Lambert routine that our initial guess of the flight-path angle is simply the average of the minimum and maximum flight-path angles derived from the escape velocity condition.

The nominal case of the previous section was rerun and detailed results for the number of iterations required appear in Table 14.2. We can see that very accurate Lambert solutions are obtained after only four iterations and that after seven iterations we are obtaining a degree of accuracy that is better than obtained

Table 14.2 Number of iterations are dramatically reduced

Iteration	Flight-path angle, deg	V_{RX} , ft/s	V_{RY} , ft/s	Flight time, s
1	33.7524947	-3764.57976	18926.02426	1239.37545
2	11.2508376	-12075.71473	18072.66484	813.53185
3	21.1038504	-8103.20444	18287.99279	979.68031
4	22.3088581	-7665.88409	18332.46792	1001.50708
5	22.2256555	-7695.84063	18329.28399	999.98566
6	22.2264396	-7695.55815	18329.31392	999.99999
7	22.2264402	-7695.55795	18329.31394	1000.00000

Listing 14.2 More efficient Lambert routine

```

IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
XLONGMDEG=45.
XLONGTDEG=90.
ALTNMT=0.
ALTNMM=0.
TF=1000.
PI=3.14159
DEGRAD=360./(2.*PI)
A=2.0926E7
GM=1.4077E16
ALTT=ALTNMT*6076.
ALTM=ALTNMM*6076.
XLONGM=XLONGMDEG/DEGRAD
XLONGT=XLONGTDEG/DEGRAD
XM=(A+ALTM)*COS(XLONGM)
YM=(A+ALTM)*SIN(XLONGM)
XT=(A+ALTT)*COS(XLONGT)
YT=(A+ALTT)*SIN(XLONGT)
CALL LAMBERT(XM,YM,TF,XT,YT,VRXM,VRYM,XLONGM,XLONGT)
PAUSE
CLOSE(1)
END

SUBROUTINE LAMBERT(XIC,YIC,TFDES,XF,YF,VRX,VRY,XLONGM
,XLONGT)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
A=2.0926E7
GM=1.4077E16
RIC=SQRT(XIC**2+YIC**2)
RF=SQRT(XF**2+YF**2)
CPHI=(XIC*XF+YIC*YF)/(RIC*RF)
PHI=ACOS(CPHI)
SPHI=SIN(PHI)
R0=RIC
PI=3.14159
DEGRAD=360./(2.*PI)
ICOUNT=0
GMIN=ATAN2(SPHI-SQRT(2.*R0*(1.-CPHI)/RF),(1-CPHI))
GMAX=ATAN2(SPHI+SQRT(2.*R0*(1.-CPHI)/RF),(1-CPHI))
GAM=(GMIN+GMAX)/2.
DO
    TOP=GM*(1.-COS(PHI))
    TEMP=R0*COS(GAM)/RF-COS(PHI+GAM)
    BOT=R0*COS(GAM)*TEMP
    V=SQRT(TOP/BOT)
    IF (XLONGT>XLONGM) THEN
        VRX=V*COS(PI/2.-GAM+XLONGM)

```

(Contd.)

Listing 14.2 (Continued)

```

        VRY=V*SIN(PI/2.-GAM+XLONGM)
ELSE
        VRX=V*COS(-PI/2.+GAM+XLONGM)
        VRY=V*SIN(-PI/2.+GAM+XLONGM)
END IF
XLAM=R0*V*V/GM
TOP1=TAN(GAM)*(1-COS(PHI))+(1-XLAM)*SIN(PHI)
BOT1P=(1-COS(PHI))/(XLAM*COS(GAM)*COS(GAM))
BOT1=(2-XLAM)*(BOT1P+COS(GAM+PHI)/COS(GAM))
TOP2=2*COS(GAM)
BOT2=XLAM*((2/XLAM-1)**1.5)
TOP3=SQRT(2/XLAM-1)
BOT3=COS(GAM)/TAN(PHI/2)-SIN(GAM)
TEMP=(TOP2/BOT2)*ATAN2(TOP3,BOT3)
TF=R0*(TOP1/BOT1+TEMP)/(V*COS(GAM))
ICOUNT=ICOUNT+1
WRITE(9,300)ICOUNT,57.3*GAM,VRX,VRY,TF
WRITE(1,300)ICOUNT,57.3*GAM,VRX,VRY,TF
300  FORMAT(I4,1X,F12.7,1X,F12.5,1X,F12.5,1X,F12.5)
IF((ABS(TFDES-TF).LE.(.00000001*TFDES)).OR.
1   ICOUNT.GT.100)THEN
        EXIT
ENDIF
IF(TF>TFDES)THEN
        GMAX=GAM
ELSE
        GMIN=GAM
ENDIF
IF(ICOUNT.EQ.1)THEN
        XNEXT=(GMAX+GMIN)/2.
ELSE
        XNEXT=GAM+(GAM-GOLD)*(TFDES-TF)/(TF-TOLD)
        IF(XNEXT>GMAX.OR.XNEXT<GMIN)THEN
                XNEXT=(GMAX+GMIN)/2.
        ENDIF
ENDIF
GOLD=GAM
TOLD=TF
GAM=XNEXT
REPEAT
RETURN
END

```

with 1084 iterations in the previous section using the brute force approach. *The new Lambert routine is not only more accurate than the one in the previous section but it is also more than two orders of magnitude faster!*

Reference 4 makes extensive tests on this efficient numerical solution to Lambert's problem and shows that it is competitive with the best numerical approaches. In addition, Ref. 4 also shows how this efficient solution of Lambert's problem can be extended to parabolic and hyperbolic trajectories.

Booster Steering

Thus far we have seen that, given that we know where we are and where we want to go and given an arrival time, the Lambert subroutine will tell us the orientation of the velocity vector for an impulsive missile to satisfy the problem. Since we do not have impulsive missiles (missiles that get up to speed immediately), it is desirable to find out if the Lambert subroutine could be of use in enabling a nonimpulsive missile or booster to reach its target. If we neglect the atmosphere, the solution to the problem is quite simple and is known as Lambert guidance.¹

Consider the vector diagram shown in Fig. 14.5. All that has to be done at small time increments, while the missile is boosting, is to find the desired velocity from the Lambert subroutine V_{Lambert} and subtract the current missile velocity V_M . The difference in velocities is known as the velocity to be gained ΔV . If the boosting missile thrust vector is aligned with the velocity to be gained vector, then the desired velocity will be obtained in a feedback fashion. When the desired velocity is achieved, the engine is cut off and the missile flies ballistically to the intended target.

Mathematically, we are saying that the components of the velocity to be gained are

$$\Delta V_x = V_{\text{Lambert}x} - V_{Mx}$$

$$\Delta V_y = V_{\text{Lambert}y} - V_{My}$$

Therefore, the total velocity to be gained is simply

$$\Delta V = (\Delta V_x^2 + \Delta V_y^2)^{0.5}$$

If the magnitude of the current thrust acceleration is given by a_T , then the direction of the thrust acceleration at each instant of time should be aligned with the velocity to be gained vector, or

$$a_{Tx} = a_T \Delta V_x / \Delta V$$

$$a_{Ty} = a_T \Delta V_y / \Delta V$$

Listing 14.3 presents a FORTRAN simulation of a two-stage booster using Lambert guidance during the boost phase. Actually, the scenario is unrealistic because g loading and range safety considerations have been ignored,² but it is useful for demonstrating how Lambert guidance works. The booster considered in this example has a capability of reaching a velocity of 20,000 ft/s. The booster is assumed to have two stages, each of which has a fuel mass fraction of 0.9 and

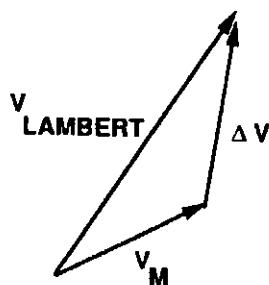


Fig. 14.5 Basis of Lambert guidance.

Listing 14.3 Booster simulation with Lambert guidance

```

IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER STEP
LOGICAL LEFT,QBOOST
LEFT=.TRUE.
QBOOST=.TRUE.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
OPEN(2,STATUS='UNKNOWN',FILE='LAMFIL')
XISP1=300.
XISP2=300.
XMF1=.90
XMF2=.90
WPAY=100.
DELV=20000.
DELV1=.3333*DELV
DELV2=.6667*DELV
AMAX1=20.
AMAX2=20.
TOP2=WPAY*(EXP(DELV2/(XISP2*32.2))-1.)
BOT2=1/XMF2-((1.-XMF2)/XMF2)*EXP(DELV2/(XISP2*32.2))
WP2=TOP2/BOT2
WS2=WP2*(1-XMF2)/XMF2
WTOT2=WP2+WS2+WPAY
TRST2=AMAX2*(WPAY+WS2)
TB2=XISP2*WP2/TRST2
TOP1=WTOT2*(EXP(DELV1/(XISP1*32.2))-1.)
BOT1=1/XMF1-((1.-XMF1)/XMF1)*EXP(DELV1/(XISP1*32.2))
WP1=TOP1/BOT1
WS1=WP1*(1-XMF1)/XMF1
WTOT=WP1+WS1+WTOT2
TRST1=AMAX1*(WTOT2+WS1)
TB1=XISP1*WP1/TRST1
DELVK=DELV/1000.
H=.01
T=0.
S=0.
A=2.0926E7
GM=1.4077E16
ALTNM=0.
ALT=ALTNM*6076.
ANGDEG=30.
ANG=ANGDEG/57.3
XLONGM=ANG
X=(A+ALT)*COS(ANG)
Y=(A+ALT)*SIN(ANG)
ALT=SQRT(X**2+Y**2)-A
XFIRST=X
YFIRST=Y
X1=0.

```

(Contd.)

Listing 14.3 (Continued)

```

Y1=0.
AXT=0.
AXT=0.
AYT=0.
XLONGTDEG=45.
XLONGT=XLONGTDEG/57.3
XF=A*COS(XLONGT)
YF=A*SIN(XLONGT)
TF=500.
10IF(ALT<0..AND.T>10.)GOTO 999
XOLD=X
YOLD=Y
X1OLD=X1
Y1OLD=Y1
STEP=1
GOTO 200
66 STEP=2
X=X+H*XD
Y=Y+H*YD
X1=X1+H*X1D
Y1=Y1+H*Y1D
T=T+H
GOTO 200

55 CONTINUE
X=(XOLD+X)/2+.5*H*XD
Y=(YOLD+Y)/2+.5*H*YD
X1=(X1OLD+X1)/2+.5*H*X1D
Y1=(Y1OLD+Y1)/2+.5*H*Y1D
ALT=SQRT(X**2+Y**2)-A
S=S+H
IF(QBOOST)THEN
    TGOLAM=TF-T
    XLONGM=ATAN2(Y,X)
    CALL LAMBERT(X,Y,TGOLAM,XF,YF,VRX,VRY,XLONGM,XLONGT,
1      ICOUNT)
    DELX=VRX-X1
    DELY=VRY-Y1
    DEL=SQRT(DELX**2+DELY**2)
    WRITE(2,99)T,VRX,X1,VRY,Y1,DISTNM,ALTNM
    IF(TRST>0..AND.DEL>500.)THEN
        AXT=AT*DELX/DEL
        AYT=AT*DELY/DEL
    ELSEIF(DEL<500.)THEN
        TRST=0.
        QBOOST=.FALSE.
        AXT=0.
        AYT=0.
        X1=VRX

```

(Contd.)

Listing 14.3 (Continued)

```

        Y1=VRY
        X1OLD=X1
        Y1OLD=Y1
    ELSE
        QBOOST=.FALSE.
        AXT=0.
        AYT=0.
    ENDIF
ENDIF
99 FORMAT(7F10.3)
IF(S.LT.9.99999)GOTO 10
S=0.
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTNM)
ALTNM=(SQRT(X**2+Y**2)-A)/6076.
VELK=SQRT(X1**2+Y1**2)/1000.
GAMDEG=57.3*ATAN2(Y1,X1)
XNM=X/6076.
YNM=Y/6076.
WRITE(*,97)T,DISTNM,ALTNM
WRITE(1,97)T,DISTNM,ALTNM
GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
IF(T<TB1)THEN
    WGT=-WP1*T/TB1+WTOT
    TRST=TRST1
ELSEIF(T<(TB1+TB2))THEN
    WGT=-WP2*T/TB2+WTOT2+WP2*TB1/TB2
    TRST=TRST2
ELSE
    WGT=WPAY
    TRST=0.
ENDIF
AT=32.2*TRST/WGT
XD=X1
YD=Y1
TEMBOT=(X**2+Y**2)**1.5
X1D=-GM*X/TEMBOT+AXT
Y1D=-GM*Y/TEMBOT+AYT
IF(STEP-1)66,66,55
999 CONTINUE
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTNM)
ALTNM=(SQRT(X**2+Y**2)-A)/6076.
VELK=SQRT(X1**2+Y1**2)/1000.
WRITE(*,98)T,DISTNM,ALTNM,VELK
WRITE(1,97)T,DISTNM,ALTNM
98 FORMAT(4F10.3)
PAUSE
CLOSE(1)
CLOSE(2)
END

```

Listing 14.3 (Continued)

```

SUBROUTINE DISTANCE(XT,YT,XF,YF,DISTNM)
SAVE
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
R=SQRT(XT**2+YT**2)
A=2.0926E7
CBETA=(XT*XF+YT*YF)/(R*A)
IF(CBETA<1.)THEN
    BETA=ACOS(CBETA)
    DISTNM=A*BETA/6076.
ELSE
    DISTNM=(XF-XT)/6076.
ENDIF
RETURN
END

SUBROUTINE LAMBERT(XIC,YIC,TFDES,XF,YF,VRX,VRY,XLONGM
1 ,XLONGT,ICOUNT)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
A=2.0926E7
GM=1.4077E16
RIC=SQRT(XIC**2+YIC**2)
RF=SQRT(XF**2+YF**2)
CPHI=(XIC*XF+YIC*YF)/(RIC*RF)
PHI=ACOS(CPHI)
SPHI=SIN(PHI)
R0=RIC
PI=3.14159
DEGRAD=360./(2.*PI)
ICOUNT=1
GMIN=ATAN2((SPHI-SQRT(2.*R0*(1.-CPHI)/RF)),(1-CPHI))
GMAX=ATAN2((SPHI+SQRT(2.*R0*(1.-CPHI)/RF)),(1-CPHI))
GAM=(GMIN+GMAX)/2.
DO
    TOP=GM*(1.-COS(PHI))
    TEMP=R0*COS(GAM)/RF-COS(PHI+GAM)
    BOT=R0*COS(GAM)*TEMP
    V=SQRT(TOP/BOT)
    IF (XLONGT>XLONGM) THEN
        VRX=V*COS(PI/2.-GAM+XLONGM)
        VRY=V*SIN(PI/2.-GAM+XLONGM)
    ELSE
        VRX=V*COS(-PI/2.+GAM+XLONGM)
        VRY=V*SIN(-PI/2.+GAM+XLONGM)
    END IF
    XLAM=R0*V*V/GM
    TOP1=TAN(GAM)*(1-COS(PHI))+(1-XLAM)*SIN(PHI)
    BOT1P=(1-COS(PHI))/(XLAM*COS(GAM)*COS(GAM))

```

(Contd.)

Listing 14.3 (Continued)

```

BOT1=(2-XLAM)*(BOT1P+COS(GAM+PHI)/COS(GAM))
TOP2=2*COS(GAM)
BOT2=XLAM*((2/XLAM-1)**1.5)
TOP3=SQRT(2/XLAM-1)
BOT3=COS(GAM)/TAN(PHI/2)-SIN(GAM)
TEMP=(TOP2/BOT2)*ATAN2(TOP3,BOT3)
TF=R0*(TOP1/BOT1+TEMP)/(V*COS(GAM))
IF((ABS(TFDES-TF)<=.0000001*TFDES).OR.ICOUNT>100)THEN
    EXIT
ENDIF
IF(TF>TFDES)THEN
    GMAX=GAM
ELSE
    GMIN=GAM
ENDIF
IF(ICOUNT.EQ.1)THEN
    XNEXT=(GMAX+GMIN)/2.
ELSE
    XNEXT=GAM+(GAM-GOLD)*(TFDES-TF)/(TF-TOLD)
    IF(XNEXT<GMAX.OR.XNEXT<GMIN)THEN
        XNEXT=(GMAX+GMIN)/2.
    ENDIF
ENDIF
GOLD=GAM
TOLD=TF
GAM=XNEXT
ICOUNT=ICOUNT+1
REPEAT
RETURN
END

```

specific impulse of 300 s. The maximum acceleration in each stage is 20 g. One-third of the speed will be attained in the first stage, and the rest of the speed will be attained in the second stage. Burnout of the second stage will be completed at about 60 s. It is desired that the booster, which is initially at angular location $\theta_0 = 30$ deg ($ANGDEG = 30$), reach a target at 45 deg ($XLONGTDEG = 45$) in 500 s ($TF = 500$).

The Lambert feedback loop is at the end of the integration routine and is called every integration interval. When the difference between the desired velocity and the attained velocity is less than 500 ft/s, the simulation automatically sets the actual velocity to the desired velocity to avoid making the integration interval very small in the simulation. At this time the booster cuts off and coasts. The logic in the simulation is self-explanatory.

The nominal case was run where the inputs were previously explained. Figure 14.6 displays the x component of the achieved velocity along with the desired or Lambert velocity. We can see that the two velocities converge at about 45 s. Figure 14.7 presents the y components of the achieved and desired velocities. We can see that this component is much larger than the x component. The discontinuity in

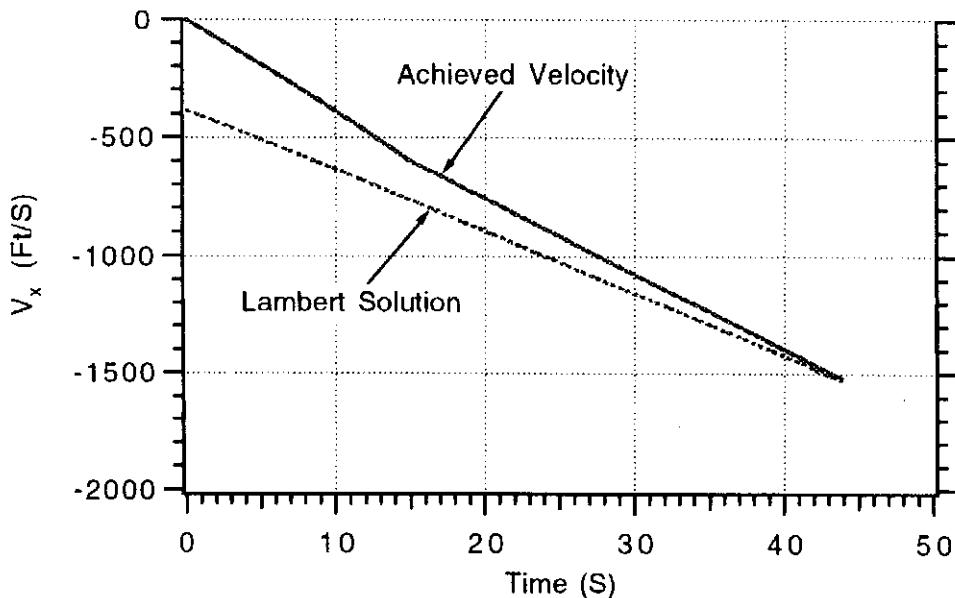


Fig. 14.6 X component of achieved velocity reaches Lambert solution.

the y component at about 15 s is due to staging, and the slight discontinuity near the end of the display is due to setting the achieved velocity to the desired velocity when the velocity to be gained was less than 500 ft/s.

Finally, Fig. 14.8 shows the resultant trajectory. The missile reaches the target at exactly 500 s. We can see from the figure that the trajectory is smooth during the boost phase of flight when Lambert guidance is used.

It is interesting to note that the Lambert solution was reached in about 45 s even though the missile was capable of burning fuel for nearly 60 s. Thus, we can see that Lambert guidance can be used to steer a strategic missile with a thrust termination system in the absence of atmospheric effects. The Lambert guidance principal can be used for interceptors that fly ballistically to hit stationary targets. Lambert guidance can also be used for guided interceptors that must hit moving

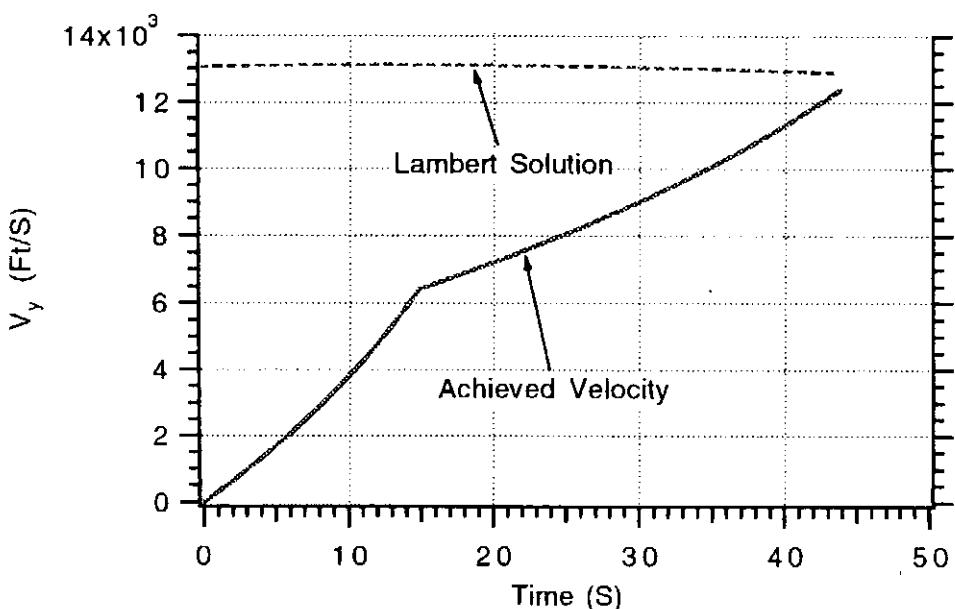


Fig. 14.7 Y component of achieved velocity reaches Lambert solution.

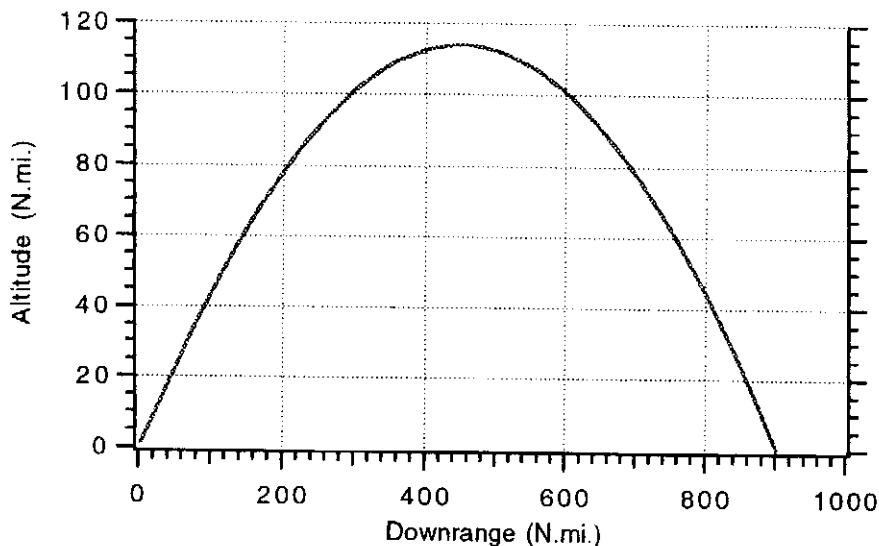


Fig. 14.8 Booster reaches target with Lambert guidance.

and accelerating targets. In this case, the purpose of Lambert guidance is to place the interceptor on a collision triangle at the end of the boost phase.

General Energy Management (GEM) Steering^{2,5}

We have seen in the previous section how it was possible to steer a boosting strategic interceptor to a desired intercept point using Lambert guidance. In the example presented in the last section, the thrust had to be terminated before the end of burn in order to achieve the desired Lambert solution. Often there is a restriction, in the absence of a thrust termination system, that all the booster fuel must be consumed. In this case a method other than Lambert guidance must be employed to waste some of the booster's excess energy. A popular energy wasting technique is known as general energy management (GEM) steering.

To explain the concept of energy wasting, consider the simplified geometry of Fig. 14.9. In this figure we have the arc of a circle whose length is denoted V_{cap} . This arc represents the velocity capability of the booster. The radius of the circle forming the arc is denoted r , and the central angle is denoted 2θ . A chord is drawn connecting both ends of the arc. The chord length represents the velocity to be gained (subtraction of achieved velocity from Lambert solution velocity) and is

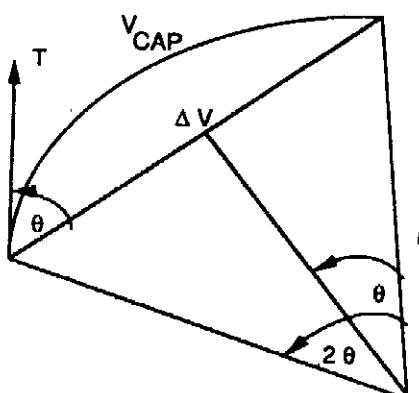


Fig. 14.9 Basic angles in general energy management.

denoted ΔV . If the thrust vector is drawn tangent to the chord at the beginning of the arc, it is easy to show from geometry that the thrust vector is at an angle of θ with respect to the chord. Finally, a perpendicular is dropped from the chord to the center of the circle. It is also easy to show that the perpendicular bisects the chord and the central angle.

From Fig. 14.9 we can see that the arc length is related to the central angle according to

$$V_{\text{cap}} = 2\theta r$$

Since the perpendicular bisects the central angle, we can also say that

$$\Delta V = 2r \sin \theta$$

Therefore, we can ratio the two velocity expressions, yielding

$$\frac{\Delta V}{V_{\text{cap}}} = \frac{2r \sin \theta}{2r\theta} = \frac{\sin \theta}{\theta}$$

Expanding the sine term into a two-term Taylor series leads to

$$\frac{\Delta V}{V_{\text{cap}}} = \left(\theta - \frac{\theta^3}{6} \right) / \theta = 1 - \frac{\theta^2}{6}$$

Solving for the angle yields

$$\theta = \sqrt{6 \left(1 - \frac{\Delta V}{V_{\text{cap}}} \right)}$$

The formula suggests that if, at each instant of time, we ensure that the thrust vector is at an angle of θ with respect to the velocity to be gained vector, then we can still achieve the Lambert solution at the end of burn and hit the target.

Figure 14.10 shows the proper relationship between the thrust and velocity to be gained vectors relative to the inertial Earth-centered coordinate system. We can see that for counterclockwise travel the components of the thrust acceleration are given by

$$a_{XT} = a_T \cos(\phi - \theta)$$

$$a_{YT} = a_T \sin(\phi - \theta)$$

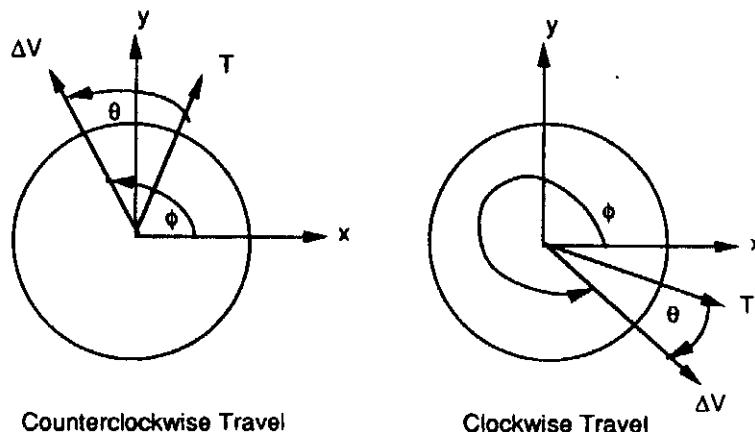


Fig. 14.10 Sign conventions for GEM.

where θ is the angle between the thrust vector and the velocity to be gained vector, and ϕ is the angle between the velocity to be gained vector and the x axis. For clockwise travel the thrust acceleration components become

$$a_{XT} = a_T \cos(\phi + \theta)$$

$$a_{YT} = a_T \sin(\phi + \theta)$$

Listing 14.4 presents a simulation of a booster intercepting a ground target using general energy management guidance. This simulation and the nominal operating conditions are identical to that of Listing 14.3 except for the GEM logic after the integration routine. We can see from the listing that the axial acceleration capability of the booster is continually being computed according to

$$V_{cap} = V_{cap} - Ha_T$$

where H is the integration step size and a_T the instantaneous axial acceleration of the booster. To avoid numerical problems, the GEM logic is terminated when the velocity to be gained drops below 50 ft/s. We can see from the listing that it is still necessary to use the Lambert subroutine in order to implement the general energy management guidance technique.

A nominal case was run to see how the GEM guidance logic performed. We can see from Fig. 14.11 that, although the booster burn lasts for nearly 60 s, the angle the thrust vector makes with respect to the velocity to be gained vector approaches steady state in slightly over 50 s. Also shown in Fig. 14.11 is a plot of how the velocity capability of the booster diminishes during the burn. The discontinuity in that curve is due to staging.

Figure 14.12 displays the GEM trajectory during the boost phase. It appears from the figure that the booster will never hit the target because it initially appears to be heading in the wrong direction. However, after wasting energy, the GEM-guided booster heads in the right direction. Superimposed on the figure is the Lambert guidance trajectory during boost for the same case. We can see that both trajectories are vastly different during the boost phase.

Figure 14.13 displays the GEM and Lambert trajectories for the entire flight (boost and coast phases). We can see from the figure that, although both trajectories

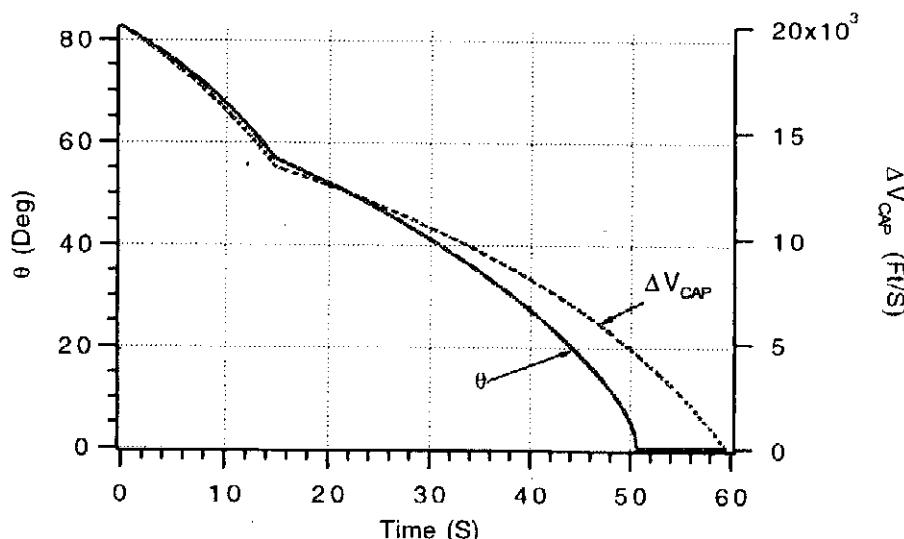


Fig. 14.11 GEM angle reaches steady state quickly.

Listing 14.4 General energy management simulation

```

IMPLICIT REAL*8(A-H)
IMPLICIT REAL*8(O-Z)
INTEGER STEP
LOGICAL LEFT,QBOOST,QZERO
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
OPEN(2,STATUS='UNKNOWN',FILE='SPECFIL')
LEFT=.TRUE.
QBOOST=.TRUE.
QZERO=.FALSE.
XISP1=300.
XISP2=300.
XMF1=.90
XMF2=.90
WPAY=100.
DELV=20000.
DELV1=.3333*DELV
DELV2=.6667*DELV
AMAX1=20.
AMAX2=20.
TOP2=WPAY*(EXP(DELV2/(XISP2*32.2))-1.)
BOT2=1/XMF2-((1.-XMF2)/XMF2)*EXP(DELV2/(XISP2*32.2))
WP2=TOP2/BOT2
WS2=WP2*(1-XMF2)/XMF2
WTOT2=WP2+WS2+WPAY
TRST2=AMAX2*(WPAY+WS2)
TB2=XISP2*WP2/TRST2
TOP1=WTOT2*(EXP(DELV1/(XISP1*32.2))-1.)
BOT1=1/XMF1-((1.-XMF1)/XMF1)*EXP(DELV1/(XISP1*32.2))
WP1=TOP1/BOT1
WS1=WP1*(1-XMF1)/XMF1
WTOT=WP1+WS1+WTOT2
TRST1=AMAX1*(WTOT2+WS1)
TB1=XISP1*WP1/TRST1
DELVK=DELV/1000.
H=.01
T=0.
S=0.
A=2.0926E7
GM=1.4077E16
ALTNM=0.
ALT=ALTNM*6076
ANGDEG=30.
ANG=ANGDEG/57.3
XLONGM=ANG
X=(A+ALT)*COS(ANG)
Y=(A+ALT)*SIN(ANG)
ALT=SQRT(X**2+Y**2)-A
X1=0.
Y1=0.
AXT=0.

```

(Contd.)

Listing 14.4 (Continued)

```

AYT=0.
XLONGTDEG=45.
XLONGT=XLONGTDEG/57.3
XF=A*COS(XLONGT)
YF=A*SIN(XLONGT)
XFIRST=XF
YFIRST=YF
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTINITNM)
TF=500.
DVCAP=DELV
10 IF(ALT<0..AND.T>10.)GOTO 999
XOLD=X
YOLD=Y
X1OLD=X1
Y1OLD=Y1
STEP=1
GOTO 200
66 STEP=2
X=X+H*XD
Y=Y+H*YD
X1=X1+H*X1D
Y1=Y1+H*Y1D
T=T+H
GOTO 200
55 CONTINUE
X=(XOLD+X)/2+.5*H*XD
Y=(YOLD+Y)/2+.5*H*YD
X1=(X1OLD+X1)/2+.5*H*X1D
Y1=(Y1OLD+Y1)/2+.5*H*Y1D
ALT=SQRT(X**2+Y**2)-A
S=S+H
TGOLAM=TF-T
DVCAP=DVCAP-H*AT
IF(QBOOST.AND.DVCAP>50.)THEN
XLONGM=ATAN2(Y,X)
CALL LAMBERT(X,Y,TGOLAM,XF,YF,VRX,VRY,XLONGM,XLONGT
,ICOUNT)
DELX=VRX-X1
DELY=VRY-Y1
DEL=SQRT(DELX**2+DELY**2)
IF(.NOT.QZERO.AND.DVCAP>DEL)THEN
THET=SQRT(6.*((1.-DEL/DVCAP)))
DEGTHET=57.3*THET
ELSE
QZERO=.TRUE.
ENDIF
PHI=ATAN2(DELY,DELX)
DEGPHI=57.3*PHI
IF(XLONGT>XLONGM)THEN
AXT=AT*COS(PHI-THET)

```

(Contd.)

Listing 14.4 (Continued)

```

AYT=AT*SIN(PHI-THET)
ELSE
AXT=AT*COS(PHI+THET)
AYT=AT*SIN(PHI+THET)
ENDIF
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTNM)
DISTNM=DISTINITNM-DISTNM
ALTNM=(SQRT(X**2+Y**2)-A)/6076.
WRITE(2,97)T,DISTNM,ALTNM,THET*57.3,DVCAP
97 FORMAT(5F10.3)
ELSEIF(QBOOST)THEN
CALL LAMBERT(X,Y,TGOLAM,XF,YF,VRX,VRY,XLONGM
1 ,XLONGT,ICOUNT)
TRST=0.
QBOOST=.FALSE.
AXT=0.
AYT=0.
X1=VRX
Y1=VRY
X1OLD=X1
Y1OLD=Y1
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTNM)
DISTNM=DISTINITNM-DISTNM
ALTNM=(SQRT(X**2+Y**2)-A)/6076.
WRITE(2,*)T,DISTNM,ALTNM,THET*57.3,DVCAP
ELSE
QBOOST=.FALSE.
AXT=0.
AYT=0.
ENDIF
IF(S.LT.9.99999)GOTO 10
S=0.
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTNM)
DISTNM=DISTINITNM-DISTNM
ALTNM=(SQRT(X**2+Y**2)-A)/6076.
VELK=SQRT(X1**2+Y1**2)/1000.
XNM=X/6076.
YNM=Y/6076.
WRITE(*,98)T,DISTNM,ALTNM
WRITE(1,98)T,DISTNM,ALTNM
GOTO 10
98 FORMAT(3F10.3)
200 CONTINUE
IF(T<TB1)THEN
WGT=-WP1*T/TB1+WTOT
TRST=TRST1
ELSEIF(T<(TB1+TB2))THEN
WGT=-WP2*T/TB2+WTOT2+WP2*TB1/TB2
TRST=TRST2
ELSE

```

(Contd.)

Listing 14.4 (Continued)

```

WGT=WPAY
TRST=0.
ENDIF
AT=32.2*TRST/WGT
XD=X1
YD=Y1
TEMBOT=(X**2+Y**2)**1.5
X1D=-GM*X/TEMBOT+AXT
Y1D=-GM*Y/TEMBOT+AYT
IF(STEP-1)66,66,55
999 CONTINUE
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTNM)
DISTNM=DISTINITNM-DISTNM
ALTNM=(SQRT(X**2+Y**2)-A)/6076.
VELK=SQRT(X1**2+Y1**2)/1000.
WRITE(*,99)T,DISTNM,ALTNM,VELK
WRITE(1,98)T,DISTNM,ALTNM
99 FORMAT(4F10.3)
PAUSE
CLOSE(1)
END

SUBROUTINE DISTANCE(XT,YT,XF,YF,DISTNM)
SAVE
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
R=SQRT(XT**2+YT**2)
A=2.0926E7
CBETA=(XT*XF+YT*YF)/(R*A)
IF(CBETA<1.)THEN
BETA=ACOS(CBETA)
DISTNM=A*BETA/6076.
ELSE
DISTNM=(XF-XT)/6076.
ENDIF
RETURN
END

SUBROUTINE LAMBERT(XIC,YIC,TFDES,XF,YF,VRX,VRY,XLONGM
1 ,XLONGT,ICOUNT)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
A=2.0926E7
GM=1.4077E16
RIC=SQRT(XIC**2+YIC**2)
RF=SQRT(XF**2+YF**2)
CPHI=(XIC*XF+YIC*YF)/(RIC*RF)
PHI=ACOS(CPHI)
SPHI=SIN(PHI)
R0=RIC

```

Listing 14.4 (Continued)

```

PI=3.14159
DEGRAD=360./(2.*PI)
ICOUNT=1
GMIN=ATAN2((SPHI-SQRT(2.*R0*(1.-CPHI)/RF)),(1-CPHI))
GMAX=ATAN2((SPHI+SQRT(2.*R0*(1.-CPHI)/RF)),(1-CPHI))
GAM=(GMIN+GMAX)/2.
DO
TOP=GM*(1.-COS(PHI))
TEMP=R0*COS(GAM)/RF-COS(PHI+GAM)
BOT=R0*COS(GAM)*TEMP
V=SQRT(TOP/BOT)
IF (XLONGT>XLONGM) THEN
VRX=V*COS(PI/2.-GAM+XLONGM)
VRY=V*SIN(PI/2.-GAM+XLONGM)
ELSE
VRX=V*COS(-PI/2.+GAM+XLONGM)
VRY=V*SIN(-PI/2.+GAM+XLONGM)
END IF
XLAM=R0*V*V/GM
TOP1=TAN(GAM)*(1-COS(PHI))+(1-XLAM)*SIN(PHI)
BOT1P=(1-COS(PHI))/(XLAM*COS(GAM)*COS(GAM))
BOT1=(2-XLAM)*(BOT1P+COS(GAM+PHI)/COS(GAM))
TOP2=2*COS(GAM)
BOT2=XLAM*((2/XLAM-1)**1.5)
TOP3=SQRT(2/XLAM-1)
BOT3=COS(GAM)/TAN(PHI/2)-SIN(GAM)
TEMP=(TOP2/BOT2)*ATAN2(TOP3,BOT3)
TF=R0*(TOP1/BOT1+TEMP)/(V*COS(GAM))
IF((ABS(TFDES-TF)<=.00000001*TFDES).OR.ICOUNT>100)THEN
EXIT
ENDIF
IF(TF>TFDES)THEN
GMAX=GAM
ELSE
GMIN=GAM
ENDIF
IF(ICOUNT.EQ.1)THEN
XNEXT=(GMAX+GMIN)/2.
ELSE
XNEXT=GAM+(GAM-GOLD)*(TFDES-TF)/(TF-TOLD)
IF (XNEXT>GMAX.OR.XNEXT<GMIN)THEN
XNEXT=(GMAX+GMIN)/2.
ENDIF
ENDIF
GOLD=GAM
TOLD=TF
GAM=XNEXT
ICOUNT=ICOUNT+1
REPEAT
RETURN
END

```

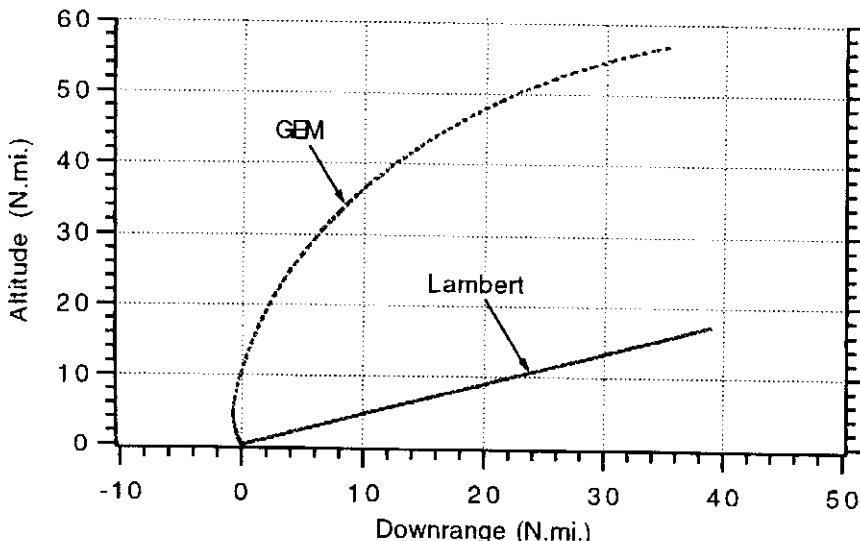


Fig. 14.12 Lambert and GEM trajectories during boost phase are vastly different.

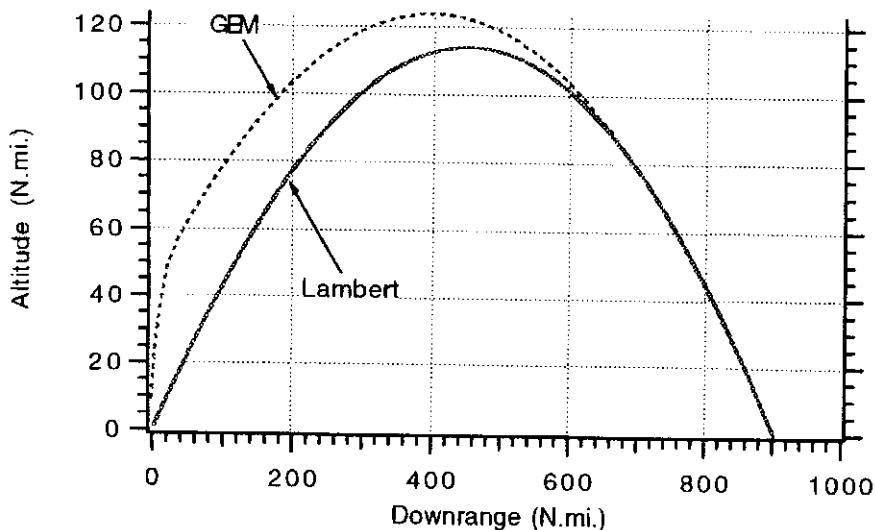


Fig. 14.13 Both Lambert and GEM trajectories hit target at the same time.

are vastly different during the boost phase, they eventually converge, and both hit the target at the same time.

Summary

In this chapter the Lambert problem was explained, and a novel numerical technique for solving the problem, based on the closed-form solutions of Chapter 11, was introduced. Two techniques were presented showing how the solution to Lambert's problem was fundamental to steering boosters. Numerical examples were presented illustrating the implementation and effectiveness of the booster steering techniques.

References

¹Battin, R. H., *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA Education Series, New York, 1987.

- ²Regan, F., *Re-Entry Vehicle Dynamics*, AIAA Education Series, New York, 1984.
- ³Acton, F. S., *Numerical Methods That Work*, Harper and Row, New York, 1970.
- ⁴Nelson, S. L., and Zarchan, P., "Alternative Approach to the Solution of Lambert's Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 15, July–Aug. 1992, pp. 1003–1009.
- ⁵Brand, T. J., "A New Approach to Lambert Guidance," Charles Stark Draper Lab., Rept. R-694, Cambridge, MA, June 1971.

Strategic Intercepts

Introduction

GUIDANCE concepts for tactical homing missiles were introduced, explained, and demonstrated in Chapters 2, 6, and 8. Strategic missiles travel much faster and farther than tactical missiles. We have shown that the coordinate system and gravity models in our tactical simulations had to be modified to handle the new speed and range regimes of strategic missiles. More specifically, we had to shift our coordinate system from the surface of a flat Earth to the center of a round Earth and use a more general formulation for gravitational acceleration (i.e., Newton's law of universal gravitation).

Since tactical missiles operate within the atmosphere, they can generate lift by moving control surfaces in order to execute guidance commands. Speed, altitude, and structural considerations limit maximum achievable acceleration levels with tactical missiles. Missile slowdown due to drag limits the tactical missile's range and maneuver capability. Strategic missiles, on the other hand, operate outside the atmosphere and must burn fuel (i.e., lateral thrusters) to respond to guidance commands. Achievable engine thrust-to-weight ratios limit maximum strategic lateral acceleration levels. In addition, when the maneuver or divert fuel is exhausted, the strategic missile cannot maneuver at all. Care must be taken to ensure that a strategic missile has sufficient divert fuel so that it can meet system objectives.

Although there are major differences between strategic and tactical missiles, there are also similarities. This chapter will show that tactical guidance laws may be suitable for strategic missiles. Useful design relationships, developed previously in the text for tactical missiles, will be modified and shown to be applicable for strategic missiles as well.

Guidance Review

In Chapter 2 we saw the effectiveness of the proportional navigation guidance law for tactical missiles. A closed-form solution for the required missile acceleration to hit a target, in the presence of heading error, was derived for a zero-lag guidance system. The required missile acceleration was shown to be

$$n_c = \frac{-V_M HE N'}{t_F} \left(1 - \frac{t}{t_F}\right)^{N'-2}$$

where V_M is the missile velocity, HE is the angular heading error, N' is the effective navigation ratio, t_F is the flight time, and t is instantaneous time.

With strategic missiles it is often more convenient to talk in terms of prediction error rather than heading error. A prelaunch calculation or prediction must be made of where the target will be at intercept. This location is known as the predicted intercept point. If the calculation is imperfect, a prediction error results, and the missile will not be fired on a perfect collision triangle. The prediction error and heading error are related by

$$\text{Pred Err} = -V_M H E t_F$$

where *Pred Err* is the prediction error in units of feet. Therefore, substitution of the preceding relationship into the closed-form solution indicates that the missile acceleration required by the proportional navigation guidance law to take out an initial prediction error is given by

$$n_c = \frac{\text{Pred Err } N'}{t_F^2} \left(1 - \frac{t}{t_F}\right)^{N'-2}$$

As was mentioned previously, strategic missiles burn fuel to maneuver. The amount of lateral divert or ΔV required is related to the missile acceleration according to

$$\Delta V = \int_0^{t_F} |n_c| dt$$

The strategic interceptor ΔV requirements are related to the total interceptor weight by the rocket equation. Increasing a missile's divert requirements can increase the total weight requirements dramatically. We can find a closed-form solution for the required divert to take out a prediction error by substituting the closed-form solution for the missile acceleration into the preceding integral. After some algebra we obtain

$$\Delta V = \frac{\text{Pred Err } N'}{(N' - 1)t_F}$$

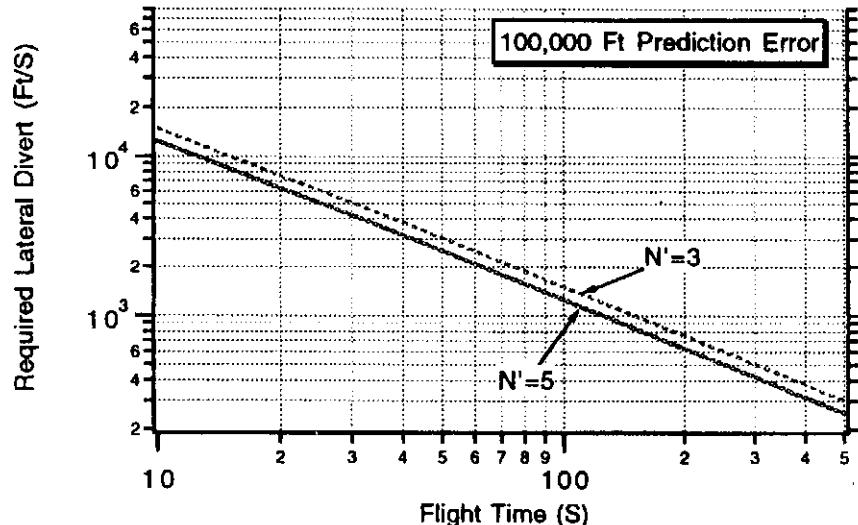


Fig. 15.1 Lateral divert requirements decrease with increasing flight time.

Thus, increasing the effective navigation ratio or increasing the flight time (or guidance time) will tend to reduce the lateral divert requirements of the interceptor. The preceding formula for the required lateral divert is plotted in Fig. 15.1 for the case in which the prediction error is 100 kft. We can see that, for an effective navigation ratio of 3, it takes more than 10,000 ft/s of divert to remove the error in 10 s, about 1050 ft/s of divert to remove the error in 100 s, and only about 300 ft/s of divert to remove the error in 500 s. Therefore, larger missile acquisition ranges result in larger guided flight times, which in turn can reduce the lateral divert requirements for a given prediction error. Increasing the effective navigation ratio to 5 only slightly reduces the lateral divert requirements. Doubling the prediction error will double the divert requirements.

Ballistic Engagement Simulation¹

We can develop a strategic ballistic missile-target engagement simulation by using an Earth-centered coordinate system as shown in Fig. 15.2. In this figure both the missile and target are in a gravity field as described by Newton's law of universal gravitation. The acceleration differential equations acting on a ballistic target were shown in Chapter 12 to be

$$\ddot{x}_T = \frac{-gm x_T}{(x_T^2 + y_T^2)^{1.5}}$$

$$\ddot{y}_T = \frac{-gm y_T}{(x_T^2 + y_T^2)^{1.5}}$$

where gm is the gravitational parameter. These differential equations are in an inertial coordinate system whose origin is at the center of the Earth. Therefore, they can be integrated directly to yield the velocity and position of the target with respect to the center of the Earth. The components of the relative position between

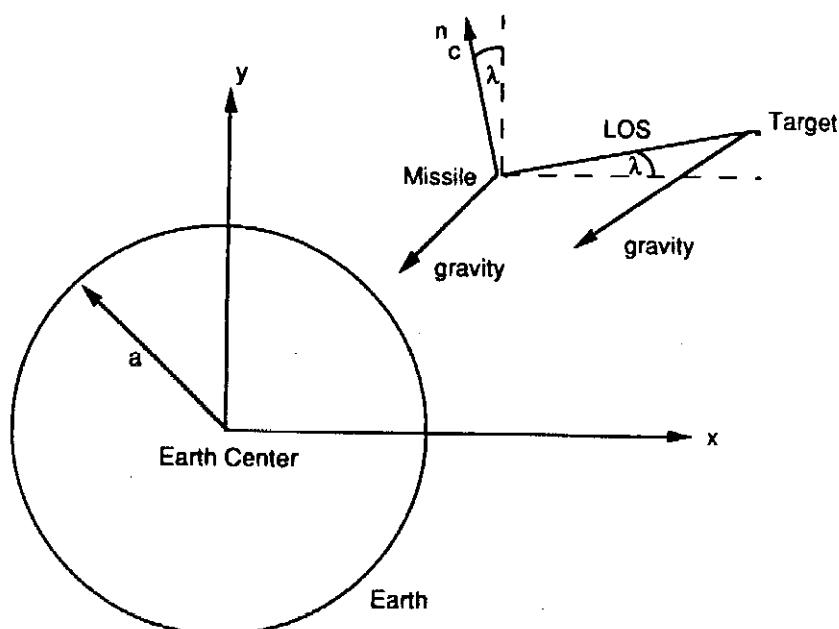


Fig. 15.2 Earth-centered coordinate system and relative engagement geometry.

the missile and target can be expressed as

$$R_{TM1} = x_T - x_M$$

$$R_{TM2} = y_T - y_M$$

and the components of the relative velocity are given by

$$V_{TM1} = \dot{x}_T - \dot{x}_M$$

$$V_{TM2} = \dot{y}_T - \dot{y}_M$$

Application of the distance formula shows that the relative separation between the missile and target can be found from

$$R_{TM} = (R_{TM1}^2 + R_{TM2}^2)^{0.5}$$

The closing velocity, which is defined as the negative rate of change of separation between missile and target, can be obtained by taking the negative derivative of the preceding expression, yielding

$$V_c = \frac{-(R_{TM1} V_{TM1} + R_{TM2} V_{TM2})}{R_{TM}}$$

The line-of-sight angle can be found by trigonometry from Fig. 15.2 as

$$\lambda = \tan^{-1} \frac{R_{TM2}}{R_{TM1}}$$

Therefore, the instantaneous value of the line-of-sight rate can be found by taking the derivative of the preceding expression, using the quotient rule, yielding

$$\dot{\lambda} = \frac{R_{TM1} V_{TM2} - R_{TM2} V_{TM1}}{R_{TM}^2}$$

We now have sufficient information to guide a strategic interceptor. The proportional navigation guidance command is proportional to the line-of-sight rate according to

$$n_c = N' V_c \dot{\lambda}$$

where N' is the effective navigation ratio and V_c the closing velocity. This guidance command is perpendicular to the line of sight. From Fig. 15.2 we can see that the components of the guidance command in the Earth-centered coordinate system can be found by trigonometry and are given by

$$a_{XM} = -n_c \sin \lambda$$

$$a_{YM} = n_c \cos \lambda$$

Therefore, the acceleration differential equations describing the missile consist of two parts: the gravitational term and the guidance command term. The

components of the missile differential equations in Earth-centered coordinates are

$$\ddot{x}_M = \frac{-gm x_M}{(x_M^2 + y_M^2)^{1.5}} + a_{XM}$$

$$\ddot{y}_M = \frac{-gm y_M}{(x_M^2 + y_M^2)^{1.5}} + a_{YM}$$

where a_{XM} and a_{YM} have already been defined.

Listing 15.1 presents a FORTRAN simulation of an engagement between an impulsive missile and a ballistic target. The simulation, which is based on the differential equations derived in this section, is similar to the tactical engagement simulations presented in the text, except that the coordinate system is Earth-centered and calculations now have to be performed using double-precision arithmetic (i.e., REAL*8). The reason for double-precision arithmetic is that distances are now very large since they are referenced with respect to the center of the Earth. Relative quantities, which are required for guidance command and miss distance computation, involve the subtraction of large numbers. Near the end of the flight the large numbers are nearly equal and single-precision arithmetic does not offer sufficient accuracy.

The program includes a prediction routine (subroutine PREDICT) to determine where the target will be at the intercept time t_F . Before the main simulation begins, this routine integrates the ballistic target equations forward in time to determine the location of the target at time t_F (i.e., predicted intercept point). The Lambert routine (subroutine LAMBERT) determines the velocity components of an impulsive strategic interceptor ($VRXM$ and $VRYM$) so that it will be on a collision triangle with the ballistic target. In other words, given an initial location, a final location, and an arrival time, the Lambert subroutine determines the correct missile velocity components so that it will collide with the target at time t_F . If the predicted intercept point is correct, then no guidance system is required for the strategic interceptor to collide with the target.

We can see from the listing that the guidance equations are virtually identical to those of the two-dimensional tactical simulation of Chapter 2. This is not surprising since the proportional navigation guidance law operates on relative quantities that should be independent of coordinate system. The differential equations for the missile and target and the guidance equations appear after statement label 200.

A nominal case was run in which the guidance system was turned off ($XNC = 0$). The resultant trajectory for the 500-s flight is shown in Fig. 15.3. In this case the missile hit the target. This means that our knowledge of the intercept point was perfect (from the prediction subroutine) and that the missile was placed on the correct collision triangle (from Lambert subroutine). The slight curvature in both missile and target trajectories is due to the fact that both objects are in a gravity field for 500 s.

The same nominal case was rerun, except this time the proportional navigation guidance system was turned on. The resultant commanded acceleration profile, which resulted in a successful intercept, along with the missile lateral divert requirements appear in Fig. 15.4. We can see from the figure that, even though the missile was initially on a collision triangle, the proportional navigation guidance system issued acceleration commands. In this case it appears that proportional

Listing 15.1 Engagement simulation with ballistic target

```

IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
XLONGMDEG=45.
XLONGTDEG=90.
ALTNMTIC=0.
ALTNMMIC=0.
TF=500.
GAMDEGT=23.
H=.01
A=2.0926E7
GM=1.4077E16
PI=3.14159
DEGRAD=360./(2.*PI)
XNP=3.
PREDERR=0.
GAMT=GAMDEGT/57.3
DISTNMT=6000.
PHIT=DISTNMT*6076./A
ALTT=ALTNMTIC*6076.
ALTM=ALTNMMIC*6076.
R0T=A+ALTT
TOP=GM*(1.-COS(PHIT))
TEMP=R0T*COS(GAMT)/A-COS(PHIT+GAMT)
BOT=R0T*COS(GAMT)*TEMP
VT=SQRT(TOP/BOT)
XLONGM=XLONGMDEG/DEGRAD
XLONGT=XLONGTDEG/DEGRAD
IF (XLONGM>XLONGT) THEN
    X1T=VT*COS(PI/2.-GAMT+XLONGT)
    Y1T=VT*SIN(PI/2.-GAMT+XLONGT)
ELSE
    X1T=VT*COS(-PI/2.+GAMT+XLONGT)
    Y1T=VT*SIN(-PI/2.+GAMT+XLONGT)
END IF
S=0.
SCOUNT=0.
XLONGM=XLONGMDEG/DEGRAD
XLONGT=XLONGTDEG/DEGRAD
XM=(A+ALTM)*COS(XLONGM)
YM=(A+ALTM)*SIN(XLONGM)
XT=(A+ALTT)*COS(XLONGT)
YT=(A+ALTT)*SIN(XLONGT)
XFIRSTT=XT
YFIRSTT=YT
T=0.
CALL PREDICT (TF,XT,YT,X1T,Y1T,XTF,YTF)
YTF=YTF+PREDERR

```

(Contd.)

Listing 15.1 (Continued)

```

CALL LAMBERT(XM,YM,TF,XTF,YTF,VRXM,VRYM,XLONGM,XLONGT
,ICOUNT)
X1M=VRXM
Y1M=VRYM
RTM1=XT-XM
RTM2=YT-YM
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=X1T-X1M
VTM2=Y1T-Y1M
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
DELV=0.
10 IF(VC<0.)GOTO 999
TGO=RTM/VC
IF(TGO>.1)THEN
    H=.01
ELSE
    H=.0001
ENDIF
XOLDT=XT
YOLDT=YT
X1OLDT=X1T
Y1OLDT=Y1T
XOLDM=XM
YOLDM=YM
X1OLDM=X1M
Y1OLDM=Y1M
DELVOLD=DELV
STEP=1
GOTO 200
66 STEP=2
XT=XT+H*XDT
YT=YT+H*YDT
X1T=X1T+H*X1DT
Y1T=Y1T+H*Y1DT
XM=XM+H*XDM
YM=YM+H*YDM
X1M=X1M+H*X1DM
Y1M=Y1M+H*Y1DM
DELV=DELV+H*DELVD
T=T+H
GOTO 200
55 XT=(XOLDT+XT)/2+.5*H*XDT
YT=(YOLDT+YT)/2+.5*H*YDT
X1T=(X1OLDT+X1T)/2+.5*H*X1DT
Y1T=(Y1OLDT+Y1T)/2+.5*H*Y1DT
XM=(XOLDM+XM)/2+.5*H*XDM
YM=(YOLDM+YM)/2+.5*H*YDM
X1M=(X1OLDM+X1M)/2+.5*H*X1DM
Y1M=(Y1OLDM+Y1M)/2+.5*H*Y1DM

```

(Contd.)

Listing 15.1 (Continued)

```

DELV=(DELVOLD+DELV)/2.+.5*H*DELVD
ALTT=SQRT(XT**2+YT**2)-A
ALTM=SQRT(XM**2+YM**2)-A
S=S+H
SCOUNT=SCOUNT+H
IF(SCOUNT.LT..99999)GOTO 10
SCOUNT=0.
XNMT=XT/6076.
YNMT=YT/6076.
XNMM=XM/6076.
YNMM=YM/6076.
ALTNMT=ALTT/6076.
CALL DISTANCE(XT,YT,XFIRSTT,YFIRSTT,DISTNMT)
ALTNMM=ALTM/6076.
XNCG=XNC/32.2
CALL DISTANCE(XM,YM,XFIRSTT,YFIRSTT,DISTNMM)
WRITE(*,97)T,DISTNMT,ALTNMT,DISTNMM,ALTNMM,XNCG,DELV
WRITE(1,97)T,DISTNMT,ALTNMT,DISTNMM,ALTNMM,XNCG,DELV
GOTO 10
97 FORMAT(7F10.4)
200 CONTINUE
TEMBOOTT=(XT**2+YT**2)**1.5
X1DT=-GM*XT/TEMBOOTT
Y1DT=-GM*YT/TEMBOOTT
XDT=X1T
YDT=Y1T
RTM1=XT-XM
RTM2=YT-YM
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=X1T-X1M
VTM2=Y1T-Y1M
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
TGO=RTM/VC
XLAM=ATAN2(RTM2,RTM1)
XLAMD=(RTM1*VTM2-RTM2*VTM1)/(RTM*RTM)
XNC=XNP*VC*XLAMD
DELVD=ABS(XNC)
AM1=-XNC*SIN(XLAM)
AM2=XNC*COS(XLAM)
TEMBOTM=(XM**2+YM**2)**1.5
X1DM=-GM*XM/TEMBOTM+AM1
Y1DM=-GM*YM/TEMBOTM+AM2
XDM=X1M
YDM=Y1M
IF(STEP-1)66,66,55
999 CONTINUE
XNMT=XT/6076.
YNMT=YT/6076.
XNMM=XM/6076.

```

(Contd.)

Listing 15.1 (Continued)

```

YNMM=YM/6076.
ALTNMT=ALTT/6076.
CALL DISTANCE(XT,YT,XFIRSTT,YFIRSTT,DISTNMT)
ALTNMM=ALTM/6076.
XNCG=XNC/32.2
CALL DISTANCE(XM,YM,XFIRSTT,YFIRSTT,DISTNMM)
WRITE(*,97)T,DISTNMT,ALTNMT,DISTNMM,ALTNMM,XNCG,DELV
WRITE(1,97)T,DISTNMT,ALTNMT,DISTNMM,ALTNMM,XNCG,DELV
WRITE(*,*)T,RTM,DELV
PAUSE
END

SUBROUTINE DISTANCE(XT,YT,XF,YF,DISTNM)
SAVE
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
R=SQRT(XT**2+YT**2)
RF=SQRT(XF**2+YF**2)
A=2.0926E7
CBETA=(XT*XF+YT*YF)/(R*RF)
IF(CBETA<1.)THEN
    BETA=ACOS(CBETA)
    DISTNM=A*BETA/6076.
ELSE
    DISTNM=(XT-XF)/6076.
ENDIF
RETURN
END

SUBROUTINE PREDICT (TF,XDUM,YDUM,X1DUM,Y1DUM,XTF,YTF)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER STEP
SAVE
H=.01
A=2.0926E7
GM=1.4077E16
T=0.
X=XDUM
Y=YDUM
X1=X1DUM
Y1=Y1DUM
10 IF(T>(TF-.00001))GOTO 999
XOLD=X
YOLD=Y
X1OLD=X1
Y1OLD=Y1
STEP=1
GOTO 200

```

(Contd.)

Listing 15.1 (Continued)

```

66 STEP=2
X=X+H*XD
Y=Y+H*YD
X1=X1+H*X1D
Y1=Y1+H*Y1D
T=T+H
GOTO 200
55 X=(XOLD+X)/2+.5*H*XD
Y=(YOLD+Y)/2+.5*H*YD
X1=(X1OLD+X1)/2+.5*H*X1D
Y1=(Y1OLD+Y1)/2+.5*H*Y1D
GOTO 10
200 CONTINUE
TEMBOT=(X**2+Y**2)**1.5
X1D=-GM*X/TEMBOT
Y1D=-GM*Y/TEMBOT
XD=X1
YD=Y1
IF(STEP-1)66,66,55
999 CONTINUE
XTF=X
YTF=Y
RETURN
END

SUBROUTINE LAMBERT(XIC,YIC,TFDES,XF,YF,VRX,VRY,XLONGM
1 ,XLONGT,ICOUNT)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
A=2.0926E7
GM=1.4077E16
RIC=SQRT(XIC**2+YIC**2)
RF=SQRT(XF**2+YF**2)
CPHI=(XIC*XF+YIC*YF)/(RIC*RF)
PHI=ACOS(CPHI)
SPHI=SIN(PHI)
R0=RIC
PI=3.14159
DEGRAD=360./(2.*PI)
ICOUNT=1
GMIN=ATAN2((SPHI-SQRT(2.*R0*(1.-CPHI)/RF));(1-CPHI))
GMAX=ATAN2((SPHI+SQRT(2.*R0*(1.-CPHI)/RF)),(1-CPHI))
GAM=(GMIN+GMAX)/2.
DO
    TOP=GM*(1.-COS(PHI))
    TEMP=R0*COS(GAM)/RF-COS(PHI+GAM)
    BOT=R0*COS(GAM)*TEMP
    V=SQRT(TOP/BOT)
    IF (XLONGT>XLONGM) THEN

```

(Contd.)

Listing 15.1 (Continued)

```

VRX=V*COS(PI/2.-GAM+XLONGM)
VRY=V*SIN(PI/2.-GAM+XLONGM)
ELSE
    VRX=V*COS(-PI/2.+GAM+XLONGM)
    VRY=V*SIN(-PI/2.+GAM+XLONGM)
END IF
XLAM=R0*V*V/GM
TOP1=TAN(GAM)*(1-COS(PHI))+(1-XLAM)*SIN(PHI)
BOT1P=(1-COS(PHI))/(XLAM*COS(GAM)*COS(GAM))
BOT1=(2-XLAM)*(BOT1P+COS(GAM+PHI)/COS(GAM))
TOP2=2*COS(GAM)
BOT2=XLAM*((2/XLAM-1)**1.5)
TOP3=SQRT(2/XLAM-1)
BOT3=COS(GAM)/TAN(PHI/2)-SIN(GAM)
TEMP=(TOP2/BOT2)*ATAN2(TOP3,BOT3)
TF=R0*(TOP1/BOT1+TEMP)/(V*COS(GAM))
IF((ABS(TFDES-TF)<=.00000001*TFDES).OR.ICOUNT>100)THEN
    EXIT
ENDIF
IF(TF>TFDES)THEN
    GMAX=GAM
ELSE
    GMIN=GAM
ENDIF
IF(ICOUNT.EQ.1)THEN
    XNEXT=(GMAX+GMIN)/2.
ELSE
    XNEXT=GAM+(GAM-GOLD)*(TFDES-TF)/(TF-TOLD)
    IF(XNEXT>GMAX.OR.XNEXT<GMIN)THEN
        XNEXT=(GMAX+GMIN)/2
    ENDIF
ENDIF
GOLD=GAM
TOLD=TF
GAM=XNEXT
ICOUNT=ICOUNT+1
REPEAT
RETURN
END

```

navigation is behaving in a counterintuitive way since we could have hit the target without any acceleration commands at all!

To understand why guidance commands were required of a missile on a collision triangle, let us review some basics. Consider the case of a constant-velocity missile and constant-velocity target on a collision triangle as shown in Fig. 15.5. If we connect lines between the missile and target at different times during the flight, we have a measure of how the line-of-sight rate changes with time. We can see from Fig. 15.5 that, when both the missile and target are traveling at constant velocities, the line-of-sight lines are parallel. In other words, the *line-of-sight rate*

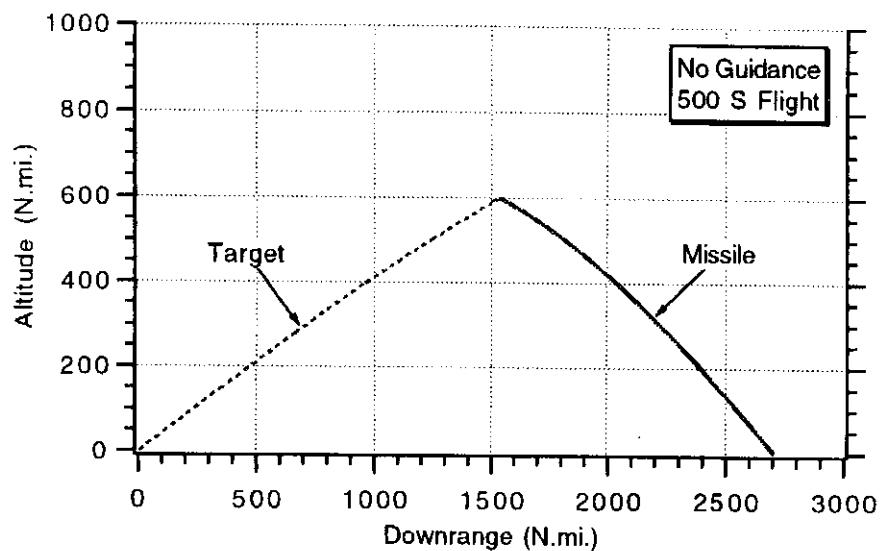


Fig. 15.3 Collision triangle geometry for nominal case.

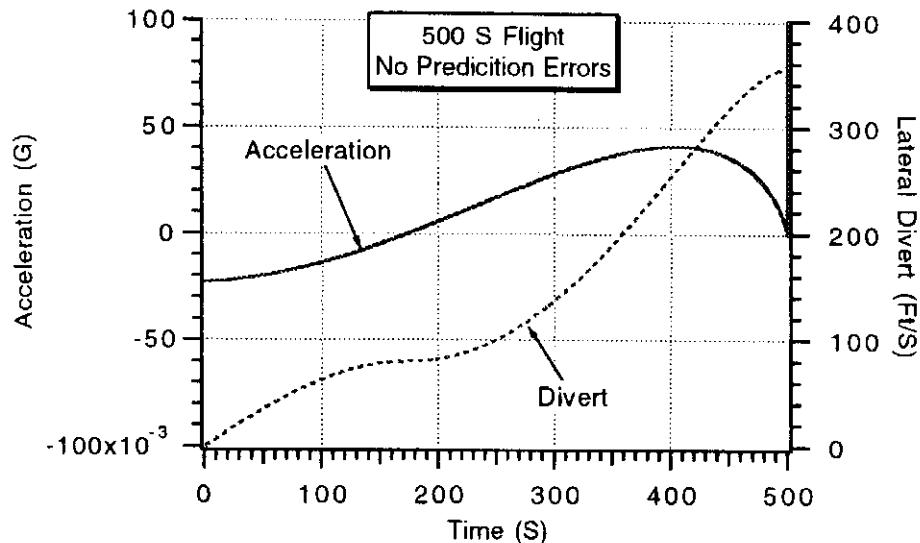


Fig. 15.4 Some divert required with proportional navigation guidance even though missile is on collision triangle.

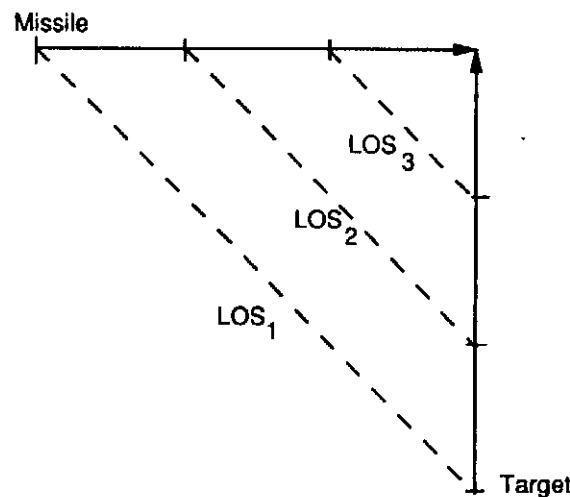


Fig. 15.5 Constant-velocity missile and target on collision triangle.

is zero! Since acceleration commands are proportional to the line-of-sight rate in a proportional navigation system, there will be no commands for a constant velocity missile and target on a collision triangle.

Figure 15.6 also shows a missile and target on a collision triangle. However, this time the missile is traveling at a constant velocity while the target velocity is nonconstant. In this case we can see that the line-of-sight lines are not parallel. Thus, we have a line-of-sight rate generated even though the missile and target are on a collision triangle. In this case a proportional navigation guidance system will generate acceleration commands, even though none are required!

In a gravity field, both the missile and target velocities vary with time. Thus, as shown in Fig. 15.6, the line-of-sight rate will not be zero, even though both the missile and target are on a collision triangle. Proportional navigation will waste some fuel in responding to the small line-of-sight rates. In our nominal case about 350 ft/s of lateral divert was required by the missile to intercept the ballistic target.

A guidance system is required since we cannot always be on a collision triangle. There will always be errors in predicting the location of the intercept point. Consider a case for a proportional navigation guidance system with an effective navigation ratio of 3 in which there is a 100-kft prediction error. This means that if we turned off the guidance system we would miss the target by 100 kft. Based on the formula derived at the beginning of this chapter, theory predicts that the required missile lateral divert should be

$$\Delta V = \frac{\text{Pred Err } N'}{(N' - 1)t_F} = \frac{100,000 * 3}{2 * 500} = 300 \text{ ft/s}$$

Figure 15.7 shows the resultant commanded acceleration and lateral divert profiles due to the 100-kft prediction error for a proportional navigation guidance system with an effective navigation ratio of 3. We can see that the missile acceleration

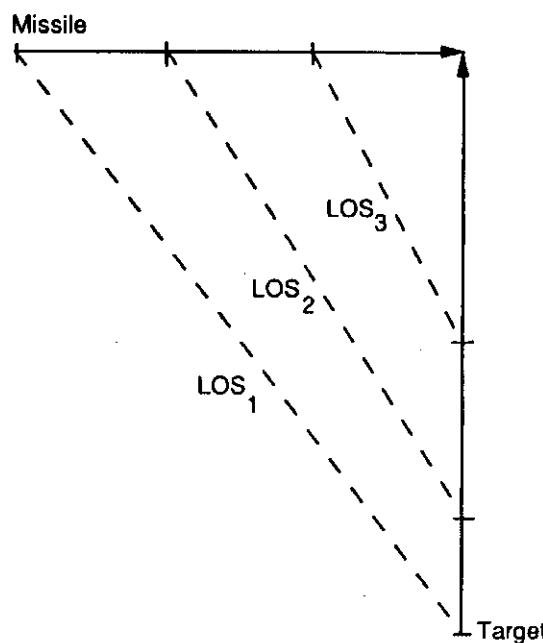


Fig. 15.6 Constant-velocity missile and variable-velocity target on collision triangle.

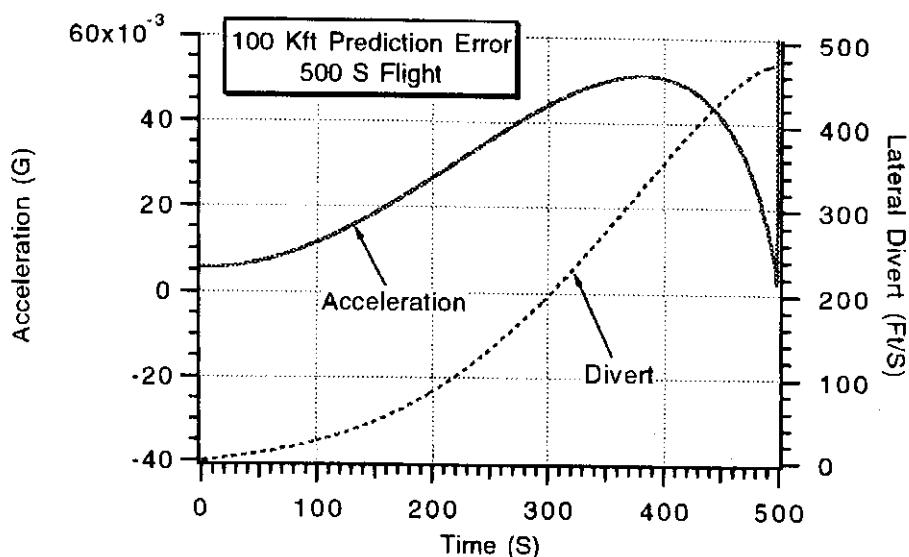


Fig. 15.7 Divert due to prediction error is close to theoretical prediction.

requirements are small (less than 0.05 g) for the flight. However, even at small acceleration levels, about 480 ft/s of lateral divert was required for a successful intercept. This value is somewhat larger than the theoretically predicted value of 300 ft/s because, as we previously saw, the gravity field also adds to the divert requirements.

Theory tells us that, for a fixed prediction error, the divert requirements will increase if the flight time is decreased. Figure 15.8 presents the engagement geometry for a 100-s flight. Note that in the simulation the flight time was reduced and the initial missile location was moved closer to the target ($TF = 100$, $XLONGMDEG = 80$). In this case both the missile and target are initially on a collision triangle.

If we introduce a 100-kft prediction error into the short-range example, theory tells us that the lateral divert requirements should increase substantially. According to the previously presented lateral divert formula, the divert requirements for a

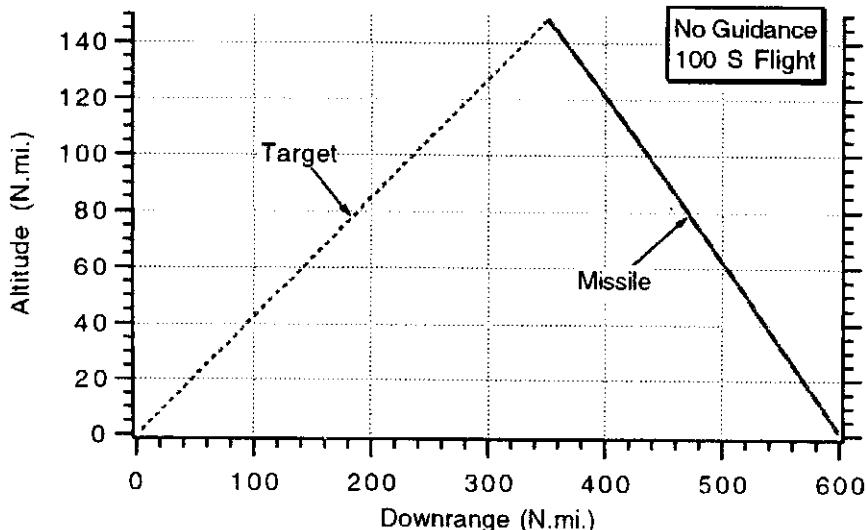


Fig. 15.8 Collision triangle geometry for shorter-range flight.

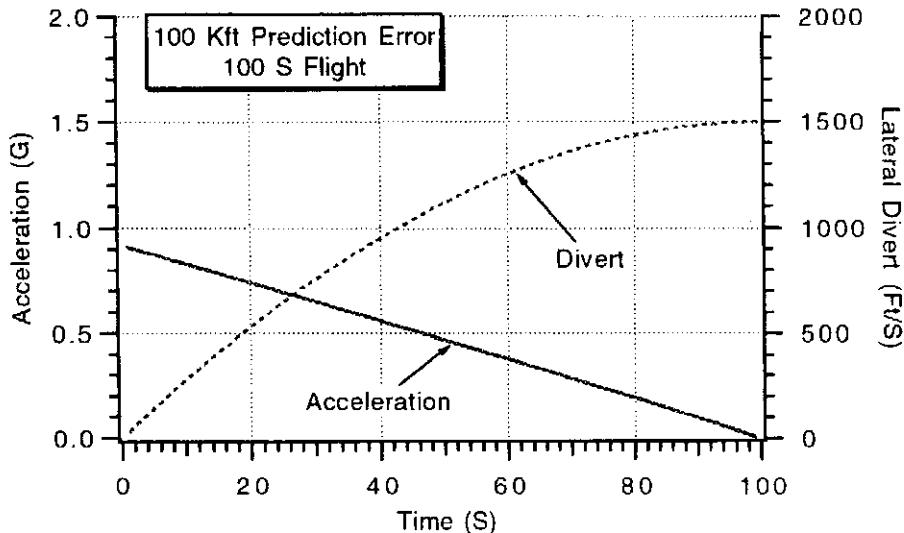


Fig. 15.9 Divert due to prediction error matches theoretical prediction for shorter flight.

100-s flight should be

$$\Delta V = \frac{\text{Pred Err } N'}{(N' - 1)t_F} = \frac{100,000 * 3}{2 * 100} = 1500 \text{ ft/s}$$

Figure 15.9 displays the commanded acceleration and actual divert requirements obtained by running the engagement simulation. We can see from the figure that the required lateral divert required is indeed nearly 1500 ft/s. Thus, we have demonstrated with nonlinear engagement simulation results that the theoretical formula is a useful and accurate indicator of divert requirements for prediction error.

Boosting Target Considerations¹

Although a booster does not execute evasive maneuvers, any longitudinal booster acceleration that is perpendicular to the line of sight will appear as a target maneuver to the missile. In Chapter 2 we saw that the closed-form solution for the acceleration required by a missile utilizing proportional navigation guidance was given by

$$n_c|_{\text{PN}} = \frac{N'}{N' - 2} \left[1 - \left(1 - \frac{t}{t_F} \right)^{N'-2} \right] n_T$$

where N' is the effective navigation ratio, t is time, t_F the flight time, and n_T the magnitude of the apparent target maneuver. From the definition of lateral divert,

$$\Delta V = \int_0^{t_F} |n_c| dt$$

we can derive an expression for the lateral divert required to hit a maneuvering target as

$$\Delta V|_{\text{PN}} = \frac{N'}{N' - 1} n_T t_F$$

Thus, the lateral divert due to a maneuvering target increases with increasing flight time and decreases with increasing effective navigation ratio.

We can develop an engagement simulation in which the target is a booster. In Chapter 13 we saw how to model a booster performing a gravity turn. We can express the longitudinal acceleration of the booster as

$$a_T = \frac{32.2T}{W}$$

where T is the booster thrust and W the booster weight. In a gravity turn the thrust and velocity vectors are aligned so that the acceleration differential equations for the booster in a gravity field become

$$\ddot{x}_T = \frac{-gm x_T}{(x_T^2 + y_T^2)^{1.5}} + \frac{a_T \dot{x}_T}{V_T}$$

$$\ddot{y}_T = \frac{-gm y_T}{(x_T^2 + y_T^2)^{1.5}} + \frac{a_T \dot{y}_T}{V_T}$$

where the target velocity V_T is given by

$$V_T = (\dot{x}_T^2 + \dot{y}_T^2)^{0.5}$$

Any component of the booster acceleration perpendicular to the line of sight will appear as an apparent target maneuver to the missile. The component of the booster acceleration perpendicular to the line of sight a_{PLOS} is given by

$$a_{\text{PLOS}} = \ddot{y}_T \cos \lambda - \ddot{x}_T \sin \lambda$$

Listing 15.2 presents a FORTRAN listing of an engagement simulation between an impulsive missile and a boosting target. The nominal numbers used to derive the booster characteristics are those of Chapter 13 in which the maximum axial booster acceleration of a two-stage booster is 20 g. The total burn time is 69.1 s and the resultant velocity and acceleration profiles can be found in Fig. 13.12. The simulation is identical to that of Listing 15.1 except that the flight time is 50 s ($TF = 50$), the initial missile location has been moved closer to the target ($XLONGMDEG = 85$), the target is a booster executing a gravity turn, and a 20-g limit has been placed on the commanded acceleration ($XNCLIM = 20$). Proportional navigation is used against the boosting target but guidance does not begin until 25 s into the flight.

A nominal case was run with the guidance system turned off to ensure that the missile and booster were on a collision triangle. Figure 15.10 shows the missile hitting the target in the nominal 50-s flight in which the booster is accelerating the entire time. The booster acceleration perpendicular to the line of sight appears in Fig. 15.11. We can see from the figure that the magnitude of the booster acceleration perpendicular to the line of sight is approximately 4 g (129 ft/s²) on the average for the last 25 s (i.e., the time for which proportional navigation is used). This means that the booster appears to the missile to be a target executing a 4-g maneuver.

The same nominal case was rerun with the proportional navigation guidance system turned on for the last 25 s. Figure 15.12 displays the missile acceleration along with the resultant lateral divert requirements. We can see that the missile lateral divert requirements for this case are approximately 5300 ft/s.

Listing 15.2 Engagement simulation for boosting target

```

IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER STEP
LOGICAL LEFT
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
LEFT=.FALSE.
XNCLIM=644.
H=.01
A=2.0926E7
GM=1.4077E16
PI=3.14159
DEGRAD=360./(2.*PI)
XNP=3.
PREDERR=0.
XISP1=250.
XISP2=250.
XMF1=.85
XMF2=.85
WPAY=100.
DELV=20000.
DELV1=.3333*DELV
DELV2=.6667*DELV
AMAX1=20.
AMAX2=20.
XKICKDEG=80.
TOP2=WPAY*(EXP(DELV2/(XISP2*32.2))-1.)
BOT2=1/XMF2-((1.-XMF2)/XMF2)*EXP(DELV2/(XISP2*32.2))
WP2=TOP2/BOT2
WS2=WP2*(1-XMF2)/XMF2
WTOT2=WP2+WS2+WPAY
TRST2=AMAX2*(WPAY+WS2)
TB2=XISP2*WP2/TRST2
TOP1=WTOT2*(EXP(DELV1/(XISP1*32.2))-1.)
BOT1=1/XMF1-((1.-XMF1)/XMF1)*EXP(DELV1/(XISP1*32.2))
WP1=TOP1/BOT1
WS1=WP1*(1-XMF1)/XMF1
WTOT=WP1+WS1+WTOT2
TRST1=AMAX1*(WTOT2+WS1)
TB1=XISP1*WP1/TRST1
XLONGMDEG=85.
XLONGTDEG=90.
ALTNMTIC=0.
ALTNMMIC=0.
XKICKDEG=80.
TF=50.
ALTT=ALTNMTIC*6076.
ALTM=ALTNMMIC*6076.
S=0.
SCOUNT=0.
XLONGM=XLONGMDEG/DEGRAD

```

(Contd.)

Listing 15.2 (Continued)

```

XLONGT=XLONGTDEG/DEGRAD
XM=(A+ALTM)*COS(XLONGM)
YM=(A+ALTM)*SIN(XLONGM)
XT=(A+ALTT)*COS(XLONGT)
YT=(A+ALTT)*SIN(XLONGT)
XFIRSTT=XT
YFIRSTT=YT
IF (LEFT) THEN
  X1T=COS(PI/2.-XKICKDEG/DEGRAD+XLONGT)
  Y1T=SIN(PI/2.-XKICKDEG/DEGRAD+XLONGT)
ELSE
  X1T=COS(-PI/2.+XKICKDEG/DEGRAD+XLONGT)
  Y1T=SIN(-PI/2.+XKICKDEG/DEGRAD+XLONGT)
END IF
XFIRSTT=XT
YFIRSTT=YT
T=0.
CALL PREDICT (TF,XT,YT,X1T,Y1T,XTF,YTF
1      ,WP1,WTOT,TB1,TRST1,TB2,WP2,WTOT2,TRST2,WPAY)
YTF=YTF+PREDERR
CALL LAMBERT(XM,YM,TF,XTF,YTF,VRXM,VRYM,XLONGM,XLONGT
      ,ICOUNT)
X1M=VRXM
Y1M=VRYM
RTM1=XT-XM
RTM2=YT-YM
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=X1T-X1M
VTM2=Y1T-Y1M
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
DELV=0.
10   IF(VC<0.)GOTO 999
TGO=RTM/VC
IF(TGO>.1)THEN
  H=.01
ELSE
  H=.0001
ENDIF
XOLDT=XT
YOLDT=YT
X1OLDT=X1T
Y1OLDT=Y1T
XOLDM=XM
YOLDM=YM
X1OLDM=X1M
Y1OLDM=Y1M
DELVOLD=DELV
STEP=1
GOTO 200
66   STEP=2

```

(Contd.)

Listing 15.2 (Continued)

```

XT=XT+H*XDT
YT=YT+H*YDT
X1T=X1T+H*X1DT
Y1T=Y1T+H*Y1DT
XM=XM+H*XDM
YM=YM+H*YDM
X1M=X1M+H*X1DM
Y1M=Y1M+H*Y1DM
DELV=DELV+H*DELVD
T=T+H
GOTO 200
55   XT=(XOLDT+XT)/2+.5*H*XDT
      YT=(YOLDT+YT)/2+.5*H*YDT
      X1T=(X1OLDT+X1T)/2+.5*H*X1DT
      Y1T=(Y1OLDT+Y1T)/2+.5*H*Y1DT
      XM=(XOLDM+XM)/2+.5*H*XDM
      YM=(YOLDM+YM)/2+.5*H*YDM
      X1M=(X1OLDM+X1M)/2+.5*H*X1DM
      Y1M=(Y1OLDM+Y1M)/2+.5*H*Y1DM
      DELV=(DELVOLD+DELV)/2+.5*H*DELVD
      ALTT=SQRT(XT**2+YT**2)-A
      ALTM=SQRT(XM**2+YM**2)-A
      S=S+H
      SCOUNT=SCOUNT+H
      IF(SCOUNT.LT..99999)GOTO 10
      SCOUNT=0.
      XNMT=XT/6076.
      YNMT=YT/6076.
      XNMM=XM/6076.
      YNMM=YM/6076.
      ALTNMT=ALTT/6076.
      CALL DISTANCE(XT,YT,XFIRSTT,YFIRSTT,DISTNMT)
      ALTNMM=ALTM/6076.
      CALL DISTANCE(XM,YM,XFIRSTT,YFIRSTT,DISTNMM)
      XNCG=XNC/32.2
      ATPLOSG=ATPLOS/32.2
      WRITE(*,97)T,DISTNMT,ALTNMT,DISTNMM,ALTNMM,XNCG,DELV
                  ,ATPLOSG
      WRITE(1,97)T,DISTNMT,ALTNMT,DISTNMM,ALTNMM,XNCG,DELV
                  ,ATPLOSG
      GOTO 10
97   FORMAT(8F10.3)
200   CONTINUE
      IF(T<TB1)THEN
          WGT=-WP1*T/TB1+WTOT
          TRST=TRST1
      ELSEIF(T<(TB1+TB2))THEN
          WGT=-WP2*T/TB2+WTOT2+WP2*TB1/TB2
          TRST=TRST2
      ELSE

```

(Contd.)

Listing 15.2 (Continued)

```

WGT=WPAY
TRST=0.
ENDIF
AT=32.2*TRST/WGT
VEL=SQRT(X1T**2+Y1T**2)
AXT=AT*X1T/VEL
AYT=AT*Y1T/VEL
TEMBOTT=(XT**2+YT**2)**1.5
X1DT=-GM*XT/TEMBOTT+AXT
Y1DT=-GM*YT/TEMBOTT+AYT
XDT=X1T
YDT=Y1T
RTM1=XT-XM
RTM2=YT-YM
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=X1T-X1M
VTM2=Y1T-Y1M
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
TGO=RTM/VC
XLAM=ATAN2(RTM2,RTM1)
XLAMD=(RTM1*VTM2-RTM2*VTM1)/(RTM*RTM)
ATPLOS=Y1DT*COS(XLAM)-X1DT*SIN(XLAM)
IF(T>25.)THEN
    XNC=XNP*VC*XLAMD+.5*XNP*ATPLOS
ELSE
    XNC=0.
ENDIF
IF(XNC>XNCLIM)XNC=XNCLIM
IF(XNC<-XNCLIM)XNC=-XNCLIM
DELVD=ABS(XNC)
AM1=-XNC*SIN(XLAM)
AM2=XNC*COS(XLAM)
TEMBOTM=(XM**2+YM**2)**1.5
X1DM=-GM*XM/TEMBOTM+AM1
Y1DM=-GM*YM/TEMBOTM+AM2
XDM=X1M
YDM=Y1M
IF(STEP-1)66,66,55
999 CONTINUE
XNMT=XT/6076.
YNMT=YT/6076.
XNMM=XM/6076.
YNMM=YM/6076.
ALTNMT=ALTT/6076.
CALL DISTANCE(XT,YT,XFIRSTT,YFIRSTT,DISTNMT)
ALTNMM=ALTM/6076.
CALL DISTANCE(XM,YM,XFIRSTT,YFIRSTT,DISTNMM)
XNCG=XNC/32.2
ATPLOSG=ATPLOS/32.2
WRITE(*,97)T,DISTNMT,ALTNMT,DISTNMM,ALTNMM,XNCG,DELV

```

(Contd.)

Listing 15.2 (Continued)

```

      ,ATPLOSG
      WRITE(1,97)T,DISTNMT,ALTNMT,DISTNM,ALTNMM,XNCG,DELV
      ,ATPLOSG
      WRITE(*,*)T,RTM,DELV
      PAUSE
      CLOSE(1)
      END

      SUBROUTINE DISTANCE(XT,YT,XF,YF,DISTNM)
      SAVE
      IMPLICIT REAL*8 (A-H)
      IMPLICIT REAL*8 (O-Z)
      R=SQRT(XT**2+YT**2)
      RF=SQRT(XF**2+YF**2)
      A=2.0926E7
      CBETA=(XT*XF+YT*YF)/(R*RF)
      IF(CBETA<1.)THEN
          BETA=ACOS(CBETA)
          DISTNM=A*BETA/6076.
      ELSE
          DISTNM=(XT-XF)/6076.
      ENDIF
      RETURN
      END

      SUBROUTINE PREDICT (TF,XDUM,YDUM,X1DUM,Y1DUM,XTF,YTF
1           ,WP1,WTOT,TB1,TRST1,TB2,WP2,WTOT2,TRST2,WPAY)
      IMPLICIT REAL*8 (A-H)
      IMPLICIT REAL*8 (O-Z)
      INTEGER STEP
      SAVE
      H=.01
      A=2.0926E7
      GM=1.4077E16
      T=0.
      X=XDUM
      Y=YDUM
      X1=X1DUM
      Y1=Y1DUM
10     IF(T>(TF-.00001))GOTO 999
      XOLD=X
      YOLD=Y
      X1OLD=X1
      Y1OLD=Y1
      STEP=1
      GOTO 200
66     STEP=2
      X=X+H*XD
      Y=Y+H*YD
      X1=X1+H*X1D

```

(Contd.)

Listing 15.2 (Continued)

```

Y1=Y1+H*Y1D
T=T+H
GOTO 200
55  X=(XOLD+X)/2+.5*H*XD
    Y=(YOLD+Y)/2+.5*H*YD
    X1=(X1OLD+X1)/2+.5*H*X1D
    Y1=(Y1OLD+Y1)/2+.5*H*Y1D
    GOTO 10
200  CONTINUE
    IF(T<TB1)THEN
        WGT=-WP1*T/TB1+WTOT
        TRST=TRST1
    ELSEIF(T<(TB1+TB2))THEN
        WGT=-WP2*T/TB2+WTOT2+WP2*TB1/TB2
        TRST=TRST2
    ELSE
        WGT=WPAY
        TRST=0.
    ENDIF
    AT=32.2*TRST/WGT
    VEL=SQRT(X1**2+Y1**2)
    AXT=AT*X1/VEL
    AYT=AT*Y1/VEL
    TEMBOTT=(X**2+Y**2)**1.5
    X1D=-GM*X/TEMBOTT+AXT
    Y1D=-GM*Y/TEMBOTT+AYT
    XD=X1
    YD=Y1
    IF(STEP-1)66,66,55
999   CONTINUE
    XTF=X
    YTF=Y
    RETURN
    END

SUBROUTINE LAMBERT(XIC,YIC,TFDES,XF,YF,VRX,VRY,XLONGM
1           ,XLONGT,ICOUNT)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
A=2.0926E7
GM=1.4077E16
RIC=SQRT(XIC**2+YIC**2)
RF=SQRT(XF**2+YF**2)
CPHI=(XIC*XF+YIC*YF)/(RIC*RF)
PHI=ACOS(CPHI)
SPHI=SIN(PHI)
R0=RIC
PI=3.14159
DEGRAD=360./(2.*PI)

```

(Contd.)

Listing 15.2 (Continued)

```

ICOUNT=1
GMIN=ATAN2((SPHI-SQRT(2.*R0*(1.-CPHI)/RF)),(1-CPHI))
GMAX=ATAN2((SPHI+SQRT(2.*R0*(1.-CPHI)/RF)),(1-CPHI))
GAM=(GMIN+GMAX)/2.
DO
    TOP=GM*(1.-COS(PHI))
    TEMP=R0*COS(GAM)/RF-COS(PHI+GAM)
    BOT=R0*COS(GAM)*TEMP
    V=SQRT(TOP/BOT)
    IF (XLONGT>XLONGM) THEN
        VRX=V*COS(PI/2.-GAM+XLONGM)
        VRY=V*SIN(PI/2.-GAM+XLONGM)
    ELSE
        VRX=V*COS(-PI/2.+GAM+XLONGM)
        VRY=V*SIN(-PI/2.+GAM+XLONGM)
    END IF
    XLAM=R0*V*V/GM
    TOP1=TAN(GAM)*(1-COS(PHI))+(1-XLAM)*SIN(PHI)
    BOT1P=(1-COS(PHI))/(XLAM*COS(GAM)*COS(GAM))
    BOT1=(2-XLAM)*(BOT1P+COS(GAM+PHI)/COS(GAM))
    TOP2=2*COS(GAM)
    BOT2=XLAM*((2/XLAM-1)**1.5)
    TOP3=SQRT(2/XLAM-1)
    BOT3=COS(GAM)/TAN(PHI/2)-SIN(GAM)
    TEMP=(TOP2/BOT2)*ATAN2(TOP3,BOT3)
    TF=R0*(TOP1/BOT1+TEMP)/(V*COS(GAM))
    IF((ABS(TFDES-TF)<=.00000001*TFDES).OR.ICOUNT>100)THEN
        EXIT
    ENDIF
    IF(TF>TFDES)THEN
        GMAX=GAM
    ELSE
        GMIN=GAM
    ENDIF
    IF(ICOUNT.EQ.1)THEN
        XNEXT=(GMAX+GMIN)/2.
    ELSE
        XNEXT=GAM+(GAM-GOLD)*(TFDES-TF)/(TF-TOLD)
        IF(XNEXT>GMAX.OR.XNEXT<GMIN)THEN
            XNEXT=(GMAX+GMIN)/2.
        ENDIF
    ENDIF
    GOLD=GAM
    TOLD=TF
    GAM=XNEXT
    ICOUNT=ICOUNT+1
REPEAT
RETURN
END

```

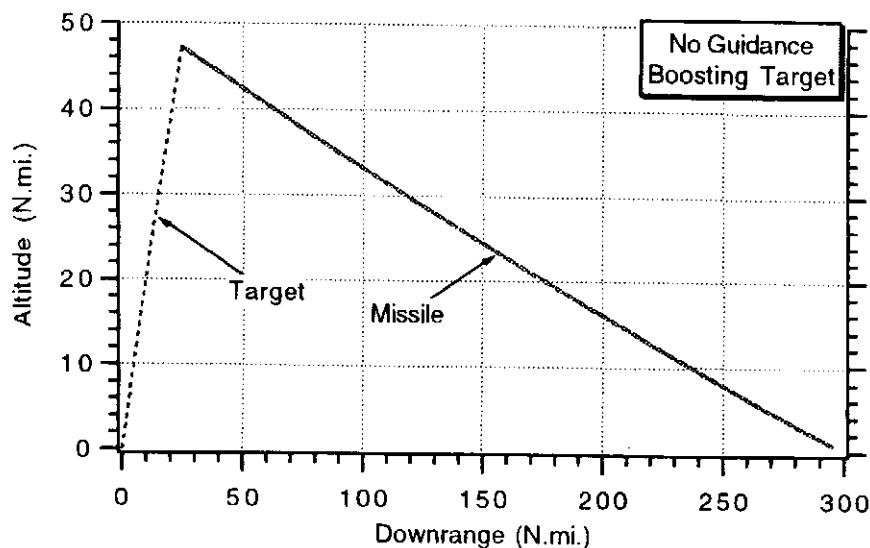


Fig. 15.10 Missile on collision triangle for boosting target.

For this case theory says that the divert requirements should be

$$\Delta V|_{\text{PN}} = \frac{N'}{N' - 1} n_T t_F = \frac{3 * 129 * 25}{2} = 4838 \text{ ft/s}$$

In other words, theory and simulation are in close agreement.

We saw that with tactical missiles the augmented proportional navigation guidance law reduced the interceptor acceleration requirements. The closed-form solution for the acceleration required to hit a maneuvering target with the augmented proportional navigation guidance law was shown to be given by

$$n_c|_{\text{APN}} = 0.5 n_T N' \left[1 - \frac{t}{t_F} \right]^{N'-2}$$

Using the fact that the lateral divert is the integral of the absolute value of the

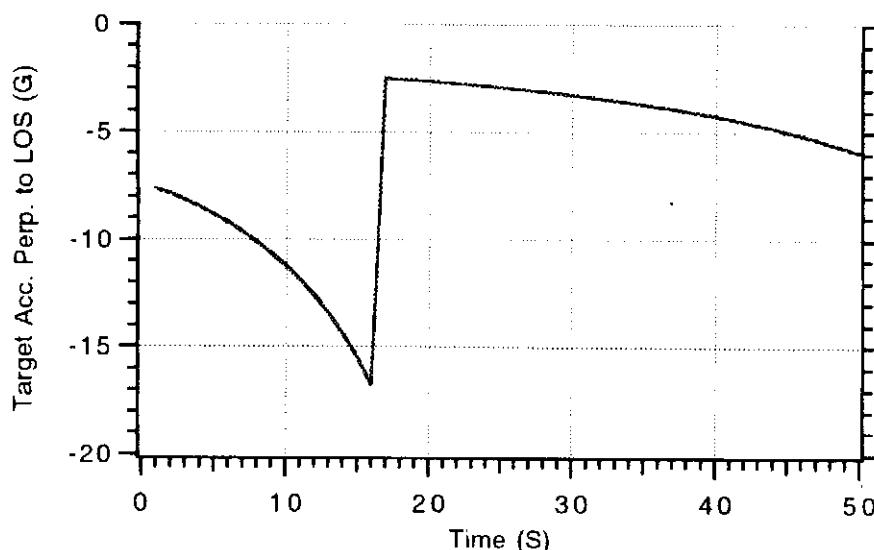


Fig. 15.11 Average target acceleration for last 25s is approximately 4 g.

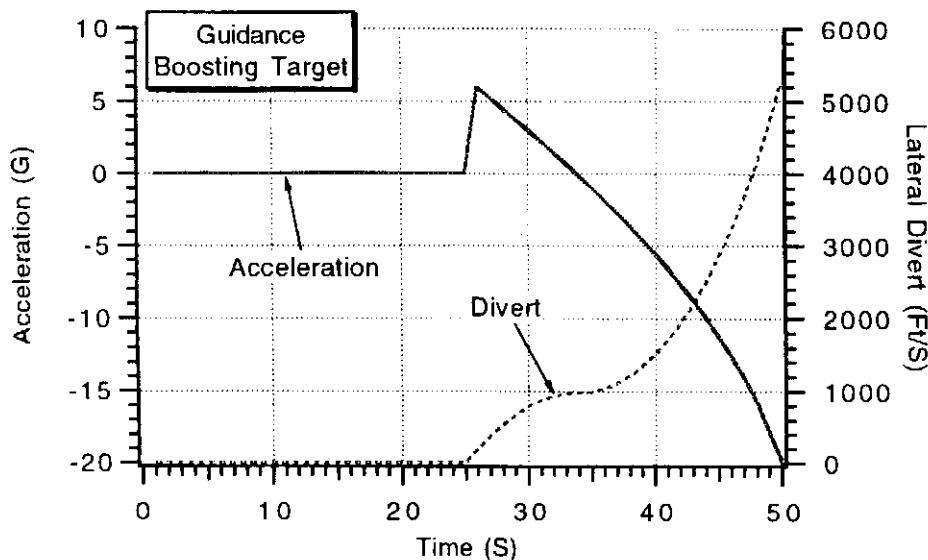


Fig. 15.12 Divert due to apparent maneuver agrees with theory.

acceleration, it is easy to show that

$$\Delta V|_{APN} = .5 \frac{N'}{N' - 1} n_T t_F = .5 \Delta V|_{PN}$$

In other words, theory says that the divert requirements for an augmented proportional navigation guidance system are half the divert requirements of a proportional navigation guidance system.

To implement augmented proportional navigation guidance in the engagement simulation, it is necessary to modify the guidance command to

$$n_c|_{APN} = N' V_c \lambda + \frac{N'}{2} a_{PLOS}$$

where a_{PLOS} is the booster acceleration perpendicular to the line of sight. The nominal simulation case was rerun, except this time the augmented proportional

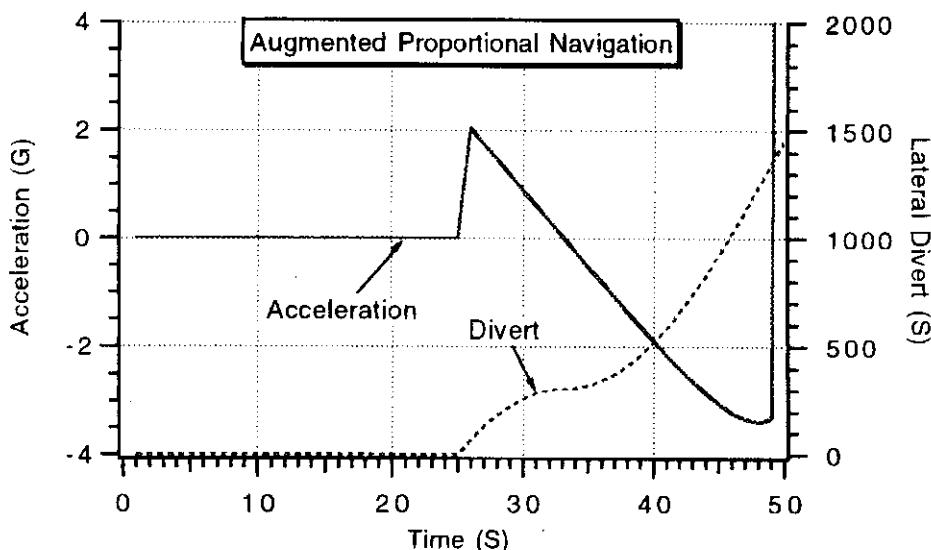


Fig. 15.13 Augmented proportional navigation reduces divert requirements due to boosting target.

navigation guidance law was used. Figure 15.13 shows that the missile lateral divert requirements were dramatically reduced to about 1500 ft/s (down from about 5200 ft/s in the proportional navigation case).

Theory says the divert requirements for the augmented proportional navigation guidance law should be

$$\Delta V|_{APN} = 0.5 \frac{N'}{N' - 1} n_T t_F = \frac{0.5 * 3 * 129 * 25}{2} = 2419 \text{ ft/s}$$

which is in approximate agreement with the simulation results.

Summary

In this chapter we have shown that the guidance concepts developed for the tactical world are applicable to the strategic world. In fact, closed-form solutions for the required missile acceleration to hit targets can be converted to lateral divert formulas. Nonlinear engagement simulation results indicate that the divert requirement formulas for prediction error, apparent target acceleration, and guidance law are not only useful but are in fact accurate indicators of strategic interceptor requirements.

Reference

¹Zarchan, P., "Space Based Interceptor Engagement Simulation and Analysis," Charles Stark Draper Lab., Rept. R-2025, Cambridge, MA, Dec. 1987.

Miscellaneous Topics

Introduction

IN this chapter we shall cover important topics that have not yet been discussed. First we shall show how lateral divert requirements can be reduced if we add an extra term to the proportional navigation guidance law to account for gravity. Next we will demonstrate that, if complete information on all the target states is available, the ultimate guidance law, predictive guidance, can be used to relax system divert requirements. A section will be devoted to showing how all the booster states can be estimated given that range and angle measurements are available. Finally, a new guidance law, known as pulsed guidance, is developed for those situations in which an interceptor does not have throttleable divert engines.

Gravity Compensation

We saw in Chapter 15 that, in the absence of prediction errors, lateral divert fuel was still required to hit a ballistic target. Changes in the missile and target velocities due to gravity caused the line-of-sight to rotate. The proportional navigation guidance law responded to the apparent line-of-sight rate with acceleration commands. If we have knowledge of gravitational acceleration, it seems reasonable that it might be possible to compensate for unnecessary accelerations via the guidance law.

Consider the ballistic missile-target model of Fig. 16.1. In this case the only acceleration acting on the missile and target is gravity. However, since the missile and target are in different locations and since gravitational acceleration is always toward the center of the Earth, the gravitational vectors for the missile and target will have different magnitudes and directions. The missile and target gravitational vectors can be expressed as

$$\mathbf{grav}_M = \mathbf{grav}_M x \mathbf{i} + \mathbf{grav}_M y \mathbf{j}$$

$$\mathbf{grav}_T = \mathbf{grav}_T x \mathbf{i} + \mathbf{grav}_T y \mathbf{j}$$

where the gravitational components for the missile and target can be found from

$$\mathbf{grav}_M x = \frac{-gm x_M}{(x_M^2 + y_M^2)^{1.5}}$$

$$\mathbf{grav}_M y = \frac{-gm y_M}{(x_M^2 + y_M^2)^{1.5}}$$

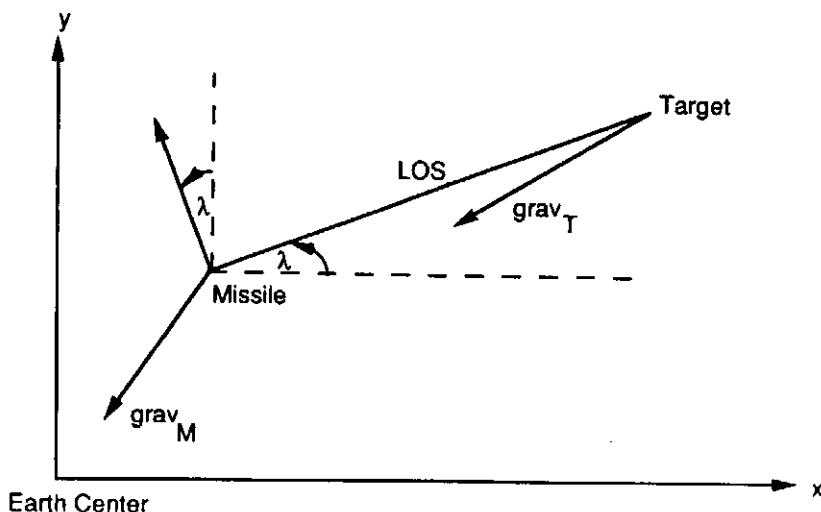


Fig. 16.1 Model for understanding gravity compensation.

$$gravx_T = \frac{-gm x_T}{(x_T^2 + y_T^2)^{1.5}}$$

$$gravy_T = \frac{-gm y_T}{(x_T^2 + y_T^2)^{1.5}}$$

From Fig. 16.1 we can see that the component of gravity perpendicular to the line of sight for both the missile and target can be found by trigonometry and is given by

$$grav_{MPLOS} = -gravx_M \sin \lambda + gravy_M \cos \lambda$$

$$grav_{TPLOS} = -gravx_T \sin \lambda + gravy_T \cos \lambda$$

The gravitational acceleration difference between the target and missile can be treated as an additional term in the zero effort miss. Therefore, we can modify the proportional navigation guidance law to account for gravity. The resultant law, which is similar to augmented proportional navigation, is

$$n_c = N' V_c \dot{\lambda} + \frac{N'}{2} (grav_{TPLOS} - grav_{MPLOS})$$

The similarity of proportional navigation with gravity compensation and augmented proportional navigation is due to the fact that the gravitational components of the missile and target are treated as an apparent residual target acceleration.

The nominal 500-s ballistic case of Chapter 15 was repeated in which the missile and target are on a collision triangle (see Figs. 15.3 and 15.4). In this case there are zero prediction errors. Figure 16.2 compares the acceleration profiles of proportional navigation, for this case, when gravity compensation is both included and excluded. We can see that in both cases the acceleration levels are low, but the gravity compensation results in even smaller acceleration levels. We can also see from the figure that the lateral divert requirements have been reduced from 356 ft/s to 114 ft/s with gravity compensation.

Although gravity compensation reduced the missile lateral divert requirements for the nominal case, the divert requirements were not very high without the

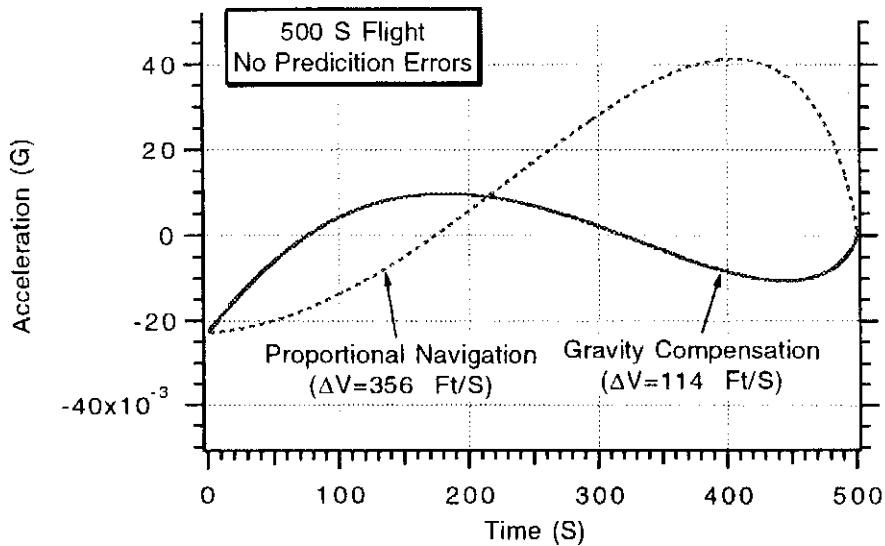


Fig. 16.2 Gravity compensation results in smaller acceleration levels for nominal case.

compensation. Gravity compensation should be even more important for longer-range flights since there is more time for a strategic missile to waste fuel. Another case was examined in which the flight time was increased from 500 s for the nominal case to 700 s for the new case. Figure 16.3 depicts the engagement geometry for the longer-range case. We can see that the longer flight time results in a more lofted missile trajectory.

Both methods of guidance (with and without gravity compensation) were compared for the new longer flight time case. Figure 16.4, which compares the commanded acceleration profiles, shows that the acceleration requirements are substantially less when gravity compensation is used. The figure indicates that the missile lateral divert requirements were reduced from 1673 ft/s to 508 ft/s when gravity compensation was used. In this case there would be a dramatic advantage in terms of fuel and weight savings in using gravity compensation.

The gravity compensation guidance law assumes that the required gravitational components of both the missile and target can be measured or estimated precisely.

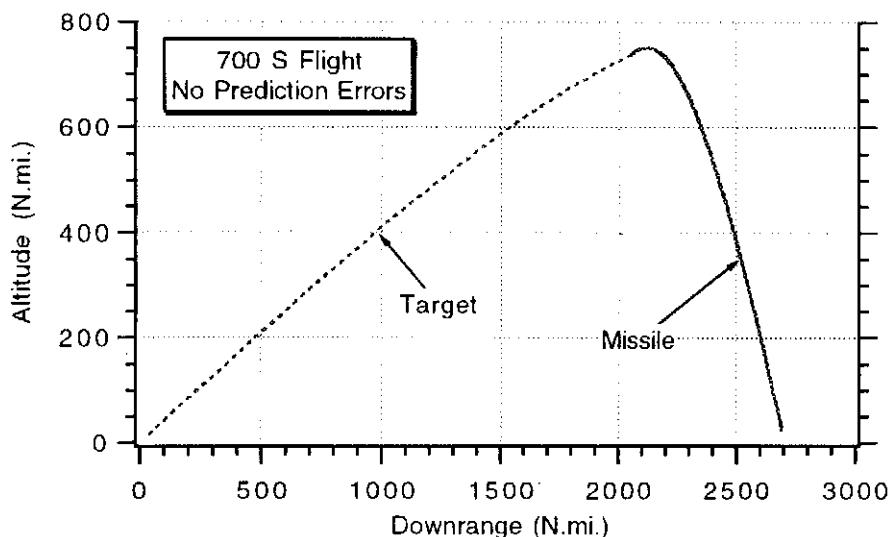


Fig. 16.3 Engagement geometry for longer-range case.

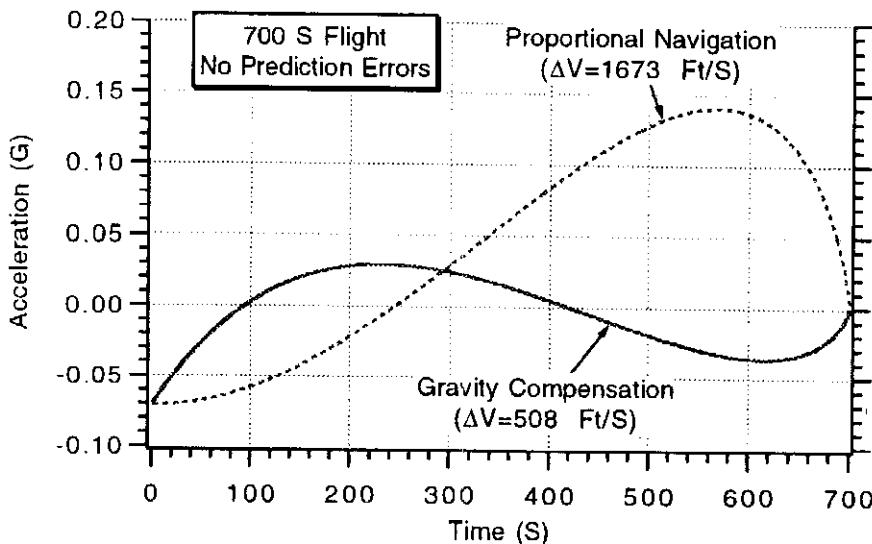


Fig. 16.4 Gravity compensation offers dramatic reduction in divert requirements for long-range case.

Errors in estimating the gravitational components will of course degrade the effectiveness of this type of compensation, perhaps to the point where its performance is the same or worse than that of proportional navigation.

Predictive Guidance

We have seen how interceptor lateral divert requirements can be reduced when extra information, if it exists, is incorporated in the guidance law. If an exact model of the target and missile dynamics were available, one could achieve the best performance with predictive guidance. The principle behind predictive guidance is quite simple. We take our dynamic models of the target and missile and numerically integrate them forward until the desired intercept time. In other words, we are predicting the future location of the missile and target. The difference between the predicted missile and target position at the intercept time is the zero effort miss. If the predicted coordinates of the missile at intercept in the Earth-centered system is given by (x_{MF}, y_{MF}) and the coordinates of the target at intercept are given by (x_{TF}, y_{TF}) , then the Earth-centered components of the zero effort miss are given by

$$ZEM_x = x_{TF} - x_{MF}$$

$$ZEM_y = y_{TF} - y_{MF}$$

We can find the component of the zero effort miss perpendicular to the line of sight by trigonometry in Fig. 16.1. The zero effort miss perpendicular to the line-of-sight is given by

$$ZEM_{PLOS} = -ZEM_x \sin \lambda + ZEM_y \cos \lambda$$

We saw from our discussion of optimal guidance that the acceleration guidance command should be proportional to the zero effort miss and inversely proportional to the square of time to go until intercept, or

$$n_c = \frac{N' ZEM_{PLOS}}{t_{go}^2}$$

Proportional navigation, augmented proportional navigation, and our previously derived optimal guidance law can all be expressed in the preceding form. In these guidance laws we have closed-form expressions for the zero effort miss. In other words, an integration of simple dynamics (assumed to be a polynomial in time) was conducted to get a closed-form expression. In predictive guidance, we ignore closed-form solutions of approximate processes and obtain the exact solution for the zero effort miss at each guidance update by numerical integration. The resultant accuracy of the computed zero effort miss depends on the size of the integration interval. Small integration intervals yield accurate answers but may take too long to be obtained in flight. Of course, the accuracy also depends on the equations used. Having inaccurate models of the target will lead to erroneous predictions of the zero effort miss, and in this case the performance of predictive guidance may be substantially worse than that of proportional navigation.

Listing 16.1 presents an engagement simulation of a ballistic missile and a boosting target. This simulation is identical to the one of Listing 15.2 except that the guidance law has changed from proportional navigation to predictive guidance. We can see from the listing that subroutine PREDICT is not only used to establish the initial estimate of the intercept point but also is now used at each guidance update to compute the zero effort miss. An examination of the prediction subroutine shows that the equations of the target and missile have been perfectly modeled. Even the integration step size is an exact match. We can see from the listing that guidance commands, based upon predictive guidance, are calculated at each guidance update throughout the flight.

The nominal 50-s boosting target case of the previous chapter (see Figs. 15.10 and 15.11) was repeated to see the effectiveness of the new guidance law. Figure 16.5 displays the commanded missile acceleration requirements for predictive guidance for the nominal case. We can see that, as expected, predictive guidance requires much less acceleration than either proportional navigation or augmented proportioned navigation. In fact, we can see that predictive guidance virtually requires zero acceleration to intercept the boosting target. The reason for this is that the missile is initially on a collision triangle with the target. Therefore, no commands are really necessary for a successful intercept. However, if for computer

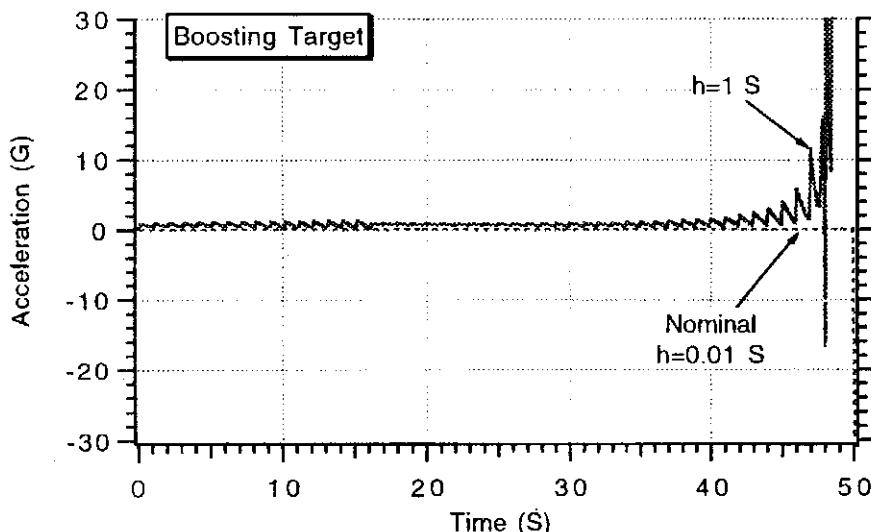


Fig. 16.5 Acceleration requirements for predictive guidance can be very small.

Listing 16.1 Predictive guidance engagement simulation of ballistic missile and booster target

```

IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER STEP
LOGICAL LEFT
OPEN (1, STATUS='UNKNOWN', FILE='DATEFIL')
LEFT=.FALSE.
H=.01.
A=2.0926E7
GM=1.4077E16
XNP=3.
AXMGUID=0.
AYMGUID=0.
PREDERR=0.
XISP1=250.
XISP2=250.
XMF1=.85
XMF2=.85
WPAY=100.
DELV=20000.
DELV1=.3333*DELV
DELV2=.6667*DELV
AMAX1=20.
AMAX2=20.
XKICKDEG=80.
TOP2=WPAY*(EXP(DELV2/(XISP2*32.2))-1.)
BOT2=1/XMF2-((1.-XMF2)/XMF2)*EXP(DELV2/(XISP2*32.2))
WP2=TOP2/BOT2
WS2=WP2*(1-XMF2)/XMF2
WTOT2=WP2+WS2+WPAY
TRST2=AMAX2*(WPAY+WS2)
TB2=XISP2*WP2/TRST2
TOP1=WTOT2*(EXP (DELV1/(XISP1*32.2))-1.)
BOT1=1/XMF1-((1.-XMF1)/XMF1)*EXP(DELV1/(XISP1*32.2))
WP1=TOP1/BOT1
WS1=WP1*(1-XMF1)/XMF1
WTOT=WP1+WS1+WTOT2
TRST1=AMAX1*(WTOT2+WS1)
TB1=XISP1*WP1/TRST1
ALTNMT=0.
ALTNMM=0.
ALTT=ALTNMT*6076.
ALTM=ALTNMM*6076.
PI=3.14159
DEGRAD=360./(2.*PI)
S=0.
SCOUNT=0.
XLONGMDEG=85.
XLONGTDEG=90.
XLONGM=XLONGMDEG/DEGRAD

```

(Contd.)

Listing 16.1 (Continued)

```

XLONGT=XLONGTDEG/DEGRAD
XM=(A+ALTM)*COS(XLONGM)
YM=(A+ALTM)*SIN(XLONGM)
XT=(A+ALTT)*COS(XLONGT)
YT=(A+ALTT)*SIN(XLONGT)
XFIRSTT=XT
YFIRSTT=YT
IF (LEFT) THEN
    X1T=COS(PI/2.-XKICKDEG/DEGRAD+XLONGT)
    Y1T=SIN(PI/2.-XKICKDEG/DEGRAD+XLONGT)
ELSE
    X1T=COS(-PI/2.+XKICKDEG/DEGRAD+XLONGT)
    Y1T=SIN(-PI/2.+XKICKDEG/DEGRAD+XLONGT)
END IF
T=0.
TF=50.
TGO=TF-T
CALL PREDICT (T,TF,XT,YT,X1T,Y1T,XTF,YTF
1      ,WP1,WTOT,TB1,TRST1,TB2,WP2,WTOT2,TRST2,WPAY
2      ,XM,YM,X1M,Y1M,ZEM1,ZEM2,TGO)
YTF=YTF+PREDERR
CALL LAMBERT(XM,YM,TF,XTF,YTF,VRXM,VRYM,XLONGM,XLONGT)
X1M=VRXM
Y1M=VRYM
RTM1=XT-XM
RTM2=YT-YM
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=X1T-X1M
VTM2=Y1T-Y1M
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
DELV=0.
10 IF(VC<0.)GOTO 999
TGO=RTM/VC
IF(TGO>.1)THEN
    H=.01
ELSE
    H=.0001
ENDIF
XOLDT=XT
YOLDT=YT
X1OLDT=X1T
Y1OLDT=Y1T
XOLDM=XM
YOLDM=YM
X1OLDM=X1M
Y1OLDM=Y1M
DELVOLD=DELV
STEP=1
GOTO 200

```

(Contd.)

Listing 16.1 (Continued)

```

66  STEP=2
    XT=XT+H*XDT
    YT=YT+H*YDT
    X1T=X1T+H*X1DT
    Y1T=Y1T+H*Y1DT
    XM=XM+H*XDM
    YM=YM+H*YDM
    X1M=X1M+H*X1DM
    Y1M=Y1M+H*Y1DM
    DELV=DELV+H*DELVD
    T=T+H
    GOTO 200
55  XT=(XOLDT+XT)/2+.5*H*XDT
    YT=(YOLDT+YT)/2+.5*H*YDT
    X1T=(X1OLDT+X1T)/2+.5*H*X1DT
    Y1T=(Y1OLDT+Y1T)/2+.5*H*Y1DT
    XM=(XOLDM+XM)/2+.5*H*XDM
    YM=(YOLDM+YM)/2+.5*H*YDM
    X1M=(X1OLDM+X1M)/2+.5*H*X1DM
    Y1M=(Y1OLDM+Y1M)/2+.5*H*Y1DM
    DELV=(DELVOLD+DELV)/2+.5*H*DELVD
    ALTT=SQRT(XT**2+YT**2)-A
    ALTM=SQRT(XM**2+YM**2)-A
    S=S+H
    SCOUNT=SCOUNT+H
    IF(SCOUNT.LT.099999)GOTO 10
    SCOUNT=0.
    CALL PREDICT (T,TF,XT,YT,X1T,Y1T,XTF,YTF
1      ,WP1,WTOT,TB1,TRST1,TB2,WP2,WTOT2,TRST2,WPAY
2      ,XM,YM,X1M,Y1M,ZEM1,ZEM2,TGO)
    ZEMPLOS=-ZEM1*SIN(XLAM)+ZEM2*COS(XLAM)
    XNC=XNP*ZEMPLOS/(TGO*TGO)
    IF (XNC>966.)XNC=966.
    IF (XNC<-966.)XNC=-966.
    XNMT=XT/6076.
    YNMT=YT/6076.
    XNMM=XM/6076.
    YNMM=YM/6076.
    ALTNMT=ALTT/6076.
    CALL DISTANCE(XT,YT,XFIRSTT,YFIRSTT,DISTNMT)
    ALTNMM=ALTM/6076.
    XNCG=XNC/32.2
    ATPLOSG=ATPLOS/32.2
    CALL DISTANCE(XM,YM,XFIRSTT,YFIRSTT,DISTNMM)
    WRITE(*,97)T,DISTNMT,ALTNMT,DISTNMM,ALTNMM,XNCG,DELV
               ,ATPLOSG
    WRITE(1,97)T,DISTNMT,ALTNMT,DISTNMM,ALTNMM,XNCG,DELV
               ,ATPLOSG
    GOTO 10

```

(Contd.)

Listing 16.1 (Continued)

```

97  FORMAT (8F10.3)
200 CONTINUE
IF(T<TB1)THEN
    WGT=-WP1*T/TB1+WTOT
    TRST=TRST1
ELSEIF(T<(TB1+TB2))THEN
    WGT=-WP2*T/TB2+WTOT2+WP2*TB1/TB2
    TRST=TRST2
ELSE
    WGT=WPAY
    TRST=0.
ENDIF
AT=32.2*TRST/WGT
VEL=SQRT(X1T**2+Y1T**2)
AXT=AT*X1T/VEL
AYT=AT*Y1T/VEL
ATPLOS=AYT*COS(XLAM)-AXT*SIN(XLAM)
TEMBOOTT=(XT**2+YT**2)**1.5
X1DT=-GM*XT/TEMBOOTT+AXT
Y1DT=-GM*YT/TEMBOOTT+AYT
XDT=X1T
YDT=Y1T
RTM1=XT-XM
RTM2=YT-YM
RTM=SQRT(RTM1**2 +RTM2**2)
VTM1=X1T-X1M
VTM2=Y1T-Y1M
VC= -(RTM1*VTM1+RTM2*VTM2)/RTM
TGO=RTM/VC
XLAM=ATAN2(RTM2,RTM1)
XLAMD=(RTM1*VTM2-RTM2*VTM1)/(RTM*RTM)
AXMGUID= -XNC*SIN(XLAM)
AYMGUID=XNC*COS(XLAM)
DELVD=ABS(XNC)
TEMBOTM=(XM** 2 +YM**2)**1.5
X1DM=-GM*XM/TEMBOTM+AXMGUID
Y1DM=-GM*YM/TEMBOTM+AYMGUID
XDM=X1M
YDM=Y1M
IF(STEP-1)66,66,55
999 CONTINUE
XNMT=XT/6076.
YNMT=YT/6076.
XNMM=XM/6076.
YNMM=YM/6076.
ALTNMT=ALTT/6076.
CALL DISTANCE(XT,YT,XFIRSTT,YFIRSTT,DISTNMT)
ALTNMM=ALTM/6076.
XNCG=XNC/32.2
ATPLOSG=ATPLOS/32.2

```

(Contd.)

Listing 16.1 (Continued)

```

CALL DISTANCE(XM, YM, XFIRSTT, YFIRSTT, DISTNMM)
WRITE(*, 97,)T,DISTNMT,ALTNMT,DISTNMM,ALTNMM,XNCG,DELV
,ATPLOSG
WRITE(1, 97)T,DISTNMT,ALTNMT,DISTNMM,ALTNMM,XNCG,DELV
,ATPLOSG
WRITE(*,*)T,RTM,DELV
PAUSE
CLOSE(1)
END

SUBROUTINE DISTANCE(XT,YT,XF,YF,DISTNM)
SAVE
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
R=SQRT(XT**2 + YT**2)
RF=SQRT(XF**2+YF**2)
A=2.0926E7
CBETA=(XT*XF+YT*YF)/(R*RF)
IF(CBETA<1.)THEN
    BETA=ACOS(CBETA)
    DISTNM=A*BETA/6076.
ELSE
    DISTNM=(XT-XF)6076.
ENDIF
RETURN
END

SUBROUTINE PREDICT (TDUM,TF,XDUM,YDUM,X1DUM,Y1DUM,XTF
1      ,YTF,WP1,WTOT,TB1,TRST1,TB2,WP2,WTOT2,TRST2,WPAY
2      ,XMDUM,YMDUM,X1MDUM,Y1MDUM,ZEM1,ZEM2,TGO)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER STEP
SAVE
IF(TGO>1)THEN
    H=.01
ELSE
    H=TGO
ENDIF
A=2.0926E7
GM=1.4077E16
T=TDUM
X=XDUM
Y=YDUM
X1=X1DUM
Y1=Y1DUM
XM=XMDUM
YM=YMDUM
X1M=X1MDUM
Y1M=Y1MDUM
10   IF(T>(TF-.00001))GOTO 999

```

Listing 16.1 (Continued)

```

XOLD=X
YOLD=Y
X1OLD=X1
Y1OLD=Y1
XOLDM=XM
YOLDM=YM
X1OLDM=X1M
Y1OLDM=Y1M
STEP=1
GOTO 200
66 STEP=2
X=X+H*XD
Y=Y+H*YD
X1=X1+H*X1D
Y1=Y1+H*Y1D
XM=XM+H*XDM
YM=YM+H*YDM
X1M=X1M+H*X1DM
Y1M=Y1M+H*Y1DM
T=T+H
GOTO 200
55 X=(XOLD+X)/2+.5*H*XD
Y=(YOLD+Y)/2+.5*H*YD
X1=(X1OLD+X1)/2+.5*H*X1D
Y1=(Y1OLD+Y1)/2+.5*H*Y1D
XM=(XOLDM+XM)/2+.5*H*XDM
YM=(YOLDM+YM)/2+.5*H*YDM
X1M=(X1OLDM+X1M)/2+.5*H*X1DM
Y1M=(Y1OLDM+Y1M)/2+.5*H*Y1DM
GOTO 10
200 CONTINUE
IF(T<TB1)THEN
    WGT=-WP1*T/TB1+WTOT
    TRST=TRST1
ELSEIF(T<(TB1+TB2))THEN
    WGT=-WP2*T/TB2+WTOT2+WP2*TB1/TB2
    TRST=TRST2
ELSE
    WGT=WPAY
    TRST=0.
ENDIF
AT=32.2*TRST/WGT
VEL=SQRT(X1**2+Y1**2)
AXT=AT*X1/VEL
AYT=AT*Y1/VEL
TEMBOOTT=(X**2 + Y**2)**1.5
X1D=-GM*X/TEMBOOTT+AXT
Y1D=-GM*Y/TEMBOOTT+AYT
XD=X1

```

(Contd.)

Listing 16.1 (Continued)

```

YD=Y1
TEMBOTM=(XM**2+YM**2)**1.5
X1DM=-GM*XM/TEMBOTM
Y1DM=-GM*YM/TEMBOTM
XDM=X1M
YDM=Y1M
IF(STEP-1)66,66,55
999 CONTINUE
XTF=X
YTF=Y
ZEM1=X-XM
ZEM2=Y-YM
RETURN
END

SUBROUTINE LAMBERT(XIC,YIC,TFDES,XF,YF,VRX,VRY,XLONGM
,XLONGT)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
A=2.0926E7
GM=1.4077E16
RIC=SQRT(XIC**2+YIC**2)
RF=SQRT(XF**2+YF**2)
CPHI=(XIC*XF+YIC*YF)/(RIC*RF)
PHI=ACOS(CPHI)
SPHI=SIN(PHI)
R0=RIC
PI=3.14159
DEGRAD=360./(2.*PI)
ICOUNT=1
GMIN=ATAN2((SPHI-SQRT(2.*R0*(1.-CPHI)/RF)), (1-CPHI))
GMAX=ATAN2((SPHI-SQRT(2.*R0*(1.-CPHI)/RF)), (1-CPHI))
GAM=(GAMIN+GMAX)/2.
DO
    TOP=GM*(1.-COS(PHI))
    TEMP=R0*COS(GAM)/RF-COS(PHI+GAM)
    BOT=R0*COS(GAM)*TEMP
    V=SQRT(TOP/BOT)
    IF (XLONGT>XLONGM) THEN
        VRX=V*COS(PI/2.-GAM+XLONGM)
        VRY=V*SIN(PI/2.-GAM+XLONGM)
    ELSE
        VRX=V*COS(-PI/2.+GAM+XLONGM)
        VRY=V*SIN(-PI/2.+GAM+XLONGM)
    END IF
    XLAM=R0*V*V/GM
    TOP1=TAN(GAM)*(1-COS(PHI))+(1-XLAM)*SIN(PHI)
    BOT1P=(1-COS(PHI))/(XLAM*COS(GAM)*COS(GAM))
    BOT1=(2-XLAM)*(BOT1P+COS(GAM+PHI)/COS(GAM))
    TOP2=2*COS(GAM)
    BOT2=XLAM*((2/XLAM-1)**1.5)

```

(Contd.)

Listing 16.1 (Continued)

```

TOP3=SQRT(2/XLAM-1)
BOT3=COS(GAM)/TAN(PHI/2)-SIN(GAM)
TEMP=(TOP2/BOT2)*ATAN2(TOP3,BOT3)
TF=R0*(TOP1/BOT1+TEMP)/(V*COS(GAM))
IF(( ABS (TFDES-TF)<=.0000001*TFDES). OR. ICOUNT>100)THEN
    EXIT
ENDIF
IF(TF>TFDES)THEN
    GMAX=GAM
ELSE
    GMIN=GAM
ENDIF
IF(ICOUNT.EQ.1)THEN
    XNEXT=(GMAX+GMIN)/2.
ELSE
    XNEXT=GAM+(GAM-GOLD)*(TFDES-TF)/TF-TOLD
    IF (XNEXT>GMAX.OR.XNEXT<GMIN)THEN
        XNEXT=(GMAX+GMIN)/2.
    ENDIF
ENDIF
GOLD=GAM
TOLD=TF
GAM=XNEXT
ICOUNT=ICOUNT+1
REPEAT
RETURN
END

```

throughput reasons the integration step size h in subroutine PREDICT had to be increased by two orders of magnitude (from .01 s to 1 s), we can see from Fig. 16.5 that the missile acceleration response would be unstable.

The divert requirements for predictive guidance are quite small because the prediction is perfect. In other words, we have a perfect model of the missile and target and a perfect numerical integration technique. The integration is perfect because the same method (second-order Runge-Kutta) and step size are used in the simulation and prediction portion of the program. In a practical application of predictive guidance, a larger integration step size h might have to be used to satisfy flight computer constraints.

Figure 16.6 presents the missile lateral divert requirement profiles when the integration step size h in the prediction subroutine is 0.01 s (nominal), 0.1 s, and 1 s. We can see that the miss distance and lateral divert increase as the integration step size grows. In fact we can see by comparing Figs. 16.6 and 15.13 that when the integration step size is 1 s the lateral divert requirements are greater with predictive guidance than for augmented proportional navigation. By comparing Fig. 16.6 with Fig. 15.12 we might conclude that the divert requirements for predictive guidance are still smaller than those of proportional navigation, but Fig. 16.6 indicates that predictive guidance misses the target with the larger integration step size.

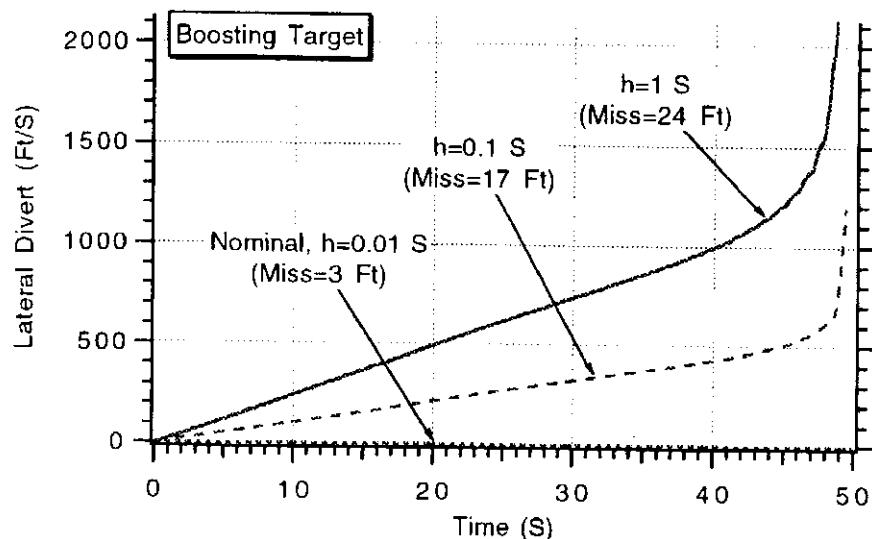


Fig. 16.6 Divert requirements for predictive increase as the integration step size increases.

Booster Estimation with Range and Angle Measurements

Thus far, except for Chapters 7 and 9, we have assumed that information concerning all the target states was available. These states were used in all of the guidance concepts used in the text. In this section we shall show a simple, if not optimal, way of estimating the booster states required for predictive guidance given that a range and angle measurement are available. Consider the model of Fig. 16.7 in which it is assumed that the missile location (x_M, y_M) is known and that range from the missile to the target R_{TM} and the line-of-sight angle λ are measured. Estimating the location, velocity, and acceleration of the target is desired.

From Fig. 16.7 we can see that the location of the target can be expressed as

$$x_T = x_M + R_{TM} \cos \lambda$$

$$y_T = y_M + R_{TM} \sin \lambda$$

If we assume that range is measured perfectly and that the line-of-sight angle measurement is contaminated with noise, λ_{NOISE} , then the measurements of target location x_T^* and y_T^* are given by

$$x_T^* = x_M + R_{TM} \cos(\lambda + \lambda_{NOISE})$$

$$y_T^* = y_M + R_{TM} \sin(\lambda + \lambda_{NOISE})$$

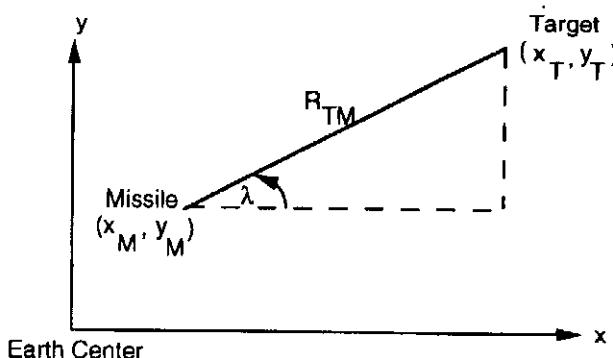


Fig. 16.7 Model for estimating booster states given range and angle information.

The preceding measurement equations are not in a suitable form for an optimal linear Kalman filter since the measurement is not a linear function of the states. However, we will ignore that fact and use the simple polynomial Kalman filters, one for x_T^* and the other for y_T^* , which were derived in Chapter 9. Before we begin we must find some way of finding the equivalent positional noise, σ_x and σ_y . The theoretical positional noise will be used in the Riccati equations to derive gains for each of the polynomial Kalman filters. From the chain rule we know that

$$\Delta x_T = \frac{\partial x_T}{\partial \lambda} \Delta \lambda = -R_{TM} \sin \lambda \Delta \lambda$$

$$\Delta y_T = \frac{\partial y_T}{\partial \lambda} \Delta \lambda = R_{TM} \cos \lambda \Delta \lambda$$

We can find the variance of the positional noise by squaring both of the preceding equations and then taking expectations yielding

$$\sigma_{x_T}^2 = E[\Delta x_T \Delta x_T^T] = R_{TM}^2 \sin^2 \lambda \sigma_\lambda^2$$

$$\sigma_{y_T}^2 = E[\Delta y_T \Delta y_T^T] = R_{TM}^2 \cos^2 \lambda \sigma_\lambda^2$$

Both polynomial Kalman filters were incorporated into the booster engagement simulation of Listing 15.2, and the resultant simulation is shown in Listing 16.2. In this simulation augmented proportional navigation guidance makes use of the actual states. The filter is only in the simulation to see how well the target states can be estimated. We can see from the simulation listing that there is 1 mr of angle noise and data is being collected every 0.1 s. The actual position and velocity filter states have been initialized perfectly, but the initial acceleration estimate has been set to zero. The actual position noise has been computed so it could be checked against the theoretical formulas that were just derived.

Before the Kalman filter estimates were evaluated, a check of the formulas for the theoretical positional noise was made. We can see from Figs. 16.8 and 16.9 that the single flight simulated positional noise lies between the previously derived formulas for the positional noise approximately 68% of the time, indicating that

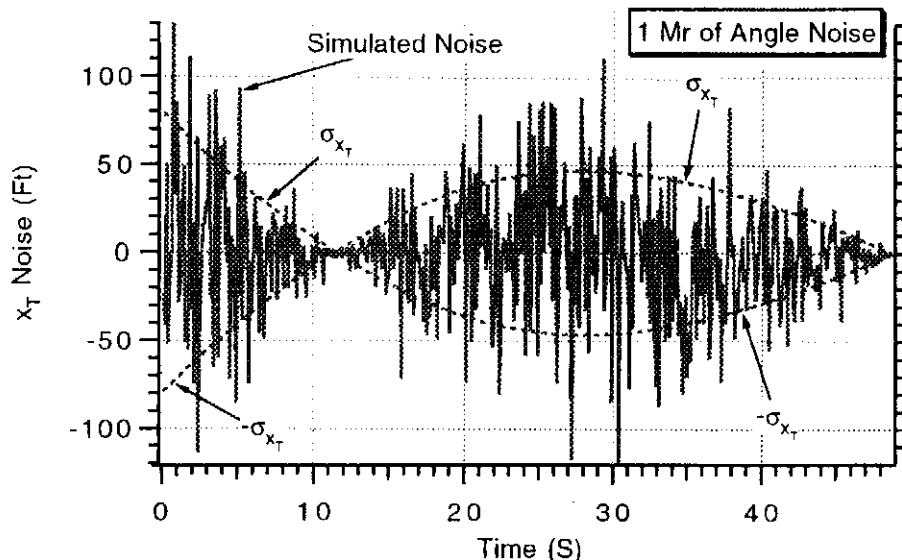


Fig. 16.8 Formula for x_T positional noise is correct.

Listing 16.2 Booster engagement simulation with Kalman filter based on range and angle measurement

```

GLOBAL DEFINE
    INCLUDE 'quickdraw.inc'
END
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
REAL*8 M11,M12,M13,M22,M23,M33,K1,K2,K3
REAL*8 M11P,M12P,M13P,M22P,M23P,M33P,K1P,K2P,K3P
INTEGER STEP
OPEN (1, STATUS='UNKNOWN', FILE = 'DATFIL')
A=2.0926E7
GM=1.4077E16
XNP=3.
TS=.1.
XLONGTDEG=90.
XLONGMDEG=85.
DEGRAD=57.3
SIGLAM=.001
PREDERR=0.
XISP1=250.
XISP2=250.
XMF1=.85
XMF2=.85
WPAY=100.
DELV=20000.
DELV1=.3333*DELV
DELV2=.6667*DELV
AMAX1=20.
AMAX2=20.
XKICKDEG=80.
TOP2=WPAY*(EXP(DELV2/(XISP2*32.2))-1.)
BOT2=1/XMF2-((1.-XMF2)/XMF2)*EXP(DELV2/(XISP2*32.2))
WP2=TOP2/BOT2
WS2=WP2*(1-XMF2)/XMF2
WTOT2=WP2+WS2+WPAY
TRST2=AMAX2*(WPAY+WS2)
TB2=XISP2*WP2/TRST2
TOP1=WTOT2*(EXP(DELV1/(XISP1*32.2))-1.)
BOT1=1/XMF1-((1.-XMF1)/XMF1)*EXP(DELV1/(XISP1*32.2))
WP1=TOP1/BOT1
WS1=WP1*(1-XMF1)/XMF1
WTOT=WP1+WS1+WTOT2
TRST1=AMAX1*(WTOT2+WS1)
TB1=XISP1*WP1/TRST1
ALTNMT=0.
ALTNMM=0.
ALTT=ALTNMT*6076.
ALTM=ALTNMM*6076.
S=0.
SCOUNT=0.

```

(Contd.)

Listing 16.2 (Continued)

```

XLONGM=XLONGMDEG/DEGRAD
XLONGT=XLONGTDEG/DEGRAD
XM=A*COS(XLONGM)
YM=A*SIN(XLONGM)
XT=(A+ALTT)*COS(XLONGT)
YT=(A+ALTT)*SIN(XLONGT)
XFIRSTT=XT
YFIRSTT=YT
X1T=COS(XKICKDEG/57.3)
Y1T=SIN(XKICKDEG/57.3)

K1=0.
K2=0.
K3=0.
K1P=0.
K2P=0.
K3P=0.
XTH=XT
XTDH=X1T
XTDDH=0.
YTH=YT
YTDH=Y1T
YTDDH=0.
T=0.
TF=50.
PHIN=100*100/TF
CALL PREDICT (TF,XT,YT,X1T,Y1T,XTF,YTF
1 ,WP1,WTOT,TB1,TRST1,TB2,WP2,WTOT2,TRST2,WPAY)
YTF=YTF+PREDERR
CALL LAMBERT(XM,YM,TF,XTF,YTF,VRXM,VRYM,XLONGM,XLONGT)
1 ,ICOUNT)
X1M=VRXM
Y1M=VRYM
RTM1=XT-XM
RTM2=YT-YM
RTM=SQRT(RTM1**2+RTM2**2)
XLAM=ATAN2(RTM2, RTM1)
SIGX2=(SIGLAM*RTM*SIN(XLAM)) **2
SIGX=SQRT(SIGX2)
P11=SIGX2
P12=0.
P13=0.
P22=0.
P23=0.
P33=100*100
SIGY2=(SIGLAM*RTM*COS(XLAM)) **2
SIGY=SQRT(SIGY2)
P11P=SIGY2
P12P=0.
P13P=0.

```

(Contd.)

Listing 16.2 (Continued)

```

P22P=0.
P23P=0.
P33P=100*100
VTM1=X1T-X1M
VTM2=Y1T-Y1M
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
DELV=0.
10 IF(VC>0.)GOTO 999
    TGO=RTM/VC
    IF(TGO>.1)THEN
        H=.01
    ELSE
        H=.0001
    ENDIF
    XOLDT=XT
    YOLDT=YT
    X1OLDT=X1T
    Y1OLDT=Y1T
    XOLDM=XM
    YOLDM=YM
    X1OLDM=X1M
    Y1OLDM=Y1M
    DELVOLD=DELV
    STEP=1
    GOTO 200
66 STEP=2
    XT=XT+H*XDT
    YT=YT+H*YDT
    X1T=X1T+H*X1DT
    Y1T=Y1T+H*Y1DT
    XM=XM+H*XDM
    YM=YM+H*YDM
    X1M=X1M+H*X1DM
    Y1M=Y1M+H*Y1DM
    DELV=DELV+H*DELVD
    T=T+H
    GOTO 200
55 XT=(XOLDT+XT)/2+.5*H*XDT
    YT=(YOLDT+YT)/2+.5*H*YDT
    X1T=(X1OLDT+X1T)/2+.5*H*X1DT
    Y1T=(Y1OLDT+Y1T)/2+.5*H*Y1DT
    XM=(XOLDM+XM)/2+.5*H*XDM
    YM=(YOLDM+YM)/2+.5*H*YDM
    X1M=(X1OLDM+X1M)/2+.5*H*X1DM
    Y1M=(Y1OLDM+Y1M)/2+.5*H*Y1DM
    DELV=(DELVOLD+DELV)/2+.5*H*DELVD
    ALTT=SQRT(XT**2+YT**2)-A
    ALTM=SQRT(XM**2+YM**2)-A
    S=S+H
    SCOUNT=SCOUNT+H

```

(Contd.)

Listing 16.2 (Continued)

```

IF(SCOUNT.LT.(TS-.00001))GOTO 10
SCOUNT=0.
TS2=TS*TS
TS3=TS2*TS
TS4=TS3*TS
TS5=TS4*TS
SIGX2=SIGLAM*RTM*SIN(XLAM)) **2
SIGY2=SIGLAM*RTM*COS(XLAM)) **2
SIGX=SQRT(SIGX2)
SIGY=SQRT(SIGY2)
M11=P11+TS*P12+.5*TS2*P13+TS*(P12+TS*P22+.5*TS2*P23)
M11=M11+.5*TS2*(P13+TS*P23+.5*TS2*P33)+TS5*PHIN/20.
M12=P12+TS*P22+.5*TS2*P23+TS*(P13+TS*P23+.5*TS2*P33)+TS4*PHIN/8.
M13=P13+TS*P23+.5*TS2*P33+PHIN*TS3/6.
M22=P22+TS*P23+TS*(P23+TS*P33)+PHIN*TS3/3.
M23=P23+TS*P33+.5*TS2*PHIN
M33=P33+PHIN*TS
BOT=M11+SIGX2
K1=M11/BOT
K2=M12/BOT
K3=M13/BOT
FACT=1.-K1
P11=FACT*M11
P12=FACT*M12
P13=FACT*M13
P22=-K2*M12+M22
P23=-K2*M13+M23
P33=-K3*M13+M33
M11P=P11P+TS*P12P+.5*TS2*P13P+TS*(P12P+TS*P22P+.5*TS2*P23P)
M11P=M11P+.5*TS2*(P13P+TS*P23P+.5*TS2*P33P)+TS5*PHIN/20.
M12P=P12P+TS*P22P+.5*TS2*P23P+TS*(P13P+TS*P23P+.5*TS2*P33P)+  

1   TS4*PHIN/8.
M13P=P13P+TS*P23P+.5*TS2*P33P+PHIN*TS3/6.
M22P=P22P+TS*P23P+TS*(P23P+TS*P33P)+PHIN*TS3/3.
M23P=P23P+TS*P33P+.5*TS2*PHIN
M33P=P33P+PHIN*TS
BOTP=M11P+SIGY2
K1P=M11P/BOTP
K2P=M12P/BOTP
K3P=M13P/BOTP
FACTP=1.-K1P
P11P=FACTP*M11P
P12P=FACTP*M12P
P13P=FACTP*M13P
P22P=-K2P*M12P+M22P
P23P=-K2P*M13P+M23P
P33P=-K3P*M13P+M33P
CALL GAUSS(XLAMNOISE,SIGLAM)
YTMEAS=YM+RTM*SIN(XLAM+XLAMNOISE)
XTMEAS=XM+RTM*COS(XLAM+XLAMNOISE)

```

Listing 16.2 (Continued)

```

XNOISE=XT-XTMEAS
YNOISE=YT-YTMEAS
RESX=XTMEAS-XTH-TS*XTDH-.5*TS2*XTDDH
XTH=K1*RESX+XTH+TS*XTDH+.5*TS2*XTDDH
XTDH=K2*RESX+XTDH+TS*XTDDH
XTDDH=K3*RESX+XTDDH
RESY=YTMEAS-YTH-TS*YTDH-.5*TS2*YTDDH
YTH=K1P*RESY+YTH+TS*YTDH+.5*TS2*YTDDH
YTDH=K2P*RESY+YTDH+TS*YTDDH
YTDDH=K3P*RESY+YTDDH
XNMT=XT/6076.
YNMT=YT/6076.
XNMM=XM/6076.
YNMM=YM/6076.
ALTNMT=ALTT/6076.
CALL DISTANCE(XT,YT,XFIRSTT,YFIRSTT,DISTNMT)
ALTNMM=ALTM/6076.
CALL DISTANCE(XM,YM,XFIRSTT,YFIRSTT,DISTNMM)
X1DTG=X1DT/32.2
XTDDHG=XTDDH/32.2
Y1DTG=Y1DT/32.2
YTDDHG=YTDDH/32.2
WRITE(*, 97)T, XNOISE, SIGX, -SIGX, YNOISE, SIGY, -SIGY, X1DTG
1 , XTDDHG, Y1DTG, YTDDHG
WRITE(1,97)T, XNOISE, SIGX, -SIGX, YNOISE, SIGY, -SIGY, X1DTG
1 , XTDDHG, Y1DTG, YTDDHG
GOTO 10
97 FORMAT (11F10.3)
200 CONTINUE
IF(T<TB1)THEN
    WGT=-WP1*T/TB1+WTOT
    TRST=TRST1
ELSEIF(T<(TB1+TB2))THEN
    WGT=-WP2*T/TB2+WTOT2+WP2*TB1/TB2
    TRST=TRST2
ELSE
    WGT=WPAY
    TRST=0.
ENDIF
AT=32.2*TRST/WGT
VEL=SQRT(X1T**2+Y1T**2)
AXT=AT*X1T/VEL
AYT=AT*Y1T/VEL
TEMBO=(XT**2+YT**2)**1.5
X1DT=-GM*XT/TEMBO+AXT
Y1DT=-GM*YT/TEMBO+AYT
ATPLOS=Y1DT*COS(XLAM)-X1DT*SIN(XLAM)
XDT=X1T
YDT=Y1T
RTM1=XT-XM

```

(Contd.)

Listing 16.2 (Continued)

```

RTM2=YT-YM
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=X1T-X1M
VTM2=Y1T-Y1M
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
TGO=RTM/VC
XLAM=ATAN2(RTM2,RTM1)
XLAMD=(RTM1*VTM2-RTM2*VTM1)/(RTM*RTM)
XNC=XNP*VC*XLAMD+.5*XNP*ATPLOS
DELVD=ABS(XNC)
AM1=-XNC*SIN(XLAM)
AM2=XNC*COS(XLAM)
TEMBOTM=(XM**2+YM**2)**1.5
X1DM=-GM*XM/TEMBOTM+AM1
Y1DM=-GM*YM/TEMBOTM+AM2
XDM=X1M
YDM=Y1M
IF(STEP-1)66,66,55
999 CONTINUE
XNMT=XT/6076.
YNMT=YT/6076.
XNMM=XM/6076.
YNMM=YM/6076.
ALTNMT=ALTT/6076.
CALL DISTANCE(XT,YT,XFIRSTT,YFIRSTT,DISTNMT)
ALTNMM=ALTM/6076.
CALL DISTANCE(XM,YM,XFIRSTT,YFIRSTT,DISTNMM)
WRITE(*,*)T,RTM,DELV
PAUSE
CLOSE(1)
END

SUBROUTINE DISTANCE(XT,YT,XF,YF,DISTNM)
SAVE
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
R=SQRT(XT**2+YT**2)
RF=SQRT(XF**2+YF**2)
A=2.0926E7
CBETA=(XT*XF+YT*YF)/(R*RF)
IF(CBETA<1.)THEN
    BETA=ACOS(CBETA)
    DISTNM=A*BETA/6076.
ELSE
    DISTNM=(XT-XF)/6076.
ENDIF
RETURN
END

```

(Contd.)

Listing 16.2 (Continued)

```

SUBROUTINE PREDICT (TF,XDUM,YDUM,X1DUM,Y1DUM,XTF,YTF
1 ,WP1,WTOT,TB1,TRST1,TB2,WP2,WTOT2,TRST2,WPAY)
1 IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER STEP
SAVE
H=.01.
A=2.0926E7
GM=1.4077E16
T=0.
X=XDUM
Y=YDUM
X1=X1DUM
Y1=Y1DUM
10 IF(T>(TF-.00001))GOTO 999
XOLD=X
YOLD=Y
X1OLD=X1
Y1OLD=Y1
STEP=1
GOTO 200
66 STEP=2
X=X+H*XD
Y=Y+H*YD
X1=X1+H*X1D
Y1=Y1+H*Y1D
T=T+H
GOTO 200
55 X=(XOLD+X)/2+.5*H*XD
Y=(YOLD+Y)/2+.5*H*YD
X1=(X1OLD+X1)/2+.5*H*X1D
Y1=(Y1OLD+Y1)/2+.5*H*Y1D
GOTO 10
200 CONTINUE
IF(T<TB1)THEN
    WGT=-WP1*T/TB1+WTOT
    TRST=TRST1
ELSEIF(T<(TB1+TB2))THEN
    WGT=-WP2*T/TB2+WTOT2+WP2*TB1/TB2
    TRST=TRST2
ELSE
    WGT=WPAY
    TRST=0.
ENDIF
AT=32.2*TRST/WGT
VEL=SQRT(X1**2+Y1**2)
AXT=AT*X1/VEL
AYT=AT*Y1/VEL
TEMBOOTT=(X**2+Y**2)**1.5
X1D=-GM*X/TEMBOOTT+AXT

```

(Contd.)

Listing 16.2 (Continued)

```

Y1D=-GM*Y/TEMBOTT+AYT
XD=X1
YD=Y1
IF(STEP-1)66,66,55
999  CONTINUE
XTF=X
YTF=Y
RETURN
END

SUBROUTINE LAMBERT(XIC,YIC,TFDES,XF,YF,VRX,VRY,XLONGM
,XLONGT,ICOUNT)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
A=2.0926E7
GM =1.4077E16
RIC=SQRT(XIC**2+YIC**2)
RF=SQRT(XF**2+YF**2)
CPHI=(XIC*XF+YIC*YF)/(RIC*RF)
PHI=ACOS(CPHI)
SPHI=SIN(PHI)
R0=RIC
PI=3.14159
DEGRAD=360./(2.*PI)
ICOUNT=1
GMIN=ATAN2 ((SPHI-SQRT(2.*R0*(1.-CPHI)/RF)), (1-CPHI)
GMAX=ATAN2 ((SPHI+SQRT(2.*R0*(1.-CPHI)/RF)), (1-CPHI)
GAM=(GMIN+GMAX)/2.
DO
    TOP=GM*(1.-COS(PHI))
    TEMP=R0*COS(GAM)/RF-COS(PHI+GAM)
    BOT=R0*COS(GAM)*TEMP
    V=SQRT(TOP/BOT)
    IF (XLONGT>XLONGM) THEN
        VRX=V*COS(PI/2.-GAM+XLONGM)
        VRY=V*SIN(PI/2.-GAM+XLONGM)
    ELSE
        VRX=V*COS(-PI/2.+GAM+XLONGM)
        VRY=V*SIN(-PI/2.+GAM+XLONGM)
    END IF
    XLAM=R0*V*V/GM
    TOP1=TAN(GAM)*(1-COS(PHI))+(1-XLAM)*SIN(PHI)
    BOT1P=(1-COS(PHI))/(XLAM*COS(GAM)*COS(GAM))
    BOT1=(2-XLAM)*(BOT1P+COS(GAM+PHI)/COS(GAM))
    TOP2=2*COS(GAM)
    BOT2=XLAM*((2/XLAM-1)**1.5)
    TOP3=SQRT(2/XLAM-1)
    BOT3=COS(GAM)/TAN(PHI/2)-SIN(GAM)
    TEMP=(TOP2/BOT2)*ATAN2(TOP3,BOT3)

```

(Contd.)

Listing 16.2 (Continued)

```

TF=R0*(TOP1/BOT1+TEMP)/(V*COS(GAM))
IF((ABS(TFDES-TF)<=.00000001*TFDES).OR.ICOUNT>100)THEN
    EXIT
ENDIF
IF(TF>TFDES) THEN
    GMAX=GAM
ELSE
    GMIN=GAM
ENDIF
IF(ICOUNT.EQ.1)THEN
    XNEXT=(GMAX+GMIN)/2.
ELSE
    XNEXT=GAM+(GAM-GOLD)*(TFDES-TF)/(TF-TOLD)
    IF(XNEXT>GMAX.OR.XNEXT<GMIN)THEN
        XNEXT=(GMAX+GMIN)/2.
    ENDIF
ENDIF
GOLD=GAM
TOLD=TF
GAM=XNEXT
ICOUNT=ICOUNT+1
REPEAT
RETURN
END

SUBROUTINE GAUSS(X,SIG)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER SUM
SUM=0
DO 14 J=1,6
IRAN=Random()
SUM=SUM+IRAN
14 CONTINUE
X=SUM/65536.
X=1.414*X*SIG
RETURN
END

```

the formulas are correct. We can also see that there is much more positional noise in y_T than there is in x_T for this particular geometry. Therefore we would expect the Kalman filter in the x coordinate to behave differently than the one in the y coordinate.

For our example the most difficult estimate is that of booster acceleration. Figures 16.10 and 16.11 show how the filter estimate of acceleration in the x coordinate compares to the filter estimate of acceleration along the y coordinate. The filter estimate along the x coordinate is much better along x than along y because there is much less positional noise along x than along y . In fact the staging event is detected along x within a few seconds, but it takes at least 10 s along y to detect the staging event.

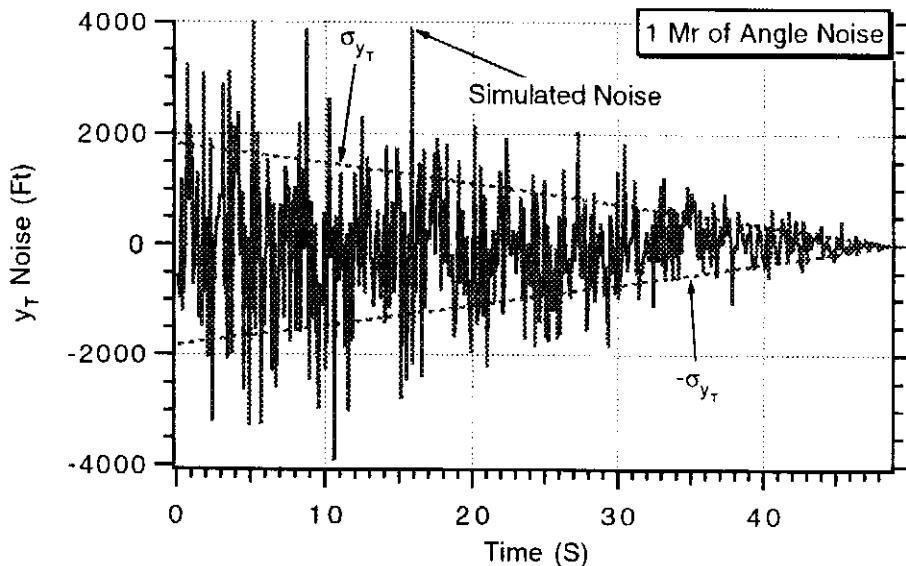


Fig. 16.9 Formula for y_T positional noise is correct.

The quality of the estimates from the Kalman filter depends on the measurement noise level. Although the filter adjusts its bandwidth to accommodate different noise intensities, in general, estimation accuracy will improve as the measurement noise level decreases. Figure 16.12 shows that if the measurement noise could be decreased by an order of magnitude (i.e., from 1 to 0.1 mr) then the detection time for the staging event in the y coordinate can be reduced from more than 10 s to nearly 5 s.

Other factors such as inaccurate range measurements would cause filter performance to degrade. In each application, filter performance must be evaluated with appropriate noise levels to determine if the estimates are of sufficient quality to be used for purposes of detection or possibly to be used with more advanced guidance techniques. If the estimates are of poor quality, the proportional navigation guidance law, which does not need acceleration information, may yield the best overall system performance.

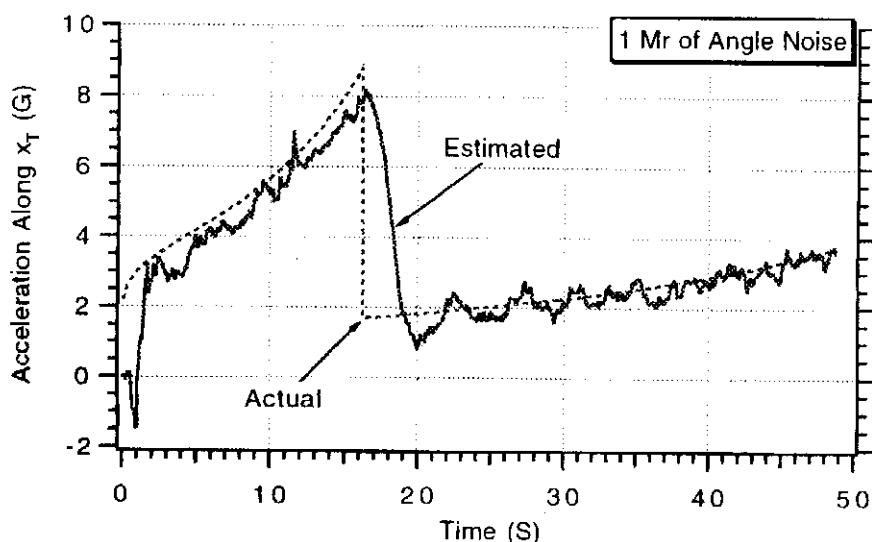


Fig. 16.10 Kalman filter in x can detect staging event within a few seconds.

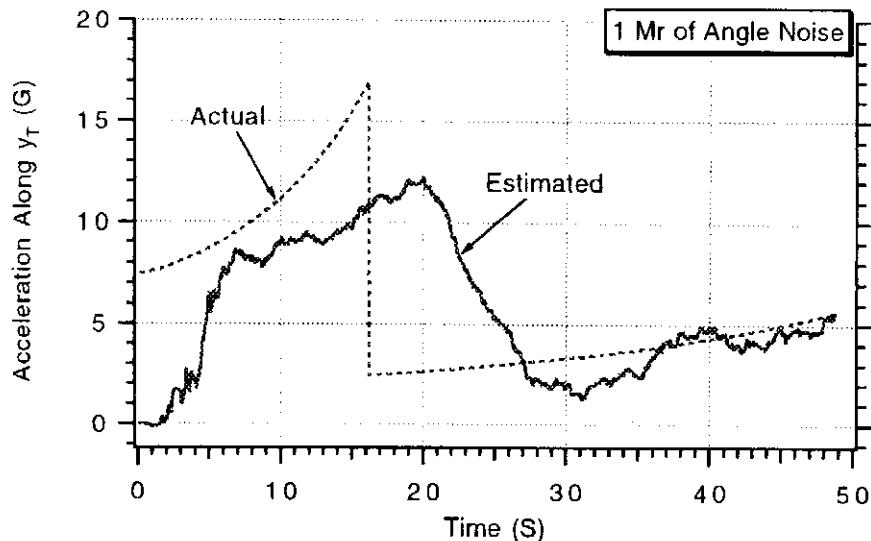


Fig. 16.11 Kalman filter in y takes more than 10 s to detect staging event.

Pulsed Guidance

A strategic interceptor maneuvers with divert engines. Sometimes the engines are effectively throttleable. This means that, by issuing divert commands in opposite directions, any effective acceleration level (within engine thrust-to-weight ratio constraints) can be reached. This is ideal for the implementation of proportional navigation type guidance laws. However, sometimes it is only possible to issue guidance commands of fixed amplitude when the engine is on. In this case we can only influence the duration of the guidance pulse by turning the engine off. In this situation it is not obvious how a guidance law can be implemented or if proportional navigation is appropriate.

Consider the acceleration diagram in Fig. 16.13. Here we have a guidance pulse of magnitude a ft 2 /s lasting for Δt seconds. Implicit in the diagram is the approximation that the acceleration level is constant for the duration of the pulse. Actually, the acceleration level would be increasing for a constant thrust level because of

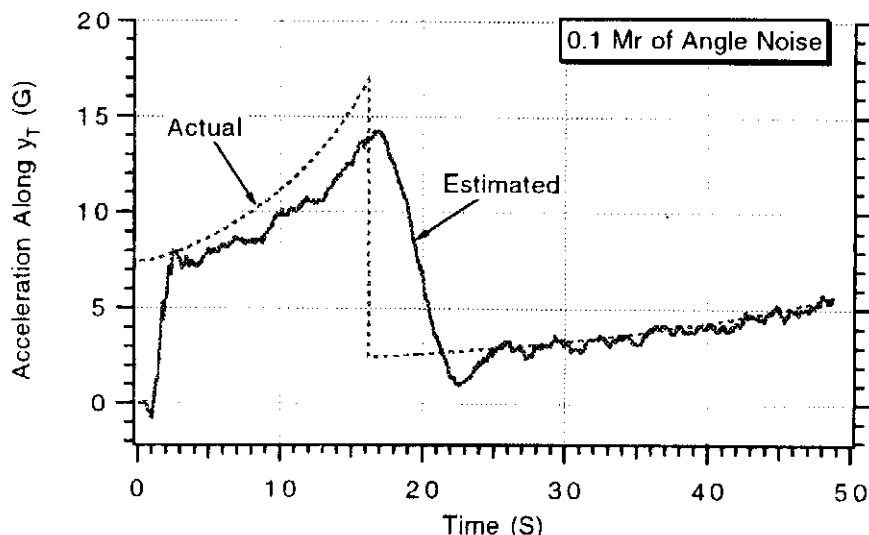


Fig. 16.12 Reducing angle noise improves filter's estimates.

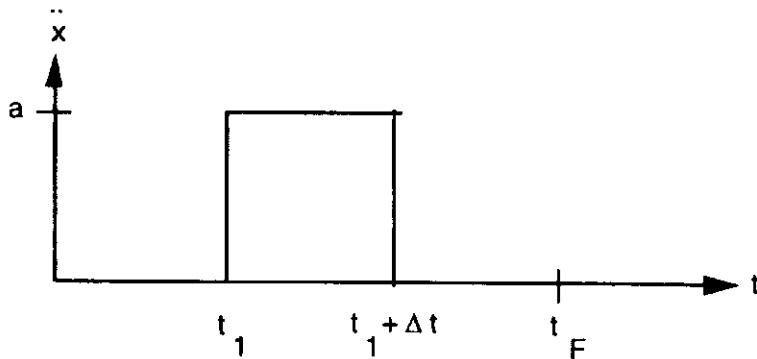


Fig. 16.13 Conceptual diagram for pulsed guidance law.

the expenditure of fuel. We wish to derive a guidance law whose output for a given acceleration magnitude a contains information concerning the duration of the guidance pulse Δt .

We have previously seen how proportional navigation is related to the zero effort miss (ZEM). In proportional navigation type guidance, we take the entire guided portion of the flight to remove the zero effort miss. If we have a single pulse, as shown in Fig. 16.13, we must remove the zero effort miss in Δt seconds. For a pulse of amplitude a and duration Δt , we can integrate the acceleration twice yielding the velocity profile of Fig. 16.14 and the position profile of Fig. 16.15. Using the information from Fig. 16.15 we can equate the distance traveled due to the acceleration pulse (derived by integrating the acceleration pulse twice) to the zero effort miss or

$$ZEM = .5a\Delta t^2 + a\Delta t(t_{go} - \Delta t)$$

However, we also know that the proportional navigation guidance law can be expressed as

$$n_c = N'V_c\dot{\lambda} = \frac{N'ZEM}{t_{go}^2}$$

Therefore, the zero effort miss can be expressed in terms of the line-of-sight rate as

$$ZEM = V_c t_{go}^2 \dot{\lambda}$$

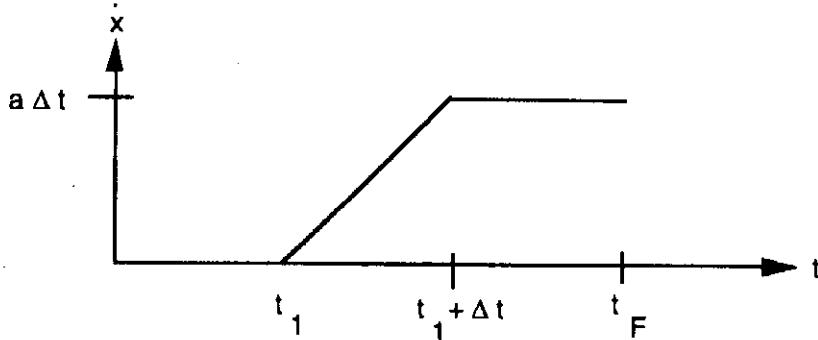


Fig. 16.14 Integrating acceleration pulse once yields velocity.

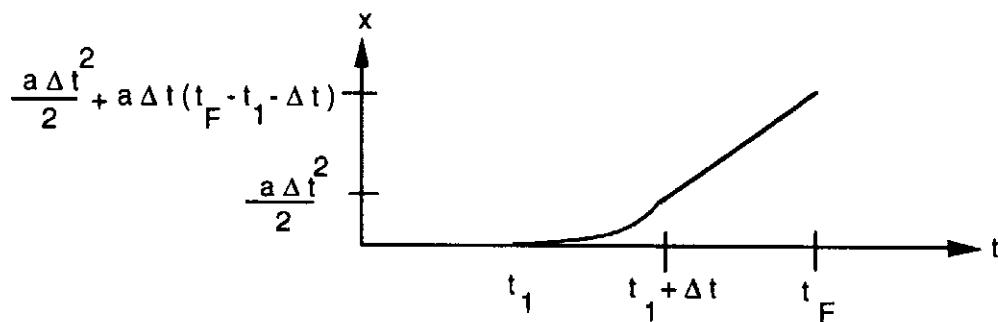


Fig. 16.15 Integrating acceleration pulse twice yields position.

Equating the expressions for the proportional navigation zero effort miss with the one for the pulse of fixed duration, we get

$$V_c t_{go}^2 \dot{\lambda} = .5a\Delta t^2 + a\Delta t(t_{go} - \Delta t)$$

This is a quadratic equation in terms of Δt . Using the quadratic formula and eliminating the unrealistic root, we obtain a closed-form expression for the pulse duration time as

$$\Delta t = t_{go} \left[1 - \sqrt{1 - \frac{2V_c}{a} \dot{\lambda}} \right]$$

To implement a pulse guidance scheme, we must know in advance the number of pulses to be used. For simplicity, let us assume that the pulses are equally distributed throughout the flight. If it is time for a pulse to commence, we calculate the pulse duration from the preceding formula. An engagement simulation, based upon Listing 15.1, with a pulsed guidance system is presented in Listing 16.3.

A nominal case was run with the pulsed guidance system in which there was 100 kft of prediction error. Figure 16.16 shows the line-of-sight rate profile for a successful intercept in which the missile had 10 guidance pulses. We can see from the plot that the pulsed guidance system is always trying to drive the line-of-sight rate to zero after each pulse is issued. This is not surprising, since the pulsed guidance law was derived with this concept in mind.

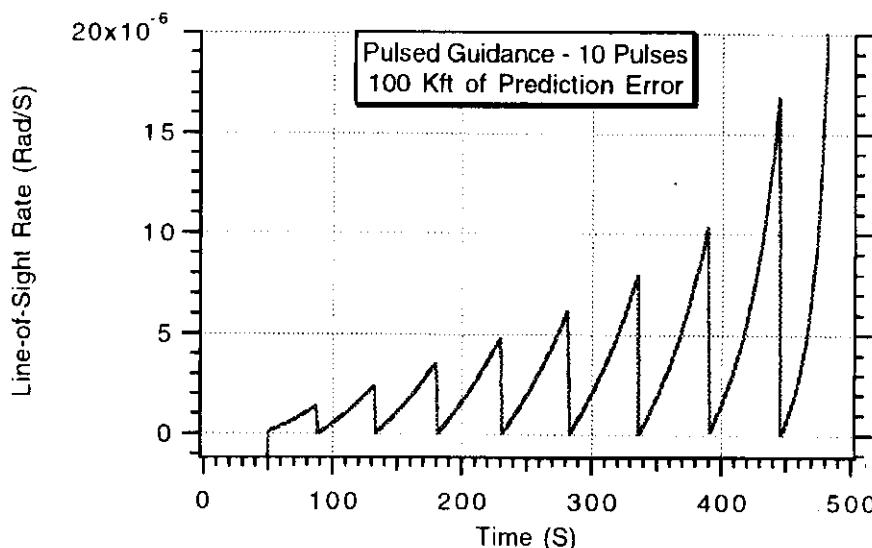


Fig. 16.16 Pulse guidance attempts to drive line-of-sight rate to zero with each pulse.

Listing 16.3 Pulsed guidance simulation

```

IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER STEP
LOGICAL PULSE_ON,ACQUIRE
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
XLONGMDEG=45.
XLONGTDEG=90.
ALTNMTIC=0.
ALTNMMIC=0.
TF=500.
GAMDEGT=23.
AMAG=64.4
PULSES=10.
PREDERR=-100000.
PULSE_ON=.FALSE.
ACQUIRE=.TRUE.
PULSE_NUM=PULSES-1.
H=.01.
A=2.0926E7
GM=1.4077E16
PI=3.14159
DEGRAD=360./(2.*PI)
XNC=0.
GAMT=GAMDEGT/57.3
DISTNMT=6000.
PHIT=DISTNMT*6076./A
ALTT=ALTNMTIC*6076.
ALTM=ALTNMMIC*6076.
R0T=A+ALTT
TOP=GM*(1.-COS(PHIT))
TEMP=R0T*COS(GAMT)/A-COS(PHIT+GAMT)
BOT=R0T*COS(GAMT)*TEMP
VT=SQRT(TOP/BOT)
XLONGM=XLONGMDEG/DEGRAD
XLONGT=XLONGTDEG/DEGRAD
IF (XLONGM>XLONGT) THEN
  X1T=VT*COS(PI/2.-GAMT+XLONGT)
  Y1T=VT*SIN(PI/2.-GAMT+XLONGT)
ELSE
  X1T=VT*COS(-PI/2.+GAMT+XLONGT)
  Y1T=VT*SIN(-PI/2.+GAMT+XLONGT)
END IF
S=0.
SCOUNT=0.
XLONGM=XLONGMDEG/DEGRAD
XLONGT=XLONGTDEG/DEGRAD
XM=(A+ALTM)*COS(XLONGM)
YM=(A+ALTM)*SIN(XLONGM)
XT=(A+ALTT)*COS(XLONGT)
YT=(A+ALTT)*SIN(XLONGT)
XFIRSTT=XT

```

Listing 16.3 (Continued)

```

YFIRSTT=YT
T=0.
CALL PREDICT (TF,XT,YT,X1T,Y1T,XTF,YTF)
YTF=YTF+PREDERR
CALL LAMBERT(XM,YM,TF,XTF,YTF,VRXM,VRYM,XLONGM,XLONGT
,ICOUNT)
X1M=VRXM
Y1M=VRYM
RTM1=XT-XM
RTM2=YT-YM
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=X1T-X1M
VTM2=Y1T-Y1M
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
DELV=0.
10 IF(VC<0.)GOTO 999
TGO=RTM/VC
IF(TGO>.1)THEN
    H=.01
ELSE
    H=.0001
ENDIF
XOLDT=XT
YOLDT=YT
X1OLDT=X1T
Y1OLDT=Y1T
XOLDM=XM
YOLDM=YM
X1OLDM=X1M
Y1OLDM=Y1M
DELVOLD=DELV
STEP=1
GOTO 200
66 STEP=2
XT=XT+H*XDT
YT=YT+H*YDT
X1T=X1T+H*X1DT
Y1T=Y1T+H*Y1DT
XM=XM+H*XDM
YM=YM+H*YDM
X1M=X1M+H*X1DM
Y1M=Y1M+H*Y1DM
DELV=DELV+H*DELVD
T=T+H
GOTO 200
55 XT=(XOLDT+XT)/2+.5*H*XDT
YT=(YOLDT+YT)/2+.5*H*YDT
X1T=(X1OLDT+X1T)/2+.5*H*X1DT
Y1T=(Y1OLDT+Y1T)/2+.5*H*Y1DT

```

(Contd.)

Listing 16.3 (Continued)

```

XM=(XOLDM+XM)/2+.5*H*XDM
YM=(YOLDM+YM)/2+.5*H*YDM
X1M=(X1OLDM+X1M)/2+.5*H*X1DM
Y1M=(Y1OLDM+Y1M)/2+.5*H*Y1DM
DELV=(DELVOLD+DELV)/2+.5*H*DELVD
ALTT=SQRT(XT**2+YT**2)-A
ALTM=SQRT(XM**2+YM**2)-A
IF(PULSE_ON)THEN
    IF(T>TOFF)THEN
        PULSE_ON=.FALSE.
        XNC=0.
    ENDIF
ENDIF
IF(PULSE_NUM>0.)THEN
    IF(TGO<=(TF - 0.)* PULSE_NUM/(PULSES - 1.))THEN
        PULSE_NUM=PULSE_NUM-1.
        PULSE_ON=.TRUE.
        DISC=1.-2.*VC*ABS(XLAMD)/AMAG
        IF(DISC>0.)THEN
            TPULSE=TGO*(1.-SQRT(DISC))
            IF(XLAMD>0.)THEN
                XNC=AMAG
            ELSE
                XNC=-AMAG
            ENDIF
        ELSE
            TPULSE=0.
        ENDIF
        IF(TGO<TPULSE)THEN
            TOFF=9999999.
        ELSE
            TOFF=T+TPULSE
        ENDIF
    ENDIF
ELSE
    DISC=1.-2.*VC*ABS(XLAMD)/AMAG
    IF(DISC>0.)THEN
        TPULSE=TGO*(1.-SQRT(DISC))
    ELSE
        TPULSE=999999.
    ENDIF
    IF (TGO<=TPULSE) THEN
        IF(XLAMD>0.)THEN
            XNC=AMAG
        ELSE
            XNC=-AMAG
        ENDIF
        PULSE_ON=.TRUE.
        TOFF=999999.
    END IF

```

(Contd.)

Listing 16.3 (Continued)

```

END IF
S=S+H
SCOUNT=SCOUNT+H
IF(SCOUNT.LT..99999)GOTO 10
SCOUNT=0.
XNMT=XT/6076.
YNMT=YT/6076.
XNMM=XM/6076.
YNMM=YM/6076.
ALTNMT=ALTT/6076.
CALL DISTANCE(XT,YT,XFIRSTT,YFIRSTT,DISTNMT)
ALTNMM=ALTM/6076.
CALL DISTANCE(XM,YM,XFIRSTT,YFIRSTT,DISTNMM)
WRITE(*,97)T,XNC,XLAMD,DELV
WRITE(1,97)T,XNC,XLAMD,DELV
GOTO 10
97 FORMAT (F413.7)
200 CONTINUE
TEMBO=(XT**2+YT**2)**1.5
X1DT=-GM*XT/TEMBO
Y1DT=-GM*YT/TEMBO
XDT=X1T
YDT=Y1T
RTM1=XT-XM
RTM2=YT-YM
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=X1T-X1M
VTM2=Y1T-Y1M
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
TGO=RTM/VC
XLAM=ATAN2(RTM2,RTM1)
XLAMD=(RTM1*VTM2-RTM2*VTM1)/(RTM*RTM)
DELVD=ABS(XNC)
AM1=-XNC*SIN(XLAM)
AM2=XNC*COS(XLAM)
TEMBOTM=(XM**2+YM**2)**1.5
X1DM=-GM*XM/TEMBOTM+AM1
Y1DM=-GM*YM/TEMBOTM+AM2
XDM=X1M
YDM=Y1M
IF(STEP-1)66,66,55
999 CONTINUE
XNMT=XT/6076.
YNMT=YT/6076.
XNMM=XM/6076.
YNMM=YM/6076.
ALTNMT=ALTT/6076.
CALL DISTANCE(XT,YT,XFIRSTT,YFIRSTT,DISTNMT)
ALTNMM=ALTM/6076.
CALL DISTANCE(XM,YM,XFIRSTT,YFIRSTT,DISTNMM)
WRITE(*, 97)T,XNC,XLAMD,DELV

```

Listing 16.3 (Continued)

```

        WRITE(*,*)T,RTM,DELV
        PAUSE
        CLOSE(1)
        END

        SUBROUTINE DISTANCE(XT,YT,XF,YF,DISTNM)
        SAVE
        IMPLICIT REAL*8 (A-H)
        IMPLICIT REAL*8 (O-Z)
        R=SQRT(XT**2+YT**2)
        RF=SQRT(XF**2+YF**2)
        A=2.0926E7
        CBETA=(XT*XF+YT*YF)/(R*RF)
        IF(CBETA<1.)THEN
            BETA=ACOS(CBETA)
            DISTNM=A*BETA/6076.
        ELSE
            DISTNM=(XT-XF)/6076.
        ENDIF
        RETURN
        END

        SUBROUTINE PREDICT (TF,XDUM,YDUM,X1DUM,Y1DUM,XTF,YTF)
        IMPLICIT REAL*8 (A-H)
        IMPLICIT REAL*8 (O-Z)
        INTEGER STEP
        SAVE
        H=.01.
        A=2.0926E7
        GM=1.4077E16
        T=0.
        X=XDUM
        Y=YDUM
        X1=X1DUM
        Y1=Y1DUM
10     IF(T>(TF-.00001))GOTO 999
        XOLD=X
        YOLD=Y
        X1OLD=X1
        Y1OLD=Y1
        STEP=1
        GOTO 200
66     STEP=2
        X=X+H*XD
        Y=Y+H*YD
        X1=X1+H*X1D
        Y1=Y1+H*Y1D
        T=T+H
        GOTO 200
55     X=(XOLD+X)/2+.5*H*XD
        Y=(YOLD+Y)/2+.5*H*YD

```

(Contd.)

Listing 16.3 (Continued)

```

X1=(X1OLD+X1)/2+.5*H*X1D
Y1=(Y1OLD+Y1)/2+.5*H*Y1D
GOTO 10
200 CONTINUE
TEMBOT=(X**2+Y**2)**1.5
X1D=-GM*X/TEMBOT
Y1D=-GM*Y/TEMBOT
XD=X1
YD=Y1
IF(STEP-1)66,66,55
999 CONTINUE
XTF=X
YTF=Y
RETURN
END

SUBROUTINE LAMBERT(XIC,YIC,TFDES,XF,YF,VRX,VRY,XLONGM
,XLONGT,ICOUNT)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
A=2.0926E7
GM=1.4077E16
RIC=SQRT(XIC**2+YIC**2)
RF=SQRT(XF**2+YF**2)
CPHI=(XIC*XF+YIC*YF)/(RIC*RF)
PHI=ACOS(CPHI)
SPHI=SIN(PHI)
R0=RIC
PI=3.14159
DEGRAD=360./(2.*PI)
ICOUNT=1
GMIN=ATAN2((SPHI-SQRT(2.*RO*(1.-CPHI/RF)),(1-CPHI))
GMAX=ATAN2((SPHI-SQRT(2.*RO*(1.-CPHI/RF)),(1-CPHI))
GAM=(GMIN+GMAX)/2
DO
    TOP=GM*(1.-COS(PHI))
    TEMP=R0*COS(GAM)/RF-COS(PHI+GAM)
    BOT=R0*COS(GAM)*TEMP
    V=SQRT(TOP/BOT)
    IF (XLONGT>XLONGM) THEN
        VRX=V*COS(PI/2.-GAM+XLONGM)
        VRY=V*SIN(PI/2.-GAM+XLONGM)
    ELSE
        VRX=V*COS(-PI/2.+GAM+XLONGM)
        VRY=V*SIN(-PI/2.+GAM+XLONGM)
    END IF
    XLAM=R0*V*V/GM
    TOP1=TAN(GAM)*(1-COS(PHI))+(1-XLAM)*SIN(PHI)
    BOT1P=(1-COS(PHI))/(XLAM*COS(GAM)*COS(GAM))
    BOT1=(2-XLAM)*(BOT1P+COS(GAM+PHI)/COS(GAM))
    TOP2=2*COS(GAM)

```

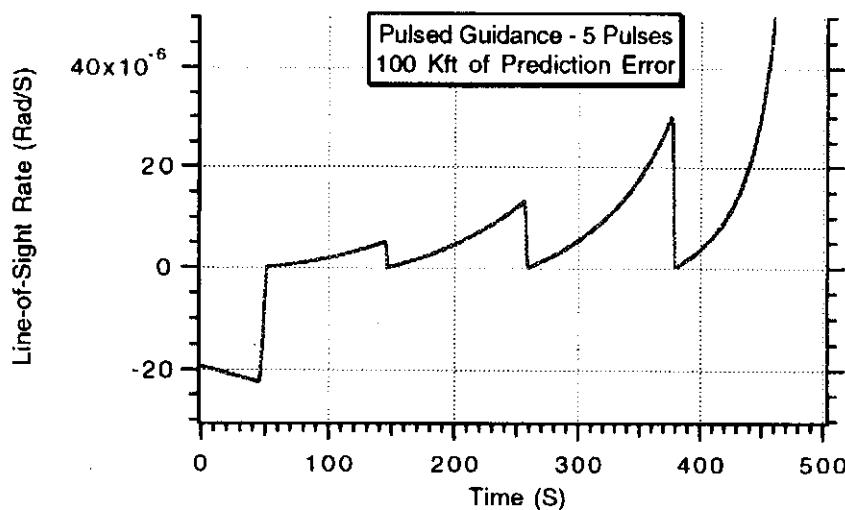
(Contd.)

Listing 16.3 (Continued)

```

BOT2=XLAM*((2/XLAM-1)**1.5)
TOP3=SQRT(2/XLAM-1)
BOT3=COS(GAM)/TAN(PHI/2)-SIN(GAM)
TEMP=(TOP2/BOT2)*ATAN2(TOP3,BOT3)
TF=R0*(TOP1/BOT1+TEMP)/(V*COS(GAM))
IF ((ABS(TFDES-TF)<=.00000001*TFDES).OR.ICOUNT>100)THEN
    EXIT
ENDIF
IF(TF>TFDES)THEN
    GMAX=GAM
ELSE
    GMIN=GAM
ENDIF
IF (ICOUNT. EQ.1) THEN
    XNEXT=(GMAX +GMIN)/2.
ELSE
    XNEXT =GAM +(GAM-GOLD)*(TFDES -TF)/(TF-TOLD)
    IF (XNEXT>GMAX. OR. XNEXT<GMIN) THEN
        XNEXT =(GMAX+GMIN)/2.
    ENDIF
ENDIF
GOLD=GAM
TOLD=TF
GAM=XNEXT
ICOUNT=ICOUNT +1
REPEAT
RETURN
END

```

**Fig. 16.17 Pulsed guidance works with fewer pulses.**

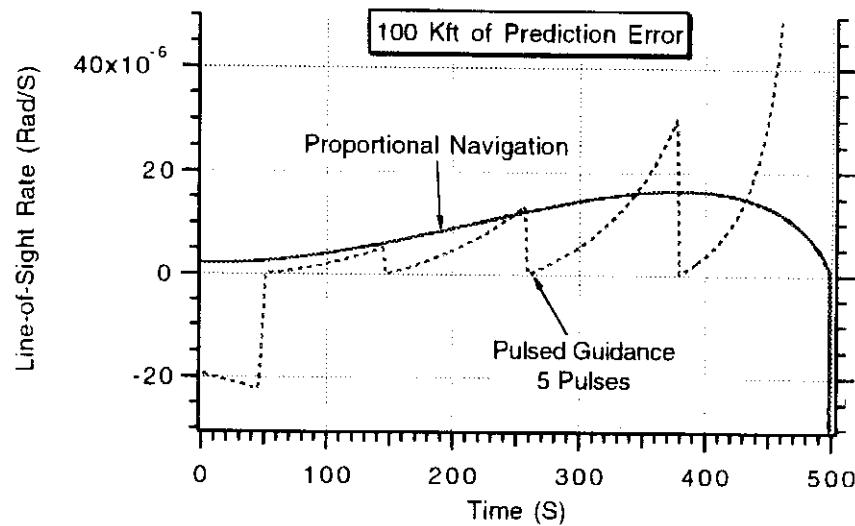


Fig. 16.18 Comparison of proportional navigation and pulsed guidance.

Figure 16.17 shows a similar case, except only five guidance pulses are used. We can see that the pulsed guidance law is still attempting to drive the line-of-sight rate to zero after each pulse is issued. However, since there are fewer pulses, the line-of-sight rate builds up to larger values in between pulses.

The five pulse results are overlayed with the proportional navigation results for the identical case and are shown in Figure 16.18. We can see that proportional navigation drives the line-of-sight rate to zero at the end of the flight only.

Although each of the cases studied so far resulted in successful intercepts, the divert requirements for each guidance concept are different. Figure 16.19 displays the required lateral divert profiles for each of the cases. We can see that proportional navigation has the smallest divert requirements. Increasing the number of pulses in a pulsed guidance system does not appear to influence the lateral divert requirements.

We use pulsed guidance when the divert engine characteristics make it infeasible to use proportional navigation. The price paid is somewhat higher for lateral divert requirements.

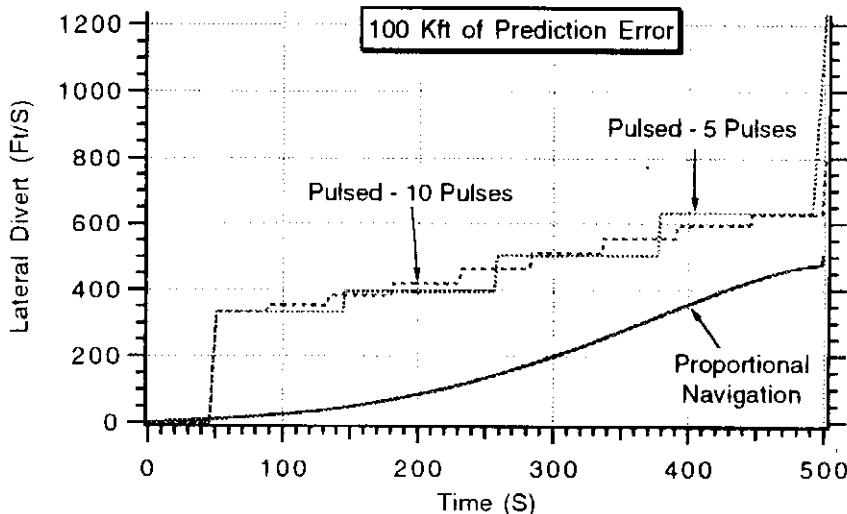


Fig. 16.19 Pulsed guidance does not reduce lateral divert requirements.

Ballistic Target Properties

Introduction^{1,2}

ALTHOUGH surface-to-surface missiles were used as terror weapons during World War II and the Iran-Iraq War, most of the world became familiar with the ballistic missile threat during the 1991 Persian Gulf War. Hundreds of millions of TV viewers will never forget the wail of sirens and images of Scud missiles glowing in the night-time skies over Tel Aviv and Dhahran as they decelerated through the atmosphere toward their civilian targets after having traveled hundreds of miles from their launch sites in Iraq. Viewers, regardless of nationality, were riveted by the drama of the almost nightly duels between the Patriot interceptor and its intended prey—the Scud ballistic target.

In the chapters pertaining to tactical guidance the interceptor's intended target was considered to be an aircraft, whereas in the chapters pertaining to strategic guidance the engagement threat was either considered to be a booster or an exoatmospheric ballistic target. In the next two chapters of the text we will consider the special problems encountered in intercepting an endoatmospheric ballistic target.

Ballistic Target Model

When a ballistic target re-enters the atmosphere after having traveled a long distance, its speed is high and the remaining time to ground impact is relatively short. The small distances traveled by ballistic targets after they re-enter the atmosphere enable us to accurately model these threats using the flat-Earth, constant gravity approximation as was done in modeling tactical interceptors. This simplification is important because it will lead to useful closed-form solutions for ballistic targets.

Figure 17.1 presents the flat-Earth, constant gravity model for the ballistic endoatmospheric threat. In this model, only drag and gravity act on the ballistic target.³ We can see from Fig. 17.1 that the target has velocity V_T and is initially at re-entry angle γ_T . Note that drag F_{drag} acts in a direction opposite to the velocity vector and gravity g always acts downward in the flat-Earth model. Therefore, if the effect of drag is greater than that of gravity, the target will slow up or decelerate. Since we will eventually consider the ballistic target as an interceptor threat, the magnitude of the target deceleration will be of interest to us.

From Fig. 17.1 we can see that the target re-entry angle γ_T can be computed, using trigonometry, from the two inertial components of the target velocity as

$$\gamma_T = \tan^{-1} \frac{-V_{T2}}{V_{T1}}$$

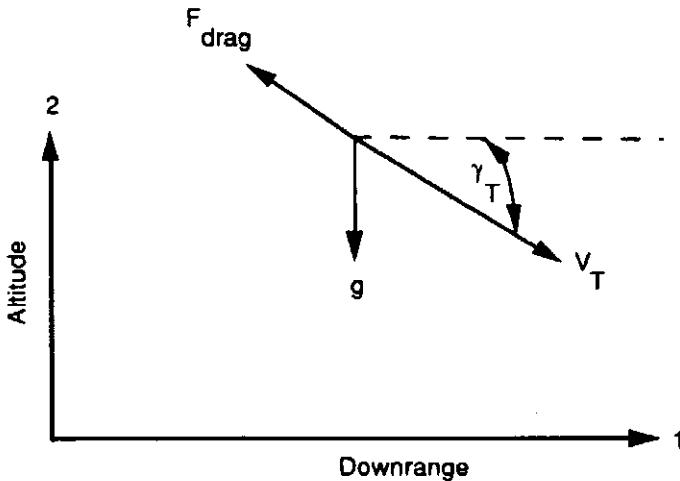


Fig. 17.1 Ballistic target geometry.

The acceleration components of the ballistic target in the inertial downrange and altitude directions of Fig. 17.1 can either be expressed in terms of the target weight W , reference area S_{ref} , zero lift drag C_{D0} , and gravity g or more simply in terms of the ballistic coefficient β (defined in Chapter 11) according to

$$\frac{dV_{T1}}{dt} = \frac{-F_{\text{drag}}}{m} \cos \gamma_T = \frac{-QS_{\text{ref}}C_{D0}g}{W} \cos \gamma_T = \frac{-Qg}{\beta} \cos \gamma_T$$

$$\frac{dV_{T2}}{dt} = \frac{F_{\text{drag}}}{m} \sin \gamma_T - g = \frac{QS_{\text{ref}}C_{D0}g}{W} \sin \gamma_T - g = \frac{Qg}{\beta} \sin \gamma_T - g$$

where Q is the dynamic pressure. Recall that the dynamic pressure has been previously defined as

$$Q = 0.5\rho V_T^2$$

where V_T is the total target velocity, which can be expressed in terms of component velocities as

$$V_T = \sqrt{V_{T1}^2 + V_{T2}^2}$$

and ρ is the air density measured in slug/ft³ and was shown to be accurately approximated exponentially in Chapter 11 as

$$\rho = 0.0034e^{\frac{-R_{T2}}{22,000}}$$

above 30,000 ft and

$$\rho = 0.002378e^{\frac{-R_{T2}}{30,000}}$$

below 30,000 ft. The target altitude R_{T2} is measured in feet. Since the acceleration equations are in a fixed or inertial frame, they can be integrated directly to yield velocity and position.

Ballistic Target Experiments

A simulation of a ballistic target, based on the acceleration differential equations of the previous section, appears in Listing 17.1. The ballistic target acceleration

Listing 17.1 Ballistic target simulation

```

INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
RT1=0.
RT2=100000.
VT=6000.
GAMTDEG=45.
BETA=500.
VT1=VT*COS(GAMTDEG/57.3)
VT2=-VT*SIN(GAMTDEG/57.3)
T=0.
H=.01
S=0.
5 IF(RT2<0.)GOTO 999
S=S+H
RT1OLD=RT1
RT2OLD=RT2
VT1OLD=VT1
VT2OLD=VT2
STEP=1
GOTO 200
66 STEP=2
RT1=RT1+H*VT1
RT2=RT2+H*VT2
VT1=VT1+H*AT1
VT2=VT2+H*AT2
T=T+H
GOTO 200
55 CONTINUE
RT1=.5*(RT1OLD+RT1+H*VT1)
RT2=.5*(RT2OLD+RT2+H*VT2)
VT1=.5*(VT1OLD+VT1+H*AT1)
VT2=.5*(VT2OLD+VT2+H*AT2)
IF(S.GE..09999)THEN
S=0.
ATG=SQRT(AT1**2+AT2**2)/32.2
RT1K=RT1/1000.
RT2K=RT2/1000.
VT=SQRT(VT1**2+VT2**2)
ATG=SQRT(AT1**2+AT2**2)/32.2
WRITE(*,97)T,RT1K,RT2K,ATG,VT
WRITE(1,97)T,RT1K,RT2K,ATG,VT
END IF
GOTO 5
97 FORMAT(5F10.3)
200 CONTINUE
IF(RT2.LE.30000.)THEN
RHO=.002378*EXP(-RT2/30000.)
ELSE
RHO=.0034*EXP(-RT2/22000.)
ENDIF

```

(Contd.)

Listing 17.1 (Continued)

```

VT=SQRT(VT1**2+VT2**2)
Q=.5*RHO*VT**2
GAMT=ATAN2(-VT2,VT1)
AT1=-32.2*Q*COS(GAMT)/BETA
AT2=-32.2+32.2*Q*SIN(GAMT)/BETA
IF(STEP-1)66,66,55
999  CONTINUE
ATG=SQRT(AT1**2+AT2**2)/32.2
RT1K=RT1/1000.
RT2K=RT2/1000.
VT=SQRT(VT1**2+VT2**2)
ATG=SQRT(AT1**2+AT2**2)/32.2
WRITE(*,97)T,RT1K,RT2K,ATG,VT
WRITE(1,97)T,RT1K,RT2K,ATG,VT
CLOSE(1)
PAUSE
END

```

differential equations appear after statement label 200, and the initial conditions, required for the integration of the differential equations, appear at the beginning of the simulation (before statement label 5). We can see that the program is initialized with a target altitude of 100 kft, a target velocity of 6000 ft/s, and a re-entry angle of 45 deg. The nominal ballistic coefficient for the target is 500 lb/ft². From statement label 5 we can see that the simulation stops when the ballistic threat hits the ground ($R_{T2} < 0$). Every tenth of a second the target location is printed in kft, acceleration in g , and velocity in ft/s. The simulation integration step size of 0.01 s ($H = 0.01$) is sufficiently small to get accurate answers with the second-order Runge-Kutta numerical integration technique.

The nominal case of Listing 17.1 was run, and Fig. 17.2 presents the resultant trajectory of the ballistic target. We can see from the figure that the target trajectory

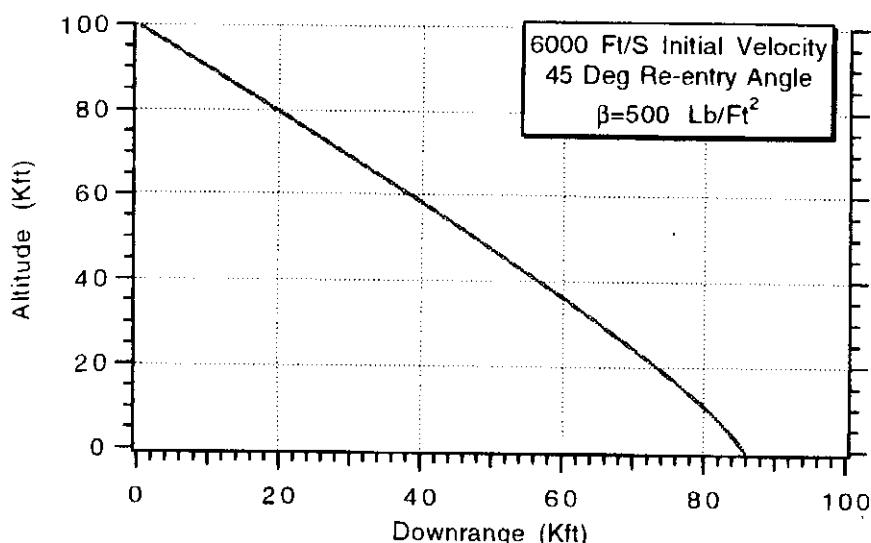


Fig. 17.2 Nominal ballistic target trajectory is approximately a straight line.

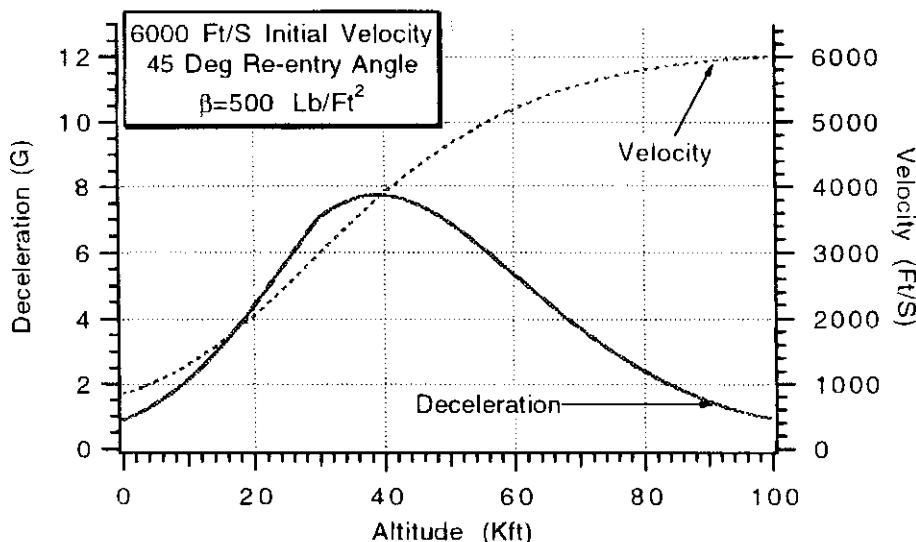


Fig. 17.3 Peak target deceleration occurs at 40 kft.

is approximately a straight line (we shall exploit this observation later). At the lower altitudes there is slight curvature in the trajectory due to both drag and gravity.

Figure 17.3 displays the deceleration and velocity of the nominal target as a function of altitude. At 100-kft altitude the target has an initial velocity of 6000 ft/s and there is 1 g of acceleration due to gravity (there is too little atmosphere at 100 kft to cause substantial drag). The drag deceleration increases and target velocity decreases as the target descends in altitude. At approximately 40 kft altitude the target deceleration peaks and is nearly 8 g. At this altitude of maximum deceleration the target speed is approximately 63% of its original value (i.e., $3800 = 0.63 * 6000$).

The simulation of Listing 17.1 was rerun, and this time the initial target velocity at 100-kft altitude was made a parameter. We can see from the simulation results, displayed in Fig. 17.4, that target deceleration increases as the target speed increases. This should not be surprising since the acceleration differential equations tell us that deceleration is proportional to the dynamic pressure (i.e., target

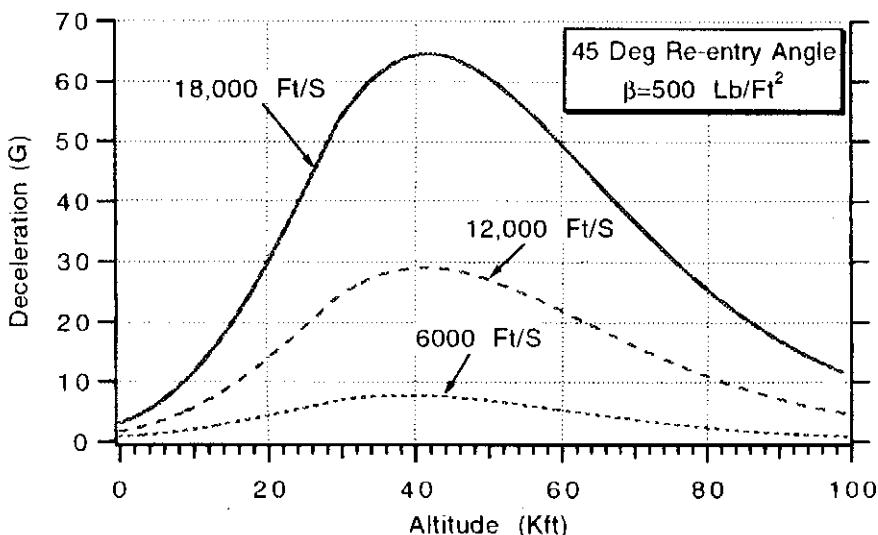


Fig. 17.4 Peak target deceleration increases with increasing target speed.

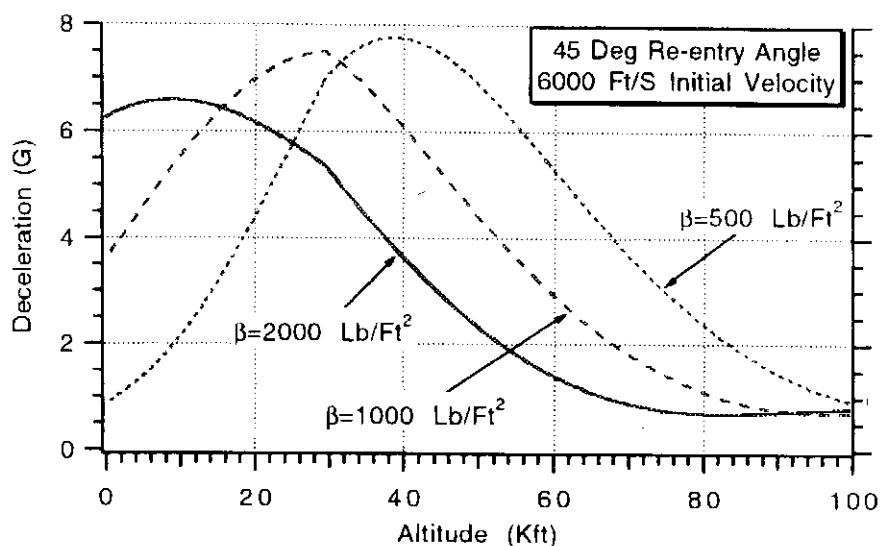


Fig. 17.5 Peak target deceleration is approximately independent of ballistic coefficient.

velocity squared). Therefore very fast ballistic threats can cause enormous decelerations. Surprisingly, the simulation results in Fig. 17.4 also indicate that the altitude of maximum target deceleration appears to be approximately independent of the target speed!

The simulation of Listing 17.1 was again rerun, and this time the target ballistic coefficient was made a parameter. Simulation results, displayed in Fig. 17.5, appear to indicate that the peak target deceleration is approximately independent of ballistic coefficient. This is surprising since there is more drag with lower ballistic coefficients. However, these simulation results also indicate that the altitude at which the peak target deceleration occurs decreases with increasing ballistic coefficient.

Another experiment was conducted using Listing 17.1. This time the initial target velocity and ballistic coefficient were fixed and the re-entry angle was made a parameter. Simulation results, displayed in Fig. 17.6, indicate that the peak target

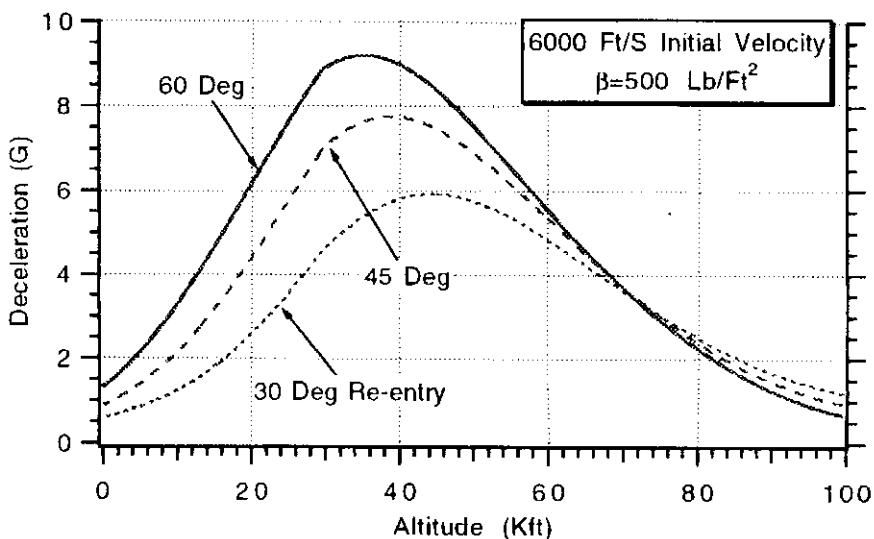


Fig. 17.6 Peak target deceleration increases with increasing re-entry angle.

deceleration increases with increasing re-entry angle. This is reasonable because as the re-entry angle increases the resultant target trajectory tends to become more vertical and more drag is experienced. In addition, Fig. 17.6 shows that the altitude at which the peak target deceleration occurs decreases with increasing re-entry angle.

Closed-Form Solutions for Ballistic Targets

In this section we will derive some useful closed-form solutions for ballistic targets and compare the theoretical solutions to the simulation results of the previous section. If we neglect gravity, Newton's second law says that we can express the drag force acting on a ballistic target in terms of the dynamic pressure, reference area, and zero lift drag according to

$$F_{\text{drag}} = m \frac{dV_T}{dt} = -QS_{\text{ref}}C_{D0}$$

Using definitions of dynamic pressure and ballistic coefficient, we can rewrite the preceding differential equation as

$$\frac{dV_T}{dt} = \frac{-\rho g V_T^2}{2\beta}$$

If we assume that the target trajectory is always a straight line (as appeared to be the case in Fig. 17.2), then the re-entry angle γ_T is a constant. Figure 17.1 indicates that for the constant re-entry angle assumption, the altitude component of velocity can be expressed in terms of the total velocity as

$$V_{T2} = \frac{dR_{T2}}{dt} = -V_T \sin \gamma_T$$

According to the chain rule we can express the rate of change of the total velocity as

$$\frac{dV_T}{dt} = \frac{dV_T}{dR_{T2}} \frac{dR_{T2}}{dt} = -\frac{dV_T}{dR_{T2}} V_T \sin \gamma_T$$

Substitution of the expression for the rate of change of the total velocity into the preceding equation yields

$$\frac{-\rho g V_T^2}{2\beta} = -\frac{dV_T}{dR_{T2}} V_T \sin \gamma_T$$

Assuming that the exponential approximation for air density

$$\rho = 0.0034 e^{-\frac{R_{T2}}{22,000}}$$

applies everywhere (actually another approximation is better below 30,000 ft and is indicated in the previous section), we can rearrange the preceding differential equation so that velocity terms are on one side and altitude terms are on the other side. Rewriting the resultant differential equation with integrals yields

$$\int_{V_{T_{IC}}}^{V_T} \frac{dV_T}{V_T} = \frac{0.0034g}{2\beta \sin \gamma_T} \int_{R_{T2_{IC}}}^{R_{T2}} e^{-\frac{R_{T2}}{22,000}} dR_{T2}$$

where gravity, the ballistic coefficient, and the re-entry angle have been brought outside the integral because they are considered to be constants. Integrating the preceding expression yields the velocity formula where the target velocity is a function of its initial velocity, re-entry angle, ballistic coefficient, and altitude (or air density) according to

$$V_T = V_{T_{IC}} e^{-\frac{22,000g\rho}{2\beta \sin \gamma_T}}$$

The maximum deceleration experienced by the target will occur at an altitude in which the dynamic pressure is a maximum. Substituting the velocity formula into the definition of dynamic pressure yields

$$Q = 0.5\rho V_T^2 = 0.5\rho V_{T_{IC}}^2 e^{-\frac{22,000g\rho}{\beta \sin \gamma_T}}$$

We can find when the dynamic pressure is a maximum by taking its derivative with respect to the air density and setting the resultant expression to zero or

$$\frac{dQ}{d\rho} = 0 = 0.5V_{T_{IC}}^2 e^{-\frac{2,000g\rho}{\beta \sin \gamma_T}} - 0.5\rho V_{T_{IC}}^2 \frac{22,000g}{\beta \sin \gamma_T} e^{-\frac{22,000g\rho}{\beta \sin \gamma_T}}$$

After some algebra we find that the maximum dynamic pressure condition is

$$\beta \sin \gamma_T = 22,000\rho g$$

The velocity of the target at the maximum dynamic pressure condition can be found by substituting the preceding expression into the velocity formula yielding

$$V_T|_{\max Q} = V_{T_{IC}} e^{-\frac{22,000g\rho}{2\beta \sin \gamma_T}} = V_{T_{IC}} e^{-0.5} = 0.606V_{T_{IC}}$$

In other words, the velocity of the target is always 61% of its initial value when the dynamic pressure is a maximum! This important result was also observed empirically in the simulation results of Fig. 17.3.

To find the altitude at which maximum target deceleration occurs, we must first find the altitude or air density at which the dynamic pressure is greatest. The air density at maximum dynamic pressure can be found from the maximum dynamic pressure condition to be

$$\rho_{\max Q} = \frac{\beta \sin \gamma_T}{22,000g}$$

Since the the altitude at maximum dynamic pressure is related to the air density at maximum dynamic pressure according to

$$\rho_{\max Q} = 0.0034 e^{-\frac{R_{T2_{\max Q}}}{22,000}}$$

we can solve for the altitude at this important flight condition. After some algebra we obtain

$$R_{T2_{\max Q}} = 22,000 \ln \frac{0.0034 * 22,000g}{\beta \sin \gamma_T} = 22,000 \ln \frac{2409}{\beta \sin \gamma_T}$$

where the altitude at which maximum target deceleration occurs is expressed in units of feet. From the preceding relationship we can see that the altitude of maximum target deceleration does not depend on target velocity but only on the

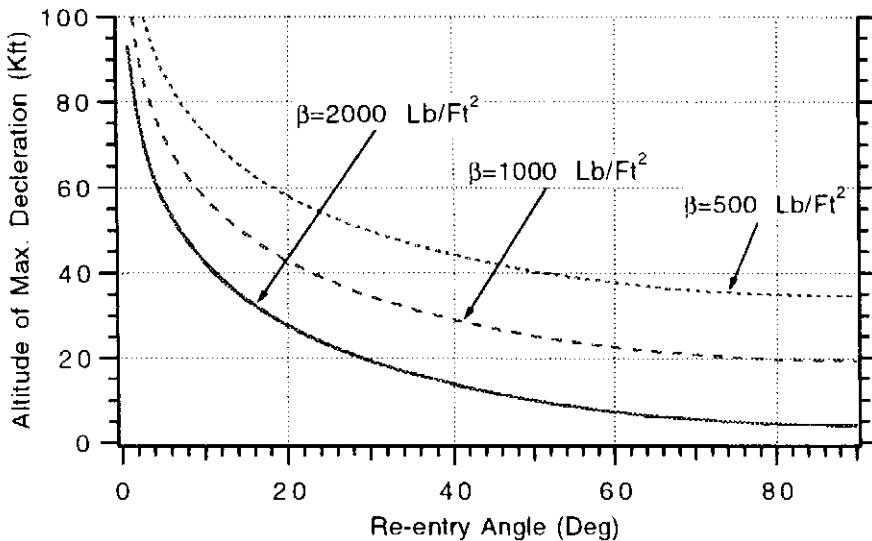


Fig. 17.7 Theory indicates that altitude at which maximum deceleration occurs is independent of velocity.

ballistic coefficient and re-entry angle! This observation is in agreement with the simulation results of Fig. 17.4.

The closed-form solution for the altitude of maximum target deceleration is displayed as a function of the re-entry angle in Fig. 17.7. Here we can see that the altitude of maximum target deceleration increases with decreasing re-entry angle and decreasing ballistic coefficient. The theoretical results of Fig. 17.7, which neglects gravity and approximates the atmosphere below 30,000 ft, are in excellent agreement with the simulation results of Figs. 17.4–17.6.

Since the maximum target deceleration, expressed in units of g , is proportional to the maximum dynamic pressure and inversely proportional to the target ballistic coefficient, we obtain

$$\frac{a}{g} \Big|_{\max} = \frac{-Q_{\max}}{\beta} = \frac{-0.5 \rho_{\max} Q V_T^2 \Big|_{\max Q}}{\beta} = \frac{-0.5 \beta \sin \gamma_T}{22,000 g \beta} 0.606^2 V_{T_{IC}}^2$$

where the negative sign indicates target deceleration rather than target acceleration. Simplification of the preceding formula yields

$$\frac{a}{g} \Big|_{\max} = -2.6 * 10^{-7} V_{T_{IC}}^2 \sin \gamma_T$$

Thus we can see that maximum deceleration does not depend on the target ballistic coefficient but only on the velocity and re-entry angle as was also observed empirically in the simulation results displayed in Fig. 17.5! The maximum deceleration formula is displayed as a function of the re-entry angle in Fig. 17.8. We can see that the maximum target deceleration increases with increasing target velocity (actually as the square of target velocity) and increasing re-entry angle. The theoretical results of Fig. 17.8, which neglects gravity and approximates the atmosphere below 30,000 ft, are also in excellent agreement with the simulation results of Figs. 17.4–17.6.

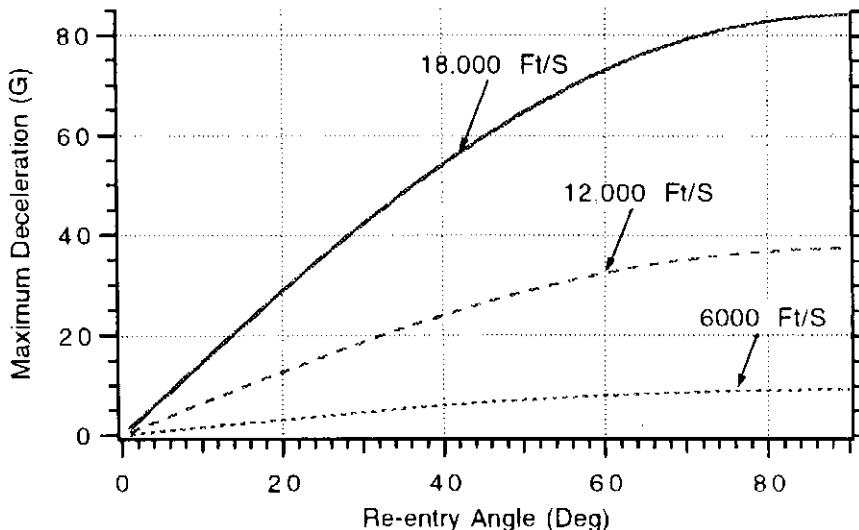


Fig. 17.8 Theory tells us that maximum deceleration is independent of ballistic coefficient.

Missile Aerodynamics

We have just observed the deceleration properties of a ballistic target as a function of its ballistic coefficient, velocity, altitude, and re-entry angle. In this section we want to get an idea of the generic acceleration capability of a pursuing interceptor so that we can better understand ballistic target engagements. To get a first-order estimate of the aerodynamic capability of a missile, we shall treat the interceptor as a cylinder with length L and diameter D .

Basic aerodynamic theory tells us that the lift coefficient C_L for a cylinder is⁴

$$C_L = 2\alpha + \frac{1.5S_{\text{plan}}\alpha^2}{S_{\text{ref}}}$$

where α is the angle of attack or the angle between the missile body and its velocity vector. The planform area S_{plan} and reference area S_{ref} are related to the geometry of a cylinder according to

$$S_{\text{plan}} \approx LD$$

$$S_{\text{ref}} = \frac{\pi D^2}{4}$$

From Chapter 11 we know that the relationship between acceleration and the lift coefficient is given by

$$F = ma = \frac{Wn_L}{g} = QS_{\text{ref}}C_L$$

where W is the missile weight, n_L the lateral missile acceleration, g the acceleration of gravity, and Q the dynamic pressure or

$$Q = 0.5\rho V_M^2$$

The air density ρ can be found from the exponential approximation discussed in both this chapter and Chapter 11. Substitution of the lift coefficient and dynamic pressure into Newton's second law yields the formula for the acceleration capability

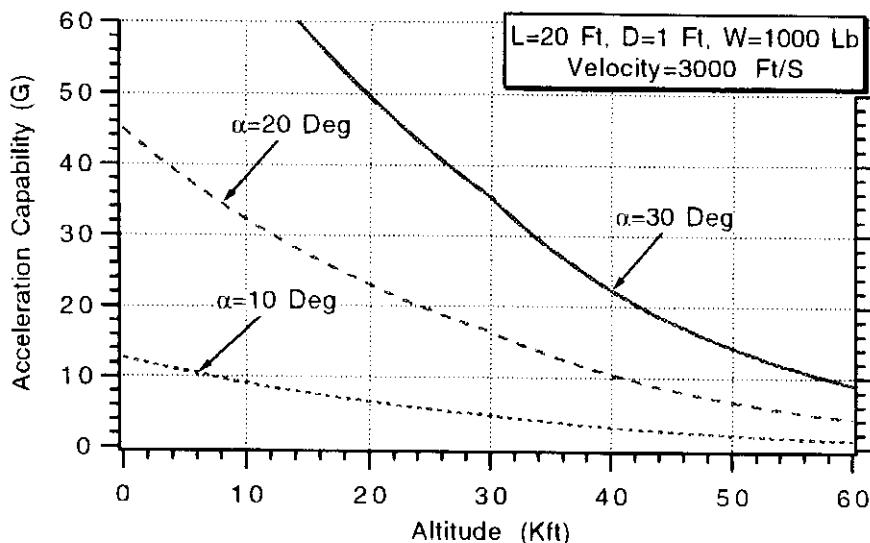


Fig. 17.9 Missile acceleration capability decreases with increasing altitude and decreasing angle of attack.

in units of g of a flying telephone pole (or cylinder) as a function of missile velocity, angle of attack, and altitude (or air density) for a given missile length, diameter, and weight.

$$\frac{n_L}{g} = \frac{QS_{\text{ref}}C_L}{W} = \frac{0.5\rho V_M^2 S_{\text{ref}}}{W} \left[2\alpha + \frac{1.5S_{\text{plan}}\alpha^2}{S_{\text{ref}}} \right]$$

To get a better understanding of the preceding acceleration equation, let us consider a numerical example in which the missile weighs 1000 lb and is 20 ft long and 1 ft in diameter. Figure 17.9 displays the resultant acceleration capability of a cylindrical missile without wings and tails, using the preceding equation, with velocity 3000 ft/s and three different angles of attack, as a function of altitude. We can see that interceptor acceleration capability decreases as altitude increases and as angle of attack decreases. This is why interceptors have a reduced acceleration capability at the higher altitudes and is also why they must operate at higher angles of attack at the higher altitudes in order to maintain the same acceleration capability they had at the lower altitudes. We can see that for this example, the missile has only a 7-g capability at 50-kft altitude if its maximum angle of attack is limited to 20 deg. Increasing the maximum angle of attack capability to 30 deg will double the acceleration capability of the interceptor at 50-kft altitude while reducing the maximum angle of attack by 10 deg halves the acceleration capability at that altitude.

Figure 17.10 shows that the interceptor acceleration capability increases as missile velocity increases. Increasing the missile velocity by 1000 ft/s at 50-kft altitude nearly doubles the interceptor acceleration capability. Reducing the interceptor velocity by 1000 ft/s approximately halves the acceleration capability of the interceptor at that altitude.

We can compare the lateral acceleration capability of our generic interceptor (or flying telephone pole) to the deceleration levels of the ballistic target. Figure 17.11 compares a 3000-ft/s cylindrical interceptor with a maximum angle-of-attack capability of 20 deg to a 6000-ft/s ballistic target with a 500-lb/ft² ballistic coefficient and a 45-deg re-entry angle. We can see that the interceptor acceleration capability

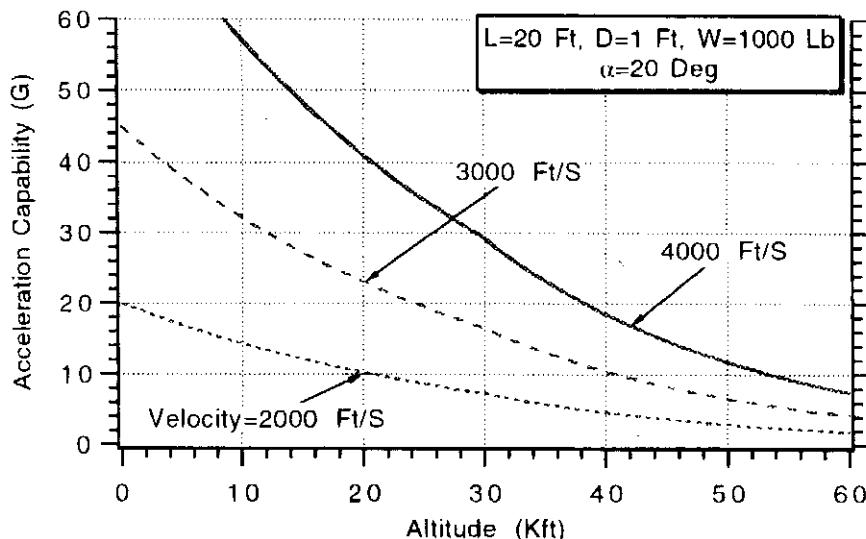


Fig. 17.10 Missile acceleration capability increases with increasing missile velocity.

increases with decreasing altitude whereas the target deceleration capability increases with decreasing altitude until it peaks at 40 kft. At 50-kft altitude the interceptor acceleration capability and target deceleration characteristics are matched at approximately 7 g. From an interceptor acceleration capability point of view, the ideal intercept should take place at very low altitude where the interceptor has enormous capability and a considerable acceleration advantage over the target. In fact, from a missile point of view, the ideal intercept altitude is near sea level where the interceptor acceleration capability is largest and target deceleration capability smallest. However, practical considerations may require the interceptor to engage the ballistic target at much higher altitudes.

Intercepting a Ballistic Target

In this chapter we have spent considerable time simulating and understanding the properties of ballistic targets. We have seen that fast ballistic targets can

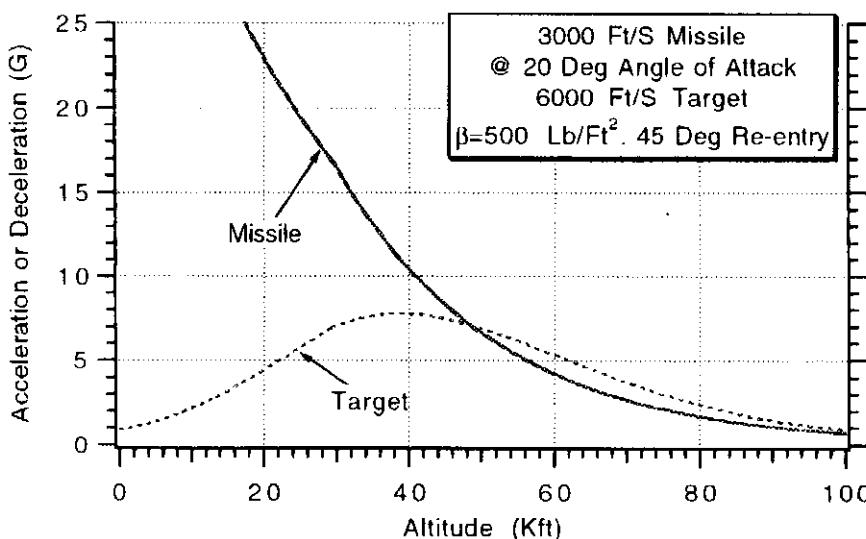


Fig. 17.11 Cylindrical interceptor acceleration capability matches target deceleration characteristics at 50-kft altitude.

go through tremendous deceleration levels as they re-enter the atmosphere. The component of target deceleration perpendicular to the line of sight appears as a target maneuver to a pursuing interceptor. In this section we will investigate some of the difficulties and potential solutions associated with an interceptor trying to engage a ballistic target within the atmosphere.

Listing 17.2 presents the FORTRAN source code for an engagement simulation involving a constant speed interceptor and decelerating ballistic target. At the beginning of the simulation, the target is defined as having a 6000-ft/s initial velocity at 200-kft altitude with a re-entry angle of 45 deg and ballistic coefficient of 500 lb/ft². It is desired to fire an interceptor immediately and have the intercept take place at 50-kft altitude ($RT2DES = 50,000$).

Subroutine INITIAL is called to predict the location of the target at intercept ($RT1F$, $RT2F$) and to compute the time $TFDES$ it will take the target to reach the intercept altitude. Essentially subroutine INITIAL is a mini-simulation of the target. Note that we must tell this subroutine the estimated ballistic coefficient $BETEST$ of the target. If the estimated ballistic coefficient is in error, interceptor launch errors will result. From the outputs of subroutine INITIAL the interceptor launch angle $GAMMDEG$ and total velocity VM are computed so that the interceptor will be on a perfect collision course with the target (assuming missile is fired when target is at 200-kft altitude). For simplicity, gravity and drag effects are not included on the interceptor.

The missile and target differential equations appear after statement label 200. The ballistic target differential equations are identical to those that have already been modeled in this chapter. The constant speed interceptor differential equations are identical to those found in Chapter 2. Three interceptor guidance options appear. The parameter APN determines which guidance law is used and $XNCLIMG$ determines the interceptor acceleration capability (nominally set to infinity). If APN is 0, we get the proportional navigation guidance law, and if APN is 1, we get augmented proportional navigation where the augmented term includes the component of target acceleration (or deceleration) perpendicular to the line-of-sight $ATPLOS$.

A nominal case was run in which a proportional navigation interceptor guides on a ballistic target as shown in Fig. 17.12. The geometry is considered near head-on

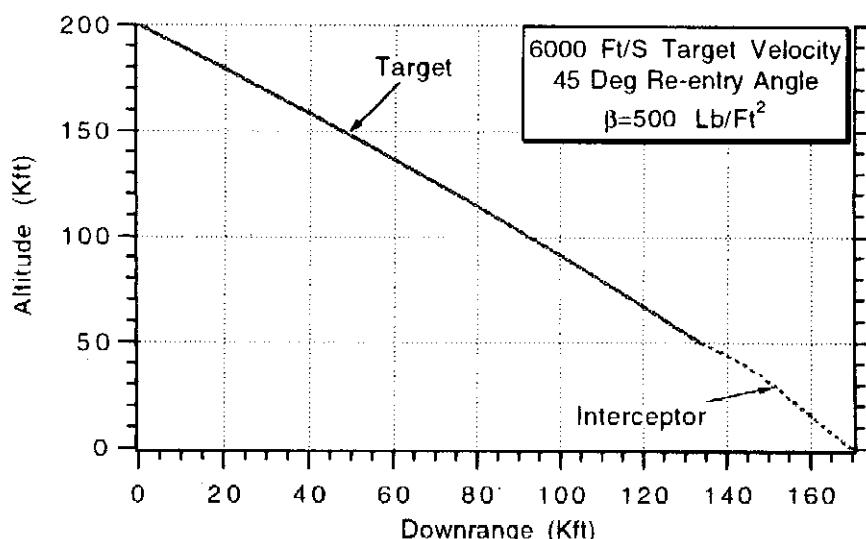


Fig. 17.12 Geometry for near inverse trajectory.

Listing 17.2 Ballistic target engagement simulation

```

INTEGER STEP,APN
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
APN=0
XNP=3.
RT1=0.
RT2=200000.
RM1=170000.
RM2=0.
VT=6000.
RT2DES=50000.
GAMTDEG=45.
BETA=500.
BETEST=500.
XNCLIMG=7.
XNCLIM=XNCLIMG*32.2
VT1=VT*COS(GAMTDEG/57.3)
VT2=-VT*SIN(GAMTDEG/57.3)
CALL INITIAL(RT2DES,RT1,RT2,VT1,VT2,BETEST,RT1F,RT2F,TFDES)
RTM1F=RT1F-RM1
RTM2F=RT2F-RM2
GAMMDEG=57.3*ATAN2(RTM2F,RTM1F)
RTMF=SQRT(RTM1F**2+RTM2F**2)
VM=RTMF/TFDES
VM1=VM*COS(GAMMDEG/57.3)
VM2=VM*SIN(GAMMDEG/57.3)
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
T=0.
H=.01
S=0.
XNC=0.
ZEMPLOS=0.
ZEM1=0.
ZEM2=0.
5 IF(VC<0.)GOTO 999
IF(RTM<1000.)THEN
    H=.0002
ELSE
    H=.01
ENDIF
S=S+H
RT1OLD=RT1
RT2OLD=RT2
VT1OLD=VT1
VT2OLD=VT2
RM1OLD=RM1

```

(Contd.)

Listing 17.2 (Continued)

```

RM2OLD=RM2
VM1OLD=VM1
VM2OLD=VM2
STEP=1
GOTO 200
66 STEP=2
RT1=RT1+H*VT1
RT2=RT2+H*VT2
VT1=VT1+H*AT1
VT2=VT2+H*AT2
RM1=RM1+H*VM1
RM2=RM2+H*VM2
VM1=VM1+H*AM1
VM2=VM2+H*AM2
T=T+H
GOTO 200
55 CONTINUE
RT1=.5*(RT1OLD+RT1+H*VT1)
RT2=.5*(RT2OLD+RT2+H*VT2)
VT1=.5*(VT1OLD+VT1+H*AT1)
VT2=.5*(VT2OLD+VT2+H*AT2)
RM1=.5*(RM1OLD+RM1+H*VM1)
RM2=.5*(RM2OLD+RM2+H*VM2)
VM1=.5*(VM1OLD+VM1+H*AM1)
VM2=.5*(VM2OLD+VM2+H*AM2)
IF(S.GE..09999)THEN
  S=0.
  ATG=SQRT(AT1**2+AT2**2)/32.2
  RT1K=RT1/1000.
  RT2K=RT2/1000.
  RM1K=RM1/1000.
  RM2K=RM2/1000.
  XNCG=XNC/32.2
  ATPLOSG=ATPLOS/32.2
  VM=SQRT(VM1**2+VM2**2)
  CADEG=180.-57.3*ACOS((VT1*VM1+VT2*VM2)/(VT*VM))
  WRITE(*,97)T,RT1K,RT2K,RM1K,RM2K,ATG,XNCG,ATPLOSG
  WRITE(1,97)T,RT1K,RT2K,RM1K,RM2K,ATG,XNCG,ATPLOSG
END IF
GOTO 5
97 FORMAT(8F10.3)
200 CONTINUE
IF(RT2.LE.30000.)THEN
  RHO=.002378*EXP(-RT2/30000.)
ELSE
  RHO=.0034*EXP(-RT2/22000.)
ENDIF
VT=SQRT(VT1**2+VT2**2)
Q=.5*RHO*VT**2
GAMT=ATAN2(-VT2,VT1)

```

(Contd.)

Listing 17.2 (Continued)

```

AT1=-32.2*Q*COS(GAMT)/BETA
AT2=-32.2+32.2*Q*SIN(GAMT)/BETA
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
XLAM=ATAN2(RTM2,RTM1)
XLAMD=(RTM1*VTM2-RTM2*VTM1)/(RTM*RTM)
ATPLOS=-AT1*SIN(XLAM)+AT2*COS(XLAM)
IF(APN.EQ.1)THEN
    XNC=XNP*VC*XLAMD+.5*XNP*ATPLOS
ELSE
    XNC=XNP*VC*XLAMD
ENDIF
IF(XNC>XNCLIM)XNC=XNCLIM
IF(XNC<-XNCLIM)XNC=-XNCLIM
AM1=-XNC*SIN(XLAM)
AM2=XNC*COS(XLAM)
IF(STEP-1)66,66,55
999 CONTINUE
ATG=SQRT(AT1**2+AT2**2)/32.2
RT1K=RT1/1000.
RT2K=RT2/1000.
RM1K=RM1/1000.
RM2K=RM2/1000.
XNCG=XNC/32.2
ATPLOSG=ATPLOS/32.2
VM=SQRT(VM1**2+VM2**2)
CADEG=180.-57.3*ACOS((VT1*VM1+VT2*VM2)/(VT*VM))
WRITE(*,97)T,RT1K,RT2K,RM1K,RM2K,ATG,XNCG,ATPLOSG
WRITE(1,97)T,RT1K,RT2K,RM1K,RM2K,ATG,XNCG,ATPLOSG
WRITE(*,*)RTM
CLOSE(1)
PAUSE
END
SUBROUTINE INITIAL(RT2DES,RT1IC,RT2IC,VT1IC,VT2IC,BETA,
1      RT1F,RT2F,TFDES)
SAVE
INTEGER STEP
RT1=RT1IC
RT2=RT2IC
VT1=VT1IC
VT2=VT2IC
T=0.
H=.01
5 IF(RT2<=RT2DES)GOTO 999
RT1OLD=RT1
RT2OLD=RT2

```

(Contd.)

Listing 17.2 (Continued)

```

VT1OLD=VT1
VT2OLD=VT2
STEP=1
GOTO 200
66 STEP=2
RT1=RT1+H*VT1
RT2=RT2+H*VT2
VT1=VT1+H*AT1
VT2=VT2+H*AT2
T=T+H
GOTO 200
55 CONTINUE
RT1=.5*(RT1OLD+RT1+H*VT1)
RT2=.5*(RT2OLD+RT2+H*VT2)
VT1=.5*(VT1OLD+VT1+H*AT1)
VT2=.5*(VT2OLD+VT2+H*AT2)
GOTO 5
200 CONTINUE
IF(RT2.LE.30000.)THEN
    RHO=.002378*EXP(-RT2/30000.)
ELSE
    RHO=.0034*EXP(-RT2/22000.)
ENDIF
VT=SQRT(VT1**2+VT2**2)
Q=.5*RHO*VT**2
GAMT=ATAN2(-VT2,VT1)
AT1=-32.2*Q*COS(GAMT)/BETA
AT2=-32.2+32.2*Q*SIN(GAMT)/BETA
IF(STEP-1)66,66,55
999 CONTINUE
RT1F=RT1
RT2F=RT2
TFDES=T
RETURN
END

```

and is also known as an inverse trajectory. The target trajectory is much longer than the missile trajectory since the target is traveling at a much higher velocity.

Figure 17.13 displays the important accelerations for the nominal case. We can see that the target deceleration is approximately 8 g at intercept (or 50 kft). This is in accordance with the ballistic target simulation results of Fig. 17.4 and the theoretical results of Fig. 17.8. We can see that since the engagement geometry is near inverse, there is no target deceleration perpendicular to the line of sight. From a missile point of view, the target does not appear to be maneuvering. Since the missile is initially on a collision course and there is no apparent target maneuver, very little acceleration is required by an interceptor using proportional navigation to hit the target.

A more stressing geometry was considered in which the interceptor is initially at 50 kft downrange ($RM1 = 50,000$). The engagement geometry, shown in

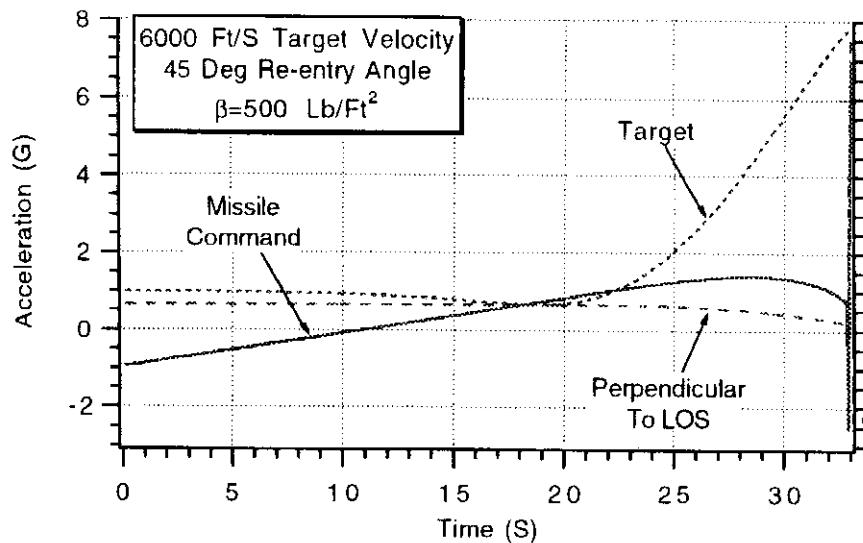


Fig. 17.13 Missile guidance commands are small because very little of target deceleration is perpendicular to line of sight.

Fig. 17.14, is no longer inverse. The intercept still takes place at 50-kft altitude and the missile has a speed of 3000 ft/s in order to be on a collision course. It is assumed that the interceptor has a maximum angle of attack limit of 20 deg, yielding a maximum acceleration capability of 7 g ($XNCLIMG = 7$) at the intercept altitude based on the results of the previous section for our flying telephone pole ($W = 1000 \text{ lb}$, $L = 20 \text{ ft}$, $D = 1 \text{ ft}$).

Figure 17.15 shows that the target deceleration is unchanged for this new engagement geometry and approaches 8 g near intercept. However, the component of target deceleration perpendicular to the line of sight is much larger than it was for the inverse trajectory case, and the peak value is in excess of 3 g. Therefore it is not surprising that the interceptor requires more than 7 g (missile acceleration limit) to hit the apparent 3-g target maneuver. Acceleration saturation follows causing a large miss distance.

Augmented proportional navigation can be used to relax the acceleration requirements of the interceptor under this stressing engagement geometry. Figure 17.16

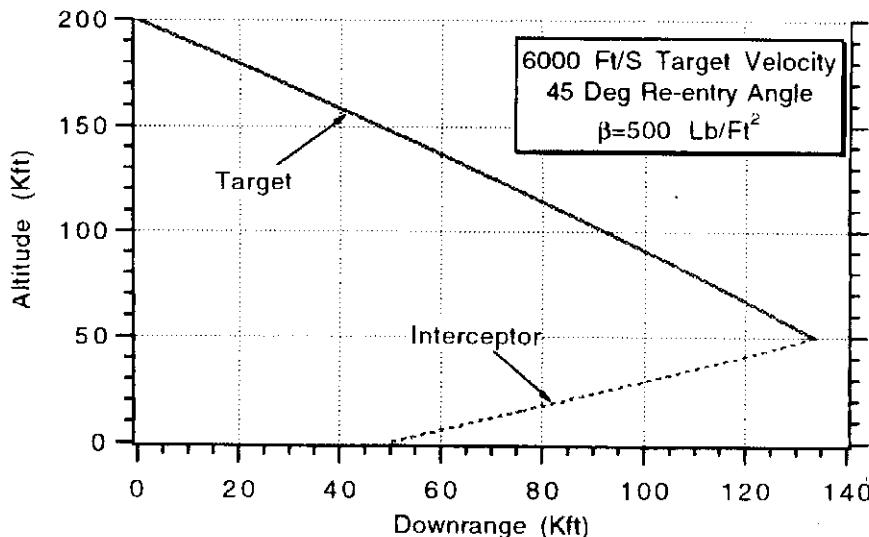


Fig. 17.14 Example of more stressing trajectory.

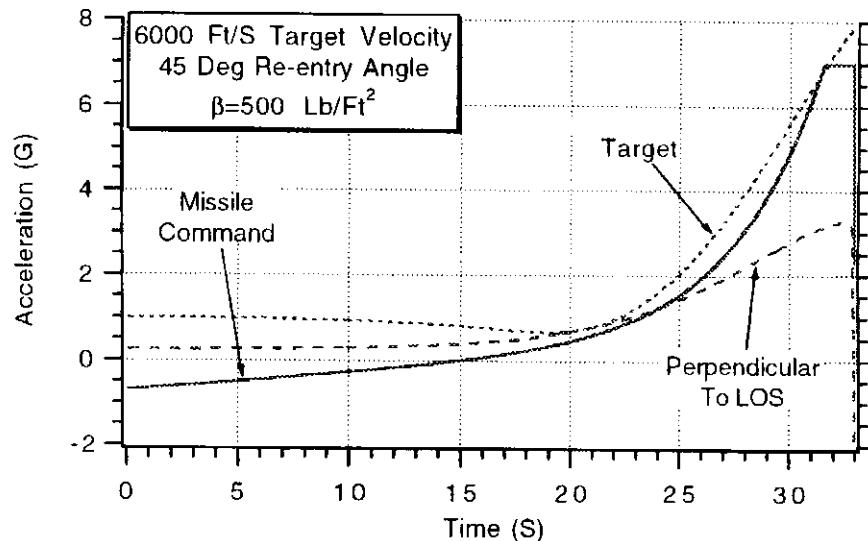


Fig. 17.15 More missile acceleration is required in stressing trajectory because more target deceleration is perpendicular to line of sight.

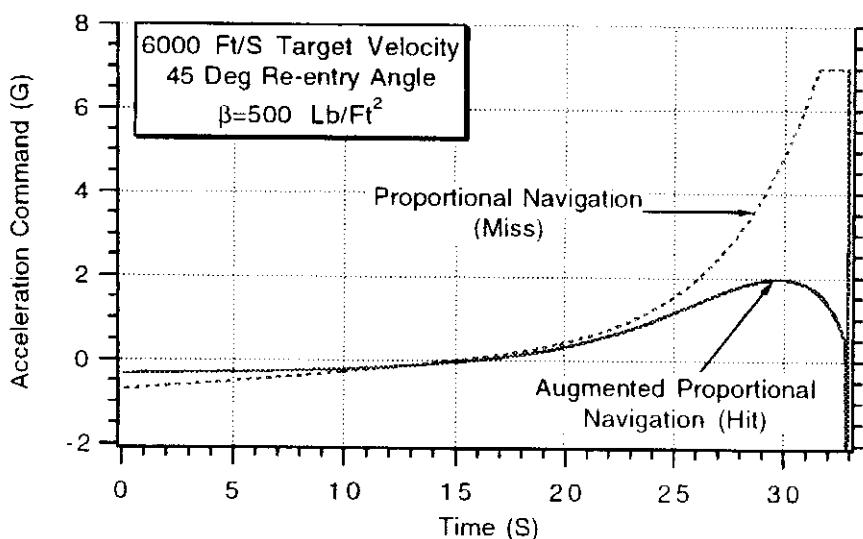


Fig. 17.16 More advanced guidance laws offer significant benefits for stressing trajectory.

shows that the previously unsuccessful intercept can be made successful with this advanced guidance law. However, augmented proportional navigation requires more information than does proportional navigation in order to operate successfully.

Summary

We have seen that ballistic targets can go through enormous decelerations as they re-enter the atmosphere. The magnitude of the target deceleration increases with increasing target speed and increasing target re-entry angle. Any target deceleration that is perpendicular to the line of sight will appear as a target maneuver to the interceptor. It is best for a pursuing interceptor to engage the target on an inverse trajectory where little of the target deceleration is perpendicular to the line of sight.⁵ If for practical reasons the target must be engaged under stressing conditions, the

interceptor must be sized to have adequate acceleration capability if proportional navigation guidance is used. Advanced guidance laws such as predictive guidance can significantly relax the interceptor acceleration requirements if missile-target range information is available and if the target ballistic coefficient is either known or can be estimated accurately.

References

- ¹Riezenman, M., "Revising the Script After Patriot," *IEEE Spectrum*, Sept. 1991, pp. 49-52.
- ²Canavan, G., "Strategic Defense in Past and Future Conflicts," *The Journal of Practical Applications In Space*, Vol. 2, No. 3, Spring 1991, pp. 1-42.
- ³Regan, F., *Re-Entry Vehicle Dynamics*, AIAA Education Series, AIAA, New York, 1984.
- ⁴Jerger, J. J., *System Preliminary Design*, Van Nostrand, Princeton, NJ, 1960.
- ⁵Lin, C. F., *Modern Navigation, Guidance, and Control Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1991.

Extended Kalman Filtering and Ballistic Coefficient Estimation

Introduction

KNOWLEDGE of the target ballistic coefficient can be used in advanced guidance laws such as predictive guidance to relax the interceptor acceleration requirements in stressing engagement geometries. In addition, knowledge of the target ballistic coefficient is required for fire control due to the importance of accurate intercept point predictions in launching the interceptor on a collision course. Therefore the accurate estimation of the ballistic coefficient of a target re-entering the atmosphere is very important for both guidance and fire control purposes. In this chapter we shall show, in detail, how extended Kalman filtering concepts can be applied to ballistic coefficient estimation.

Theoretical Equations¹

To apply extended Kalman filtering techniques, it is first necessary to describe the real world by a set of nonlinear differential equations. One standard dynamical model of the system or real world is given by

$$\dot{x} = f(x) + w$$

where x is a vector of the system states, $f(x)$ is a nonlinear function of those states, and w is a random zero mean process. The process noise matrix describing the random process w for the preceding model is given by

$$Q = E(ww^T)$$

Finally, the measurement equation, required for the application of an extended Kalman filter, is considered a nonlinear function of the states according to

$$z = h(x) + v$$

where v is a random zero mean process described by the measurement noise matrix R , which is defined as

$$R = E(vv^T)$$

For systems in which the measurements are discrete we can rewrite the measurement equation as

$$z_k = h(x_k) + v_k$$

The discrete measurement noise matrix \mathbf{R}_k consists of measurement noise source variances. Since the system and measurement equations are nonlinear, a first-order approximation is used in the Riccati equations for the systems dynamic matrix \mathbf{F} and measurement matrix \mathbf{H} . The matrices are related to the system and measurement equations according to

$$\mathbf{F} = \left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}}$$

$$\mathbf{H} = \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}}$$

The fundamental matrix, also required for the Riccati equations, is usually approximated by the first two terms of the Taylor series expansion $\exp(\mathbf{FT}_s)$ and is given by

$$\Phi_k \approx \mathbf{I} + \mathbf{FT}_s$$

where T_s is the sampling time and \mathbf{I} is the identity matrix. Note that the approximations to the systems dynamics matrix, measurement matrix, and fundamental matrix are time-varying and nonlinear because they depend on the system state estimates. The Riccati equations, needed for the computation of the Kalman gains, are still given by the matrix difference equations of Chapter 9 and are repeated for convenience as

$$\begin{aligned} \mathbf{M}_k &= \Phi_k \mathbf{P}_{k-1} \Phi_k^T + \mathbf{Q}_k \\ \mathbf{K}_k &= \mathbf{M}_k \mathbf{H}^T [\mathbf{HM}_k \mathbf{H}^T + \mathbf{R}_k]^{-1} \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{M}_k \end{aligned}$$

where \mathbf{P}_k is a covariance matrix representing errors in the state estimates before an update and \mathbf{M}_k is the covariance matrix representing errors in the state estimates after an update. Since Φ_k and \mathbf{H} are nonlinear functions of the state estimates, the Kalman gains cannot be computed off line as is possible with a linear Kalman filter. The discrete process noise matrix \mathbf{Q}_k can still be found from the continuous process noise matrix \mathbf{Q} and the fundamental matrix according to

$$\mathbf{Q}_k = \int_0^{T_s} \Phi(\tau) \mathbf{Q} \Phi^T(\tau) d\tau$$

If the dynamical model of a linear Kalman filter is matched to the real world, the covariance matrix \mathbf{P}_k can not only be used to calculate Kalman gains but can also provide exact predictions of the errors in the state estimates. The extended Kalman filter offers no such guarantees and in fact the Riccati equation covariance matrix may indicate excellent performance projections when the filter is performing poorly or is even broken.

The preceding approximations only have to be used in the computation of Kalman gains. The actual extended Kalman filtering equations do not have to use those approximations but instead can be written in terms of the nonlinear measurement equation where the new estimate is the old estimate plus a gain times a

residual, or

$$\hat{x}_{k+1} = \hat{x}_k + K_k [z_k - h(\hat{x}_k)]$$

In the preceding equation the residual is the difference between the actual measurement and the nonlinear measurement equation. The new state estimates do not have to be propagated forward from the old estimate with the fundamental matrix but instead can be obtained directly by integrating the actual nonlinear differential equations at each sampling interval. For example, Euler integration (see Appendix A) can be applied to the nonlinear system differential equations yielding

$$\dot{\hat{x}} = f(\hat{x}_{k-1})$$

$$\hat{x}_k = \hat{x}_{k-1} + \dot{\hat{x}} T_s$$

where state estimates are used instead of the actual states and the sampling time T_s is used as an integration step size.

Differential Equation for One-Dimensional Ballistic Target

To illustrate how extended Kalman filtering concepts can be applied, let us consider the one-dimensional tracking problem originally considered by Gelb¹ and illustrated in Fig. 18.1. In this example a ballistic target is falling on a straight-line path directly toward a surface-based tracking radar. Only drag and gravity act on the ballistic target. This is equivalent to the case in the previous chapter in which the target re-entry angle is 90 deg. In this problem the tracking radar measures the distance from the radar to the target every T_s s. In addition, the tracking radar has the incentive of working well in this application because it is directly in the path of the ballistic target.

We can see from Fig. 18.1 that drag acts upward whereas gravity acts downward. The total acceleration acting on the ballistic target can be expressed in terms of a zero lift drag C_{D0} or a ballistic coefficient β as

$$\frac{dV_{T2}}{dt} = \frac{F_{\text{drag}}}{m} - g = \frac{Q S_{\text{ref}} C_{D0} g}{W} - g = \frac{Qg}{\beta} - g$$

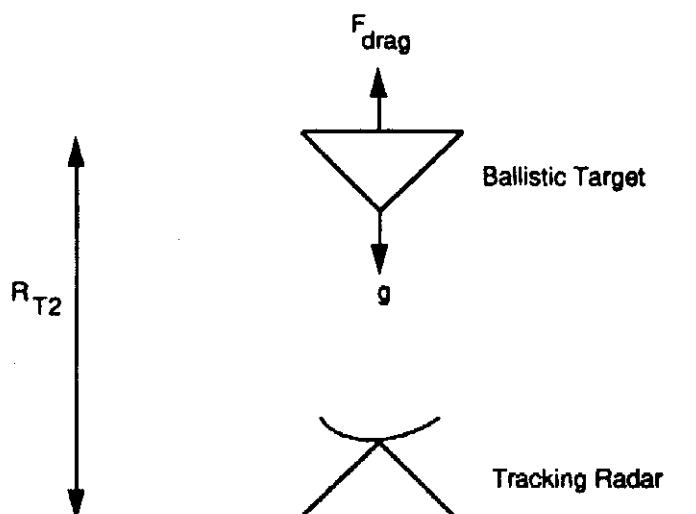


Fig. 18.1 Forces acting on a one-dimensional ballistic target.

where g is the acceleration of gravity and Q is the dynamic pressure. The dynamic pressure can be expressed in terms of the air density ρ and the ballistic target velocity V_{T2} as

$$Q = 0.5\rho V_{T2}^2$$

For purposes of simplicity we can assume that the air density is measured in slug/ft³ and is exponentially related to altitude R_{T2} measured in feet according to

$$\rho = 0.0034e^{-\frac{R_{T2}}{22,000}}$$

and ignore the fact that the coefficients of the exponential approximation change below 30,000 ft. If we assume that the ballistic coefficient of the target is a constant, its derivative must be zero. Therefore the three differential equations that govern the one-dimensional ballistic target can be summarized as

$$\begin{aligned}\dot{R}_{T2} &= V_{T2} \\ \dot{V}_{T2} &= \frac{0.0034e^{-\frac{R_{T2}}{22,000}} g V_{T2}^2}{2\beta} - g \\ \dot{\beta} &= 0\end{aligned}$$

If we want to account for the fact that there may be a large uncertainty in the ballistic coefficient or that it might actually change with time, we could modify the third state equation to be

$$\dot{\beta} = u_s$$

where u_s is white process noise with spectral density Φ_s .

Extended Kalman Filter for One-Dimensional Ballistic Target

In the previous section we showed that the differential equations governing the one-dimensional ballistic target could be expressed in terms of position R_{T2} , velocity V_{T2} , and ballistic coefficient β . Therefore a plausible candidate for the system state vector is

$$\mathbf{x} = \begin{bmatrix} R_{T2} \\ V_{T2} \\ \beta \end{bmatrix}$$

In the Gelb example the tracking radar measures position directly. Therefore the measurement equation in this example is a linear function of the states or

$$R_{T2}^* = R_{T2} + v_k = \underbrace{[1 \ 0 \ 0]}_H \begin{bmatrix} R_{T2} \\ V_{T2} \\ \beta \end{bmatrix} + v_k$$

where the uncertainty in the position measurement is simply the scalar variance or

$$\mathbf{R}_k = E(\mathbf{v}_k \mathbf{v}_k^T) = \sigma_k^2$$

The systems dynamics matrix can be obtained from the three differential equations describing the target according to the definition of the theoretical section as

$$\mathbf{F} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}} = \begin{bmatrix} \frac{\partial \dot{R}_{T2}}{\partial R_{T2}} & \frac{\partial \dot{R}_{T2}}{\partial V_{T2}} & \frac{\partial \dot{R}_{T2}}{\partial \beta} \\ \frac{\partial \dot{V}_{T2}}{\partial R_{T2}} & \frac{\partial \dot{V}_{T2}}{\partial V_{T2}} & \frac{\partial \dot{V}_{T2}}{\partial \beta} \\ \frac{\partial \dot{\beta}}{\partial R_{T2}} & \frac{\partial \dot{\beta}}{\partial V_{T2}} & \frac{\partial \dot{\beta}}{\partial \beta} \end{bmatrix}_{\mathbf{x}=\hat{\mathbf{x}}}$$

After taking partial derivatives of the three system differential equations we obtain

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{-\hat{\rho}g\hat{V}_{T2}^2}{44,000\hat{\beta}} & \frac{\hat{\rho}g\hat{V}_{T2}}{\hat{\beta}} & \frac{-\hat{\rho}g\hat{V}_{T2}^2}{2\hat{\beta}^2} \\ 0 & 0 & 0 \end{bmatrix}$$

where the estimated air density is given by

$$\hat{\rho} = 0.0034e^{-\frac{\hat{R}_{T2}}{22,000}}$$

The fundamental matrix can be obtained from the systems dynamics matrix as

$$\Phi_k \approx \mathbf{I} + \mathbf{F}T_s = \begin{bmatrix} 1 & T_s & 0 \\ \frac{-\hat{\rho}g\hat{V}_{T2}^2T_s}{44,000\hat{\beta}} & 1 + \frac{\hat{\rho}g\hat{V}_{T2}T_s}{\hat{\beta}} & \frac{-\hat{\rho}g\hat{V}_{T2}^2T_s}{2\hat{\beta}^2} \\ 0 & 0 & 1 \end{bmatrix}$$

whereas the continuous process noise matrix can be found from

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Phi_s \end{bmatrix}$$

where Φ_s is the spectral density of the process noise. The discrete process noise matrix can be obtained from the continuous process noise matrix according to the relationship

$$\mathbf{Q}_k = \int_0^{T_s} \Phi(\tau) \mathbf{Q} \Phi^T(\tau) d\tau$$

If we substitute τ for T_s in the previous fundamental matrix approximation, we get

$$\Phi(\tau) = \begin{bmatrix} 0 & \tau & 0 \\ f_{21}\tau & 1 + f_{22}\tau & f_{23}\tau \\ 0 & 0 & 1 \end{bmatrix}$$

where f_{21} , f_{22} , and f_{23} are defined in terms of the state estimates as

$$f_{21} = \frac{-\hat{\rho}g\hat{V}_{T2}^2}{44,000\hat{\beta}}$$

$$f_{22} = \frac{\hat{\rho}g\hat{V}_{T2}}{\hat{\beta}}$$

$$f_{23} = \frac{-\hat{\rho}g\hat{V}_{T2}^2}{2\hat{\beta}^2}$$

Assuming that f_{21} , f_{22} , and f_{23} are approximately constant over the sampling interval, we can integrate with respect to τ and obtain the discrete process noise matrix as

$$\mathbf{Q}_k = \Phi_s \begin{bmatrix} 0 & 0 & 0 \\ 0 & f_{23}^2 \frac{T_s^3}{3} & f_{23} \frac{T_s^2}{2} \\ 0 & f_{23} \frac{T_s^2}{2} & T_s \end{bmatrix}$$

If we want to neglect process noise in our model, then Φ_s is set to zero.

Since the measurement noise matrix \mathbf{R}_k is a scalar in this three-state system, there will only be three Kalman gains at each update (i.e., K_1 , K_2 , and K_3). The Kalman gains will also depend on the state estimates because the fundamental matrix depends on the system state estimates. As was mentioned previously, the new extended Kalman filter states will simply be the old states propagated forward by Euler integration, plus a gain times a residual, or

$$\begin{aligned} \text{Residual} &= R_{T2}^* - \hat{R}_{T2_{k-1}} - \overline{\dot{R}_{T2}} T_s \\ \hat{R}_{T2_k} &= \hat{R}_{T2_{k-1}} + \overline{\dot{R}_{T2}} T_s + K_1 * \text{Residual} \\ \hat{V}_{T2_k} &= \hat{V}_{T2_{k-1}} + \overline{\dot{V}_{T2}} T_s + K_2 * \text{Residual} \\ \hat{\beta}_k &= \hat{\beta}_{k-1} + K_3 * \text{Residual} \end{aligned}$$

The barred quantities in the preceding set of difference equations represent the derivatives required by Euler integration and are obtained directly from the non-linear system equations as

$$\begin{aligned} \overline{\dot{R}_{T2}} &= \hat{V}_{T2_{k-1}} \\ \overline{\dot{V}_{T2}} &= \frac{0.0034e^{-\frac{\hat{R}_{T2_{k-1}}}{22.000}} g \hat{V}_{T2_{k-1}}^2}{2\hat{\beta}_{k-1}} - g \end{aligned}$$

We now have all of the equations necessary to simulate an extended Kalman filter for the one-dimensional tracking problem.

Numerical Example

The same numerical example considered by Gelb¹ is presented here in which a target with ballistic coefficient 500 lb/ft² is initially at 100-kft altitude and is traveling downward at a speed of 6000 ft/s. A surface-based radar measures the range from the radar to the target (i.e., altitude in this example) every 0.05 s with measurement variance 500 ft². The initial estimate of position is 100,025 ft (25-ft error), of velocity is 6150 ft/s (150-ft/s error), and of the ballistic coefficient is 800 lb/ft² (300-lb/ft² error). Uncertainties in the initial state estimates are also reflected in the initial covariance matrix. The first diagonal element of the initial covariance matrix represents the variance of the error in the initial estimate of position and is taken to be the variance of the measurement noise or 500 ft². The second diagonal element of the initial covariance matrix represents the variance of the error in the initial estimate of velocity and is taken to be 20,000 ft²/s² (slightly less than 150²). The third diagonal element of the initial covariance matrix represents the variance of the error in the initial estimate of ballistic coefficient and is taken to be the square of the initial error in estimating the ballistic coefficient or 90,000 lb²/ft⁴ (or 300²). The off-diagonal elements of the initial covariance matrix are set to zero and it is assumed that there is no process noise.

Listing 18.1 presents the resultant one-dimensional extended Kalman filter for ballistic coefficient estimation derived in the previous section. Matrix subroutines² are included for the solution of the discrete Riccati equations so that we do not have to manually perform all matrix operations as was done in Chapter 9. In addition, double-precision arithmetic is used to ensure the accuracy of the results. The initial conditions for the actual ballistic target, filter state estimates, and initial covariance matrix reflect the nominal case. Since Q_{33} (or Φ_s) is set to zero, there is no process noise. The second-order Runge-Kutta numerical integration technique is used for solving the actual nonlinear differential equations representing the ballistic target with integration step size of 0.001 s. The exponential approximation for the air density, used in calculating the drag on the actual ballistic target, matches the assumption made in the extended Kalman filter derivation. The actual ballistic coefficient and its estimate are printed every sampling interval. In addition, the actual errors in the estimate of the ballistic coefficient are computed and compared to the square root of the third diagonal element of the covariance matrix. This diagonal element represents the extended Kalman filter's internal prediction of the error in the estimate of the ballistic coefficient.

The nominal case of Listing 18.1 was run, and the estimated and actual ballistic coefficients are displayed versus altitude in Fig. 18.2. At 100-kft altitude (i.e., beginning of the estimation process) the initial estimate of the ballistic coefficient is on the high side by 300 lb/ft². As the target descends in altitude, the filter's estimate of the ballistic coefficient appears to be continually improving. Below 60-kft altitude, the extended Kalman filter has an excellent estimate of the target's ballistic coefficient.

Figure 18.3 compares single flight results for the actual error in the estimate of the ballistic coefficient (labeled simulation) with the theoretical predictions of the covariance matrix (labeled σ_{THEORY}). Note that the covariance matrix thinks that, as the ballistic target descends in altitude and more measurements are taken, the estimates continually improve (or the error in the estimate of the ballistic coefficient goes to zero). Therefore it appears that the single flight results agree with the covariance matrix predictions in this example.

Listing 18.1 One-dimensional extended Kalman filter for ballistic coefficient estimation

```

GLOBAL DEFINE
    INCLUDE 'quickdraw.inc'
END
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
REAL*8 PHI(3,3),P(3,3),M(3,3),PHIP(3,3),PHIPPHIT(3,3),GAIN(3,1)
REAL*8 Q(3,3),HMAT(1,3),HM(1,3),MHT(3,1)
REAL*8 PHIT(3,3)
REAL*8 HMHT(1,1),HT(3,1),KH(3,3),IDN(3,3),IKH(3,3)
INTEGER ORDER,STEP
RT2=100000.
VT2=-6000.
BETA=500.
RT2H=100025.
VT2H=-6150.
BETAH=800.
OPEN(2,STATUS='UNKNOWN',FILE='COVFIL')
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
ORDER=3
TS=.05
TF=30.
Q33=0./TF
T=0.
S=0.
H=.001
SIGNOISE=SQRT(500.)
DO 1000 I=1,ORDER
DO 1000 J=1,ORDER
    PHI(I,J)=0.
    P(I,J)=0.
    Q(I,J)=0.
    IDN(I,J)=0.
1000 CONTINUE
IDN(1,1)=1.
IDN(2,2)=1.
IDN(3,3)=1.
P(1,1)=SIGNOISE*SIGNOISE
P(2,2)=20000.
P(3,3)=300.**2
DO 1100 I=1,ORDER
    HMAT(1,I)=0.
    HT(I,1)=0.
1100 CONTINUE
HMAT(1,1)=1.
HT(1,1)=1.
10 IF(RT2<0.)GOTO 999
RT2OLD=RT2
VT2OLD=VT2
STEP=1

```

(Contd.)

Listing 18.1 (Continued)

```

GOTO 200
66 STEP=2
RT2=RT2+H*RT2D
VT2=VT2+H*VT2D
T=T+H
GOTO 200
55 CONTINUE
RT2=.5*(RT2OLD+RT2+H*RT2D)
VT2=.5*(VT2OLD+VT2+H*VT2D)
S=S+H
IF(S.LE.(TS-.00001))GOTO 10
S=0.
RHOH=.0034*EXP(-RT2H/22000.)
F21=-32.2*RHOH*VT2H*VT2H/(2.*22000.*BETAH)
F22=RHOH*32.2*VT2H/BETAH
F23=-RHOH*32.2*VT2H*VT2H/(2.*BETAH*BETAH)
PHI(1,1)=1.
PHI(1,2)=TS
PHI(2,1)=F21*TS
PHI(2,2)=1.+F22*TS
PHI(2,3)=F23*TS
PHI(3,3)=1.
Q(2,2)=F23*F23*Q33*TS*TS*TS/3.
Q(2,3)=F23*Q33*TS*TS/2.
Q(3,2)=Q(2,3)
Q(3,3)=Q33*TS
CALL MATTRN(PHI,ORDER,ORDER,PHIT)
CALL MATMUL(PHI,ORDER,ORDER,P,ORDER,ORDER,PHIP)
CALL MATMUL(PHIP,ORDER,ORDER,PHIT,ORDER,ORDER,PHIPPHIT)
CALL MATADD(PHIPPHIT,ORDER,ORDER,Q,M)
CALL MATMUL(HMAT,1,ORDER,M,ORDER,ORDER,HM)
CALL MATMUL(HM,1,ORDER,HT,ORDER,1,HMHT)
HMHTR=HMHT(1,1)+SIGNOISE*SIGNOISE
HMHTRINV=1./HMHTR
CALL MATMUL(M,ORDER,ORDER,HT,ORDER,1,MHT)
DO 150 I=1,ORDER
    GAIN(I,1)=MHT(I,1)*HMHTRINV
150 CONTINUE
CALL MATMUL(GAIN,ORDER,1,HMAT,1,ORDER,KH)
CALL MATSUB(IDN,ORDER,ORDER,KH,IKH)
CALL MATMUL(IKH,ORDER,ORDER,M,ORDER,ORDER,P)
CALL GAUSS(XNOISE,SIGNOISE)
RT2DB=VT2H
VT2DB=.0034*32.2*VT2H*VT2H*EXP(-RT2H/22000.)/(2.*BETAH)-32.2
RES=RT2+XNOISE-(RT2H+RT2DB*TS)
RT2H=RT2H+RT2DB*TS+GAIN(1,1)*RES
VT2H=VT2H+VT2DB*TS+GAIN(2,1)*RES
BETAH=BETAH+GAIN(3,1)*RES

ERRY=RT2-RT2H
SP11=SQRT(P(1,1))

```

(Contd.)

Listing 18.1 (Continued)

```

ERRV=VT2-VT2H
SP22=SQRT(P(2,2))
ERRBETA=BETA-BETAH
SP33=SQRT(P(3,3))
RT2K=RT2/1000.
WRITE(*,97)T,RT2K,RT2,RT2H,VT2,VT2H,BETA,BETAH
WRITE(1,97)T,RT2K,RT2,RT2H,VT2,VT2H,BETA,BETAH
WRITE(2,98)T,RT2K,ERRY,SP11,-SP11,ERRV,SP22,-SP22,ERRBETA,SP33,
1      -SP33
      GOTO 10
97    FORMAT(8F10.3)
98    FORMAT(11F10.3)
200   CONTINUE
      RT2D=VT2
      VT2D=.0034*32.2*VT2*VT2*EXP(-RT2/22000.)/(2.*BETA)-32.2
      IF(STEP-1)66,66,55
999   CONTINUE
      PAUSE
      END

      SUBROUTINE GAUSS(X,SIG)
      IMPLICIT REAL*8(A-H)
      IMPLICIT REAL*8(O-Z)
      INTEGER SUM
      SUM=0
      DO 14 J=1,6
      IRAN=Random()
      SUM=SUM+IRAN
14    CONTINUE
      X=SUM/65536.
      X=1.414*X*SIG
      RETURN
      END

      SUBROUTINE MATTRN(A,IROW,ICOL,AT)
      IMPLICIT REAL*8 (A-H)
      IMPLICIT REAL*8 (O-Z)
      REAL*8 A(IROW,ICOL),AT(ICOL,IROW)
      DO 105 I=1,IROW
      DO 105 J=1,ICOL
      AT(J,I)=A(I,J)
105   CONTINUE
      RETURN
      END

      SUBROUTINE MATMUL(A,IROW,ICOL,B,JROW,JCOL,C)
      IMPLICIT REAL*8 (A-H)
      IMPLICIT REAL*8 (O-Z)
      REAL*8 A(IROW,ICOL),B(JROW,JCOL),C(IROW,JCOL)
      DO 110 I=1,IROW

```

(Contd.)

Listing 18.1 (Continued)

```

DO 110 J=1,JCOL
C(I,J)=0.
DO 110 K=1,ICOL
C(I,J)= C(I,J)+A(I,K)*B(K,J)
110 CONTINUE
RETURN
END

SUBROUTINE MATADD(A,IROW,ICOL,B,C)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
REAL*8 A(IROW,ICOL),B(IROW,ICOL),C(IROW,ICOL)
DO 120 I=1,IROW
DO 120 J=1,ICOL
C(I,J)=A(I,J)+B(I,J)
120 CONTINUE
RETURN
END

SUBROUTINE MATSUB(A,IROW,ICOL,B,C)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
REAL*8 A(IROW,ICOL),B(IROW,ICOL),C(IROW,ICOL)
DO 120 I=1,IROW
DO 120 J=1,ICOL
C(I,J)=A(I,J)-B(I,J)
120 CONTINUE
RETURN
END

```

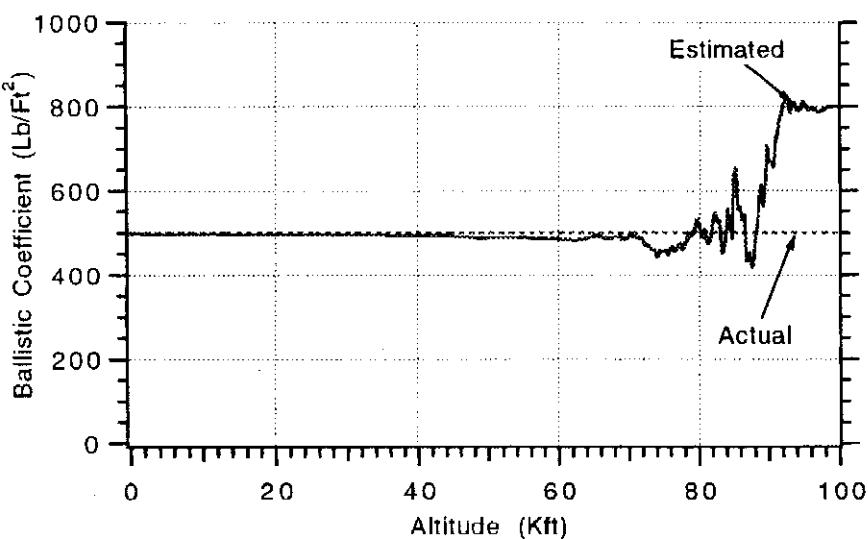


Fig. 18.2 After a while extended Kalman filter is able to estimate ballistic coefficient

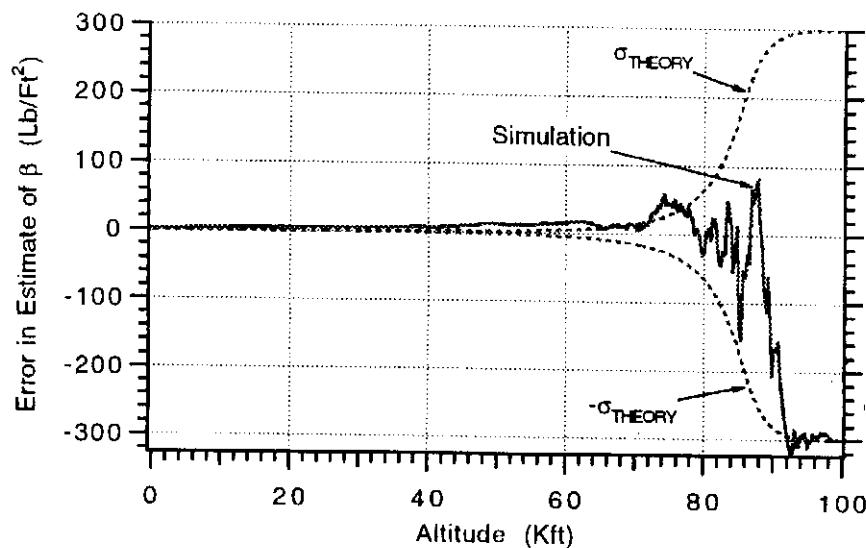


Fig. 18.3 Theory and single flight results appear to agree.

We can rerun the nominal case but start at 200-kft altitude rather than 100-kft altitude. Figure 18.4 indicates that both the theoretical and actual errors in the estimate of the ballistic coefficient do not improve from the initial guess until the ballistic target descends below 150-kft altitude. The lack of estimation capability is due to the absence of drag in the high altitude regime. In other words, at the higher altitudes the ballistic coefficient is not observable from just position measurements. This result can be very important if we must predict the future location of the ballistic target in the atmosphere based on estimates of the ballistic coefficient at very high altitudes.

Another case was run where the ballistic target started at the nominal altitude of 100 kft. However, this time the initial estimate of the ballistic coefficient was 1500 lb/ft² (i.e., 1000-lb/ft² error) rather than 800 lb/ft² (i.e., 300-lb/ft² error). The third diagonal element of the initial covariance matrix was increased to 1000² to reflect the larger initial uncertainty in the ballistic coefficient. Figure 18.5 shows

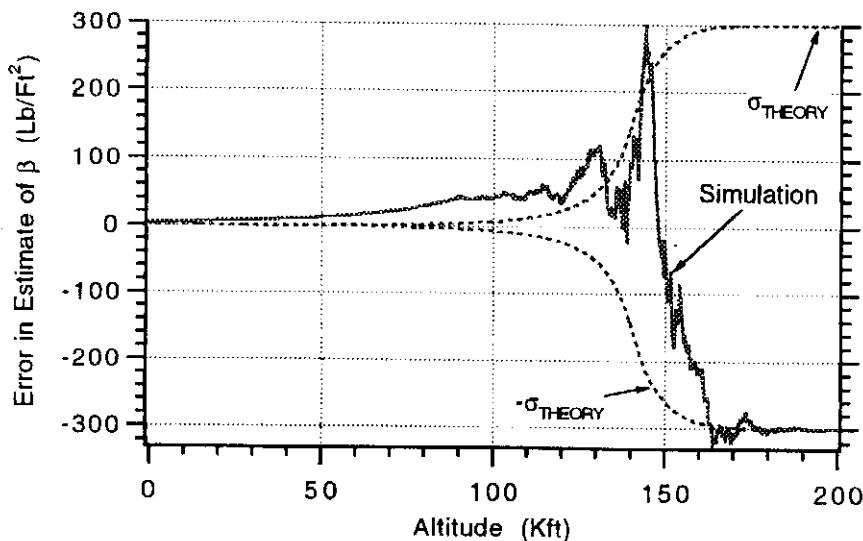


Fig. 18.4 Extended Kalman filter unable to estimate ballistic coefficient above 150 kft.

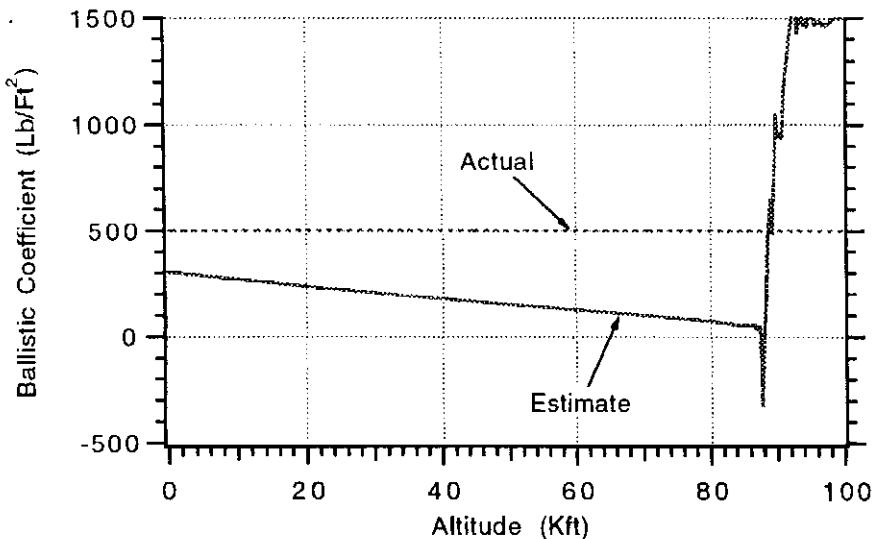


Fig. 18.5 Extended Kalman filter breaks if we severely overestimate ballistic coefficient.

that under these circumstances, the extended Kalman filter is unable to estimate the ballistic coefficient before the ballistic target hits the tracking radar. Unlike a linear Kalman filter, the extended Kalman filter's performance is highly dependent on initial conditions!

Figure 18.6 shows that even though the extended Kalman filter is not able to estimate the ballistic coefficient when we initially severely overestimate the ballistic coefficient by 1000 lb/ft², the filter's covariance matrix predictions indicate that the errors in the estimate of the ballistic coefficient are near zero. Apparently this filter does not even realize when it is broken! Therefore we can see that although the covariance matrix is required for Kalman gain computation, its theoretical predictions are not always useful!

In the preceding example the filter is not able to recover when we initially overestimate the ballistic coefficient by a large amount. The filter's lack of robustness is due to a zero process noise matrix (i.e., $Q_k = 0$). When the process noise is zero,

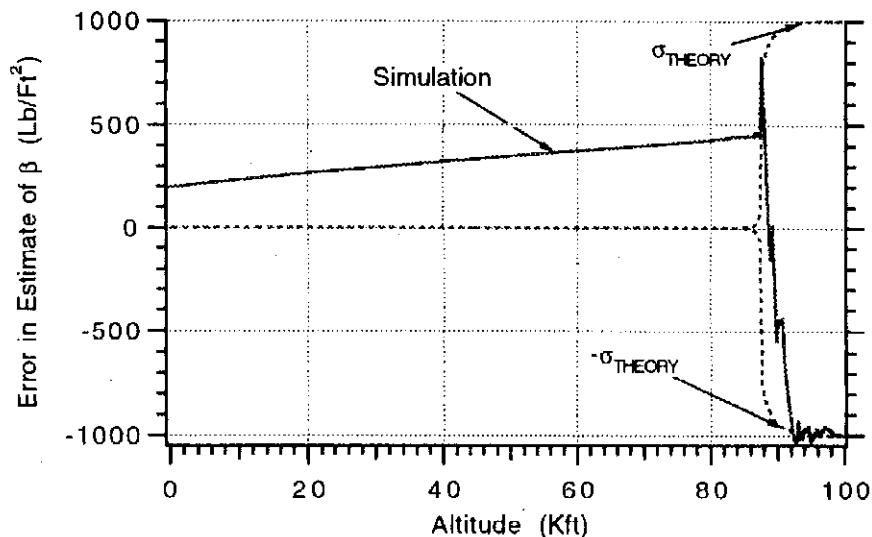


Fig. 18.6 Extended Kalman filter does not even realize it is broken.

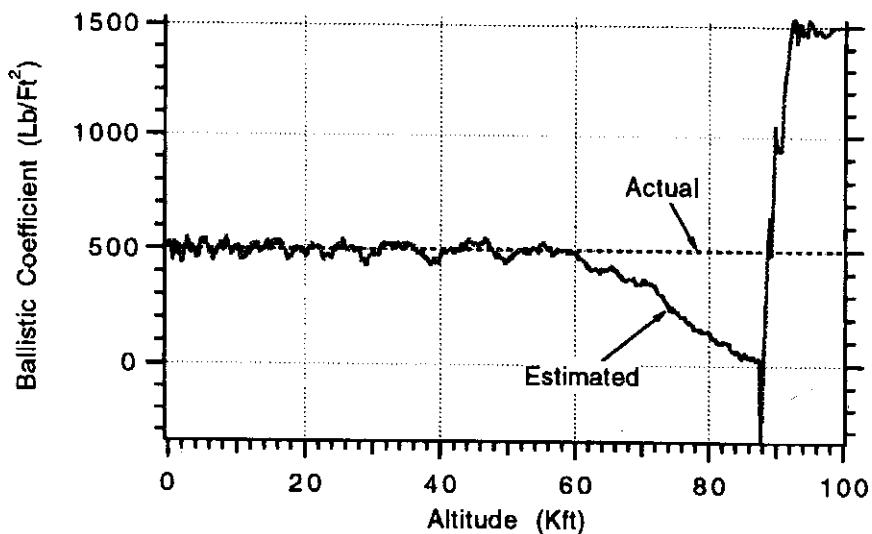


Fig. 18.7 Adding process noise enables extended Kalman filter to recover from overestimating ballistic coefficient.

the filter thinks it is very smart (i.e., it must have a terrific dynamical model) and eventually stops looking at the measurements. Under these circumstances the filter changes from an extended Kalman filter to an arrogant Kalman filter! Process noise was added to the filter with value $\Phi_s = 1000^2/30$ to indicate large uncertainty in the ballistic coefficient model. Figure 18.7 shows that when the process noise is added, the estimated and actual ballistic coefficients converge after a while.

Figure 18.8 now shows that when realistic process noise is added to reflect large uncertainties, the single flight results and covariance matrix predictions of the error in the estimate of ballistic coefficient are in agreement after the ballistic target has descended below 60 kft. Although the estimates are not great initially, they are continually improving.

We can now revisit the nominal results of Fig. 18.3 when, without a process noise matrix, the arrogant Kalman filter thought its estimates were continually improving as more measurements were taken. If we rerun the nominal case with

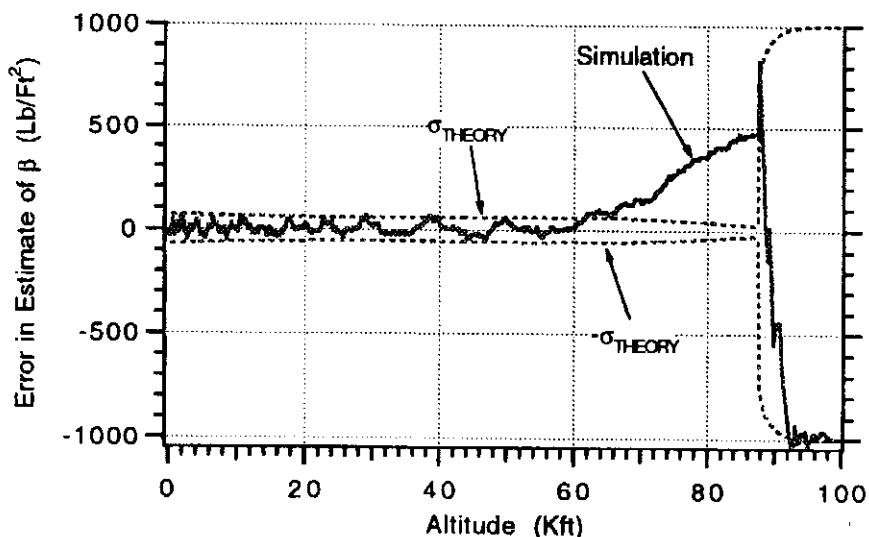


Fig. 18.8 Adding process noise makes covariance matrix predictions more meaningful.

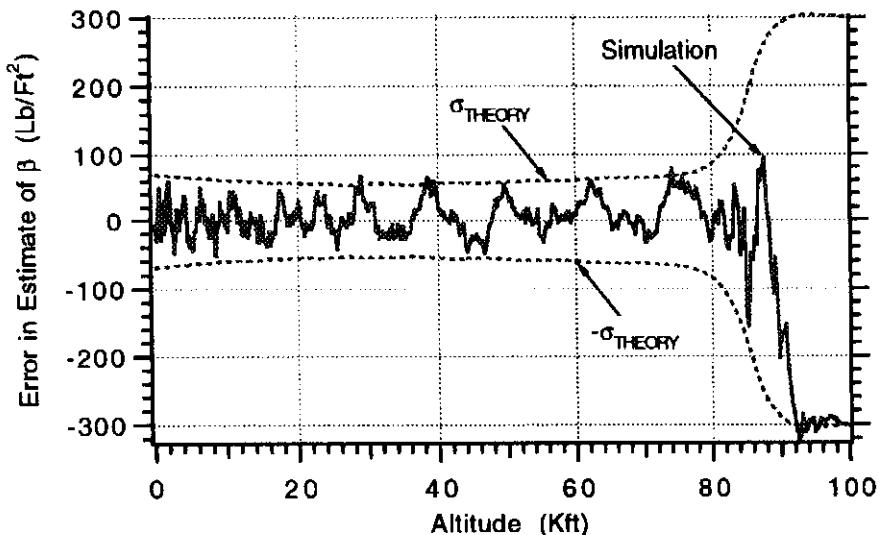


Fig. 18.9 Nominal results are worse when there is process noise but filter is more robust.

$\Phi_s = 300^2/30$ to reflect the fact that we have a smaller uncertainty in our knowledge of the ballistic coefficient, we get more sobering results as shown in Fig. 18.9. We can see that after a while the filter's estimate of the ballistic coefficient does not improve. However, our initial uncertainty in our estimate of the ballistic coefficient has been reduced from the initial guess of $300 \text{ lb}/\text{ft}^2$ to an estimate with slightly under $100 \text{ lb}/\text{ft}^2$ of error. In addition, we now have a filter that is more robust to initialization errors.

Summary

We have seen, using a simplified extended Kalman filter, the difficulties in estimating a target's ballistic coefficient—especially at high altitude where there is very little drag. We have also observed that using zero process noise in the filter gain computations leads to overly optimistic performance projections and makes the filter fragile in the presence of large initialization errors. Adding process noise to the filter's gain computation appears to be the engineering fix when there is large uncertainty. Although adding process noise degrades filter performance under benign conditions, it also enables the filter to perform adequately under more stressing conditions.

References

- ¹Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.
- ²Wolf, P. M., and Koelling, C. P., *Basic Engineering, Science and Business Programs for the Apple II and IIe*, Bradly Communications Co. Inc., Bowie, MD, 1984.



Ballistic Target Challenges

Introduction

In this chapter we shall integrate many of the text's concepts in order to illustrate, from a miss distance point of view, additional reasons why ballistic targets are challenging. First, new miss distance formulas will be derived in order to show how the miss due to noise depends on the closing velocity and guidance system time constant. Next, a new formula will be presented showing how the minimum possible guidance system time constant depends on radome slope, closing velocity, and missile turning rate time constant. For head-on scenarios, numerical examples will be presented showing how low-closing velocity aircraft engagements and high-closing velocity ballistic target engagements yield different miss distances even though the error sources may be the same.

Miss Distance Due to Noise

In Chapter 3 closed-form solutions for various deterministic error sources were derived for a single time constant proportional navigation guidance system. We demonstrated in Chapter 6 that although the miss distances generated with the low-order model of the guidance system were serious underestimates of the actual miss, the closed-form solutions were useful because the miss distance normalization factors did not change for higher order guidance systems. We shall use the same methodology in obtaining miss distance formulas due to noise error sources. First we shall obtain noise miss distance closed-form solutions for the single time constant guidance system and then use the brute force method to extend those solutions for a fifth-order binomial guidance system.

As was done with deterministic error sources in Chapter 3, we shall also use the method of adjoints for finding miss distance formulas due to various noise error sources found in homing guidance systems.¹ The noise sources considered are those usually associated with radar homing missiles. Figure 19.1 presents a generalized model of the homing loop similar to Fig. 3.16 except this time the error sources are random rather than deterministic. The first error source is glint or scintillation noise and is caused by random fluctuations of the target radar return. The spectral density of this error source is related to the physical dimensions of the target. Strictly speaking, glint should not be modeled as white noise since it may be highly correlated.² For semiactive systems, in which the target is illuminated by a transmitter not on the interceptor, range dependent noise is the thermal noise produced in the interceptor radar receiver according to the radar range equation. In this simplified model the spectral density of the noise is defined at a reference

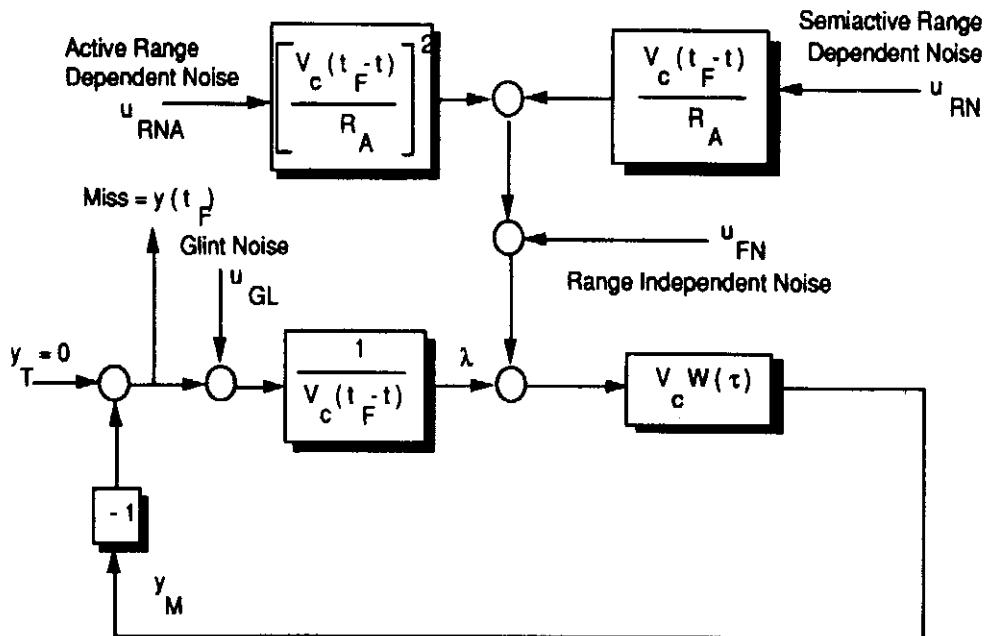


Fig. 19.1 Generalized proportional navigation guidance system with noise sources.

range R_A . The noise is proportional to the distance from the missile to the target and goes to zero at intercept.³ For active systems in which the target is illuminated by a transmitter on the interceptor, range dependent noise is proportional to the square of the distance from the missile to the target.⁴ Other noise sources are usually lumped together and termed range independent noise.

The spectral densities for the various white noise error sources are given by

$$\Phi_{FN} = \text{Spectral Density of } u_{FN}$$

$$\Phi_{RN} = \text{Spectral Density of } u_{RN}$$

$$\Phi_{RNA} = \text{Spectral Density of } u_{RNA}$$

$$\Phi_{GL} = \text{Spectral Density of } u_{GL}$$

The generalized homing loop adjoint model, which appears in Fig. 19.2, can be found from Fig. 19.1 by using the rules of adjoints developed in Chapters 3 and 4 and then applying some block diagram manipulation. Note that all white noise inputs of the original system become outputs in the adjoint system by squaring, integrating, and multiplying by the spectral density of each of the white noise error sources. Critical points in the adjoint block diagram have been labeled $H(\tau)$, $g(\tau)$, and $f(\tau)$.

To illustrate how noise miss distance formulas can be derived, let us consider a single time constant guidance system with a navigation ratio of 3. Recall from Chapter 3 that for this case W can be represented in the frequency domain as

$$W(s) = \frac{3}{s(1+sT)}$$

where T is the guidance system time constant. We showed in Chapter 3 that $1 - H$ could be found in the frequency domain from W according to the relationship

$$1 - H(s) = e^{\int W ds} = \frac{s^3}{(s + 1/T)^3}$$

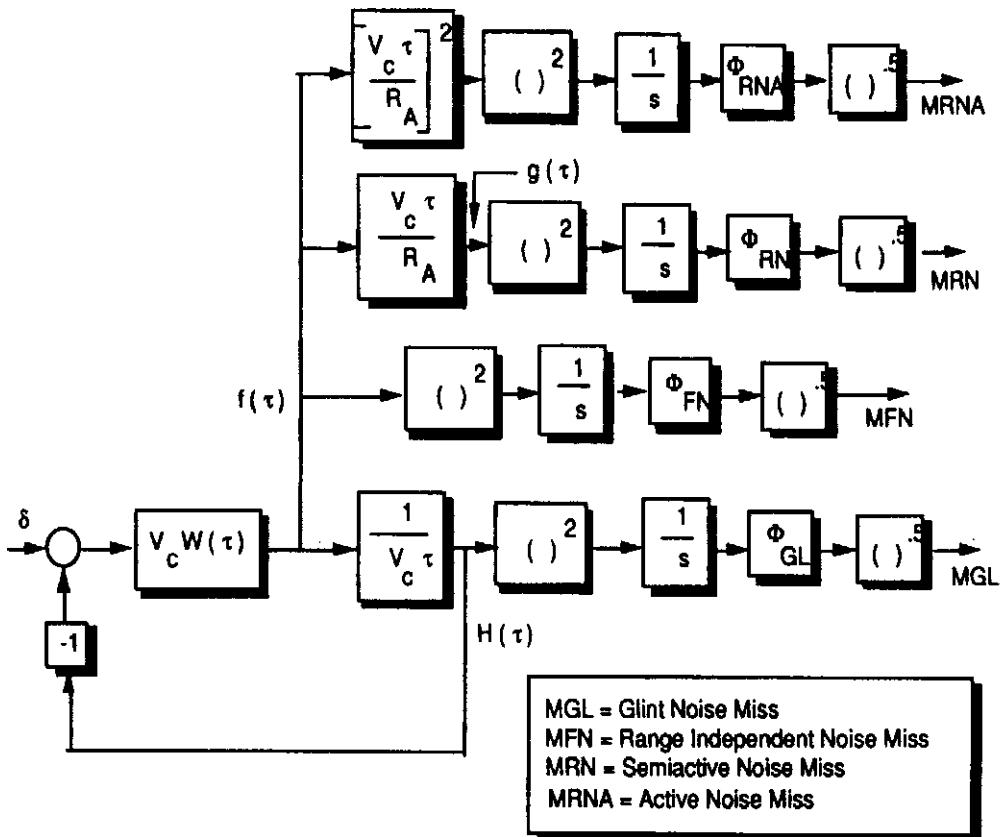


Fig. 19.2 Generalized adjoint homing loop model.

Solving for H by algebraic manipulation of the preceding equation yields

$$H(s) = \frac{1 + 3sT + 3s^2T^2}{(1 + sT)^3}$$

We can convert H from the frequency domain to the adjoint time domain by taking the inverse Laplace transform of the preceding equation yielding

$$H(\tau) = \frac{e^{-\frac{\tau}{T}}}{T} \left[3 - \frac{3\tau}{T} + \frac{\tau^2}{2T^2} \right]$$

where τ can be interpreted as the homing time or time of flight. We can see from Fig. 19.2 that the miss due to glint noise (MGL) can be found by squaring and integrating $H(\tau)$ and then multiplying the result by the square root of the glint noise spectral density. We can simplify matters and integrate from zero to infinity yielding the closed-form solution for the standard deviation of the steady-state miss due to white glint noise as⁵

$$MGL_{N=3} = \Phi_{GL}^{0.5} \sqrt{\int_0^\infty H^2(\tau) d\tau} = 1.44T^{-0.5} \Phi_{GL}^{0.5}$$

From the preceding formula we can see that unlike most other error source results we have studied, if the guidance system time constant is reduced, the miss due to glint noise will increase! We can also see that, unlike deterministic error source results, the miss due to glint noise does not go to zero as the homing time approaches infinity. In other words, there will always be some miss distance due to glint noise, no matter how much homing time we have. Closed-form miss distance

formulas for the single time constant guidance system can also be derived, in a similar manner, for different effective navigation ratios. The steady-state standard deviation of the miss due to glint noise for effective navigation ratios of four and five can be found to be

$$MGL_{N'=4} = 1.71T^{-0.5}\Phi_{GL}^{0.5}$$

$$MGL_{N'=5} = 1.94T^{-0.5}\Phi_{GL}^{0.5}$$

Note that the miss distance normalization factors do not change with different effective navigation ratios. However the miss distance coefficients due to glint noise increase slightly with increasing effective navigation ratio.

To find the steady-state standard deviation of the miss due to range independent noise (*MFN*), it is first necessary to square and integrate the expression $(1 - H)V_c W$ as can be seen from Fig. 19.2. For a single time constant guidance system with an effective navigation ratio of 3, $(1 - H)V_c W$ in the frequency domain becomes

$$[1 - H(s)]V_c W(s) = \frac{3V_c s^2}{T(s + 1/T)^4}$$

We can convert $(1 - H)V_c W$ to the adjoint time domain by taking the inverse Laplace transform of the preceding equation obtaining

$$f(\tau) = \mathcal{L}^{-1}\{[1 - H(s)]V_c W(s)\} = \frac{3V_c \tau e^{-\tau/T}}{T} \left[1 - \frac{\tau}{T} + \frac{\tau^2}{6T^2} \right]$$

Squaring and integrating the preceding expression from zero to infinity yields the steady-state formula for the standard deviation of the miss distance due to range independent noise as

$$MFN_{N'=3} = \Phi_{FN}^{0.5} \sqrt{\int_0^\infty f^2(\tau) d\tau} = 0.532V_c T^{0.5} \Phi_{FN}^{0.5}$$

where V_c is the closing velocity. In this case we can see that there is now a geometry dependence on the miss distance because the miss is proportional to closing velocity. Higher closing velocity engagement scenarios will yield more miss distance due to range independent noise. On the other hand, we can see from the preceding expression that, unlike the glint noise case, reducing the guidance system time constant will decrease the miss due to range independent noise. Closed-form miss distance formulas for range independent noise in a single time constant guidance system can also be derived in a similar manner for different effective navigation ratios. The steady-state standard deviation of the miss due to range independent noise for effective navigation ratios of four and five can be found to be

$$MFN_{N'=4} = 0.561V_c T^{0.5} \Phi_{FN}^{0.5}$$

$$MFN_{N'=5} = 0.588V_c T^{0.5} \Phi_{FN}^{0.5}$$

Here again we can see that there is a slight increase in the miss distance coefficients as the effective navigation ratio increases.

We can find the standard deviation of the steady-state miss due to semiaactive range dependent noise (*MRN*) by squaring and integrating $g(\tau)$ as shown in

Fig. 19.2. The expression for $g(\tau)$ can be found from $f(\tau)$ as

$$g(\tau) = \frac{V_c \tau f(\tau)}{R_A} = \frac{3V_c^2 \tau^2 e^{-\tau/T}}{R_A T} \left[1 - \frac{\tau}{T} + \frac{\tau^2}{6T^2} \right]$$

Squaring and integrating $g(\tau)$ from zero to infinity yields the standard deviation of the steady-state miss due to semiactive range dependent noise for an effective navigation ratio of three as

$$MRN_{N'=3} = \Phi_{RN}^{0.5} \sqrt{\int_0^\infty g^2(\tau) d\tau} = \frac{1.06 V_c^2 T^{1.5} \Phi_{RN}^{0.5}}{R_A}$$

where R_A is taken to be a reference range. We can see that the closing velocity and time constant dependence of the miss is much greater for semiactive range dependent noise than it was for range independent noise. Steady-state miss distance formulas for semiactive range dependent noise in a single time constant guidance system can be found for higher effective navigation ratios in a similar way and are

$$MRN_{N'=4} = \frac{1.10 V_c^2 T^{1.5} \Phi_{RN}^{0.5}}{R_A}$$

$$MRN_{N'=5} = \frac{1.15 V_c^2 T^{1.5} \Phi_{RN}^{0.5}}{R_A}$$

Again we can see that the miss distance coefficients increase slightly with increasing effective navigation ratio. For active systems miss distance formulas can also be derived for a single time constant system. The steady-state standard deviation of the miss due to active range dependent noise ($MRNA$) for effective navigation ratios of three, four, and five are

$$MRNA_{N'=3} = \frac{4.66 V_c^3 T^{2.5} \Phi_{RNA}^{0.5}}{R_A^2}$$

$$MRNA_{N'=4} = \frac{4.68 V_c^3 T^{2.5} \Phi_{RNA}^{0.5}}{R_A^2}$$

$$MRNA_{N'=5} = \frac{4.82 V_c^3 T^{2.5} \Phi_{RNA}^{0.5}}{R_A^2}$$

We can see that the closing velocity and time constant dependence for active range dependent noise is even stronger than it was for semiactive range dependent noise.

Fifth-Order Binomial Guidance System Miss Distances

In Chapter 6 we saw that once we had closed-form solutions for a single time constant guidance system, we could obtain solutions for higher order systems by the method of brute force because the miss distance normalization factors remained unchanged with system order (i.e., only coefficients change). In this section we shall use the same method to get noise miss distance formulas for higher order systems. The adjoint model for a fifth-order binomial guidance system, first shown in Fig. 6.5, has been modified to include the noise error sources discussed in the previous

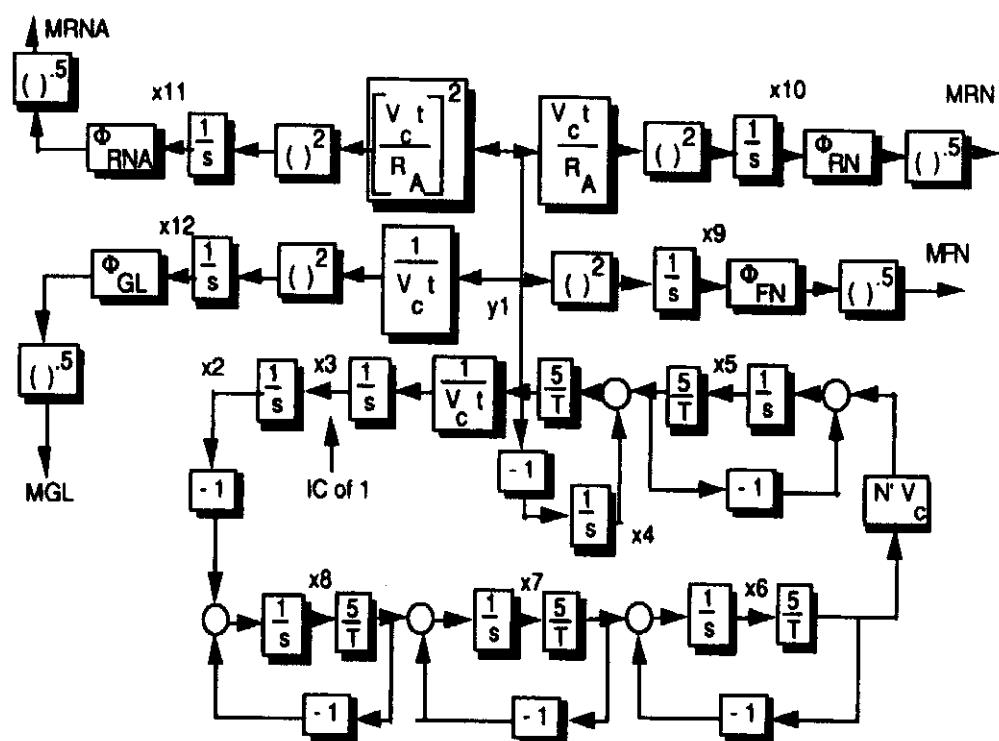


Fig. 19.3 Adjoint of fifth-order binomial guidance system for noise miss distance calculations.

section and is redrawn in Fig. 19.3. Deterministic error sources are not shown in this diagram. Note that the basic model is unchanged. We have only added new outputs to correspond to the miss due to range independent noise, glint noise, and both semi-active and active range dependent noise. This has been accomplished by squaring and integrating signals proportional to the adjoint variable y_1 shown in Fig. 19.3.

Adjoint simulation Listing 6.1 has also been modified for noise miss distance calculations and appears in Listing 19.1. The noise spectral densities, closing velocity, guidance system time constant, and reference range have been set to unity so that we can calculate the coefficients for the noise miss distance normalization factors.

Using the method of brute force with the adjoint simulation of Listing 19.1, Table 19.1 was generated for the standard deviation of the steady-state noise miss

Table 19.1 Miss distance formulas for fifth-order binomial guidance system

Error source	Normalization factor	Miss coefficient		
		$N' = 3$	$N' = 4$	$N' = 5$
Range independent noise	$\frac{\sigma_{\text{Miss}}}{V_c T^{0.5} \Phi_{FN}^{0.5}}$	3.04	5.08	8.19
Semiactive range dependent noise	$\frac{\sigma_{\text{Miss}} R_A}{V_c^2 T^{1.5} \Phi_{FN}}$	9.47	18.4	33.7
Active range dependent noise	$\frac{\sigma_{\text{Miss}} R_A^2}{V_c^3 T^{2.5} \Phi_{RNA}^{0.5}}$	41.3	89.4	182
Glint noise	$\frac{\sigma_{\text{Miss}} T^{0.5}}{\Phi_{GL}^{0.5}}$	1.68	2.35	3.21

Listing 19.1 Adjoint of fifth-order binomial guidance system for noise miss distance calculations

```

INTEGER STEP
DATA XNP,TAU,TF,VC/ 3.,1.,10.,1./
DATA PHIFN,PHIRN,PHIRNA,PHIGL,RA/1.,1.,1.,1.,1./
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
T=0.
S=0.
TP=T+.00001
X2=0
X3=1
X4=0
X5=0.
X6=0.
X7=0.
X8=0.
X9=0.
X10=0.
X11=0.
X12=0.
H=.01
10 IF(TP>(TF-.00001))GOTO 999
S=S+H
X2OLD=X2
X3OLD=X3
X4OLD=X4
X5OLD=X5
X6OLD=X6
X7OLD=X7
X8OLD=X8
X9OLD=X9
X10OLD=X10
X11OLD=X11
X12OLD=X12
STEP=1
GOTO 200
66 STEP=2
X2=X2+H*X2D
X3=X3+H*X3D
X4=X4+H*X4D
X5=X5+H*X5D
X6=X6+H*X6D
X7=X7+H*X7D
X8=X8+H*X8D
X9=X9+H*X9D
X10=X10+H*X10D
X11=X11+H*X11D
X12=X12+H*X12D
TP=TP+H
GOTO 200
55 CONTINUE

```

Listing 19.1 (Continued)

```

X2=(X2OLD+X2)/2+.5*H*X2D
X3=(X3OLD+X3)/2+.5*H*X3D
X4=(X4OLD+X4)/2+.5*H*X4D
X5=(X5OLD+X5)/2+.5*H*X5D
X6=(X6OLD+X6)/2+.5*H*X6D
X7=(X7OLD+X7)/2+.5*H*X7D
X8=(X8OLD+X8)/2+.5*H*X8D
X9=(X9OLD+X9)/2+.5*H*X9D
X10=(X10OLD+X10)/2+.5*H*X10D
X11=(X11OLD+X11)/2+.5*H*X11D
X12=(X12OLD+X12)/2+.5*H*X12D
IF(S<.09999)GOTO 10
S=0.
XMFN=SQRT(X9*PHIFN)
XMRN=SQRT(X10*PHIRN)
XMRNA=SQRT(X11*PHIRNA)
XMGL=SQRT(X12*PHIGL)
WRITE(*,97)TP,XMFN,XMRN,XMRNA,XMGL
WRITE(1,97)TP,XMFN,XMRN,XMRNA,XMGL
GOTO 10
97 FORMAT(5F10.3)
200 CONTINUE
X2D=X3
Y1=-5.*(.5*X5/TAU+X4)/TAU
TGO=TP+.00001
X3D=Y1/(VC*TGO)
X4D=-Y1
X5D=-5.*X5/TAU+5.*X6*XNP*VC/TAU
X6D=-5.*X6/TAU+5.*X7/TAU
X7D=-5.*X7/TAU+5.*X8/TAU
X8D=-5.*X8/TAU-X2
X9D=Y1**2
X10D=(Y1*VC*TGO/RA)**2
X11D=(Y1*((VC*TGO/RA)**2))**2
X12D=(Y1/(VC*TGO))**2
IF(STEP-1)66,66,55
999 CONTINUE
PAUSE
CLOSE(1)
END

```

distances for the fifth-order binomial proportional navigation guidance system under consideration. We can see from Table 19.1 that although the miss distance normalization factors are the same as they were for a single time constant guidance system, the miss distance coefficients are an order of magnitude larger! This means that the miss distances for the higher order system will also be an order of magnitude greater than that of the single time constant guidance system. In addition, we can see that unlike the single time constant guidance system, the effective navigation ratio has a strong influence on the miss distance for the higher order system. Increasing the effective navigation ratio significantly increases the miss distance due to noise.

Minimum Guidance System Time Constant

We saw in Chapter 6 that the radome aberration effects create an unwanted feedback path in the guidance system which can cause stability problems. In the presence of radome slope the guidance system transfer function can be derived from Fig. 6.21 and is given by

$$\frac{n_L}{\lambda} = N'V_c \left/ \left\{ \left[1 + \frac{sT}{5} \right]^5 + \frac{N'V_c R}{V_M} (1 + T_\alpha s) \right\} \right.$$

where R is the radome slope, V_M the missile velocity, T_α an aerodynamic parameter known as the turning rate time constant, and T the guidance system time constant. We can see from the preceding equation that if the radome slope is zero the guidance system transfer function reduces to a fifth-order binomial. For different combinations of guidance system parameters, simulation experiments conducted in Chapter 6 showed that the guidance system can be unstable. One can show mathematically⁶ that if the ratio of the turning rate time constant to the guidance system time constant is greater than unity or

$$\frac{T_\alpha}{T} > 1$$

then the guidance system transfer will be stable only if the following inequality is satisfied

$$-0.79 < \frac{N'V_c RT_\alpha}{V_M T} < 2.07$$

If the radome slope is negative, we can find from the preceding inequality that the minimum guidance system time constant to yield a stable guidance system is

$$T_{\min} = \frac{N'V_c RT_\alpha}{0.79 V_M}$$

We can see from the preceding relationship that engagements with larger closing velocities (i.e., ballistic targets) or those taking place at high altitudes (i.e., larger turning rate time constant) will require a larger guidance system time constant in order to keep the guidance system stable. However, larger guidance system time constants will also tend to increase the miss distance.

Missile Turning Rate Time Constant^{7,8}

We saw in Chapter 6 that the missile turning rate time constant T_α had a significant interaction with radome effects. The missile turning rate time constant can be defined as the amount of time it takes to turn the missile flight path angle γ through an equivalent angle of attack α or

$$T_\alpha = \frac{\alpha}{\dot{\gamma}} = \frac{\alpha V_M}{n_L}$$

where V_M is missile velocity and n_L is missile acceleration. We showed in Chapter 11 the relationship between the lift coefficient C_L and missile acceleration.

Substitution of those relationships into the preceding expression yields

$$T_\alpha = \frac{\alpha V_M W}{g Q S_{\text{ref}} C_L}$$

where W is the missile weight, S_{ref} is the missile reference area, g is the acceleration of gravity, and Q is the dynamic pressure. If we assume our missile to be a cylinder or flying telephone pole, we showed in Chapter 17 that the lift coefficient could be expressed as

$$C_L = 2\alpha + \frac{1.5 S_{\text{plan}} \alpha^2}{S_{\text{ref}}}$$

where S_{plan} is the missile planform area. Substitution of the lift coefficient expression into the turning rate time constant formula and expressing the dynamic pressure in more detail yields

$$T_\alpha = 2W \left/ \left\{ g \rho V_M S_{\text{ref}} \left[2 + \frac{1.5 S_{\text{plan}} \alpha}{S_{\text{ref}}} \right] \right\} \right.$$

We can see that the turning rate time constant depends on altitude (or air density ρ), missile velocity, and angle of attack.

Consider the cylindrical missile of Chapter 17 which was 20 ft long, 1 ft in diameter, and weighed 1000 lb. Figure 19.4 shows that for a missile velocity of 3000 ft/s, the turning rate time constant increases with increasing altitude and decreasing angle of attack. At 50-kft altitude and 20-deg angle of attack the turning rate time constant is approximately 5 s.

We can display the turning rate time constant as a function of altitude for different missile velocities as is done in Fig. 19.5. We can see that the missile turning rate time constant increases with decreasing missile velocity. We have already seen in Chapter 6 that larger turning rate time constants exacerbate the radome slope stability problem. In summary, we can say that the radome stability problem will be greatest at the high-altitude, low-missile-velocity portion of the flight envelope.

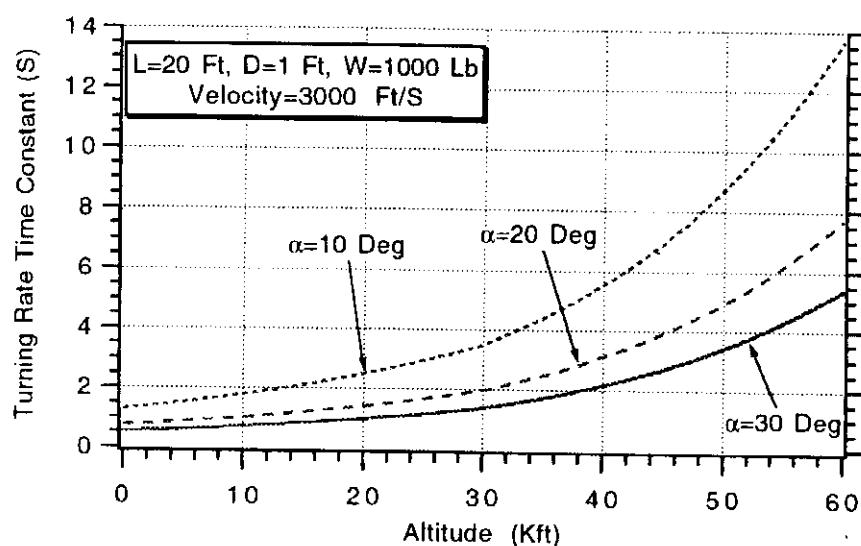


Fig. 19.4 Turning rate time constant increases with increasing altitude and decreasing angle of attack.

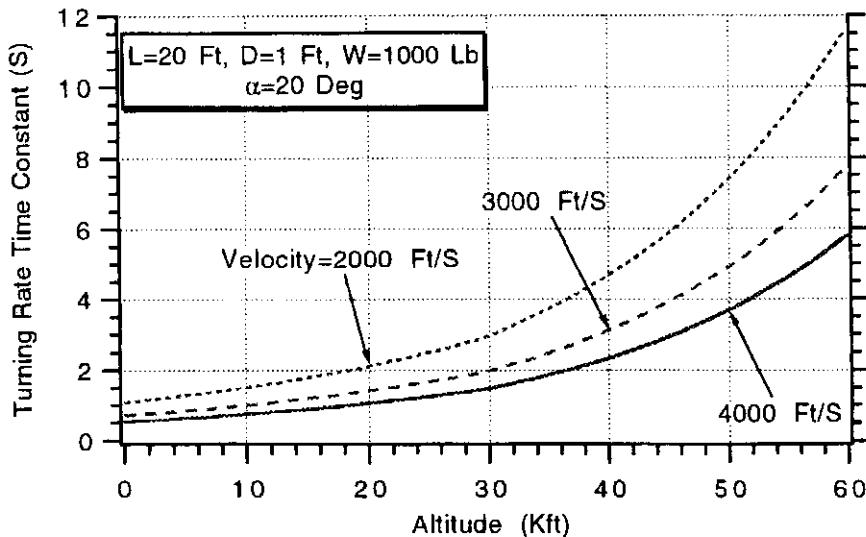


Fig. 19.5 Turning rate time constant increases with decreasing missile velocity.

Checking Minimum Guidance System Time Constant Constraints

To illustrate the increased stability problem caused by ballistic targets, let us compare missile acceleration profiles for both aircraft and ballistic targets at 50-kft altitude. We demonstrated in Chapter 17 that a cylindrical missile (i.e., $W = 1000$ lb, $L = 20$ ft, and $D = 1$ ft) has a 7-g capability for a speed of 3000 ft/s and an angle of attack of 20 deg. In addition, let us assume the radome slope for the interceptor is -0.01 . If we consider a head-on case in which an aircraft target is traveling at 1000 ft/s, the resultant closing velocity will be 4000 ft/s. In this chapter we showed that the minimum guidance system time constant permitted for a fifth-order binomial proportional navigation guidance system is

$$T_{\min} = \frac{N' V_c R T_a}{0.79 V_M}$$

From Fig. 19.4 we can see that the turning rate time constant for the cylindrical interceptor is 5 s at this flight condition. Therefore the minimum guidance system time constant at this flight condition becomes

$$T_{\min} = \frac{3 * 4000 * 0.01 * 5}{0.79 * 3000} = 0.25 \text{ s}$$

In other words, the interceptor guidance system time constant must be greater than 0.25 s when engaging this particular aircraft threat.

To test the preceding theoretical limit on the minimum allowable interceptor guidance system time constant against the aircraft threat, our multiple run simulation of a fifth-order binomial guidance system with radome effects (Listing 19.2) was modified to investigate single flights. Listing 19.2 shows that the resultant single flight simulation is set up to monitor the relative separation between missile and target y and the acceleration command n_c . The nominal inputs for the aircraft threat we are considering include the 7-g missile acceleration command limit, 4000-ft/s closing velocity, 5-s turning rate time constant, 3000-ft/s interceptor speed, -0.01 radome slope, and effective navigation ratio of 3. A 1-g target maneuver is considered as the only error source.

Listing 19.2 Simulation of bombing loop with radome effects

```

INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
VC=4000.
XNT=32.2
XNCLIMG=7.
YIC=0.
VM=3000.
HEDEG=0.
TAU=.3
XNP=3.
TA=5.
R=-.01
TF=10.
Y=YIC
YD=-VM*HEDEG/57.3
YDIC=YD
XNL=0.
ELAMDH=0.
X4=0.
X5=0.
TH=0.
THH=0.
T=0.
H=.01
S=0.
XNCLIM=XNCLIMG*32.2
10 IF(T>(TF-.0001))GOTO 999
YOLD=Y
YDOLD=YD
XNLOLD=XNL
ELAMDHOLD=ELAMDH
X4OLD=X4
X5OLD=X5
THOLD=TH
THHOLD=THH
STEP=1
GOTO 200
66 STEP=2
Y=Y+H*YD
YD=YD+H*YDD
XNL=XNL+H*XNLD
ELAMDH=ELAMDH+H*ELAMDHD
X4=X4+H*X4D
X5=X5+H*X5D
TH=TH+H*THD
THH=THH+H*THHD
T=T+H
GOTO 200
55 CONTINUE

```

(Contd.)

Listing 19.2 (Continued)

```

Y=.5*(YOLD+Y+H*YD)
YD=.5*(YDOLD+YD+H*YDD)
XNL=.5*(XNLOLD+XNL+H*XNLD)
ELAMDH=.5*(ELAMDHOLD+ELAMDH+H*ELAMDHD)
X4=.5*(X4OLD+X4+H*X4D)
X5=.5*(X5OLD+X5+H*X5D)
TH=.5*(THOLD+TH+H*THD)
THH=.5*(THHOLD+THH+H*THHD)
S=S+H
IF(S>.09999)THEN
  S=0.
  WRITE(*,97)T,Y,XNC/32.2
  WRITE(1,97)T,Y,XNC/32.2
ENDIF
GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
TGO=TF-T+.00001
XLAM=Y/(VC*TGO)
EPS=XLAM-TH-THH+R*THH
DD=5.*EPS/TAU
ELAMDHD=5.*(DD-ELAMDH)/TAU
XNC=XNP*VC*ELAMDH
IF(XNC>XNCLIM)XNC=XNCLIM
IF(XNC<-XNCLIM)XNC=-XNCLIM
X4D=5.*(XNC-X4)/TAU
X5D=5.*(X4-X5)/TAU
XNLD=5.*(X5-XNL)/TAU
THD=XNL/VM+TA*XNLD/VM
THHD=DD-THD
YDD=XNT-XNL
IF(STEP-1)66,66,55
999 CONTINUE
PAUSE
CLOSE(1)
END

```

The nominal case was run and the guidance system time constant was made a parameter. Figure 19.6 shows that when the guidance system time constant is 0.2 s (less than the minimum permissible time constant) the acceleration command oscillates between ± 7 g, indicating that the guidance system is indeed unstable. Increasing the guidance system time constant to 0.3 s (more than the minimum permissible time constant) stabilizes the acceleration command. The resultant acceleration profile is a monotonically increasing straight line, as would be expected for the response due to a step target maneuver. Thus the theoretical formula for the minimum guidance system time constant and the simulation results of Listing 19.2 appear to be in agreement.

If we fly an inverse trajectory with a 3000 ft/s interceptor against a 6000 ft/s ballistic target, the resultant closing velocity becomes 9000 ft/s. The minimum

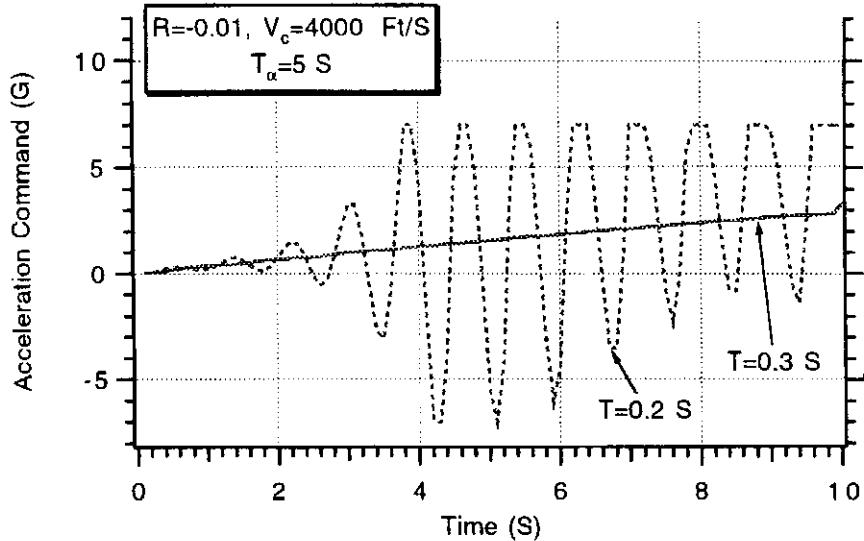


Fig. 19.6 Simulation and theory agree for aircraft threat.

guidance system time constant at this flight condition reduces to

$$T_{\min} = \frac{3 * 9000 * 0.01 * 5}{0.79 * 3000} = 0.57 \text{ s}$$

which indicates that the minimum permissible interceptor guidance system time constant against this particular ballistic threat must be greater than 0.57 s. In other words, for the case considered, the missile guidance system time constant must be much greater against a ballistic threat than it has to be against an aircraft threat because of the much higher closing velocities.

The simulation of Listing 19.2 was rerun for the case in which the closing velocity was 9000 ft/s and the guidance system time constant was again made a parameter. Figure 19.7 shows that when the guidance system time constant is 0.5 s (less than the minimum permissible time constant) the acceleration command oscillates between $\pm 7 \text{ g}$, indicating that the guidance system is unstable. When the guidance system time constant is increased to 0.7 s (more than the minimum

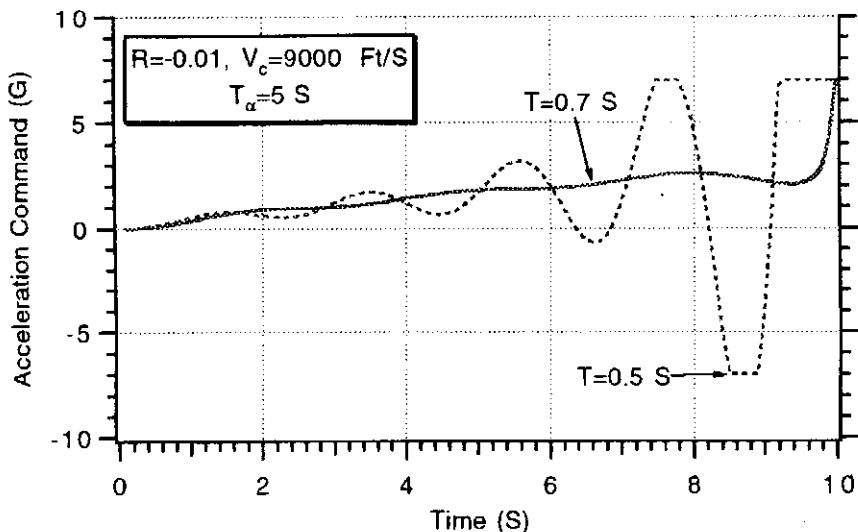


Fig. 19.7 Simulation and theory agree for ballistic threat.

permissible time constant), the acceleration command is stable and approximately monotonically increasing as would be expected for the response due to a step target acceleration. We can also see by comparing Figs. 19.6 and 19.7 that the frequency of oscillation depends on the closing velocity.

Miss Due to Noise for Aircraft and Ballistic Targets

In this chapter we have presented formulas for the miss distance due to various noise sources and have also presented stability requirements for the minimum guidance system time constant in a fifth-order binomial proportional navigation guidance system. In this section we shall illustrate, via a numerical example, how the miss distance due to noise increases when the threat changes from an aircraft to a ballistic target.

All of the formulas for the noise miss distances depended on the spectral density Φ of the noise. Often we talk about noise with a standard deviation σ entering a guidance system every T_s seconds. A useful approximation relating the noise spectral density to the standard deviation is given by

$$\Phi = \sigma^2 T_s$$

where Φ is measured in units²/Hz, σ is measured in units, and T_s is measured in seconds. This relationship is identical to the one we used for simulating white noise in Chapter 4 with the sampling time T_s being replaced by the integration interval.

Let us consider an example in which range independent noise with standard deviation 0.001 rad enters a proportional navigation guidance system ($N' = 3$) every 0.01 s.^{4,8} The miss distance formula for an effective navigation ratio of 3 expressed in terms of the noise standard deviation can be found from Table 19.1 and is given by

$$\sigma_{\text{Miss}} \left| \begin{array}{l} \text{Range} \\ \text{Independent} \\ \text{Noise} \end{array} \right. = 3.04 V_c T^{0.5} \sigma_{FN} T_s^{0.5}$$

For the aircraft threat with a closing velocity of 4000 ft/s and a minimum guidance time constant of 0.25 s, the minimum miss due to range independent noise is

$$\sigma_{\text{Miss}} \left| \begin{array}{l} \text{Range} \\ \text{Independent} \\ \text{Noise} \end{array} \right. = 3.04 * 4000 * 0.25^{0.5} * 0.001 * 0.01^{0.5} = 0.6 \text{ ft}$$

whereas for the ballistic threat with closing velocity of 9000 ft/s and a minimum guidance time constant of 0.57 s, the minimum miss is

$$\sigma_{\text{Miss}} \left| \begin{array}{l} \text{Range} \\ \text{Independent} \\ \text{Noise} \end{array} \right. = 3.04 * 9000 * 0.57^{0.5} * 0.001 * 0.01^{0.5} = 2.1 \text{ ft}$$

Although the noise miss is more than three times larger in the ballistic target case than it was in the aircraft threat case, the miss due to range independent noise is negligible.

Let us now consider the influence of semiactive range dependent noise on both targets. The standard deviation of the miss due to semiactive range dependent noise for an effective navigation ratio of three can be found from Table 19.1 as

$$\sigma_{\text{Miss}} \left| \begin{array}{l} \\ \\ \text{Semiactive} \\ \text{Range} \\ \text{Dependent} \\ \text{Noise} \end{array} \right. = \frac{9.47 V_c^2 T^{1.5} \sigma_{RN} T_s^{0.5}}{R_A}$$

We notice from the preceding formula that the dependence on closing velocity and guidance system time constant is more significant than it was in the range dependent noise case. Let us now consider a case in which there is 0.02 rad of semiactive noise at 30,000 ft entering the guidance system every 0.01 s.^{4,7} For the aircraft threat with a closing velocity of 4000 ft/s and a minimum guidance time constant of 0.25 s, the minimum miss due to semiactive range dependent noise is

$$\sigma_{\text{Miss}} \left| \begin{array}{l} \\ \\ \text{Semiactive} \\ \text{Range} \\ \text{Dependent} \\ \text{Noise} \end{array} \right. = \frac{9.47 * 4000^2 * 0.25^{1.5} * 0.02 * 0.01^{0.5}}{30,000} = 1.3 \text{ ft}$$

whereas for the ballistic threat with closing velocity of 9000 ft/s and a minimum guidance time constant of 0.57 s, the minimum miss is

$$\sigma_{\text{Miss}} \left| \begin{array}{l} \\ \\ \text{Semiactive} \\ \text{Range} \\ \text{Dependent} \\ \text{Noise} \end{array} \right. = \frac{9.47 * 9000^2 * 0.57^{1.5} * 0.02 * 0.01^{0.5}}{30,000} = 22 \text{ ft}$$

We can see that the miss due to semiactive noise is very large against ballistic targets because of the strong dependence on closing velocity and guidance system time constant. For this example the ballistic threat miss was nearly 20 times greater than the aircraft threat miss!

Finally, let us now consider the influence of active range dependent noise on both targets. The standard deviation of the miss due to active range dependent noise for an effective navigation ratio of three can be found from Table 19.1 as

$$\sigma_{\text{Miss}} \left| \begin{array}{l} \\ \\ \text{Active} \\ \text{Range} \\ \text{Dependent} \\ \text{Noise} \end{array} \right. = \frac{41.3 V_c^3 T^{2.5} \sigma_{RNA} T_s^{0.5}}{R_A^2}$$

We notice from the preceding formula that the dependence on closing velocity and guidance system time constant is even more significant than it was in the semiactive noise case. Let us now consider a case in which there is 0.02 rad of active noise at 30,000 ft entering the guidance system every 0.01 s.^{4,7} For the aircraft threat with a closing velocity of 4000 ft/s and a minimum guidance time constant of 0.25 s, the minimum miss due to active range dependent noise is

$$\sigma_{\text{Miss}} = \frac{41.3 * 4000^3 * 0.25^{2.5} * 0.02 * 0.01^{0.5}}{30,000^2} = 0.18 \text{ ft}$$

Active
Range
Dependent
Noise

whereas for the ballistic threat with closing velocity of 9000 ft/s and a minimum guidance time constant of 0.57 s, the minimum miss is

$$\sigma_{\text{Miss}} = \frac{41.3 * 9000^3 * 0.57^{2.5} * 0.02 * 0.01^{0.5}}{30,000^2} = 16.4 \text{ ft}$$

Active
Range
Dependent
Noise

Although the active noise miss is less than the semiactive noise miss it is nearly 100 times larger for the ballistic threat than for the aircraft threat because the miss depends on the cube of closing velocity and the 2.5 power of the guidance system time constant.

Summary

We have shown that from a noise induced miss distance point of view, ballistic targets are more challenging than the aircraft threat. The high-closing velocity of the ballistic target engagement significantly increases the minimum guidance system time constant required for radome slope stability. The large guidance system time constant and high-closing velocity can make ballistic target noise induced miss distances more than an order of magnitude greater than the miss distances experienced against an aircraft threat. To achieve small miss distances against high-closing velocity ballistic targets, methods for reducing the noise and effective radome slope must be found.

References

- ¹Travers, P., "Interceptor Dynamics," unpublished lecture notes, Raytheon, circa 1971.
- ²Garnell, P., and East, D. J., *Guided Weapon Control System*, Pergamon, Oxford, 1977.
- ³Macfadzean, R. H. M., *Surface-Based Air Defense System Analysis*, Artech House, Norwood, MA, 1992.
- ⁴Lin, C. F., *Modern Navigation, Guidance, and Control Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1991.

⁵Howe, R. M., "Guidance," *System Engineering Handbook*, edited by R. E. Machol, W. P. Tanner Jr., and S. N. Alexander, McGraw-Hill, New York, 1965, Chap. 19.

⁶Nesline, F. W., and Zarchan, P., "Radome Induced Miss Distance in Aerodynamically Controlled Homing Missiles," *Proceedings of AIAA Guidance and Control Conference*, AIAA, New York, Aug. 1984.

⁷Jerger, J. J., *System Preliminary Design*, Van Nostrand, Princeton, NJ, 1960.

⁸Nesline., F. W., and Zarchan, P., "Miss Distance Dynamics in Homing Missiles," *Proceedings of AIAA Guidance and Control Conference*, AIAA, New York, Aug. 1984.

Multiple Targets

Introduction and Background

IN both tactical and strategic engagements there may be an instantaneous apparent shift in the target location that can cause unacceptable miss distances. For example, for tactical intercepts, the interceptor may be guiding on the power centroid of two aircraft flying in close formation. When one of the aircraft falls outside the missile seeker beam, the other aircraft will be resolved. In this case it appears to the pursuing interceptor that the target has instantaneously shifted from the location of the power centroid to the location of the resolved aircraft. In other words, there has been an apparent step change in target position.¹ Similarly, a strategic exoatmospheric missile may be homing on one of two closely spaced objects. After a while discrimination takes place and the interceptor's software may conclude that one object is a decoy while the second object is the real target. In this case too, as far as the missile is concerned, it appears as if the target has instantaneously changed position (from first object or decoy to second object).

In both preceding examples the target displacement disturbance occurs late in the flight, which is the worst possible time from a missile guidance system point of view. Large miss distances may result because of insufficient remaining homing time. In this chapter we will develop normalized design curves for both a single time constant and fifth-order binomial proportional navigation guidance system to both illustrate and quantify the multiple target problem. Rules of thumb will be developed relating the necessary ratio of the time left for homing after resolution has taken place to the guidance system time constant and the miss due to the apparent shift in target location.

Development of a Linear Model

To develop an appropriate linear model for the multiple target phenomenon, let us first think of two aircraft flying in close formation being pursued by an interceptor as shown in Fig. 20.1. The first aircraft is at 1200 ft altitude while the second aircraft is at 800 ft altitude. For simplicity we will assume that the power centroid is located halfway between both aircraft at 1000 ft altitude. In this example both aircraft and the power centroid are traveling at 1000 ft/s. The pursuing missile is at 1000 ft altitude and is moving at 3000 ft/s toward the power centroid. At a certain time to go before intercept with the power centroid, the missile realizes that the true target is aircraft 1 and not the power centroid. At that time it will appear as if the target has instantaneously changed position (from power centroid to aircraft 1). From a simulation point of view, we only have to model the target

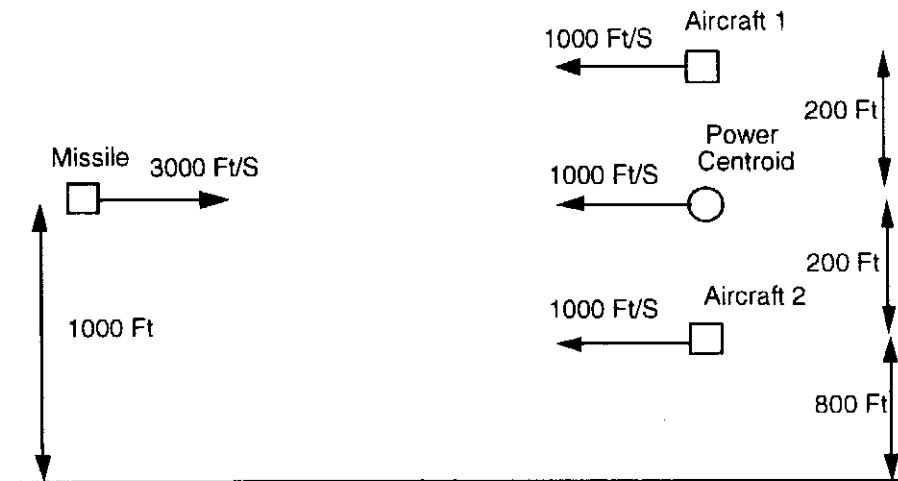


Fig. 20.1 Missile engaging two aircraft flying in formation.

the missile sees. Therefore for most of the flight the missile will be guiding on the power centroid and for the rest of the flight the missile will be guiding on aircraft 1. This can easily be modeled as if the target is taking a step displacement at a certain time to go before intercept.

A two-dimensional nonlinear missile-target engagement simulation, based on Listing 2.1 and Fig. 20.1, was developed for this problem and appears in Listing 20.1. For simplicity this simulation neglects both gravity and drag. Because the missile is on a collision triangle with the power centroid and there is no target maneuver or heading error, no acceleration commands will be required for most of the engagement. At one second before intercept (i.e., THOM = 1) the target jumps 200 ft instantaneously from the power centroid to aircraft 1 (i.e., see logic after statement label 10) and the missile responds with acceleration commands based on proportional navigation guidance (i.e., in this example using a navigation ratio of 3). We can see from Listing 20.1 that in this case there are no dynamics in the guidance system.

Figure 20.2 shows the alteration of the missile and target trajectories after the missile makes the guidance switch from the power centroid to aircraft 1. Here

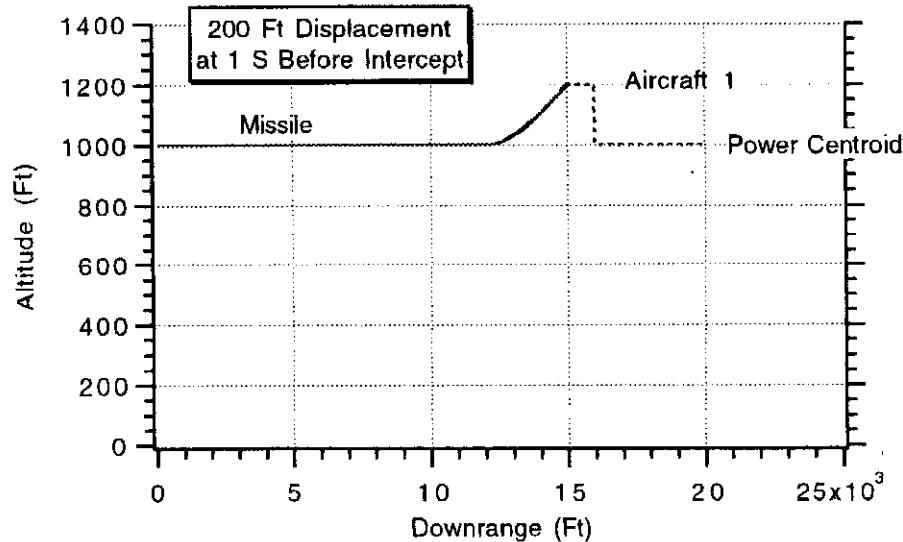


Fig. 20.2 Missile first guides on power centroid and then guides on aircraft 1.

**Listing 20.1 Nonlinear engagement simulation
with step in target displacement**

```

LOGICAL QSWITCH
INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
XNP=3.
DISPLACE=200.
THOM=1.
VM=3000.
VT=1000.
RM1=0.
RM2=1000.
RT1=20000.
RT2=1000.
QSWITCH=.FALSE.
VT1=-VT
VT2=0.
T=0.
S=0.
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
XLAM=ATAN2(RTM2,RTM1)
VM1=VM
VM2=0.
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
TGO=RTM/VC
H=.005
10 IF(VC<0.)GOTO 999
TGO=RTM/VC
IF(TGO<.3)H=.00005
IF(TGO<=THOM.AND..NOT.QSWITCH)THEN
    QSWITCH=.TRUE.
    RT2=RT2+DISPLACE
ENDIF
RT1OLD=RT1
RT2OLD=RT2
RM1OLD=RM1
RM2OLD=RM2
VM1OLD=VM1
VM2OLD=VM2
STEP=1
GOTO 200
66 STEP=2
RT1=RT1+H*VT1
RT2=RT2+H*VT2
RM1=RM1+H*VM1
RM2=RM2+H*VM2
VM1=VM1+H*AM1

```

Listing 20.1 (Continued)

```

VM2=VM2+H*AM2
T=T+H
GOTO 200
55  CONTINUE
RT1=.5*(RT1OLD+RT1+H*VT1)
RT2=.5*(RT2OLD+RT2+H*VT2)
RM1=.5*(RM1OLD+RM1+H*VM1)
RM2=.5*(RM2OLD+RM2+H*VM2)
VM1=.5*(VM1OLD+VM1+H*AM1)
VM2=.5*(VM2OLD+VM2+H*AM2)
S=S+H
IF(S<.049999)GOTO 10
S=0
WRITE(*,97)T,RM1,RM2,RT1,RT2,XNC/32.2
WRITE(1,97)T,RM1,RM2,RT1,RT2,XNC/32.2
GOTO 10
97  FORMAT(6F10.3)
200  CONTINUE
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
XLAM=ATAN2(RTM2,RTM1)
XLAMD=(RTM1*VTM2-RTM2*VTM1)/(RTM*RTM)
XNC=XNP*VC*XLAMD
AM1=-XNC*SIN(XLAM)
AM2=XNC*COS(XLAM)
IF(STEP-1)66,66,55
999  CONTINUE
WRITE(*,*)T,RTM
PAUSE
CLOSE(1)
END

```

we can see that the apparent step change in target position at one second before intercept causes the missile to respond immediately. At the end of the flight, the missile hits aircraft 1. The successful intercept should not be surprising since we can see from Listing 20.1 there was no limit on the amount of acceleration that could be used nor were there any dynamics in the guidance system to cause miss distance.

Figure 20.3 shows that the price paid for the rapid missile trajectory change is that nearly 20 g of missile acceleration is required to make the intercept successful. Apparent step target displacements occurring later in the flight (i.e., time to go before intercept is smaller) will cause even more acceleration while apparent step displacements occurring earlier in the flight will place smaller acceleration demands on the interceptor.

We now want to see if we can use our linear analysis tools to understand how an apparent step in target displacement influences guidance system performance.

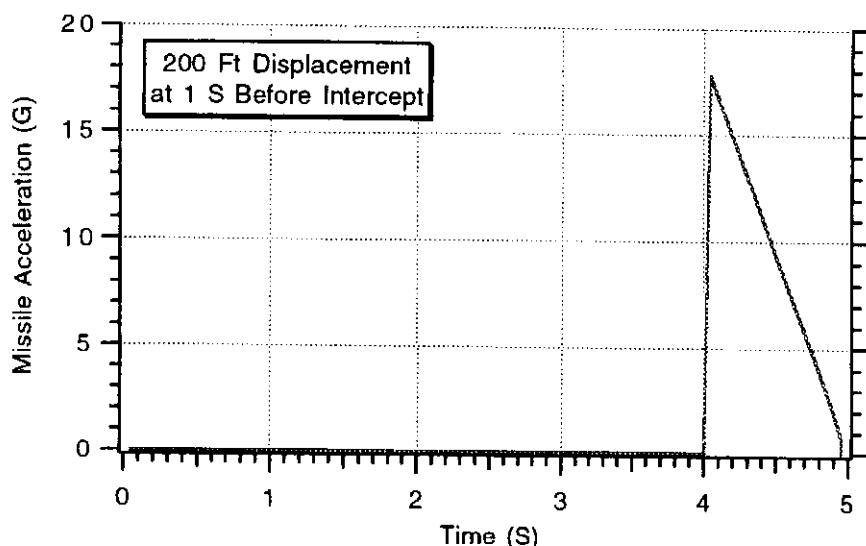


Fig. 20.3 Nearly 20 g of acceleration is required to take out 200 ft of target displacement.

First we must determine if the linearization techniques we used in Chapter 2 are valid for approximating a step in target displacement. Figure 20.4 redraws the simplest possible linearized proportional navigation homing loop of Chapter 2. In this perfect guidance system in which the geometry is linearized, models of the seeker, noise filter, guidance, and flight control systems have been considered to be perfect and without dynamics. We are modeling the shift from the power centroid to aircraft 1 as a step in target displacement y_{TIC} . As can be seen from Fig. 20.4, the step target displacement can be modeled as an initial condition on the integrator with output y since $y = y_T - y_M$ and there is no initial y_M .

Following a procedure similar to that of Chapter 2, we can solve the linear time-varying differential equation associated with Fig. 20.4 (Ref. 2). From Fig. 20.4 we can see that the relative acceleration (zero target acceleration minus missile

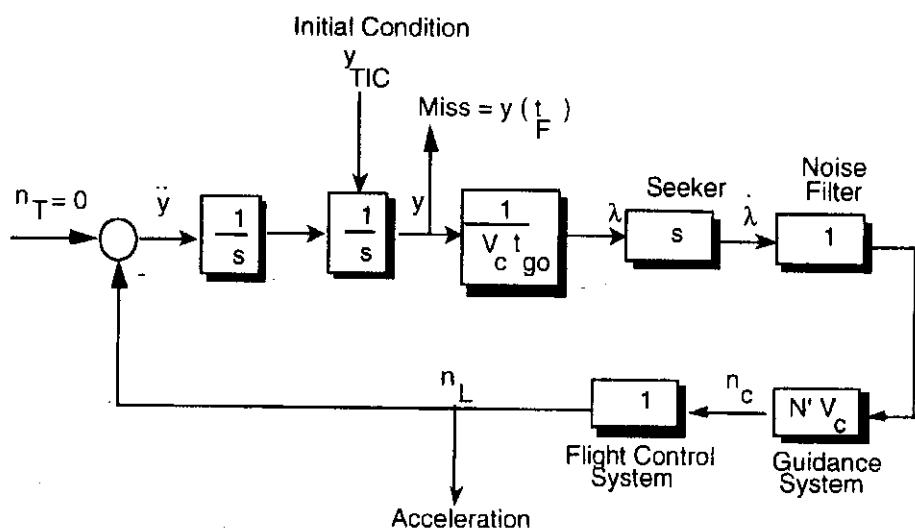


Fig. 20.4 Linear proportional navigation guidance homing loop with step in target displacement.

acceleration) can be expressed as

$$\ddot{y} = 0 - n_c = -N' V_c \dot{\lambda}$$

Integrating the preceding differential equation once yields

$$\dot{y} = -N' V_c \lambda + C_1$$

where C_1 is the constant of integration. Substitution of the line-of-sight angle definition from Fig. 20.4 into the preceding expression yields the following linear time-varying first-order differential equation

$$\frac{dy}{dt} + \frac{N' y}{V_c(t_F - t)} = C_1$$

We have seen in Chapter 2 that a linear first-order differential equation of the form

$$\frac{dy}{dt} + a(t)y = h(t)$$

has the solution

$$y = e^{-\int_0^t a(T)dT} \int_0^t h(n) e^{\int_n^t a(T)dT} dn + C_2 e^{-\int_0^t a(T)dT}$$

Therefore we now have enough information to solve the relative trajectory differential equation exactly. As was mentioned previously, an instantaneous step in target displacement means that the initial condition on the first state y is the value of the displacement or

$$y(0) = y_{TIC}$$

Under these circumstances, after much algebra, we can solve for the closed-form solutions for the relative separation between missile and target y and the missile acceleration n_c due to a step in target displacement. These closed-form solutions are given by

$$y = \frac{[N'(1 - t/t_F) - (1 - t/t_F)^{N'}]y_{TIC}}{N' - 1}$$

and

$$n_c = \frac{N'(1 - t/t_F)^{N'-2} y_{TIC}}{t_F^2}$$

where t is time and t_F is the amount of homing time remaining after the apparent step in target displacement has taken place. From the relative separation expression we can see that the miss distance $y(t_F)$ is always zero! This should not be surprising because we have a zero-lag guidance system and the miss distance for such a system should always be zero provided that the missile has sufficient acceleration capability. In addition, from the acceleration formula we observe that the magnitude of the initial missile acceleration is largest at the beginning and is proportional to the size of the target displacement and inversely proportional to the square of the homing or flight time. Doubling the apparent target displacement will double the initial missile acceleration, whereas doubling the flight time or time remaining after the apparent target displacement occurred will quarter the initial missile

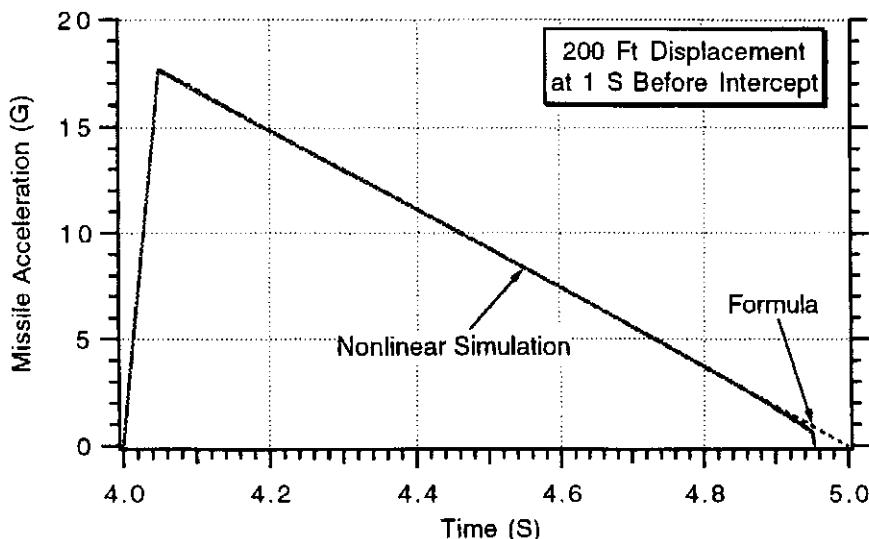


Fig. 20.5 Linear model yields accurate performance projections.

acceleration. Of course if the acceleration required by the preceding formula is not available there may be a significant miss distance.

If we overlay the preceding acceleration formula based on a linear model (i.e., $Y_{TIC} = 200$ ft, $N' = 3$, $t_F = 1$ s) with the nonlinear acceleration results of Fig. 20.3, we obtain the overlays of Fig. 20.5. Here we can see that the closed-form solution for the missile acceleration based on the linearized homing loop model approximates the nonlinear results very well. This means that from now on we can use the linear guidance system model to make performance projections when the system disturbance is an apparent step in target displacement.

The closed-form solutions for the missile acceleration response due to a step in target displacement for various effective navigation ratios are displayed in normalized form in Fig. 20.6. We can see that higher effective navigation ratios increase the missile acceleration requirements at the beginning of flight (when apparent step in target displacement occurs) and reduce the acceleration requirements towards the end of the flight. Figure 20.6 indicates that the missile acceleration is always monotonically decreasing as the flight progresses (i.e., more acceleration is needed

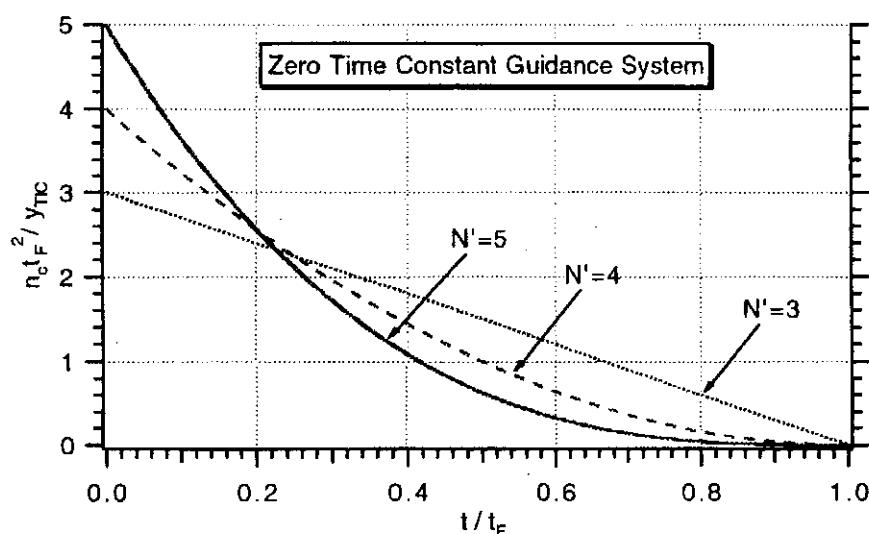


Fig. 20.6 Normalized missile acceleration due to step in target displacement.

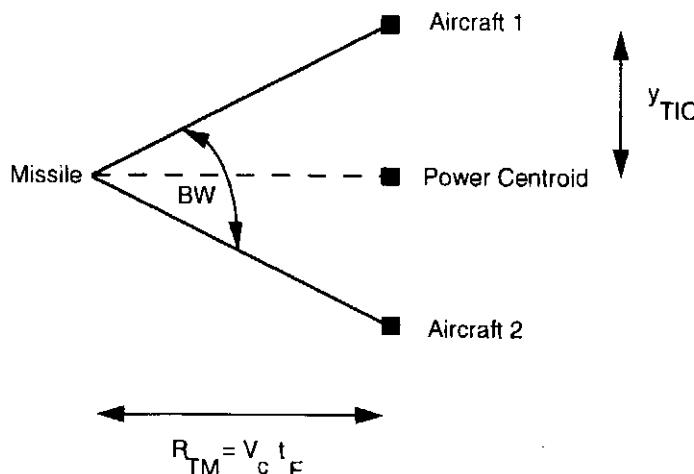


Fig. 20.7 Multiple target geometry.

at the beginning of the flight than at the end of the flight). From a system sizing point of view, the designer usually wants to ensure that the acceleration capability of the missile is adequate at the beginning of flight so that saturation can be avoided. For a fixed missile acceleration capability, Fig. 20.6 shows how requirements are placed on the minimum guidance or flight time required after final resolution (or the time remaining after the apparent step in target displacement occurs) and maximum allowable target displacement.

We can illustrate the use of the normalized acceleration curves of Fig. 20.6 and show why these curves are meaningful for the multiple target problem. Consider Fig. 20.7 in which two aircraft flying in formation are being pursued by a missile. Initially both aircraft are close enough so that the missile with seeker beamwidth BW homes on the power centroid of the two aircraft. At the point where one of the aircraft falls outside the seeker beam, resolution takes place and it appears to the missile that the aircraft has been instantaneously displaced a distance y_{TIC} (i.e., from power centroid to aircraft 1). If the missile and aircraft are traveling at constant speed with closing velocity V_c , the missile will be a distance of $V_c t_F$ from the power centroid at the point of resolution. In this example t_F is the time remaining for guidance after seeker resolution.

From trigonometry we can see that the seeker beamwidth is related to the aircraft displacement according to

$$\tan \frac{BW}{2} = \frac{y_{TIC}}{V_c t_F}$$

Using the small angle approximation and solving for the effective time remaining for guidance after seeker resolution, we get

$$t_F = \frac{2y_{TIC}}{V_c BW}$$

If a seeker has a beamwidth of 0.1 rad (nearly 6 deg), and the two aircraft are separated by 400 ft and the closing velocity is 4000 ft/s, the time remaining for guidance after resolution will be 1 s or

$$t_F = \frac{2y_{TIC}}{V_c BW} = \frac{400}{4000 * 0.1} = 1 \text{ s}$$

Assuming that the missile effective navigation ratio is 3, we can see from either the formula for missile acceleration or the normalized acceleration curves of Fig. 20.6 that the maximum acceleration occurs at the time of resolution and is given by

$$n_{c\text{MAX}} = \frac{3y_{\text{TIC}}}{t_F^2} = \frac{3 * 200}{1^2} = 600 \text{ ft/s}^2 = 18.6 \text{ g}$$

which is in agreement with the maximum acceleration indicated by the nonlinear simulation results of Fig. 20.3. In summary, this means that the missile will require nearly 20 g of acceleration to hit the resolved target when the aircraft formation spacing is 400 ft, the missile seeker beamwidth is approximately 6 deg, and the closing velocity is 4000 ft/s.

Single Time Constant Guidance System

We have observed from the closed-form solutions for a zero-lag guidance system that if the missile has sufficient acceleration capability the apparent shift in target location will not cause any miss distance. Guidance system dynamics will of course contribute to the miss distance. If all of the guidance system dynamics can be lumped together as a single time constant T as was done in Chapter 3, then the original guidance system block diagram of Fig. 20.2 is modified and is redrawn in Fig. 20.8.

In this interceptor guidance system model, the single time constant or single lag represents the combined dynamics of the seeker, noise filter, and flight-control system. Closed-form solutions for the miss distance due to a step in target displacement can be derived for a single time constant guidance system using the adjoint technique described in Chapter 3. The adjoint of the single time constant guidance system used in deriving the closed-form solutions based on Fig. 20.8 appears in Fig. 20.9.

In Fig. 20.9 we can see that the miss due to a step in target displacement is simply the second derivative of the miss due to target maneuver (i.e., proportional to x_3 in Fig. 20.9). In Chapter 3 we showed that the miss due to target maneuver

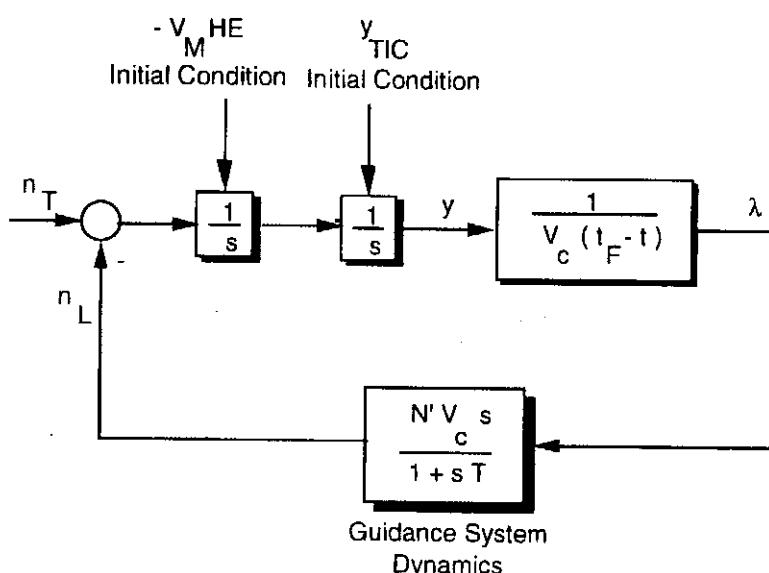


Fig. 20.8 Single-lag guidance system.

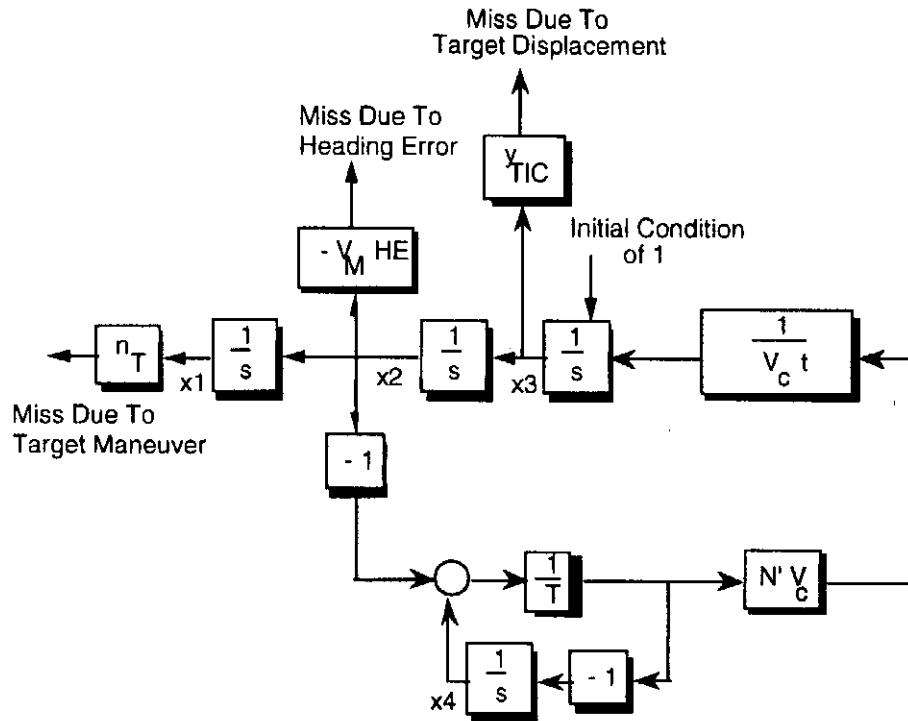


Fig. 20.9 Adjoint of single time constant guidance system with step in target displacement disturbance.

in a single time constant guidance system for various effective navigation ratios was given by

$$\frac{\text{Miss}}{n_T} \Big|_{N'=3} = 0.5 t_F^2 e^{-t_F/T}$$

$$\frac{\text{Miss}}{n_T} \Big|_{N'=4} = t_F^2 e^{-t_F/T} \left(0.5 - \frac{t_F}{6T} \right)$$

$$\frac{\text{Miss}}{n_T} \Big|_{N'=5} = t_F^2 e^{-t_F/T} \left(0.5 - \frac{t_F}{3T} + \frac{t_F^2}{24T^2} \right)$$

Therefore if we take the derivative twice with respect to t_F in the preceding three equations we get formulas for the miss due to a step in target displacement for the different effective navigation ratios or^{1,2}

$$\frac{\text{Miss}}{y_{\text{TIC}}} \Big|_{N'=3} = e^{-x} (1 - 2x + 0.5x^2)$$

$$\frac{\text{Miss}}{y_{\text{TIC}}} \Big|_{N'=4} = e^{-x} \left(1 - 3x + 1.5x^2 - \frac{x^3}{6} \right)$$

$$\frac{\text{Miss}}{y_{\text{TIC}}} \Big|_{N'=5} = e^{-x} \left(1 - 4x + 3x^2 - \frac{2x^3}{3} + \frac{x^4}{24} \right)$$

where

$$x = \frac{t_F}{T}$$

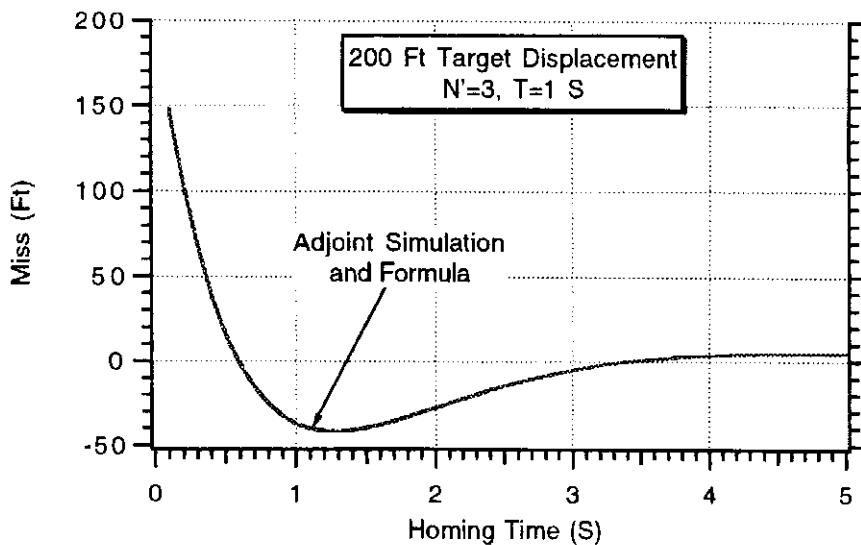


Fig. 20.10 Formula and adjoint simulation results agree.

We have seen before that the ratio of the flight time to the guidance system time constant x is also referred to as the number of guidance time constants.

To check the closed-form solutions, an adjoint simulation of a single time constant guidance was written based on Fig. 20.9 and appears in Listing 20.2. The simulation is identical to Listing 3.1 of Chapter 3 except that an additional output (i.e., called XMY in simulation) has been added in order to calculate the miss due to a step in target displacement. In addition, the formula for the miss due to a step in target displacement when the effective navigation ratio is three (THEORY) has also been added to the output calculations so that theoretical and simulation results can be compared.

We can see from Fig. 20.10 that the closed-form solution agrees with the adjoint simulation output indicating that the formula is correct.

Listing 20.3 is the nonlinear simulation of Listing 20.1 except this time there is a single time constant representation of the guidance system. If the target is above the missile at intercept (RTM2 positive), then the miss is considered to be positive whereas if the missile is above the target at intercept (RTM2 negative) the miss is considered to be negative. In addition, rather than running the simulation in the single flight mode, multiple cases are run—each one having a different time to go at which the step in target displacement occurs. Since Listing 20.3 is the brute force and nonlinear approach to the problem, the running time is considerably longer than that of the adjoint simulation.

Figure 20.11 shows that the brute force nonlinear simulation miss distance projections for the single time constant guidance system are virtually identical to those answers predicted by the miss distance formula. Therefore from now on it will be safe to use methods of linear analysis to address the step in target displacement error source.

The various closed-form solutions for the miss distance due to a step in target displacement for a single time constant proportional navigation guidance system for different effective navigation ratios are displayed in normalized form in Fig. 20.12 (Ref. 2). Superimposed on the normalized curves are the location of the power centroid and the real target (i.e., aircraft 1). A normalized miss of unity means that the missile hits the power centroid. A normalized miss of zero means that the

**Listing 20.2 Adjoint simulation of single time constant
guidance system with step in target displacement**

```

INTEGER STEP
XNP=3.
TAU=1.
TF=5.
DISP=200.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
T=0.
S=0.
TP=T+.00001
X1=0
X2=0
X3=1
X4=0
H=.01
10 IF(TP>(TF-.00001))GOTO 999
S=S+H
X1OLD=X1
X2OLD=X2
X3OLD=X3
X4OLD=X4
STEP=1
GOTO 200
66 STEP=2
X1=X1+H*X1D
X2=X2+H*X2D
X3=X3+H*X3D
X4=X4+H*X4D
TP=TP+H
GOTO 200
55 CONTINUE
X1=(X1OLD+X1)/2+.5*H*X1D
X2=(X2OLD+X2)/2+.5*H*X2D
X3=(X3OLD+X3)/2+.5*H*X3D
X4=(X4OLD+X4)/2+.5*H*X4D
IF(S<.09999)GOTO 10
S=0.
XMY=DISP*X3
X=TGO/TAU
THEORY=DISP*EXP(-X)*(1.-2.*X+.5*X*X)
WRITE(*,97)TP,XMY,THEORY
WRITE(1,97)TP,XMY,THEORY
GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
X1D=X2
X2D=X3
Y1=(X4-X2)/TAU
TGO=TP+.00001

```

(Contd.)

Listing 20.2 (Continued)

```

X3D=XNP*Y1/TGO
X4D=-Y1
IF(STEP-1)66,66,55
999  CONTINUE
PAUSE
CLOSE(1)
END

```

missile hits aircraft 1. A negative miss means that the missile overshoots aircraft 1. We can see that if the ratio of the flight time to the guidance system time constant is greater than five then the miss distance is virtually zero. Therefore we can say that for a single time constant guidance system the miss is effectively zero after five guidance time constants.

If we consider the same example of the previous section where there was no miss distance, the importance of guidance system dynamics and the normalized curves of Fig. 20.12 can be demonstrated. For a 200-ft target displacement (equivalent to 400-ft aircraft separation) 1 s of effective flight time and an overall guidance system time constant of 0.5 s ($T = 0.5$), the number of guidance time constants is 2 ($t_F/T = 1/0.5 = 2$) and the corresponding miss distance for different effective navigation ratios can be computed from Fig. 20.12 as

$$\text{Miss}_{N'=3} = -0.14 * 200 = -28 \text{ ft}$$

$$\text{Miss}_{N'=4} = -0.05 * 200 = -10 \text{ ft}$$

$$\text{Miss}_{N'=5} = 0.05 * 200 = 10 \text{ ft}$$

Thus we can see that there can be significant miss distance due to an apparent step in target displacement when system dynamics are considered. In this example a positive miss distance means that the missile is between aircraft 1 and the power

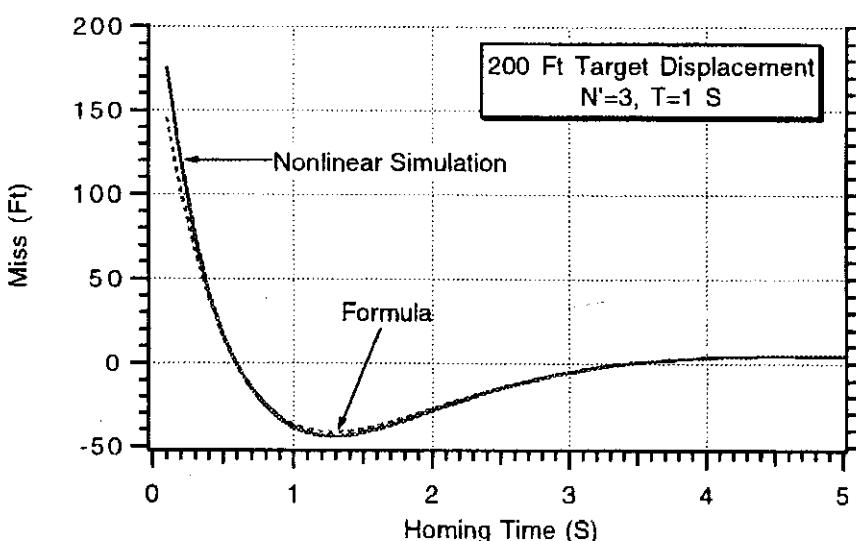


Fig. 20.11 Linear and nonlinear performance projections are identical in single time constant guidance system.

**Listing 20.3 Multiple run nonlinear single time constant
guidance system engagement simulation with step
in target displacement**

```

LOGICAL QSWITCH
XNP=3.
TAU=1.
DISPLACE=200.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
DO 22 THOM=.1,5,.1
VM=3000.
VT=1000.
RM1=0.
RM2=0.
RT1=20000.
RT2=0.
QSWITCH=.FALSE.
TV1=-VT
VT2=0
T=0.
S=0.
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
XLAM=ATAN2(RTM2,RTM1)
VM1=VM
VM2=0.
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
TGO=RTM/VC
XLAMH=0
H=.005
10 IF(VC<0.)GOTO 999
TGO=RTM/VC
IF(TGO<.3)H=.00005
IF(TGO<=THOM.AND..NOT.QSWITCH)THEN
    QSWITCH=.TRUE.
    RT2=RT2+DISPLACE
ENDIF
RT1OLD=RT1
RT2OLD=RT2
RM1OLD=RM1
RM2OLD=RM2
VM1OLD=VM1
VM2OLD=VM2
XLAMHOLD=XLAMH
STEP=1
GOTO 200

```

(Contd.)

Listing 20.3 (Continued)

```

66  STEP=2
    RT1=RT1+H*VT1
    RT2=RT2+H*VT2
    RM1=RM1+H*VM1
    RM2=RM2+H*VM2
    VM1=VM1+H*AM1
    VM2=VM2+H*AM2
    XLAMH=XLAMH+H*XLAMHD
    T=T+H
    GOTO 200
55  CONTINUE
    RT1=.5*(RT1OLD+RT1+H*VT1)
    RT2=.5*(RT2OLD+RT2+H*VT2)
    RM1=.5*(RM1OLD+RM1+H*VM1)
    RM2=.5*(RM2OLD+RM2+H*VM2)
    VM1=.5*(VM1OLD+VM1+H*AM1)
    VM2=.5*(VM2OLD+VM2+H*AM2)
    XLAMH=.5*(XLAMHOLD+XLAMH+H*XLAMHD)
    S=S+H
    GOTO 10
200 CONTINUE
    RTM1=RT1-RM1
    RTM2=RT2-RM2
    RTM=SQRT(RTM1**2+RTM2**2)
    VTM1=VT1-VM1
    VTM2=VT2-VM2
    VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
    XLAM=ATAN2(RTM2,RTM1)
    XLAMHD=(XLAM-XLAMH)/TAU
    XNC=XNP*VC*XLAMHD
    AM1=-XNC*SIN(XLAM)
    AM2=XNC*COS(XLAM)
    IF(STEP-1)66,66,55
999  CONTINUE
    X=THOM/TAU
    THEORY=DISPLACE*EXP(-X)*(1.-2.*X+.5*X*X)
    IF(RTM2>0.)THEN
        RTMP=RTM
    ELSE
        RTMP=-RTM
    ENDIF
    WRITE(*,97)THOM,RTMP,THEORY
    WRITE(1,97)THOM,RTMP,THEORY
22   CONTINUE
97   FORMAT(3F10.3)
    PAUSE
    CLOSE(1)
    END

```

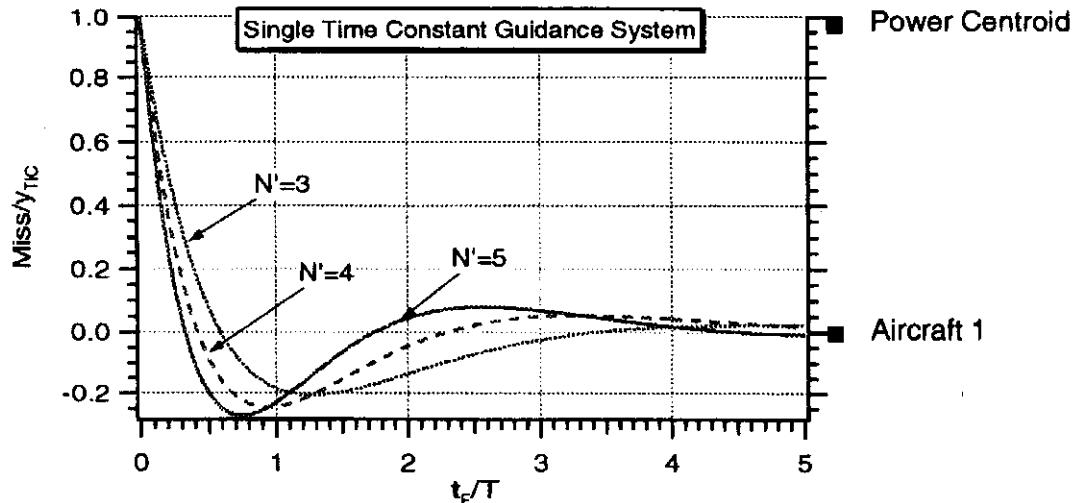


Fig. 20.12 First-order normalized miss for a step in target displacement.

centroid while a negative miss distance means the missile is below aircraft 1 in Fig. 20.12. Therefore, if the effective navigation ratio is 3, -28 ft of miss means that the missile missed aircraft 1 by 28 ft and aircraft 2 by 428 ft ($400 + 28 = 428$). We can see from Fig. 20.12 that if the missile time constant can be halved or if the seeker beamwidth can be halved then the number of guidance time constants is doubled to 4 ($t_F/T = 1/0.25 = 4$) and the miss will be reduced. The formulas and the curves indicate that large ratios of flight time to guidance system time constant yield small or near zero miss distances whereas small ratios of flight time to guidance system time constant can yield large miss distances.

Higher-Order Guidance System Dynamics

It has already been demonstrated that guidance system dynamics contributes to the miss distance. We chose a single time constant representation of the guidance system because closed-form solutions for the miss distance could be derived. The single time constant approximation to a missile guidance system is useful because the resultant closed-form solutions suggest normalization factors for the miss distance. We already know from Chapter 6 that the single time constant representation of the guidance system also seriously underestimates the miss distance. It was shown in Chapter 6 that a much better and more convenient representation of a missile guidance system transfer function is a canonic fifth-order binomial given by

$$\frac{n_L}{\lambda} = \frac{N' V_c s}{(1 + sT/5)^5}$$

where T is the total guidance system time constant, n_L is the achieved missile acceleration, and λ is the line-of-sight angle. As was pointed out in Chapter 6, one time constant represents the seeker, another represents the noise filter, and the other three time constants represent the flight-control system dynamics (aerodynamics plus autopilot). With this canonic guidance system model the overall guidance system time constant is simply the sum of the five individual time constants or T . For convenience the fifth-order binomial missile homing loop of Chapter 6 is also repeated in Fig. 20.13.

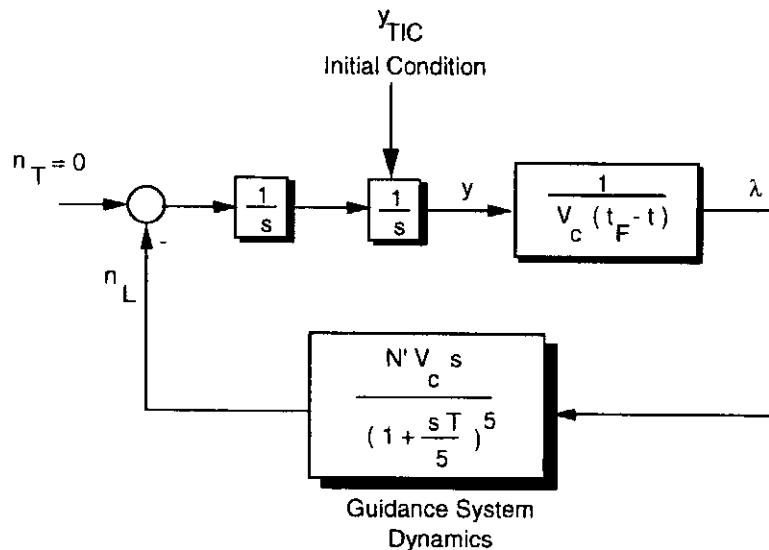


Fig. 20.13 Fifth-order binomial guidance system.

An adjoint block diagram for finding the miss due to a step in target displacement can be constructed from Fig. 20.13 directly or by modifying the adjoint block diagram of Fig. 6.5. The resultant miss distance adjoint block diagram appears in Fig. 20.14.

An adjoint simulation of the fifth-order binomial guidance system can be constructed directly from Fig. 20.14 and appears in Listing 20.4. This adjoint simulation is nearly identical to that of Listing 6.1 except the states corresponding to the miss due to the ramp and parabolic maneuvers have been removed and the miss due to a step in target displacement XMYT has been added.

Normalized miss distance curves for different effective navigation ratios were generated from Listing 20.4. Because the adjoint technique was used, it was also implicitly assumed that the missile had infinite acceleration capability. As was done in Chapter 6, it was also assumed that the adjoint curves generated with

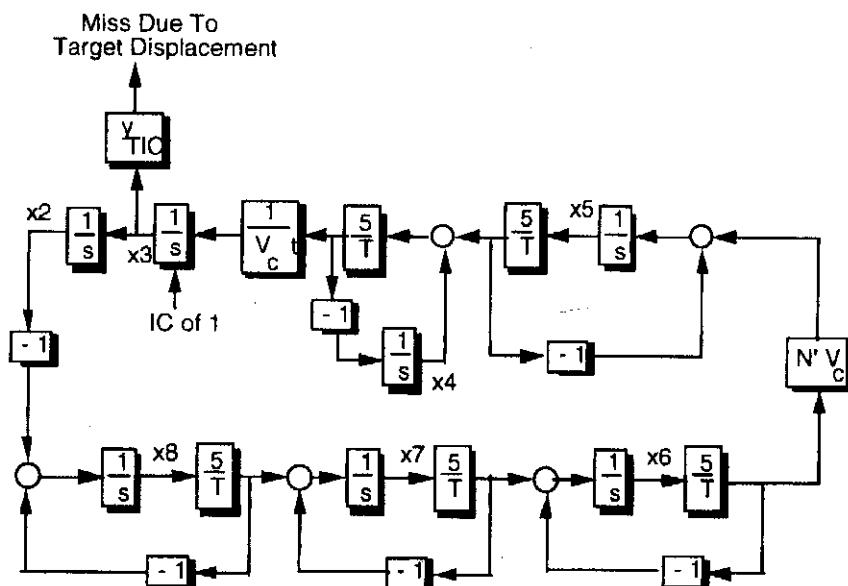


Fig. 20.14 Adjoint of fifth-order binomial guidance system.

Listing 20.4 Adjoint of fifth-order binomial guidance system used to find normalized miss due to step in target displacement

```

INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
XNP=3.
TAU=1.
TF=10.
VC=4000.
YTIC=1.
T=0.
S=0.
TP=T+.00001
X2=0
X3=1
X4=0
X5=0.
X6=0.
X7=0.
X8=0.
H=.01
10 IF(TP>(TF-.00001))GOTO 999
S=S+H
X2OLD=X2
X3OLD=X3
X4OLD=X4
X5OLD=X5
X6OLD=X6
X7OLD=X7
X8OLD=X8
STEP=1
GOTO 200
66 STEP=2
X2=X2+H*X2D
X3=X3+H*X3D
X4=X4+H*X4D
X5=X5+H*X5D
X6=X6+H*X6D
X7=X7+H*X7D
X8=X8+H*X8D
TP=TP+H
GOTO 200
55 CONTINUE
X2=(X2OLD+X2)/2+.5*H*X2D
X3=(X3OLD+X3)/2+.5*H*X3D
X4=(X4OLD+X4)/2+.5*H*X4D
X5=(X5OLD+X5)/2+.5*H*X5D
X6=(X6OLD+X6)/2+.5*H*X6D
X7=(X7OLD+X7)/2+.5*H*X7D
X8=(X8OLD+X8)/2+.5*H*X8D

```

(Contd.)

Listing 20.4 (Continued)

```

IF(S<.09999)GOTO 10
S=0.
XMYT=YTC*X3
WRITE(*,97)TP,XMYT
WRITE(1,97)TP,XMYT
GOTO 10
97 FORMAT(2F10.3)
200 CONTINUE
X2D=X3
Y1=5.*X5/TAU+X4)/TAU
TGO=TP+.00001
X3D=Y1/(VC*TGO)
X4D=-Y1
X5D=-5.*X5/TAU+5.*X6*XNP*VC/TAU
X6D=-5.*X6/TAU+5.*X7/TAU
X7D=-5.*X7/TAU+5.*X8/TAU
X8D=-5.*X8/TAU-X2
IF(STEP-1)66,66,55
999 CONTINUE
PAUSE
CLOSE(1)
END

```

Listing 20.4 had the same normalization factors as those curves for the single time constant guidance system. The hypothesis was checked by running numerous cases with Listing 20.4 using various combinations of guidance system time constant and homing time. The hypothesis was found to be true and the resultant normalized curves are displayed in Fig. 20.15 (Ref. 2). Superimposed on Fig. 20.15 is an indication of where the power centroid is in relation to the target we are supposed to be guiding on (aircraft 1 in this case). In other words a normalized miss of unity means we are hitting the power centroid and a normalized miss of zero means we are hitting aircraft 1. By comparing Figs. 20.12 and 20.15 we can conclude that

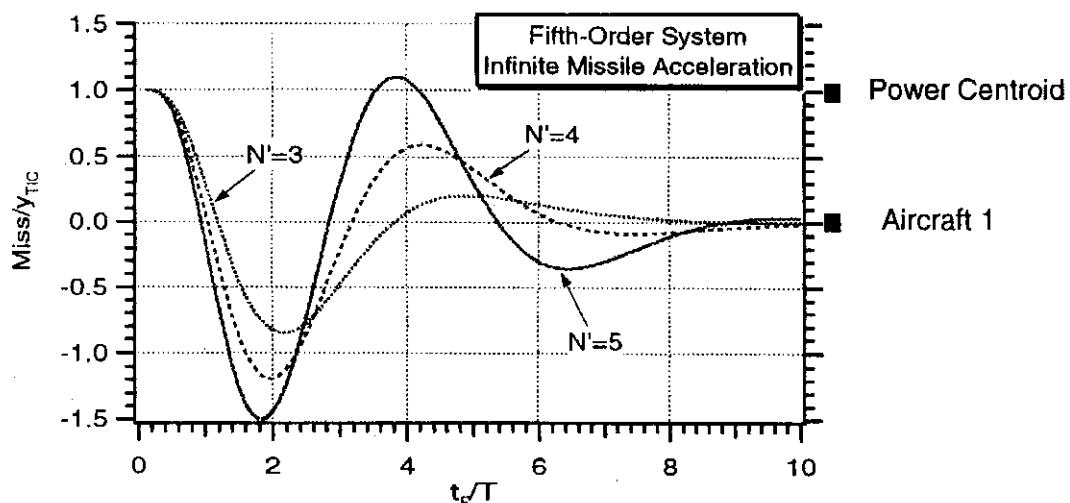


Fig. 20.15 Fifth-order normalized miss for a step in target displacement.

in general the miss distances for the fifth-order guidance system are much larger than the miss for a single time constant guidance system. In addition, Fig. 20.15 also shows that the ratio of the flight time t_F (or time remaining after the apparent step in target displacement has occurred) to the guidance system time constant T must now be greater than 10 for there to be negligible miss distance. Recall that for the single time constant system the number of guidance time constants had to be greater than five for the miss distance to be negligible. If the number of guidance time constants is less than 10, it is really a matter of luck as to how large or small the miss distance will be. Luck is involved because in reality the point at which resolution occurs for a specific engagement is random.

If we consider the same example of the previous section, where the miss distances were smaller, the importance of higher order guidance system dynamics can be seen to be even more important. For a 200-ft target displacement (equivalent to 400-ft aircraft separation), 1 s of effective flight time ($t_F = 1$) and an overall guidance system time constant of 0.5 s ($T = 0.5$), the number of guidance time constants is 2 ($t_F/T = 1/0.5 = 2$) and the corresponding miss distance for different effective navigation ratios can be computed from Fig. 20.15 as

$$\text{Miss}_{N'=3} = -0.827 * 200 = -165 \text{ ft}$$

$$\text{Miss}_{N'=4} = -1.19 * 200 = -238 \text{ ft}$$

$$\text{Miss}_{N'=5} = -1.44 * 200 = -288 \text{ ft}$$

Note that these miss distances are an order of magnitude greater than those of the single time constant guidance system! Thus we can see that the miss distances can be enormous and in fact (for cases in which the effective navigation ratio was 4 and 5) can be even greater than the original apparent target displacement. For example, if the effective navigation ratio is 4, -238 ft of miss means that the missile missed aircraft 1 (target it was homing on) by 238 ft and aircraft 2 (second target) by 638 ft ($400 + 238 = 638$). If the missile time constant can be halved, or if the seeker beamwidth can be halved, then the number of guidance time constants is doubled to 4 ($t_F/T = 1/0.25 = 4$) thus considerably reducing the miss or

$$\text{Miss}_{N'=3} = 0.069 * 200 = 14 \text{ ft}$$

$$\text{Miss}_{N'=4} = 0.565 * 200 = 113 \text{ ft}$$

$$\text{Miss}_{N'=5} = 1.08 * 200 = 216 \text{ ft}$$

Thus we can see that the ratio of the flight time remaining after resolution has occurred to the guidance system time constant is critical in determining the expected miss distance.

Acceleration Saturation

We have observed in the previous two sections that both the guidance system dynamics and effective navigation ratio play an important role in determining the miss distance due to a step in target displacement. The finite acceleration capability of the interceptor is also important in determining the miss distance. Normalized miss distance curves can also be developed when missile acceleration saturation effects are considered. In this case it is hypothesized that miss distance

normalization factors remain unchanged but new curves have to be developed for the nondimensional ratio

$$\text{Ratio} = \frac{0.5n_{\text{LIM}} T^2}{y_{\text{TIC}}}$$

where n_{LIM} is the value of the acceleration limit.³

For a fixed level of target displacement and guidance system time constant, a missile with more limited acceleration capability (smaller acceleration limit n_{LIM}) has a smaller ratio. Using the preceding ratio and the normalization factors for miss due to a step in target displacement, we can derive normalized miss distance curves by the method of brute force. In other words, we can generate normalized miss distance curves by simulating all of the possibilities. We can then infer performance by making extrapolations from the normalized miss distance curves as was done in Chapter 6 when design curves were developed for the target maneuver miss. Of course, detailed checks have been made to ensure that the normalization factors are correct. Listing 20.5 presents a brute force simulation with linear geometry and nonlinear acceleration saturation. The simulation is for the fifth-order binomial guidance system of Fig. 20.13. The program has an internal loop so when the simulation is run, the miss distance for flight times ranging from 0.1 s to 10 s in steps of 0.1 s are evaluated.

Figures 20.16–20.18 present the normalized miss distances due to a step target displacement for effective navigation ratios ranging from three to five respectively.² These curves were generated with Listing 20.5. As expected, we can see that less acceleration capability (ratio smaller) requires more guidance system time constants to move from the power centroid (normalized miss is one) to the resolved target (normalized miss is zero). On the other hand, we can also see that missiles with less acceleration capability have less of a tendency to overshoot the target (less negative normalized miss) because there is less energy available. *In other words if the number of guidance time constants is less than 10, it is possible for a missile with less acceleration capability to have a smaller miss distance than a missile with a larger acceleration capability!* We can also see by comparing the normalized curves of Figs. 20.16–20.18 that larger effective navigation ratios tend

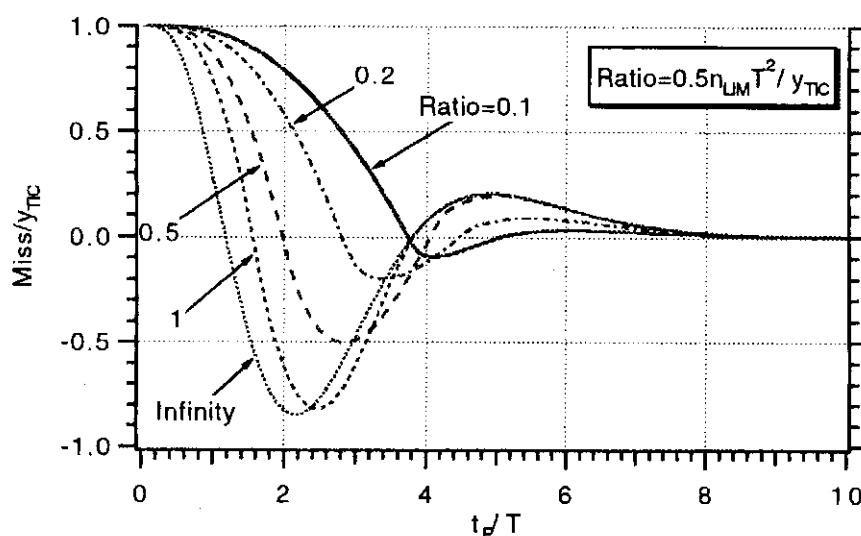


Fig. 20.16 Normalized miss due to saturation and target displacement for $N' = 3$.

Listing 20.5 Brute force simulation for finding normalized miss due to step in target displacement for fifth-order binomial guidance system

```

INTEGER STEP
VC=4000.
XNT=0.
DISPLACE=1.
VM=3000.
TAU=1.
XNP=3.
XNCLIM=99999999.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
DO 20 TF=.1,10.,.1
Y=DISPLACE
YD=0.
XNL=0.
D=0.
ELAMDH=0.
X4=0.
X5=0.
T=0.
H=.01
S=0.
10 IF(T>(TF-.0001))GOTO 999
YOLD=Y
YDOLD=YD
XNLOLD=XNL
DOLD=D
ELAMDHOLD=ELAMDH
X4OLD=X4
X5OLD=X5
STEP=1
GOTO 200
66 STEP=2
Y=Y+H*YD
YD=YD+H*YDD
XNL=XNL+H*XNLD
ELAMDH=ELAMDH+H*ELAMDHD
D=D+H*DD
X4=X4+H*X4D
X5=X5+H*X5D
T=T+H
GOTO 200
55 CONTINUE
Y=.5*(YOLD+Y+H*YD)
YD=.5*(YDOLD+YD+H*YDD)
XNL=.5*(XNLOLD+XNL+H*XNLD)
D=.5*(DOLD+D+H*DD)
ELAMDH=.5*(ELAMDHOLD+ELAMDH+H*ELAMDHD)
X4=.5*(X4OLD+X4+H*X4D)

```

(Contd.)

Listing 20.5 (Continued)

```

X5=.5*(X5OLD+X5+H*X5D)
GOTO 10
200 CONTINUE
TGO=TF-T+.00001
XLAM=Y/(VC*TGO)
DD=5.*(XLAM-D)/TAU
ELAMDHD=5.*(DD-ELAMDH)/TAU
XNC=XNP*VC*ELAMDH
IF(XNC>XNCLIM)XNC=XNCLIM
IF(XNC<-XNCLIM)XNC=-XNCLIM
X4D=5.*(XNC-X4)/TAU
X5D=5.*(X4-X5)/TAU
XNLD=5.*(X5-XNL)/TAU
YDD=XNT-XNL
IF(STEP-1)66,66,55
999 CONTINUE
WRITE(1,97)TF,Y
WRITE(*,97)TF,Y
20 CONTINUE
97 FORMAT(2F10.3)
CLOSE(1)
PAUSE
END

```

to increase the number of guidance time constants required for the miss to approach zero (i.e., approximately six guidance time constants are required for $N' = 3$ and 10 guidance time constants are required for a navigation ratio of five).

To demonstrate the utility of the normalized curves of Figs. 20.16–20.18, let us again consider the same example of the previous section where there was an apparent 200-ft step target displacement (equivalent to 400-ft aircraft separation) and 1 s of effective flight time (equivalent to 0.1 rad seeker beamwidth and 400-ft

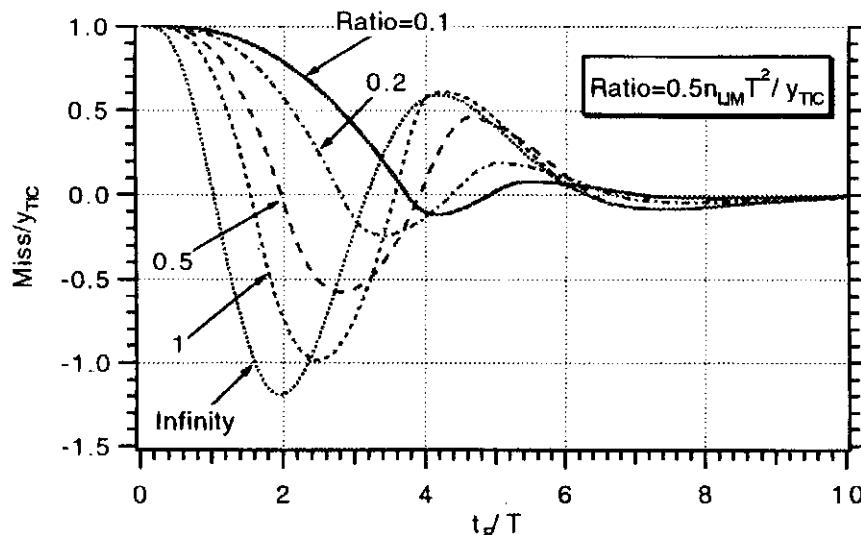


Fig. 20.17 Normalized miss due to saturation and target displacement for $N' = 4$.

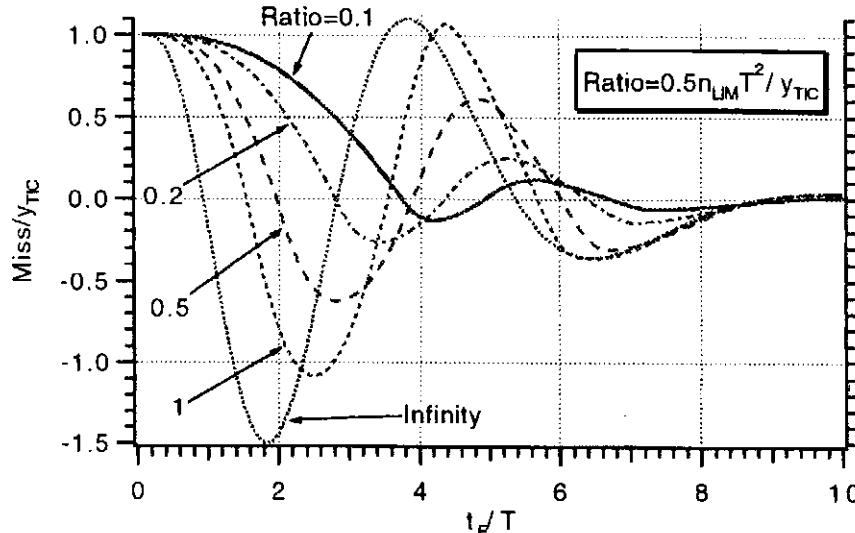


Fig. 20.18 Normalized miss due to saturation and target displacement for $N' = 5$.

aircraft spacing) and an overall guidance system time constant of 0.5 s ($T = 0.5$). In this case the number of guidance time constants is still two ($t_F/T = 1/0.5 = 2$), and the corresponding miss distance for an effective navigation ratio of 3 and different values of missile acceleration limits can be read from the normalized curves of Fig. 20.16. For example, if the missile acceleration limit is 320 ft/s^2 ($9.9 g$) then the ratio can be computed as

$$\text{Ratio} = \frac{0.5n_{\text{LIM}}T^2}{y_{\text{TIC}}} = \frac{0.5 * 320 * 0.5^2}{200} = 0.2$$

Therefore for two guidance time constants the normalized miss corresponding to a ratio of 0.2 can be read from Fig. 20.16 as 0.6. We can calculate the actual miss distance for this example to be

$$\text{Miss}_{9.9g} = 0.6 * 200 = 120 \text{ ft}$$

Increasing the missile acceleration limit to 800 ft/s^2 ($24.8 g$) increases the ratio to 0.5 or

$$\text{Ratio} = \frac{0.5n_{\text{LIM}}T^2}{y_{\text{TIC}}} = \frac{0.5 * 800 * 0.5^2}{200} = 0.5$$

Therefore for two guidance time constants the normalized miss corresponding to a ratio of 0.5 can be read from Fig. 20.16 as zero. In this case the actual miss distance is zero or

$$\text{Miss}_{24.8g} = 0 * 200 = 0 \text{ ft}$$

Increasing the missile acceleration limit again to 1600 ft/s^2 ($49.7 g$) increases the ratio to one or

$$\text{Ratio} = \frac{0.5n_{\text{LIM}}T^2}{y_{\text{TIC}}} = \frac{0.5 * 1600 * 0.5^2}{200} = 1$$

Therefore for two guidance time constants the normalized miss corresponding to a ratio of one can be read from Fig. 20.15 as -0.6 . The actual miss distance for

this example is simply

$$\text{Miss}_{49.7g} = -0.6 * 200 = -120 \text{ ft}$$

Finally, if the missile had infinite acceleration capability the ratio would be infinity. For two guidance time constants the normalized miss corresponding to infinite ratio can be read from Fig. 20.16 as -0.8. Therefore the actual miss distance for a missile with infinite acceleration capability increases to

$$\text{Miss}_{\infty g} = -0.8 * 200 = -160 \text{ ft}$$

Thus we can see that a missile with more acceleration capability does not necessarily have better miss distance performance against a step in target displacement!

Summary

Normalized miss distance curves were presented showing the designer how to calculate the miss distance due to an apparent step in target displacement. The importance of guidance system dynamics and missile acceleration saturation effects were illustrated with additional design curves and examples.

References

- ¹Travers, P., "Interceptor Dynamics," unpublished lecture notes, Raytheon Co., circa 1971.
- ²Zarchan, P., "When Bad Things Happen to Good Missiles," *Proceedings of AIAA Guidance, Navigation and Control Conference*, AIAA, Washington, DC, Aug. 1993.
- ³Shinar, J., and Steinberg, D., "Analysis of Optimal Evasive Maneuvers Based on a Linearized Two-Dimensional Model," *Journal of Aircraft*, Vol. 14, Aug. 1977, pp. 795-802.



Weaving Targets

Introduction and Background

We have seen in Chapter 6 that large miss distances could be induced by the target if a maximum acceleration maneuver was initiated at the proper time to go before intercept. It was also shown that the barrel roll or weave maneuver could also generate large miss distances. Because it is well known that tactical ballistic missiles (TBMs) can spiral or weave into resonance (i.e., TBM roll rate equals vehicles natural pitch frequency) as they re-enter the atmosphere due to either mass or configurational asymmetries, the weave maneuver is of particular interest to the guidance system designer.^{1,2}

In this chapter we will first study the influence of the target weave maneuver on a single time constant proportional navigation guidance system. Closed-form solutions for the peak steady-state miss distance as a function of the effective navigation ratio, guidance system time constant, weave maneuver amplitude and frequency will be derived.^{3,4} Because we have already shown that the single time constant guidance system seriously underestimates the miss distance, a more realistic, higher-order guidance system will be used to develop normalized miss distance design curves using the normalization factors from the single time constant target maneuver miss distance solutions. The finite acceleration capability of the interceptor also plays an important role in determining system performance. The normalized design curves, which assumed infinite missile acceleration capability, are updated to show how the missile acceleration advantage over the target plays a key role in determining system performance. Finally, methods for improving missile system performance against weaving targets will be explored.

Weave Maneuver in Single Time Constant Guidance System

Periodic maneuver sequences such as a sinusoidal or weaving target present a challenge for a missile guidance system designer. A planar representation of a weaving target is given by

$$\text{Target Maneuver} = n_T \sin \omega_T t$$

where n_T is the maneuver amplitude, ω_T is the target weave frequency, and t is time. The miss due to a weaving target as a function of flight time can be found using the method of brute force. A nonlinear two-dimensional engagement simulation, based on Listing 21.1, of a missile guiding on a weaving target appears in Listing 21.1. We can see that the listing is based on a single time constant proportional navigation guidance system and that for the nominal case the missile time constant is 1 s,

Listing 21.1 Nonlinear engagement simulation for a single time constant guidance system with weaving target

```

XNP=3.
TAU=1.
XNT=193.2
W=3.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
DO 22 RT1IC=500.,40000.,500.
VM=3000.
VT=1000.
RM1=0.
RM2=0.
RT1=RT1IC
RT2=0.
BETA=0.
VT1=-VT*COS(BETA)
VT2=VT*SIN(BETA)
T=0.
S=0.
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
XLAM=ATAN2(RTM2,RTM1)
VM1=VM
VM2=0.
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
TGO=RTM/VC
XLAMH=0.
H=.005
10 IF(VC<0.) GOTO 999
IF(RTM<1000.)H=.00005
BETAOLD=BETA
RT1OLD=RT1
RT2OLD=RT2
RM1OLD=RM1
RM2OLD=RM2
VM1OLD=VM1
VM2OLD=VM2
XLAMHOLD=XLAMH
STEP=1
GOTO=200
66 STEP=2
BETA=BETA+H*BETAD
RT1=RT1+H*VT1
RT2=RT2+H*VT2
RM1=RM1+H*VM1
RM2=RM2+H*VM2
VM1=VM1+H*AM1
VM2=VM2+H*AM2

```

(Contd.)

Listing 21.1 (Continued)

```

XLAMH=XLAMH+H*XLAMHD
T=T+H
GOTO 200
55 CONTINUE
BETA=.5*(BETAOLD+BETA+H*BETAD)
RT1=.5*(RT1OLD+RT1+H*VT1)
RT2=.5*(RT2OLD+RT2+H*VT2)
RM1=.5*(RM1OLD+RM1+H*VM1)
RM2=.5*(RM2OLD+RM2+H*VM2)
VM1=.5*(VM1OLD+VM1+H*AM1)
VM2=.5*(VM2OLD+VM2+H*AM2)
XLAMH=.5*(XLAMHOLD+XLAMH+H*XLAMHD)
S=S+H
GOTO 10
200 CONTINUE
VT1=-VT*COS(BETA)
VT2=VT*SIN(BETA)
BETAD=XNT*SIN(W*T)/VT
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
XLAM=ATAN2(RTM2,RTM1)
XLAMHD=(XLAM-XLAMH)/TAU
XNC=XNP*VC*XLAMHD
AM1=-XNC*SIN(XLAM)
AM2=XNC*COS(XLAM)
IF(STEP-1)66,66,55
IF(RTM2>0.)THEN
    RTMP=RTM
ELSE
    RTMP=-RTM
ENDIF
WRITE(*,97)T,RTMP
WRITE(1,97)T,RTMP
22 CONTINUE
97 FORMAT(2F10.3)
PAUSE
CLOSE(1)
END

```

the effective navigation ratio is 3, the target weave frequency is 3 rad/s while the target maneuver amplitude is 193.2 ft/s² or 6 g. The target always initiates its maneuver at the beginning of flight in this simulation. The program is set up to run in the brute force mode so that the miss distance results for many flight times can be evaluated. In this nonlinear engagement simulation the initial target downrange position, which is equivalent to the initial missile-target separation, is varied from 500 ft to 40,000 ft in steps of 500 ft, which in the linear world is

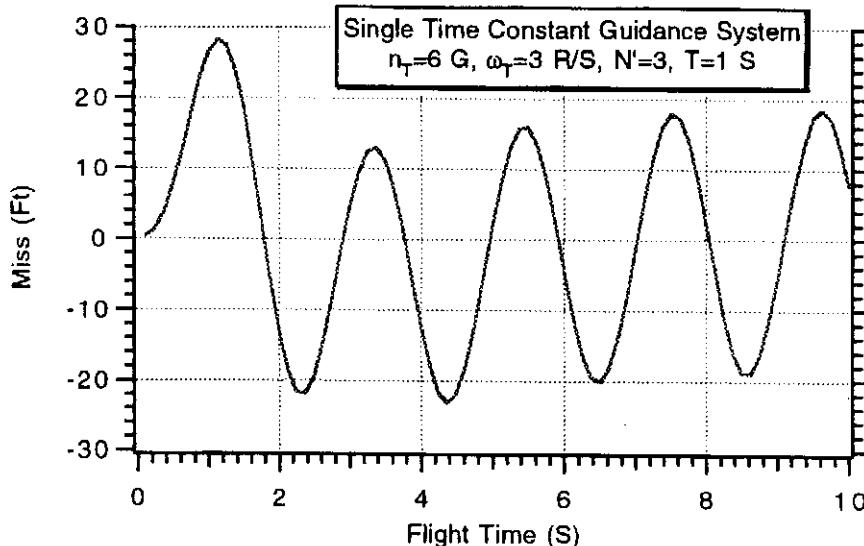


Fig. 21.1 Nonlinear results indicate that weaving target causes miss to oscillate at target weave frequency.

equivalent to varying the flight time from 0.05 s to 10 s in steps of 0.05 s because the closing velocity is approximately 4000 ft/s. After each run the homing time and miss distance are tabulated. We can also see from the simulation listing that if the target is above the missile at intercept, the miss is considered to be positive whereas if the target is below the missile at intercept the miss is considered to be negative.

The nominal case of Listing 21.1 was run, and the miss distance results as a function of flight time appear in Fig. 21.1. We can see that unlike the step target maneuver results of Chapters 3 and 6, *the miss distance due to weaving target does not approach zero as the homing time increases!* Depending on the flight time, the miss distance for this example can be as large as 28 ft or as small as zero when the effective navigation ratio is 3. Also note that after an initial transient period the miss is sinusoidal in nature with frequency 3 rad/s, which is identical to the target weave frequency.

Could these multiple run nonlinear results be generated with one adjoint run? To find out, we first have to assume that the geometry of the engagement can be linearized in a manner similar to that of Chapters 2, 3, and 20. In addition, because adjoint theory requires that all inputs to the original system appear as impulses, we also have to find some way of making the sinusoidal maneuver look like an impulse through a linear shaping network. Fortunately, in Table 1.1 of Chapter 1 we showed that a sinusoidal maneuver could be represented as an impulse through a second-order shaping network since

$$\mathcal{L}[n_T \sin \omega_T t] = \frac{\omega_T n_T}{s^2 + \omega_T^2}$$

Figure 21.2 shows a linearized representation of a single time constant linear proportional navigation guidance system driven by a weave target maneuver (i.e., impulsive input through second-order shaping network).

Figure 21.2 is drawn in such a way that the adjoint can be taken by inspection. The new adjoint diagram was constructed from Fig. 21.2 using the adjoint rules of Chapter 3 and appears in Fig. 21.3. In this particular adjoint diagram we are evaluating one disturbance only, the miss due to a weaving target.

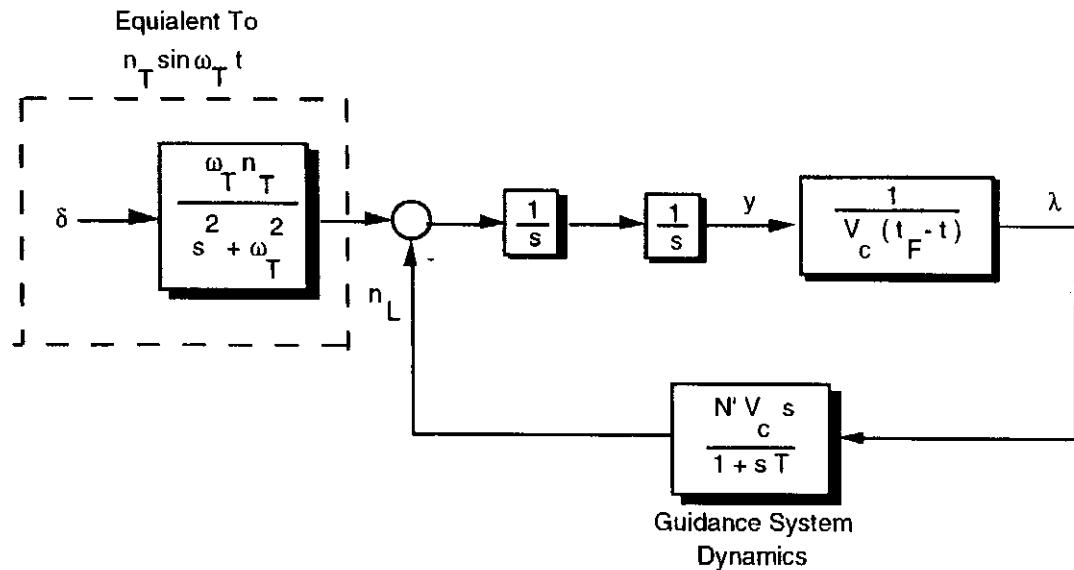


Fig. 21.2 Linearized single time constant guidance system with weave maneuver.

An adjoint simulation, based on Fig. 21.3, was constructed and appears in Listing 21.2. Here we can see that the nominal case is set up to be identical to that of the brute force nonlinear engagement simulation of Listing 21.1. However, with the adjoint simulation only one run has to be made to find out how the miss distance varies with flight time.

The nominal case of Listing 21.2 was run, and the adjoint miss distance results were plotted alongside the nonlinear, multiple run results of Fig. 21.1. We can see from Fig. 21.4 that the adjoint results are virtually identical to the nonlinear miss distance results thus validating the simplified linear model. In the rest of this chapter, we will use the linearized model of the guidance system to generate performance evaluations and to suggest ways of improving performance.

Another case was run with the adjoint simulation in which the target weave frequency was decreased from 3 rad/s to 1.5 rad/s. We can see from Fig. 21.5 that the miss distance increases significantly with the lower weave frequency and the oscillation of the miss with flight time changes to match the new target weave frequency. We now would like to get a deeper understanding of how performance is related to the target weave frequency, guidance system time constant, and effective navigation ratio.

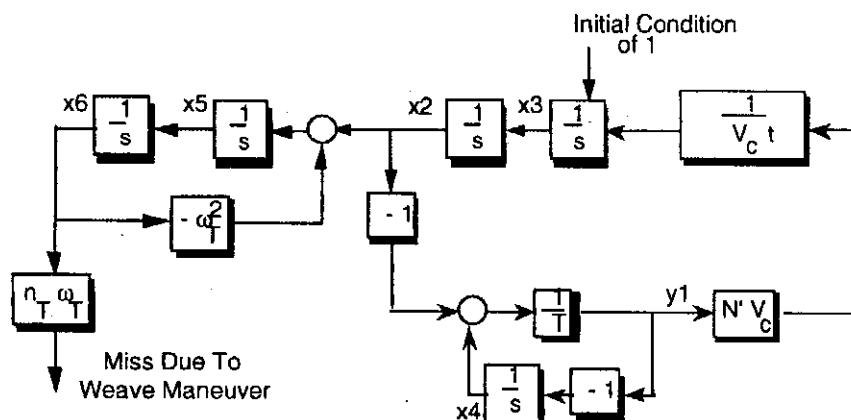


Fig. 21.3 Adjoint of single time constant guidance system with weaving target.

**Listing 21.2 Adjoint simulation of single time
constant guidance system and weaving target**

```

INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
XNT=193.2
XNP=3.
TAU=1.
TF=10.
VC=4000.
W=3.
T=0.
S=0.
TP=T+.00001
X2=0
X3=1
X4=0.
X5=0.
X6=0.
H=.01
10 IF(TP>(TF-.00001))GOTO 999
S=S+H
X2OLD=X2
X3OLD=X3
X4OLD=X4
X5OLD=X5
X6OLD=X6
STEP=1
GOTO 200
66 STEP=2
X2=X2+H*X2D
X3=X3+H*X3D
X4=X4+H*X4D
X5=X5+H*X5D
X6=X6+H*X6D
TP=TP+H
GOTO 200
55 CONTINUE
X2=(X2OLD+X2)/2+.5*H*X2D
X3=(X3OLD+X3)/2+.5*H*X3D
X4=(X4OLD+X4)/2+.5*H*X4D
X5=(X5OLD+X5)/2+.5*H*X5D
X6=(X6OLD+X6)/2+.5*H*X6D
IF(S<.09999)GOTO 10
S=0.
XMWEAVE=XNT*W*X6
WRITE(*,97)TP,XMWEAVE
WRITE(1,97)TP,XMWEAVE
GOTO 10
97 FORMAT(2F10.3)
200 CONTINUE
X2D=X3

```

(Contd.)

Listing 21.2 (Continued)

```

Y1=(-X2+X4)/TAU
TGO=TP+.00001
X3D=Y1*XNP/TGO
X4D=-Y1
X5D=X2-W*W*X6
X6D=X5
IF(STEP-1)66,66,55
999 (CONTINUE)
PAUSE
CLOSE(1)
END

```

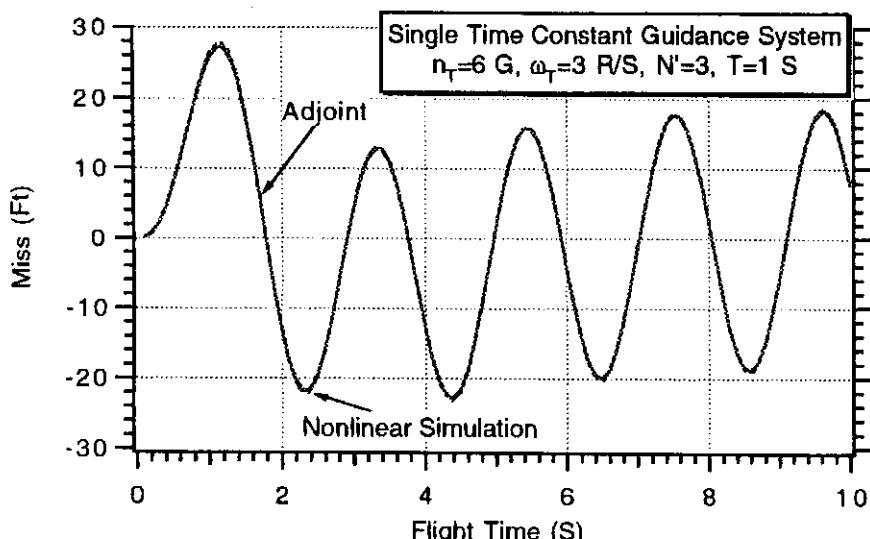


Fig. 21.4 Adjoint simulation results agree with the nonlinear results for the weaving target disturbance.

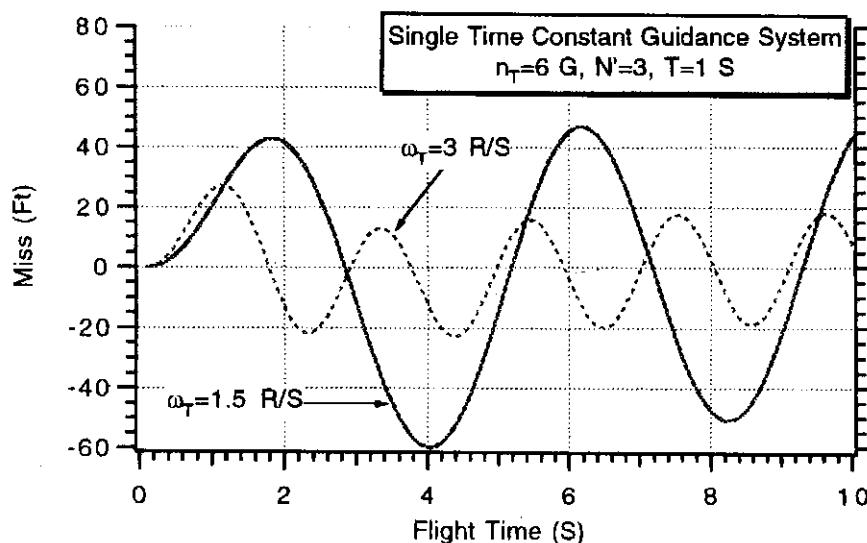


Fig. 21.5 Decreasing weave frequency increases miss distance for this example.

Closed-Form Solutions for Miss Distance

Because we have already shown that the miss due to a weaving target is a sinusoidal function of the flight time, it only makes sense to look at steady-state miss distances in order to quantify system performance. Closed-form solutions for the miss due to a weaving target can be obtained in the steady state (i.e., at large flight times when transients die out). Recall that all of the miss distance formulas for a single time constant guidance system were derived in Chapter 3 from the generalized adjoint diagram of Fig. 3.16. Figure 3.16 is redrawn and updated to include the miss due to a weaving target MWEAVE as shown in Fig. 21.6.

In Chapter 3 we found that the miss due to a step target maneuver MNT, expressed in the Laplace transform domain, was given by

$$MNT(s) = \frac{1 - H(s)}{s^3} * n_T$$

Therefore the miss due to a weaving target can be found by inspection from Fig. 21.6 and can be expressed as

$$MWEAVE(s) = \frac{1 - H(s)}{s^2} * \frac{n_T \omega_T}{s^2 + \omega_T^2}$$

where $1 - H(s)$ was shown in Chapter 3 for the single time constant guidance system to be given by

$$1 - H(s) = \left[\frac{s}{s + \frac{1}{T}} \right]^{N'}$$

Therefore the miss due to a weaving target can be expressed as

$$MWEAVE(s) = \frac{1}{s^2} * \left[\frac{s}{s + \frac{1}{T}} \right]^{N'} * \frac{n_T \omega_T}{s^2 + \omega_T^2}$$

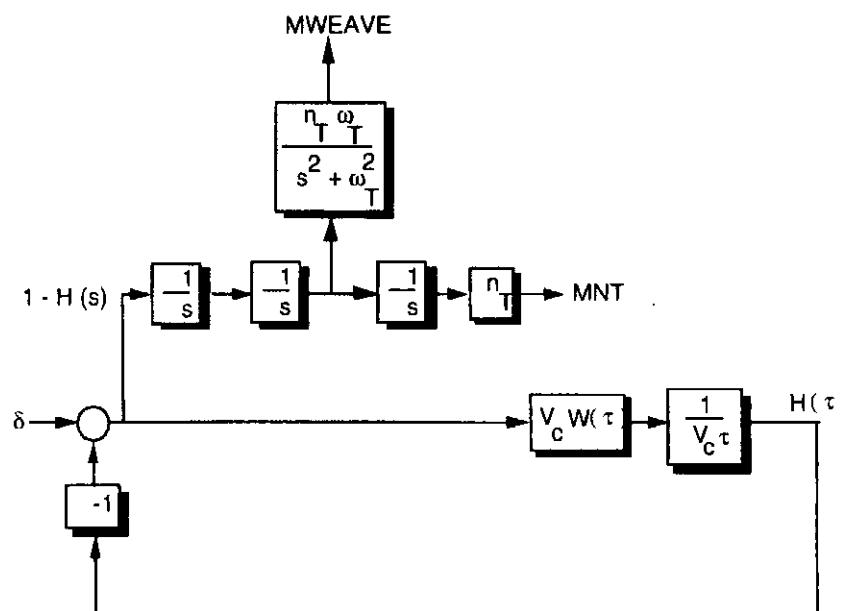


Fig. 21.6 Generalized adjoint diagram showing miss due to weaving target.

The miss distance in the Laplace transform domain can be evaluated directly by first doing a partial fraction expansion of the terms on the right side of the preceding equation and then taking the inverse Laplace transform to find the miss in the time domain. Some of the terms in the resultant complex expression would be transient in nature while other terms would be sinusoidal. In the steady state the transient terms would go to zero and only the sinusoidal terms would be left.

If we are only interested in the steady-state solution, much work can be saved using a simple technique from electrical engineering. We can rewrite the preceding expression as

$$\frac{\text{Miss}}{\text{weave } n_T}(s) = \frac{32.2}{s^2} \left[\frac{s}{s + \frac{1}{T}} \right]^{N'}$$

where n_T is now the maneuver magnitude in units of g and "weave n_T " reminds us that the target maneuver is sinusoidal. If a linear system has a sine wave input with frequency ω_T in units of rad/s, the output in the steady state will also be a sinusoid of the same frequency but of different magnitude and phase. From basic steady-state electrical engineering circuit analysis techniques it can be shown that the magnitude and phase of the sinusoidal output can be found by replacing s with $j\omega_T$ in the preceding transfer function and then finding the magnitude and phase of the resultant complex transfer function.⁵ For example, if the effective navigation ratio is three, the preceding transfer function becomes

$$\frac{\text{Miss}}{\text{weave } n_T} \Big|_{N'=3}(s) = \frac{32.2s}{(s + 1/T)^3}$$

Therefore, the complex weave miss distance transfer function can be derived from the preceding equation by substitution (i.e., $s = j\omega_T$) as

$$\frac{\text{Miss}}{\text{weave } n_T} \Big|_{N'=3}(j\omega_T) = \frac{32.2j\omega_T}{(j\omega_T + 1/T)^3}$$

The magnitude and phase of this complex transfer function can be written by inspection as

$$\text{Magnitude} \Big|_{N'=3} = \frac{32.2\omega_T}{(\omega_T^2 + 1/T^2)^{1.5}}$$

$$\text{Phase} \Big|_{N'=3} = \frac{\pi}{2} - 3 \tan^{-1} \omega_T T$$

Therefore the steady-state miss distance due to a weaving target can be written in the time domain as

$$\frac{\text{Miss}}{\text{weave } n_T} \Big|_{N'=3, \text{Steady-State}} = \text{Magnitude} \Big|_{N'=3} \sin (\omega_T t_F + \text{Phase} \Big|_{N'=3})$$

or

$$\frac{\text{Miss}}{\text{weave } n_T} \Big|_{N'=3, \text{Steady-State}} = \frac{32.2\omega_T}{(\omega_T^2 + \frac{1}{T^2})^{1.5}} \sin \left(\omega_T t_F + \frac{\pi}{2} - 3 \tan^{-1} \omega_T T \right)$$

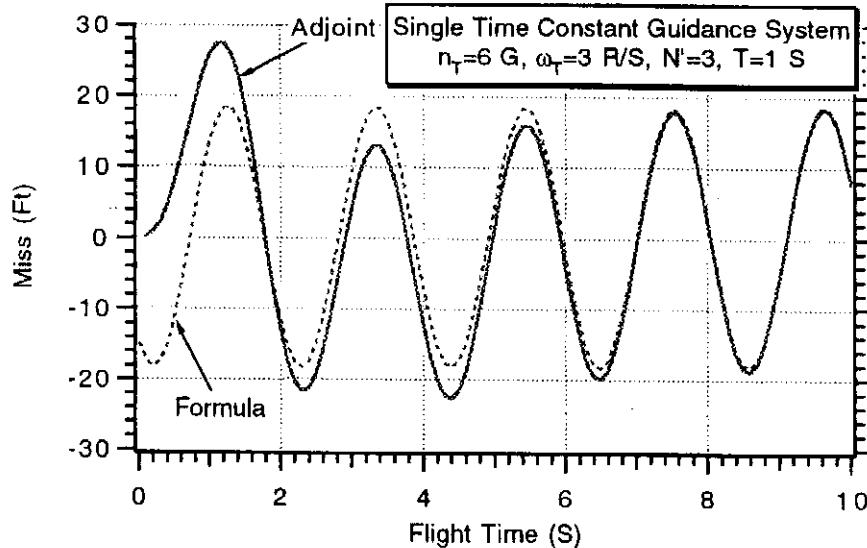


Fig. 21.7 Closed-form miss distance solution agrees with adjoint results.

Figure 21.7 presents again the adjoint miss distance results as a function of flight time for the case in which the target weave frequency is 3 rad/s while the missile guidance system time constant is 1 s. Superimposed on the figure is the preceding closed-form solution for the miss distance. We can see that after an initial transient period the closed-form steady-state miss distance solution and computer generated adjoint results are in excellent agreement thus confirming the validity of the steady-state analysis.

We have shown mathematically and by simulation that the miss distance due to a weaving target is a sinusoidal function of the flight time. Therefore it is really a matter of luck on how large or small the miss distance will be. Of particular concern to the missile guidance system designer is the maximum or peak value of the sinusoidal miss distance function. The peak value of the miss due to a weave maneuver is simply the magnitude of the steady-state miss distance sinusoid. Therefore the peak miss due to a weave maneuver in a single time constant proportional navigation guidance system with an effective navigation ratio of three is given by

$$\left. \frac{\text{Peak Miss}}{\text{weave } n_T} \right|_{N'=3} = \frac{32.2 \omega_T}{(\omega_T^2 + 1/T^2)^{1.5}} = \frac{32.2 \omega_T T^3}{(1 + \omega_T^2 T^2)^{1.5}}$$

Dividing both sides of the equation by T^2 yields

$$\left. \frac{\text{Peak Miss}}{\text{weave } n_T T^2} \right|_{N'=3} = \frac{32.2 \omega_T T}{(1 + \omega_T^2 T^2)^{1.5}}$$

If we let x be the normalized target weave frequency where

$$x = \omega_T T$$

the peak miss distance formula simplifies further to

$$\left. \frac{\text{Peak Miss}}{\text{weave } n_T T^2} \right|_{N'=3} = \frac{32.2 x}{(1 + x^2)^{1.5}}$$

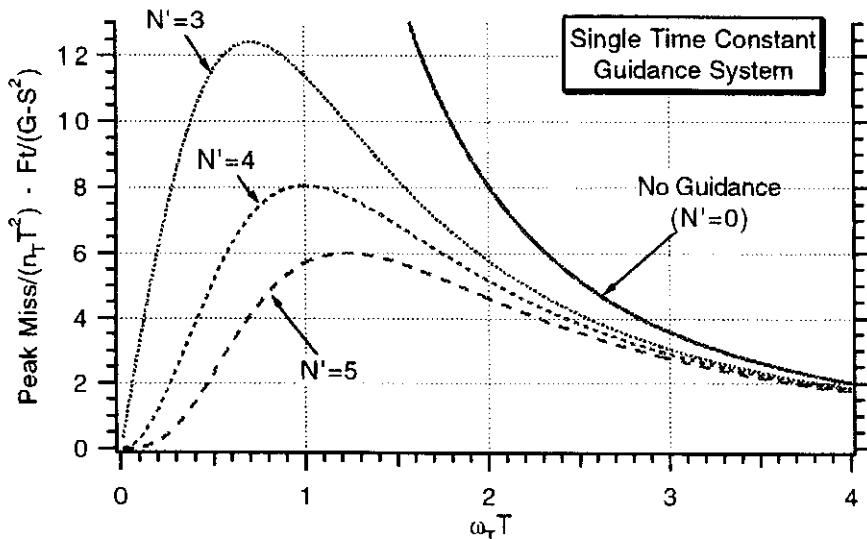


Fig. 21.8 Peak miss distance is maximum when normalized weave frequency is near unity.

Similar expressions can be found for the peak miss distance due to a weave maneuver when the effective navigation ratios are four and five and can be shown to be

$$\frac{\text{Peak Miss}}{\text{weave } n_T T^2} \Big|_{N'=4} = \frac{32.2 x^2}{(1+x^2)^2}$$

$$\frac{\text{Peak Miss}}{\text{weave } n_T T^2} \Big|_{N'=5} = \frac{32.2 x^3}{(1+x^2)^{2.5}}$$

Figure 21.8 graphically displays the preceding formulas and shows how the steady-state normalized peak miss distance varies with the normalized target maneuver frequency (i.e., product of the target weave frequency and the missile guidance system time constant). We can see from Fig. 21.8 that the peak miss distance is close to a maximum when the normalized target maneuver frequency is near unity. Large weave frequencies do not cause much miss distance because very little target displacement is created. On the other hand, small weave frequencies look like step target maneuvers and thus in the steady-state (large flight times) cause very little miss distance. If we were on a collision triangle with the target (i.e., no heading error) and we coasted to the target by turning off the guidance (i.e., $N' = 0$), the peak miss distance would simply be the peak displacement n_T / ω_T^2 caused by the weaving target. Superimposed on Fig. 21.8 is the peak displacement or induced miss distance with no missile guidance (i.e., $N' = 0$) caused by the weaving target. We can see that for the single time constant guidance system, guiding with proportional navigation always yields a smaller miss against a weaving target than coasting without guidance. *However, for large values of normalized weave frequency the miss distance with and without guidance is approximately the same!*

To illustrate the use of the normalized miss distance curves of Fig. 21.8, let us consider a numerical example in which there is a 6-g weaving target with a weave frequency of 2 rad/s. Assuming that the missile guidance system time constant is 1 s and effective navigation ratio is 3, we first compute the normalized weave

frequency as

$$\omega_T T = 2 * 1 = 2$$

which results in a normalized miss of approximately 5.5. Therefore from the ordinate of Fig. 21.8 we can compute the peak steady-state miss distance to be

$$\text{Peak Miss} \approx 5.5 n_T T^2 = 5.5 * 6 * 1^2 = 33 \text{ ft}$$

Reducing the guidance system time constant to 0.5 s changes both the normalized weave frequency and the normalized miss. The new normalized weave frequency is

$$\omega_T T = 2 * 0.5 = 1$$

which results in an increased normalized miss of approximately 11.5. However, the new peak steady-state miss distance is reduced because the guidance system time constant has been halved or

$$\text{Peak Miss} \approx 11.5 n_T T^2 = 11.5 * 6 * 0.5^2 \approx 17 \text{ ft}$$

Keeping the guidance system constant fixed to 0.5 s but increasing the target weave frequency to 4 rad/s increases the normalized weave frequency back to 2 or

$$\omega_T T = 4 * 0.5 = 2$$

which again results in a normalized miss of approximately 5.5. The new peak steady-state miss distance becomes

$$\text{Peak Miss} \approx 5.5, n_T T^2 = 5.5 * 6 * 0.5^2 \approx 8 \text{ ft}$$

Thus we can see that both the guidance system time constant and target weave frequency are important factors in determining the peak steady-state miss distance.

Higher-Order Guidance System Dynamics

The single time constant guidance system model, used in the previous section, was useful because it could be used to derive closed-form solutions for the miss distance due to a weave maneuver. The single time constant guidance system miss distance formulas also suggest normalization factors for the miss distance. We have already shown in Chapters 6 and 20 that the disadvantage of the single time constant representation of a missile guidance system is that the miss distance can be seriously underestimated. We have seen that a much better and equally convenient representation of a proportional navigation missile guidance system transfer function is a canonic fifth-order binomial given by

$$\frac{n_L}{\lambda} = \frac{N' V_c s}{(1 + sT/5)^5}$$

where T is the total guidance system time constant, n_L is the achieved missile acceleration, and λ is the line-of-sight angle. As was mentioned in Chapters 6 and 20 for this generic interceptor guidance system model, one time constant represents the seeker, another represents the noise filter, and the three other time constants represent the flight-control system dynamics (aerodynamics plus autopilot). It is easy to show that with this canonic guidance system model, the overall guidance system

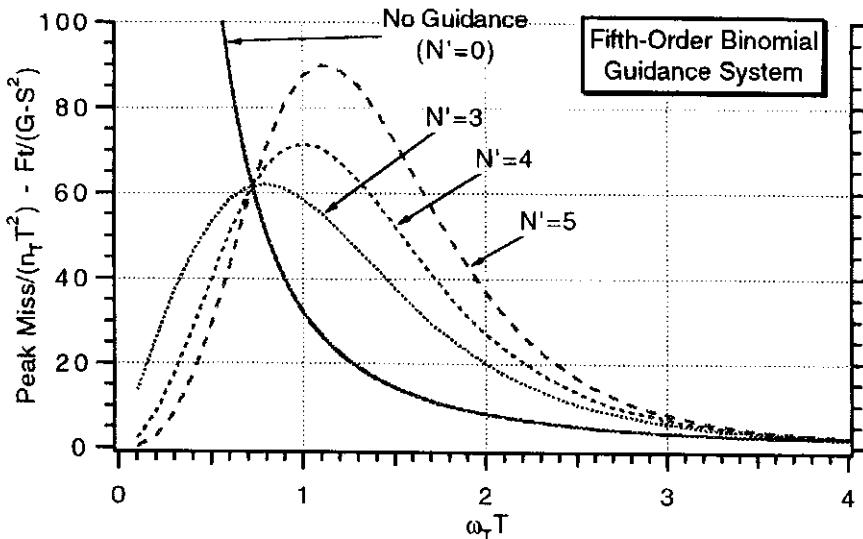


Fig. 21.9 Steady-state peak miss due to weave maneuver is much larger with fifth-order binomial guidance system.

time constant is simply the sum of the five individual time constants or T . The peak steady-state miss distance due to a weaving target for the fifth-order binomial missile homing loop can either be evaluated using the method of adjoints or the method of brute force. Because the adjoint simulation would have to be extensively modified to figure out when steady-state was reached and special logic would then have to be developed to capture the maximum miss distance, it was considered easier to use the brute force approach. In addition, the brute force approach can easily be extended to the case where there are significant nonlinearities whereas the adjoint method would no longer be valid. Listing 21.3 presents the brute force simulation based on linearized geometry, which we have already shown to be valid for the weaving target case. We can see from Listing 21.3 that acceleration saturation effects can be included by simply reducing the value of the acceleration limit XNCLIM from its near infinite value. The listing shows how the time constant is reduced when the flight times are short to ensure that we are in steady-state. In addition, special logic after statement label 999 is used to capture the maximum steady-state miss distances. The simulation is set up to generate normalized miss distance curves as a function of the normalized target weave frequency.

Figure 21.9 shows how the steady-state normalized peak miss distance due to a weave maneuver varies with the normalized target weave frequency for the fifth-order binomial guidance system. The curves in this figure are similar in shape to the ones of Fig. 21.8, but as expected, the normalized miss distances are much larger. It is interesting to note the steady-state peak miss distance is still maximum when the normalized target weave frequency is approximately unity. Superimposed on Fig. 21.9 is the zero guidance miss distance or peak displacement n_T / ω_T^2 caused by the weaving target. Surprisingly, we can see that for the fifth-order guidance system, proportional navigation only yields a smaller miss than coasting (i.e., $N' = 0$) when the normalized weave frequency is less than 0.7 (i.e., $\omega_T T < 0.7$). *In other words, for normalized weave frequencies greater than 0.7, the weaving target nullifies the effectiveness of a proportional navigation guidance system!*

To illustrate the use of the normalized miss distance curves of Fig. 21.9, let us reconsider the numerical example of the previous section in which there is a

**Listing 21.3 Brute force simulation for generating normalized design curves
against weaving target**

```

VC=4000.
XNT=32.2
XNP=3.
XNCLIM=99999.

WRITE(*,*)'
WRITE(*,*)"TARGET ACCELERATION AND ACC LIMIT"
WRITE(*,*)"XNT= ",XNT,'FT/SEC2','XNCLIM=',XNCLIM,'FT/SEC2'
WRITE(*,*)'
READ(*,*)XNT,XNCLIM

WRITE(*,*)'
WRITE(*,*)"EFFECTIVE NAVIGATION RATIO"
WRITE(*,*)"XNP= ",XNP,'NO UNIT'
WRITE(*,*)'
READ(*,*)XNP

OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
DO 30 X=.1,4.,1
IF(X < .5)THEN
    W=1.
    TAU=X/W
ELSE
    W=X
    TAU=1.
ENDIF
XMWEAVEOLD=0.
XMWEAVEMAX=0.
DO 20 TF=.2,20.,2
PHASE=0.
Y=0.
YD=0.
XNL=0.
D=0.
ELAMDH=0.
X4=0.
X5=0.
T=0.
H=.01
S=0.
10 IF(T>(TF-.0001))GOTO 999
YOLD=Y
YDOLD=YD
XNLOLD=XNL
DOLD=D
ELAMDHOLD=ELAMDH
X4OLD=X4
X5OLD=X5
STEP=1

```

(Contd.)

Listing 21.3 (Continued)

```

GOTO 200
66 STEP=2
Y=Y+H*YD
YD=YD+H*YDD
XNL=XNL+H*XNLD
ELAMDH=ELAMDH+H*ELAMDHD
D=D+H*DD
X4=X4+H*X4D
X5=X5+H*X5D
T=T+H
GOTO 200
55 CONTINUE
Y=.5*(YOLD+Y+H*YD)
YD=.5*(YDOLD+YD+H*YDD)
XNL=.5*(XNLOLD+XNL+H*XNLD)
D=.5*(DOLD+D+H*DD)
ELAMDH=.5*(ELAMDHOLD+ELAMDH+H*ELAMDHD)
X4=.5*(X4OLD+X4+H*X4D)
X5=.5*(X5OLD+X5+H*X5D)
S=S+H
GOTO 10
200 CONTINUE
YTDD=XNT*SIN(W*T)
TGO=TF-T+.00001
XLAM=Y/(VC*TGO)
DD=5.*(XLAM-D)/TAU
ELAMDHD=5.*(DD-ELAMDH)/TAU
XNC=XNP*VC*ELAMDH
IF(XNC>XNCLIM)XNC=XNCLIM
IF(XNC<-XNCLIM)XNC=-XNCLIM
X4D=5.*(XNC-X4)/TAU
X5D=5.*(X4-X5)/TAU
XNLD=5.*(X5-XNL)/TAU
YDD=YTDD-XNL
IF(STEP-1)66,66,55
999 CONTINUE
XMWEAVE=Y
IF(XMWEAVE>XMWEAVEOLD.AND.XMWEAVE>XMWEAVEMAX
.AND.TF>10.)THEN
    XMWEAVEMAX=XMWEAVE
ENDIF
XMWEAVEOLD=XMWEAVE
20 CONTINUE
IF(X<.5)THEN
    XMWEAVEMAX=XMWEAVEMAX/TAU**2
ENDIF
WRITE(*,97)X,XMWEAVEMAX
WRITE(1,97)X,XMWEAVEMAX
30 CONTINUE
97 FORMAT(2F10.3)
CLOSE(1)
PAUSE
END

```

6-g weaving target with a weave frequency of 2 rad/s. Assuming that the missile guidance system time constant is 1 s and effective navigation ratio is 3, we first compute the normalized weave frequency as

$$\omega_T T = 2 * 1 = 2$$

which results in a normalized miss of approximately 20. Therefore we can compute the peak steady-state miss distance to be

$$\text{Peak Miss} \approx 20 n_T T^2 = 20 * 6 * 1^2 = 120 \text{ ft}$$

which is four times larger than the miss in a single time constant guidance system (i.e., 120 ft vs 33 ft). Reducing the guidance system time constant to 0.5 s changes both the normalized weave frequency and the normalized miss. The new normalized weave frequency is

$$\omega_T T = 2 * 0.5 = 1$$

which results in an increased normalized miss of approximately 60. The new peak steady-state miss distance becomes

$$\text{Peak Miss} \approx 60 n_T T^2 = 60 * 6 * 0.5^2 \approx 90 \text{ ft}$$

which is five times larger than the miss in a single time constant guidance system (i.e., 90 ft vs 17 ft). Keeping the guidance system constant fixed to 0.5 s but increasing the weave frequency to 4 rad/s increases the normalized weave frequency back to 2 or

$$\omega_T T = 4 * 0.5 = 2$$

which again results in a normalized miss of approximately 20. The new peak steady-state miss distance becomes

$$\text{Peak Miss} \approx 20 n_T T^2 = 20 * 6 * 0.5^2 \approx 30 \text{ ft}$$

which is approximately four times larger than the miss induced with a single time constant guidance system (i.e., 30 ft vs 8 ft). Thus we can see that the higher-order guidance system dynamics of the fifth-order binomial guidance system yield much larger miss distances due to a weaving target than does the single time constant representation of the guidance system.

Acceleration Saturation

We have observed in the preceding two sections that both the guidance system dynamics and effective navigation ratio play an important role in determining the miss distance due to a weaving target. The finite acceleration capability of the interceptor is also important in determining the miss distance. Normalized miss distance curves can also be developed when missile acceleration saturation effects are considered. In this case it is hypothesized that miss distance normalization factors remain unchanged but new curves have to be developed for the nondimensional ratio of the missile to target acceleration advantage or

$$\text{Ratio} = n_{\text{LIM}} / n_T$$

where n_{LIM} is the interceptor acceleration limit.

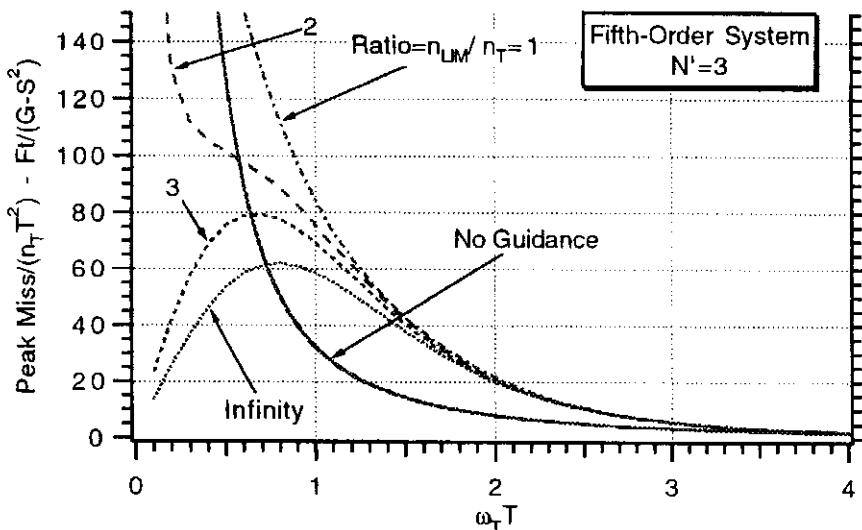


Fig. 21.10 Normalized steady-state peak miss due to weaving target and saturation effects for an effective navigation ratio of three.

Using the preceding ratio and the normalization factors for the steady-state peak miss due to a weaving target, we can derive normalized miss distance curves by the method of brute force with Listing 21.3. In other words, we can generate normalized miss distance curves by simulating all of the possibilities. We can then infer performance by making extrapolations from the normalized miss distance curves. Of course, detailed checks have to be made to ensure that the normalization factors are correct. Figures 21.10–21.12 present the normalized steady-state peak miss distances due to a weaving target for effective navigation ratios ranging from three to five respectively. As expected, we can see that less missile acceleration capability (smaller ratio) means larger miss distances. We can see that at the larger effective navigation ratios (i.e., $N' = 5$), increasing the missile acceleration capability may not always reduce the miss (i.e., $\omega_T T = 2$). Under these circumstances the weaving target is causing proportional navigation to be ineffective. This should not be surprising since we know that when the normalized weave frequency is greater than 0.7, doing nothing or $n_{LIM}/n_T = 0$ is optimal.

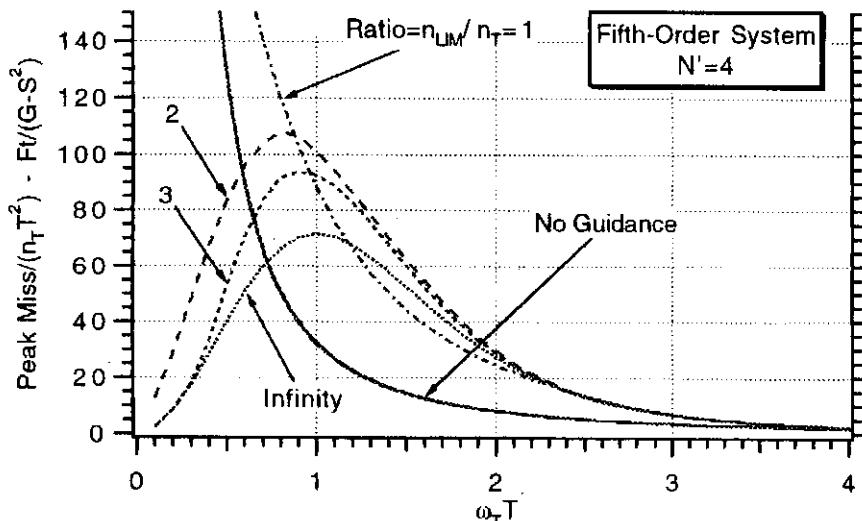


Fig. 21.11 Normalized steady-state peak miss due to weaving target and saturation effects for an effective navigation ratio of four.

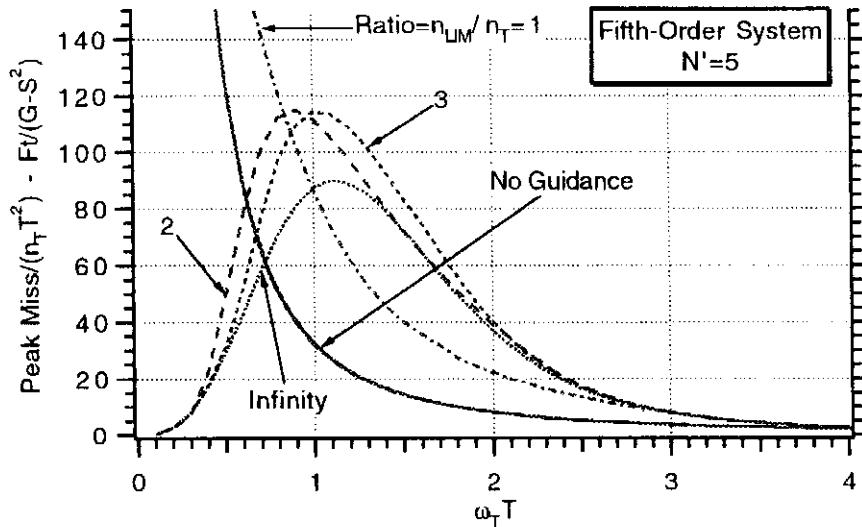


Fig. 21.12 Normalized steady-state peak miss due to weaving target and saturation effects for an effective navigation ratio of five.

To demonstrate the use of the normalized curves of Figs. 21.10–21.12 let us again consider the same example of the previous section in which there was a 6-g target weave maneuver with weave frequency of 2 rad/s and a proportional navigation missile guidance system with overall time constant of 0.5 s and effective navigation ratio of 3 (i.e., $n_T = 6$, $\omega_T = 2$, $T = 0.5$, $N' = 3$). In this case the normalized weave frequency is 1 ($\omega_T T = 2 * 0.5 = 1$). If the missile acceleration limit is infinite, then the ratio is infinite and we can read from Fig. 21.8 that the steady-state peak miss is 90 ft or

$$\text{Peak Miss}_g = 60 n_T T^2 = 60 * 6 * 0.5^2 = 90 \text{ ft}$$

Reducing the acceleration limit to 18 g reduces the ratio to 3 or

$$\text{Ratio} = n_{\text{LIM}}/n_T = 18/6 = 3$$

For a normalized weave frequency of 1, the new steady-state peak miss increases to 105 ft or

$$\text{Peak Miss}_{18 \text{ g}} = 70 n_T T^2 = 70 * 6 * 0.5^2 = 105 \text{ ft}$$

Reducing the acceleration limit further to 12 g reduces the ratio to 2 or

$$\text{Ratio} = n_{\text{LIM}}/n_T = 12/6 = 2$$

For a normalized weave frequency of 1, the new steady-state peak miss increases to 113 ft or

$$\text{Peak Miss}_{12 \text{ g}} = 75 n_T T^2 = 75 * 6 * 0.5^2 = 113 \text{ ft}$$

Finally reducing the acceleration limit even further to 6 g reduces the ratio to 12 or

$$\text{Ratio} = n_{\text{LIM}}/n_T = 6/6 = 1$$

For a normalized weave frequency of 1, the new steady-state peak miss increases to 128 ft or

$$\text{Peak Miss}_{6g} = 85 n_T T^2 = 75 * 6 * 0.5^2 = 128 \text{ ft}$$

In this example, if the missile had no acceleration capability or if the guidance system was turned off, the peak miss would be the maximum value of the weave displacement n_T / ω_T^2 or only 48.4 ft.

If the target weave frequency were increased to 4 rad/s and everything else remained the same, the new normalized weave frequency would be doubled to 2 ($\omega_T T = 4 * 0.5 = 2$). In this case we can see from Fig. 21.10 that the miss is independent of the missile to target acceleration advantage and that the miss would reduce to 30 ft or

$$\text{Peak Miss}_{\infty \sigma, 18 G, 12 G, 6 G} = 20 n_T T^2 = 20 * 6 * 0.5^2 = 30 \text{ ft}$$

Again, turning the guidance system off would make the peak miss equivalent to the maximum value of the weave displacement n_T / ω_T^2 or only 12.1 ft.

Reducing the Time Constant to Improve Performance

In general, the safest and most effective method for improving the performance of a proportional navigation guidance system against the weaving target is to reduce the overall guidance system time constant and to increase the missile to target acceleration advantage. In aerodynamically controlled missiles, the major contributor to the guidance system time constant is usually the flight-control system time constant, and the limitation on missile acceleration capability is a function of the maximum angle of attack in which a missile can operate without causing flight catastrophe. The ability to speed up the missile flight-control system and the challenge in increasing the missile's maneuverability depends on advances in flight-control system technology. Radome effects will set a lower limit on how small the missile flight-control system time constant can be made without causing stability problems⁶ and flight-control system pitch-yaw-roll cross-coupling will place an upper limit on maximum permissible angle of attack. Although a thorough discussion of the challenges in speeding up a flight-control system and safely achieving high angles of attack are beyond the scope of this text, two numerical examples will be presented in this section showing the benefits to system performance if these goals can be met.

To illustrate the importance of reducing the guidance system time constant, a nonnormalized, non-steady-state example was chosen in which there was a 6-g weaving target with a weave frequency of 2 rad/s. Figure 21.13 shows that the miss distance induced by a weaving target on a fifth-order binomial proportional navigation guidance system dramatically decreases with decreasing guidance system time constant. In fact, when the guidance system time constant is 0.1 s there is virtually no miss due to the weaving target!

If we fix the guidance system time constant at 0.1 s, we can see from Fig. 21.14 that although increasing the target weave frequency increases the miss, the miss is still small. We also could have calculated the maximum peak steady-state miss in this example from the normalized curves of Fig. 21.9. For an effective navigation ratio of 3, the curve of Fig. 21.9 is a maximum when the normalized weave frequency is 0.7. That means for this example the actual target weave frequency

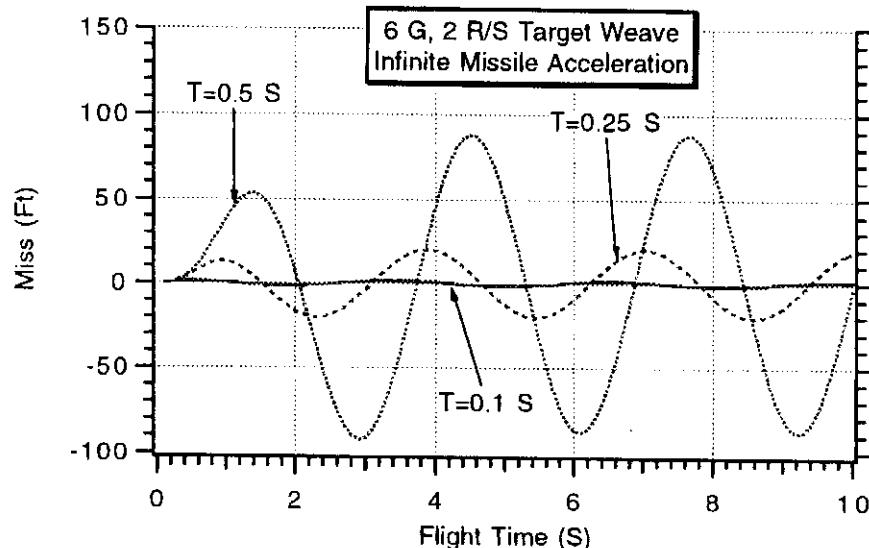


Fig. 21.13 Reducing guidance system time constant dramatically reduces miss.

is 7 rad/s (i.e., $\omega_T T = 7 * 0.1 = 0.7$). From Fig. 21.9 we can see that the actual maximum peak miss is approximately 4 ft or

$$\text{Peak Miss} \Big|_{N'=3} = 63 n_T T^2 = 63 * 6 * 0.1 \approx 4 \text{ ft}$$

Thus we can see that a very small miss distance can be achieved against this difficult maneuver if the guidance system time constant can be reduced to 0.1 s. Of course we can also see from Fig. 21.9 that turning the guidance system off would also yield the same miss.

Both previous examples assumed that the missile had infinite acceleration capability. Figure 21.15 shows that when the missile to target acceleration advantage decreases from infinity to only two the miss increases. However, since the guidance system time constant is small the maximum miss distance is not large. Of course, if the missile guidance system were turned off, the miss would only be approximately 4 ft regardless of acceleration limit.

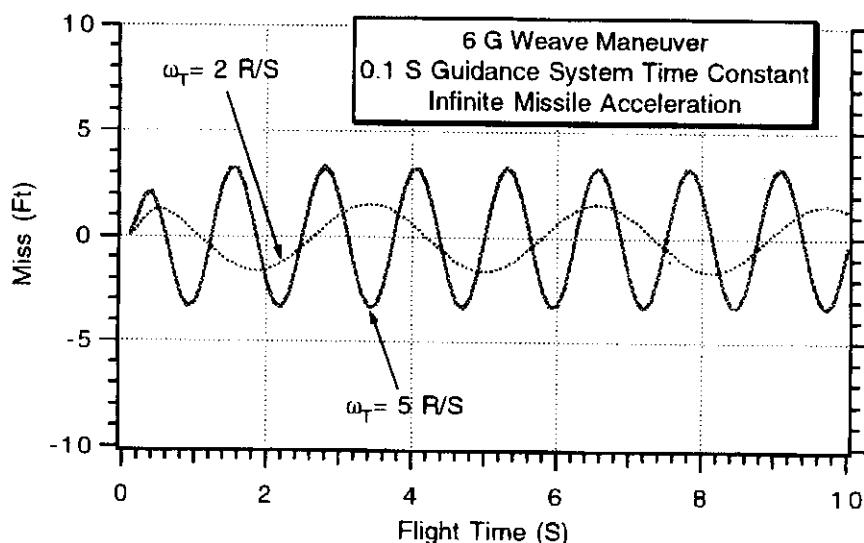


Fig. 21.14 Small guidance time constant yields good performance even when weave frequency increases.

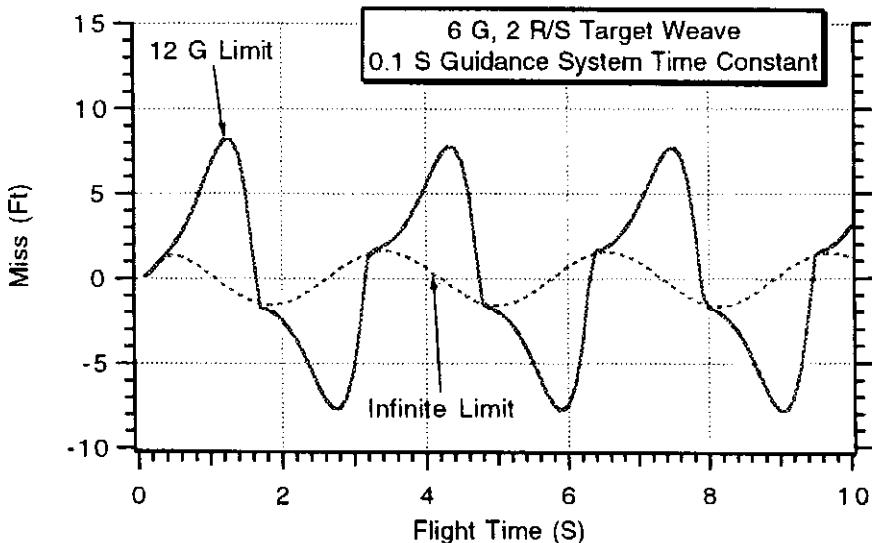


Fig. 21.15 Small miss distances can be achieved even when there is only 2 to 1 acceleration advantage.

Advanced Guidance Techniques to Improve Performance

Traditional guidance laws are a form of proportional navigation (PN) in which the acceleration command is proportional to the measured line-of-sight rate. As we saw in Chapter 2, proportional navigation can also be thought of as a guidance law in which the acceleration command is proportional to the zero effort miss and inversely proportional to the square of the time to go until intercept or

$$n_c = \frac{N'}{t_{go}^2} [y + \dot{y} t_{go}] = N' V_c \dot{\lambda}$$

The zero effort miss can be thought of as a prediction of how much the missile would miss the target by if the target continued to perform as it had done in the past and the missile issued no further acceleration commands (zero effort). We can see from the preceding equation that the zero effort miss term (bracketed quantity) in proportional navigation assumes that the target is not maneuvering. This does not mean that proportional navigation cannot hit a maneuvering target; it just means that this guidance law is not optimal in the sense that it requires the least acceleration when the target is maneuvering.

If it is known that the target is maneuvering in a step-wise fashion, we saw in Chapter 8 that the zero effort miss could be calculated exactly and a new guidance law result, known as augmented proportional navigation (APN). Mathematically this means that the zero effort miss has an acceleration term based on a constant maneuver or

$$n_c = \frac{N'}{t_{go}^2} [y + \dot{y} t_{go} + 0.5 t_{go}^2 \ddot{y}_T] = N' V_c \dot{\lambda} + 0.5 N' \ddot{y}_T$$

Although augmented proportional navigation can hit targets maneuvering in different ways (i.e., not step maneuvers), it is only optimal for the step target maneuver in the sense that it requires the least acceleration. From an implementation point of view augmented proportional navigation has one term proportional to the line-of-sight rate and another term proportional to the target acceleration. Therefore when

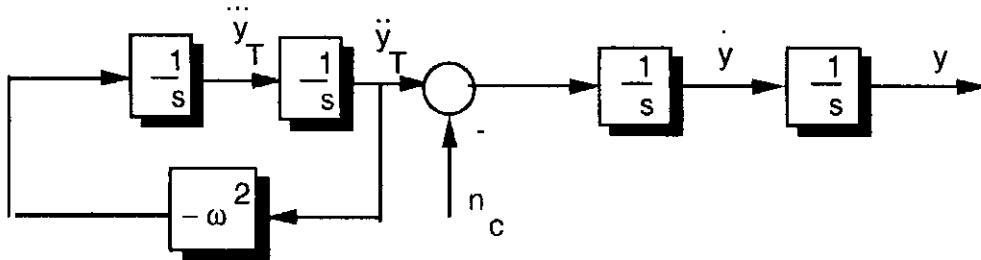


Fig. 21.16 Model for weave guidance law derivation.

augmented proportional navigation is implemented, a special filter is required to provide an estimate of both the line-of-sight rate and the instantaneous value of the target acceleration.

We can also derive a special guidance law if it is known in advance that the target is weaving.^{7,8} In this case a simple model, similar to the ones of Chapter 8 for guidance law development, is shown in Fig. 21.16. The second-order shaping network shown in Fig. 21.16 represents the weaving or sinusoidal target maneuver (i.e., see Chapter 1). As was the case for other guidance laws, we are still trying to derive a guidance law that will yield zero miss distance and at the same time minimize the integral of the acceleration squared or

$$y(t_F) = 0 \quad \text{subject to minimizing} \quad \int_0^{t_F} n_c^2(t) dt$$

Figure 21.16 can be expressed in state-space form as

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{y}_T \\ \ddot{y}_T \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \dot{y}_T \\ \ddot{y}_T \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} n_c$$

As was the case in Chapter 8, if a system is expressed in state-space form we can also express the final state of the system at any time according to

$$x(t_F) = \Phi(t_F - t)x(t) + \int_t^{t_F} \Phi(t_F - \lambda)G(\lambda)u(\lambda) d\lambda$$

where $x(t)$ is the system state vector and $\Phi(t)$ is the fundamental matrix. Because the systems dynamics matrix F in this example is time-invariant, the fundamental matrix can be found directly from F according to

$$\Phi(t) = \mathcal{L}^{-1}[(sI - F)^{-1}]$$

yielding

$$\Phi(t) = \begin{bmatrix} 1 & \frac{(1 - \cos \omega t)}{\omega^2} & \frac{(\omega t - \sin \omega t)}{\omega^3} \\ 0 & \frac{\sin \omega t}{\omega} & \frac{(1 - \cos \omega t)}{\omega^2} \\ 0 & \cos \omega t & \frac{\sin \omega t}{\omega} \\ 0 & -\omega \sin \omega t & \cos \omega t \end{bmatrix}$$

Substitution of the Φ and G matrices into the matrix expression for the final state yields four scalar equations in this example. The first of these scalar equations is given by

$$\begin{aligned} y(t_F) = & y(t) + (t_F - t)\dot{y}(t) + \frac{[1 - \cos \omega(t_F - t)]}{\omega^2} \ddot{y}_T(t) \\ & + \frac{[\omega(t_F - t) - \sin \omega(t_F - t)]}{\omega^3} \ddot{\ddot{y}}_T(t) - \int_t^{t_F} (t_F - \lambda)n_c(\lambda) d\lambda \end{aligned}$$

We can use the same shorthand notation of Chapter 8 and define f_1 and h_1 as

$$\begin{aligned} f_1(t_F - t) = & y(t) + (t_F - t)y(t) + \frac{[1 - \cos \omega(t_F - t)]}{\omega^2} \ddot{y}_T(t) \\ & + \frac{[\omega(t_F - t) - \sin \omega(t_F - t)]}{\omega^3} \ddot{\ddot{y}}_T(t) \end{aligned}$$

and

$$h_1(t_F - \lambda) = t_F - \lambda$$

so that we can say that

$$y(t_F) = f_1 - \int_t^{t_F} h_1(t_F - \lambda)n_c(\lambda) d\lambda$$

In Chapter 8 we showed via the Schwartz inequality that the general form of the resultant optimal guidance law based on the preceding formulation is given by

$$n_c(\lambda) = kh_1(t_F - \lambda)$$

where

$$k = f_1(t_F - t) \Big/ \int_t^{t_F} h_1^2(t_F - \lambda) d\lambda$$

After some algebra we find that the optimal weave guidance law is given by

$$\begin{aligned} n_c = & \frac{3}{t_{go}^2} \left[y + \dot{y}t_{go} + \frac{1 - \cos \omega t_{go}}{\omega^2} \ddot{y}_T + \frac{\omega t_{go} - \sin \omega t_{go}}{\omega^3} \ddot{\ddot{y}}_T \right] \\ = & 3V_c \dot{\lambda} + \frac{3}{t_{go}^2} \left[\frac{1 - \cos \omega t_{go}}{\omega^2} \right] \ddot{y}_T + \frac{3}{t_{go}^2} \left[\frac{\omega t_{go} - \sin \omega t_{go}}{\omega^3} \right] \ddot{\ddot{y}}_T \end{aligned}$$

We can see that the weave guidance law is similar to other optimal laws we have derived in Chapter 8 in that guidance commands are still proportional to the zero effort miss and inversely proportional to the square of time to go until intercept. From an implementation point of view, assuming that the target weave frequency can be estimated off line and the time to go until intercept is measured, the weave guidance law consists of three terms: one term proportional to the line-of-sight rate, another term proportional to the target acceleration, and a third term proportional to target jerk.

To better understand the relationship between the new guidance law and its predecessors, let us consider the case in which the target weave frequency approaches zero. One can show using Taylor series approximations that the weave guidance law at zero frequency simplifies to

$$\lim_{\omega \rightarrow 0} n_{c\text{weave}} = \frac{3}{t_{go}^2} \left[y + \dot{y}t_{go} + \frac{t_{go}^2}{2} \ddot{y}_T + \frac{t_{go}^3}{6} \dddot{y}_T \right]$$

which is simply augmented proportional navigation with an effective navigation ratio of 3 plus an extra term to account for target jerk. The bracketed term can be recognized as the Taylor series expansion for the zero effort miss for constant target jerk.

It is important to note that the new guidance law requires additional information—an estimate of the target weave frequency, target jerk, and the time to go until intercept. A four-state Kalman filter similar to the three-state filter of Chapter 9 can be used to provide estimates of the target acceleration and jerk. The target weave frequency estimate can be derived from either the homing sensors measurements using an extended Kalman filter or from an external sensor (i.e., ground radar).

Because we have already demonstrated that dynamics within the guidance system will cause miss distance, the preceding guidance law must be modified to account for guidance system lags. With endoatmospheric interceptors, the flight-control system dynamics constitute the bulk of the overall guidance system time constant. If it is known that the target maneuver is sinusoidal in nature, the weave guidance can be modified to compensate for the known dynamics of the interceptor flight-control system. The compensated weave guidance law^{7,8} is very similar to the optimal guidance law derived in Chapter 8 for a single time constant guidance system, which can be expressed as

$$n_{c\text{weave}} = \frac{N'}{t_{go}^2} \left[y + \dot{y}t_{go} + \frac{1 - \cos \omega t_{go}}{\omega^2} \ddot{y}_T + \frac{\omega t_{go} - \sin \omega t_{go}}{\omega^3} \dddot{y}_T - n_L T^2 (e^{-x} + x - 1) \right]$$

where x is given by

$$x = \frac{t_{go}}{T}$$

with t_{go} being the time to go until intercept and T being defined as the approximate time constant of the flight control system. The effective navigation ratio in the compensated weave guidance law is now time-varying and is given by

$$N' = \frac{6x^2(e^{-x} - 1 + x)}{2x^3 + 3 + 6x - 6x^2 - 12xe^{-x} - 3e^{-2x}}$$

The effective navigation ratio for the compensated weave guidance law is identical to the effective navigation ratio of the optimal guidance law of Chapter 8 and is displayed in normalized form in Fig. 21.17. We can see that at the beginning of the flight (long time to go before intercept) the effective navigation ratio is

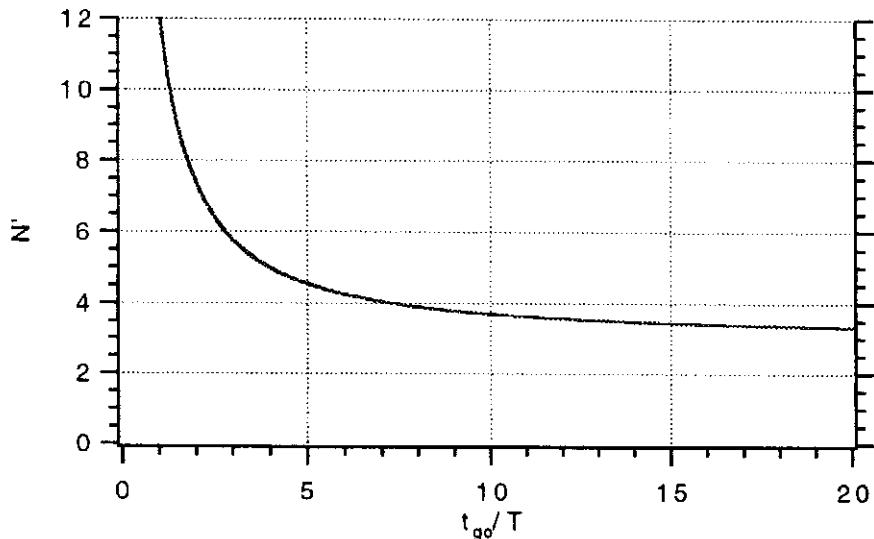


Fig. 21.17 Normalized effective navigation ratio for compensated weave guidance law.

approximately constant and is approaching three. As we get closer to intercept (small time to go), the effective navigation ratio grows considerably.

To test the effectiveness of the weave and compensated weave guidance laws, the brute force simulation of Listing 21.2 was modified to include various guidance law options (i.e., after statement label 200). We can see from Listing 21.4 that the guidance system under consideration is a fifth-order binomial even though the compensated weave guidance law assumes a single time constant guidance system. In other words, the compensated weave guidance is actually suboptimal in Listing 21.4

Figure 21.18 shows that proportional navigation can have substantial miss distances against a 6-g, 2 rad/s weaving target in a fifth-order binomial guidance system with a time constant of 0.25 s. We can also see that uncompensated weave guidance (guidance lags are not accounted for) can substantially reduce the miss.

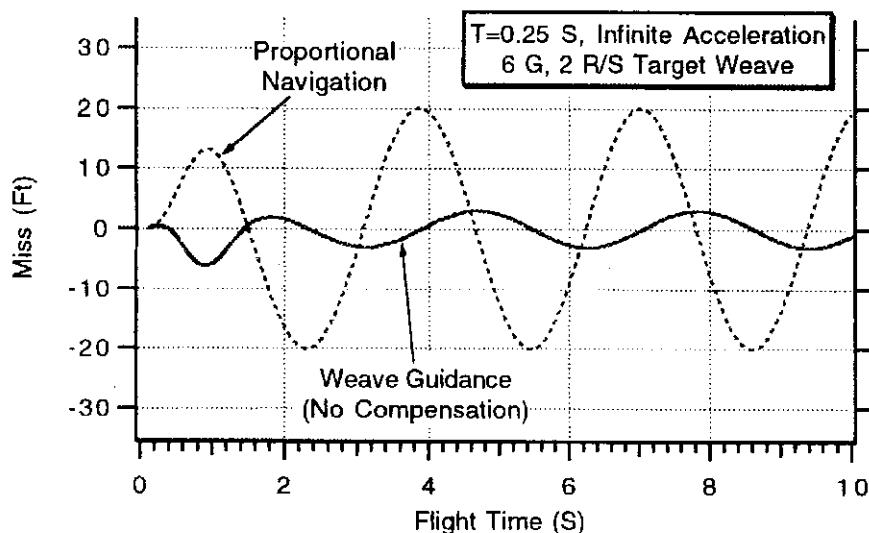


Fig. 21.18 Time lags cause miss—even for uncompensated weave guidance.

**Listing 21.4 Brute force simulation for guidance
law evaluation against weaving target**

```

INTEGER APN
VC=4000.
XNT=193.2
XNP=3.
XNCLIM=99999999.
TAU=.25
W=2.
WH=2.
APN=1
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
DO 20 TF=.1,10.,1
Y=0.
YD=0.
XNL=0.
D=0.
ELAMDH=0.
X4=0.
X5=0.
T=0.
H=.01
S=0.
10 IF(T>(TF-.0001))GOTO 999
YOLD=Y
YDOLD=YD
XNLOLD=XNL
DOLD=D
ELAMDHOLD=ELAMDH
X4OLD=X4
X5OLD=X5
STEP=1
GOTO 200
66 STEP=2
Y=Y+H*YD
YD=YD+H*YDD
XNL=XNL+H*XNLD
ELAMDH=ELAMDH+H*ELAMDHD
D=D+H*DD
X4=X4+H*X4D
X5=X5+H*X5D
T=T+H
GOTO 200
55 CONTINUE
Y=.5*(YOLD+Y+H*YD)
YD=.5*(YDOLD+YD+H*YDD)
XNL=.5*(XNLOLD+XNL+H*XNLD)
D=.5*(DOLD+D+H*DD)
ELAMDH=.5*(ELAMDHOLD+ELAMDH+H*ELAMDHD)
X4=.5*(X4OLD+X4+H*X4D)

```

(Contd.)

Listing 21.4 (Continued)

```

X5=.5*(X5OLD+X5+H*X5D)
S=S+H
GOTO 10
200 CONTINUE
YTDD=XNT*SIN(W*T)
YTDDD=W*XNT*COS(W*T)
TGO=TF-T+.00001
XLAM=Y/(VC*TGO)
DD=5.*(XLAM-D)/TAU
ELAMDHD=5.*(DD-ELAMDH)/TAU
IF(APN.EQ.1)THEN
    XNC=XNP*VC*ELAMDH
ELSEIF(APN.EQ.2.)THEN
    XP=WH*TGO
    XNC=XNP*VC*ELAMDH+XNP*YTDD*(1.-COS(XP))/XP**2+
        XNP*YTDDD*(XP-SIN(XP))/(XP*XP*WH)
1 ELSE
    X=TGO/TAU
    XP=WH*TGO
    TOP=6.*X*X*(EXP(-X)-1.+X)
    BOT1=2*X*X*X+3.+6.*X-6.*X*X
    BOT2=-12.*X*EXP(-X)-3.*EXP(-2.*X)
    XNPP=TOP/(.0001+BOT1+BOT2)
    XNC=XNPP*VC*ELAMDH+XNPP*YTDD*(1.-COS(XP))/XP**2+
1     XNPP*YTDDD*(XP-SIN(XP))/(XP*XP*WH)-
2     XNPP*XNL*TAU*TAU*(EXP(-X)+X-1.)/TGO**2
ENDIF
IF(XNC>XNCLIM)XNC=XNCLIM
IF(XNC < -XNCLIM)XNC=-XNCLIM
X4D=5.*(XNC-X4)/TAU
X5D=5.*(X4-X5)/TAU
XNLD=5.*(X5-XNL)/TAU
YDD=YTDD-XNL
IF(STEP-1)66,66,55
999 CONTINUE
WRITE(*,97)TF,Y
WRITE(1,97)TF,Y
20 CONTINUE
97 FORMAT(2F10.3)
CLOSE(1)
PAUSE
END

```

Although weave guidance has small miss distances compared to proportional navigation, there is still room for improvement when there are significant guidance system lags. Figure 21.19 shows that when there is imperfect compensation (guidance law optimal for single time constant and there are five time constants in guidance system being tested) the new guidance law reduces the miss distance even more.

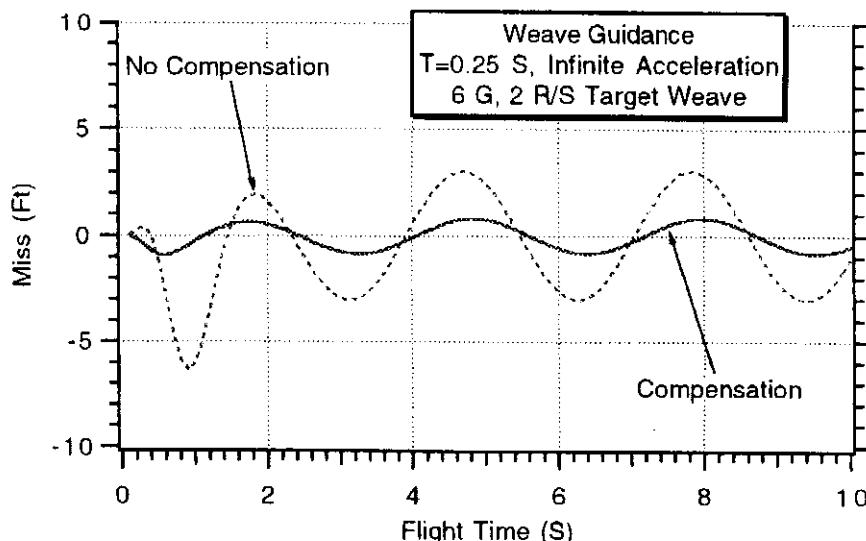


Fig. 21.19 Compensating for guidance system dynamics reduces the miss distance.

Summary

Normalized design curves have been presented showing how a weaving target influences the miss distance of a generic proportional navigation guidance system. This chapter demonstrated how the target weave frequency and amplitude, the missile guidance system time constant, effective navigation ratio and acceleration capability all play an important role in determining system performance. It was demonstrated that, in general, speeding up a missile guidance system and increasing the missile to target acceleration advantage will help reduce the miss distance due to a weaving target. It was also shown how special guidance laws that require more information than proportional navigation can be used to improve system performance.

References

- ¹Platus, D. H., "Ballistic Re-entry Vehicle Flight Dynamics," *Journal of Guidance, Control, and Dynamics*, Vol. 5, Jan.–Feb. 1982, pp. 4–16.
- ²Chadwick, W. R., and Zarchan, P., "Interception of Spiraling Ballistic Missiles," *Proceedings of American Control Conference*, Seattle, WA, June 1995.
- ³Zarchan, P., "Proportional Navigation and Weaving Targets," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 5, 1995, pp. 969–974.
- ⁴Ohlmeyer, E. J., "Root-Mean-Square Miss Distance of Proportional Navigation Missile Against Sinusoidal Target," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 563–568.
- ⁵Javid, M., and Brenner, E., *Analysis, Transmission and Filtering of Signals*, McGraw-Hill, New York, 1963.
- ⁶Nesline, F. W., and Zarchan, P., "Radome Induced Miss Distance in Aerodynamically Controlled Homing Missiles," *Proceedings of AIAA Guidance and Control Conference*, AIAA, New York, Aug. 1984.
- ⁷Forte, I., and Shinar, J., "Can a Mixed Guidance Strategy Improve Missile Performance," *Journal of Guidance, Control, and Dynamics*, Vol. 11, Jan.–Feb. 1988, pp. 53–59.
- ⁸Zarchan, P., "The Challenge of Intercepting Spiraling Tactical Ballistic Missiles," *Proceedings of ION Conference*, Cambridge, MA, June 1996.

Representing Missile Airframe with Transfer Functions

Introduction

So far we have seen that the missile guidance system time constant is extremely important in determining system performance. In aerodynamic missiles the major portion of the total guidance system time constant is governed by the time constant of the flight-control system. In Chapters 6, 20, and 21 for simplicity, we have treated the flight-control system in our fifth-order binomial guidance system as three equal time constants. In this chapter we shall show how the nonlinear force and moment equations determine how the missile airframe responds to control surface deflections. In addition, we shall show one way of linearizing the force and moment equations so that we can find airframe transfer functions. The transfer function representation of the airframe is the necessary first step in the design of the missile autopilot.

In practice both proprietary computer codes and wind tunnel-generated aerodynamic data are used to derive missile airframe transfer functions. However, it is interesting to note that wind tunnel data was not available when the original missiles were designed. Simplified aerodynamic shapes were chosen for the initial designs so that linear theory could be used to derive the necessary airframe transfer functions.

The nonlinear force and moment equations in this chapter are first expressed in terms of physical missile parameters (i.e., weight, length, etc.) rather than in terms of wind tunnel-generated functions. Although this is an approximation to reality, it will enable the reader to get a clearer understanding of how the geometry of the missile influences the force and moment equations. Next the force and moment equations are linearized, and it is shown how the airframe can then be represented by various transfer functions. Finally, using numerical examples, this chapter shows that the transfer function representation of the airframe is an excellent approximation to the nonlinear equations for angles of attack less than 20 to 30 deg.

Force and Moment Equations

A typical tail-controlled, aerodynamic winged missile is shown in Fig. 22.1. This type of missile generates lift by moving control surfaces. In this endoatmospheric missile the movable control surface or tail can be deflected about the hinge line through a fin angle δ in order to help the missile develop an angle of attack α .

The fixed surface or wing plus the missile body help the missile develop additional acceleration. The normal force acts through the center of pressure (CP). We

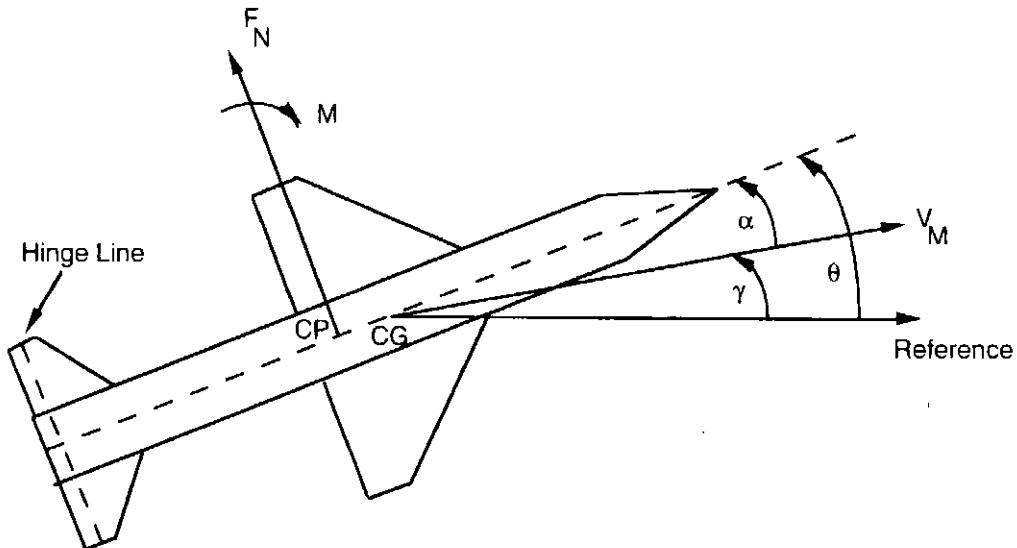


Fig. 22.1 Tail-controlled missile.

can express the normal force equation as

$$F_N = Q S_{\text{ref}} C_N$$

where C_N is the normal force coefficient, Q is the dynamic pressure, and S_{ref} is the reference area. The dynamic pressure and reference area are given by

$$Q = 0.5 \rho V_M^2$$

$$S_{\text{ref}} = \frac{\pi d^2}{4}$$

where ρ is the air density in units of slug/ft³, V_M is the missile velocity in units of ft/s, and d is the missile diameter in units of ft.

The total force acting on the missile body consists of component forces on the body, wing tail, and nose as shown in Fig. 22.2. In this simplified diagram all interference effects are neglected, and the total force is simply the sum of the individual forces. Each of the component forces act through their own centers of pressure. The centers of pressure for the body, wing, and nose are denoted X_{CPB} , X_{CPW} , and X_{CPN} , respectively. The force acting on the tail acts through the hinge line X_{HL} . Neglecting interference effects, the normal force coefficient can be approximated as^{1,2}

$$C_N = 2\alpha + \frac{1.5 S_{\text{PLAN}} \alpha^2}{S_{\text{ref}}} + \frac{8 S_W \alpha}{\beta S_{\text{ref}}} + \frac{8 S_T (\alpha + \delta)}{\beta S_{\text{ref}}}$$

Nose	Body	Wing	Tail
------	------	------	------

where α is the angle of attack, δ is the control surface deflection, and S_W , S_T , and S_{PLAN} are per panel wing, tail, and planform areas, respectively. Because the wing and tail are approximated by trapezoids in Fig. 22.2, their panel areas are given by

$$S_W = 0.5 h_W (C_{TW} + C_{RW})$$

$$S_T = 0.5 h_T (C_{TT} + C_{RT})$$

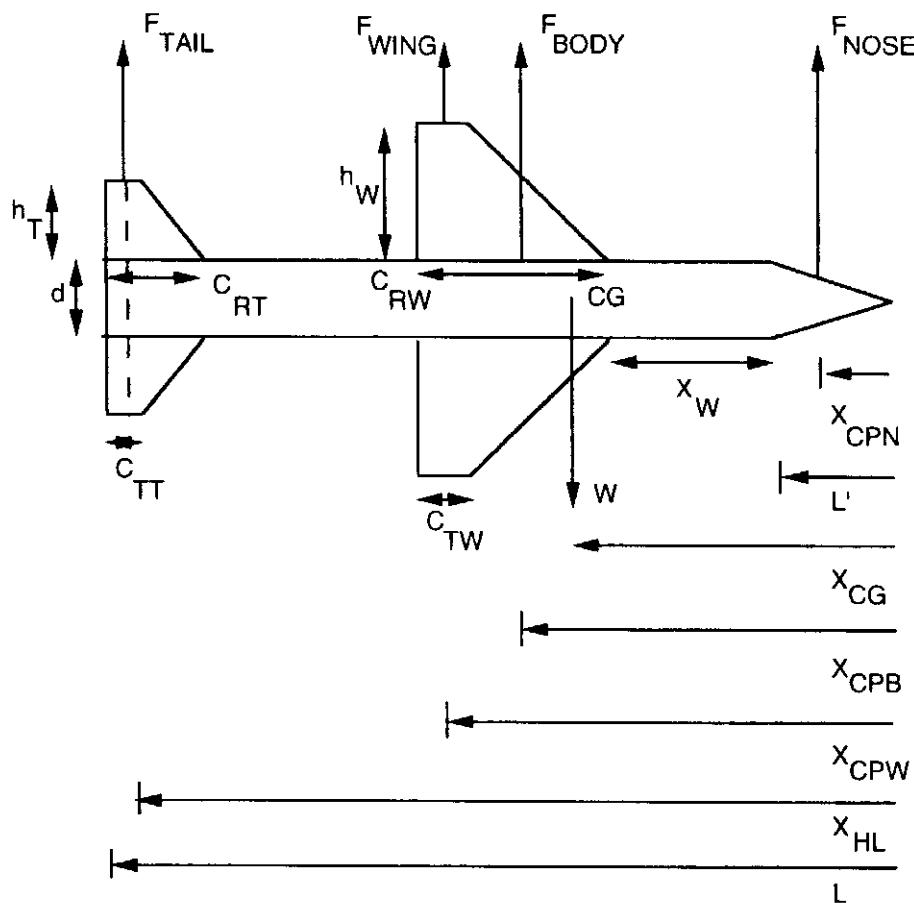


Fig. 22.2 Forces on a tail-controlled missile.

where the subscript T denotes a tip chord and the subscript R denotes a root chord. For a cylindrical missile body with a parabolic nose (radome), the planform area can be approximated as

$$S_{PLAN} = (L - L')d + 0.67L'd \approx Ld$$

where L is the missile length and L' is the radome length. The parameter β in the normal force coefficient equation is a normalized speed and for supersonic travel is given by

$$\beta = \sqrt{\text{Mach}^2 - 1}$$

The missile Mach number is simply the missile speed divided by the speed of sound. Although the speed of sound is altitude dependent, we shall assume for simplicity that the speed of sound is always 1000 ft/s. Multiplying the force by its moment arm yields the developed moment or

$$M = F_N * \text{Moment Arm}$$

Therefore the moment coefficient can be approximated as

$$C_M = 2\alpha(X_{CG} - X_{CPN}) + \frac{1.5S_{PLAN}\alpha^2}{S_{ref}}(X_{CG} - X_{CPB})$$

$$+ \frac{8S_W\alpha}{\beta S_{ref}}(X_{CG} - X_{CPW}) + \frac{8S_T(\alpha + \delta)}{\beta S_{ref}}(X_{CG} - X_{HL})$$

where X_{CG} is the distance from the nose to the missile center of gravity and X_{CPN} , X_{CPB} , and X_{CPW} are the distances from the nose to the centers of pressure for the nose, body, and wing, respectively. The preceding expression assumes a tail-controlled missile and X_{HL} is the distance from the nose to the missile hinge line. The nose, body, and wing centers of pressure (referenced with respect to the nose) can be analytically approximated because of their geometrical shape and can be shown to be³

$$X_{CPN} = 0.67L'$$

$$X_{CPW} = L' + X_W + 0.7C_{RW} - 0.2C_{TW}$$

$$X_{CPB} = \frac{0.67A_N L' + A_B[L' + 0.5(L - L')]}{A_N + A_B}$$

where X_W is the distance from the wing to the radome tangency point as shown in Fig. 22.2. The nose and body areas are given by

$$A_N = 0.67L'd$$

$$A_B = (L - L')d$$

Again for simplicity we will assume that the missile center of gravity is approximately in the center of the missile or

$$X_{CG} = 0.5L$$

The total moment M on the missile can be expressed in terms of the moment coefficient according to

$$M = Q S_{ref} d C_M$$

We now have enough information to express the normal and angular accelerations acting on the missile in terms of the geometry of the missile configuration. The acceleration normal to the missile body can be expressed in terms of the normal force according to

$$n_B = \frac{F_N g}{W} = \frac{g Q S_{ref} C_N}{W}$$

where W is the missile weight. The angular acceleration acting on the missile can be expressed in terms of the moment according to

$$\ddot{\theta} = \frac{M}{I_{yy}} = \frac{Q S_{ref} d C_M}{I_{yy}}$$

where I_{yy} is the missile moment of inertia. If the missile body is approximated as a cylinder, the formula for the moment of inertia is given by⁴

$$I_{yy} = \frac{W[3(0.5d)^2 + L^2]}{12g} \approx \frac{WL^2}{12g}$$

Finally from Fig. 22.1 we can see that the angle of attack can be expressed in terms of the missile body and flight path angles according to

$$\alpha = \theta - \gamma$$

Taking derivatives of both sides of the equation and recognizing that the flight path rate can also be expressed in terms of the missile acceleration yields

$$\dot{\alpha} = \dot{\theta} - \dot{\gamma} = \dot{\theta} - (n_L / V_M)$$

If we assume that the angle of attack is small, the missile acceleration perpendicular to the velocity n_L is approximately the same as the missile acceleration perpendicular to the body n_B . Therefore the derivative of the angle of attack can be expressed as

$$\dot{\alpha} = \dot{\theta} - \dot{\gamma} = \dot{\theta} - (n_B / V_M)$$

In the next three chapters we shall assume that the acceleration perpendicular to the body is approximately the same as the missile acceleration perpendicular to the velocity vector.

Airframe Simulation

We now have enough information to simulate the force and moment equations and thus find out how the missile airframe responds when the tail is deflected. Consider the hypothetical 1000-lb tail-controlled missile, shown in Fig. 22.3, similar to the one first considered by Jerger.² In this example both the wing and tail are triangular in shape. The locations of the wing, hinge line, and center of gravity along with all the other airframe dimensions are indicated in Fig. 22.3.

A simulation was written utilizing the preceding nonlinear force and moment equations for the hypothetical missile of Fig. 22.3. The angle of attack rate and missile angular acceleration differential equations, which involve the force and moment equations, are integrated and the simulation appears in Listing 22.1. We can see from the inputs at the beginning of the simulation that the airframe inputs and data of Fig. 22.3 are consistent. The simulation assumes that the speed of sound is always equal to 1000 ft/s. As was mentioned previously, this is an approximation since the speed of sound is altitude dependent and can be as much as 10% different than the number used in the simulation. We can also see that we are using the exponential approximation to the atmosphere that was first introduced in

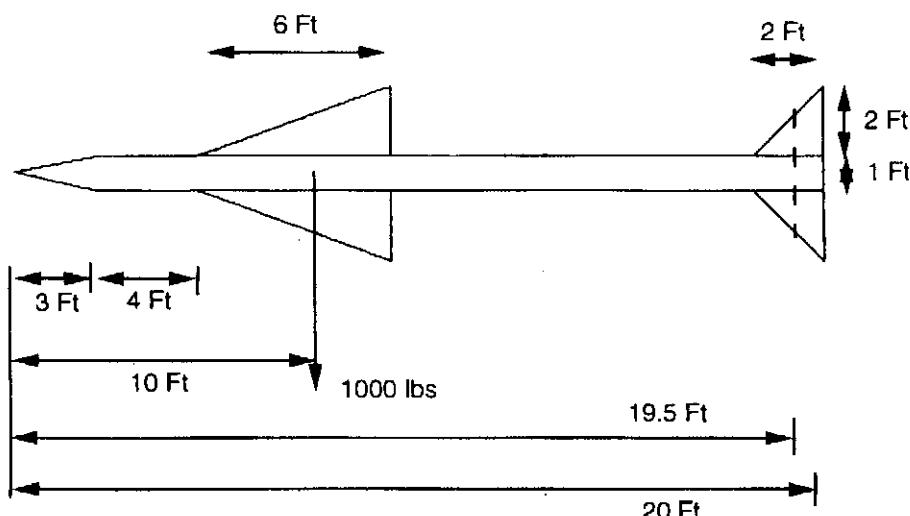


Fig. 22.3 Hypothetical missile.

Listing 22.1 Airframe simulation

```

VM=3000.
DEL=5./57.3
ALT=0.
A=1000.
DIAM=1.
FR=3.
XL=20.
CTW=0.
CRW=6.
HW=2.
CTT=0.
CRT=2.
HT=2.
XN=4.
XCG=10.
XHL=19.5
WGT=1000.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
IF(ALT.LE.30000.)THEN
    RHO=.002378*EXP(-ALT/30000.)
ELSE
    RHO=.0034*EXP(-ALT/22000.)
ENDIF
SWING=.5*HW*(CTW+CRW)
STAIL=.5*HT*(CTT+CRT)
SREF=3.1416*DIAM*DIAM/4.
XLP=FR*DIAM
SPLAN=(XL-XLP)*DIAM+1.33*XLP*DIAM/2.
XCPN=2*XLP/3
AN=.67*XLP*DIAM
AB=(XL-XLP)*DIAM
XCPB=(.67*AN*XLP+AB*(XLP+.5*(XL-XLP)))/(AN+AB)
XCPW=XLP+XN+.7*CRW-.2*CTW
XMACH=VM/A
XIYY=WGT*(3*((DIAM/2)**2)+XL*XL)/(12*32.2)
TMP1=(XCG-XCPW)/DIAM
TMP2=(XCG-XHL)/DIAM
TMP3=(XCG-XCPB)/DIAM
TMP4=(XCG-XCPN)/DIAM
B=SQRT(XMACH**2-1)
Q=.5*RHO*VM*VM
THD=0
ALF=0
T=0
H=.0025
S=0.
10 IF(T>1.99999) GOTO 999
S=S+H
THDOLD=THD

```

(Contd.)

Listing 22.1 (Continued)

```

ALFOLD=ALF
STEP=1
GOTO 200
66 STEP=2
THD=THD+H*THDD
ALF=ALF+H*ALFD
T=T+H
GOTO 200
55 CONTINUE
THD=.5*(THDOLD+THD+H*THDD)
ALF=.5*(ALFOLD+ALF+H*ALFD)
IF(S.LE..0099999)GOTO 10
S=0.
XNLG=XNL/32.2
ALFDEG=ALF*57.3
WRITE(1,97)T,XNLG,ALFDEG
WRITE(*,97)T,XNLG,ALFDEG
GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
CN=2*ALF+1.5*SPLAN*ALF*ALF/SREF+8*SWING*ALF/(B*SREF)
1      +8*STAIL*(ALF+DEL)/(B*SREF)
CM=2*ALF*TMP4+1.5*SPLAN*ALF*ALF*TMP3/SREF
1      +8*SWING*ALF*TMP1/(B*SREF)+8*STAIL*(ALF+DEL)
2      *TMP2/(B*SREF)
THDD=Q*SREF*DIAM*CM/XIYY
XNL=32.2*Q*SREF*CN/WGT
ALFD=THD-XNL/VM
IF(STEP-1)66,66,55
999 CONTINUE
CLOSE(1)
PAUSE
END

```

Chapter 11. The airframe differential equations for angle of attack rate and angular body acceleration appear after statement label 200.

A case was run for the hypothetical missile of Fig. 22.3 in which the missile fin was deflected 5 deg when the missile was at sea level traveling at 3000 ft/s. We can see from Fig. 22.4 that a 5-deg fin deflection in the positive direction causes the missile to build up to a negative angle of attack. The angle of attack initially approaches -8 deg, oscillates, and finally approaches a steady-state or trim value of approximately -5 deg after several seconds. The decaying oscillations indicate that the missile airframe has low damping. The transient values of the angle of attack due to the fin deflection are important to the designer because excessive values could cause flight catastrophe.

The buildup in angle of attack enables the missile to accelerate. We can see from Fig. 22.5 that the steady-state acceleration due to a 5-deg fin deflection is approximately 13 g at this flight condition. Again note the oscillatory nature of the achieved acceleration due to a fixed fin deflection.

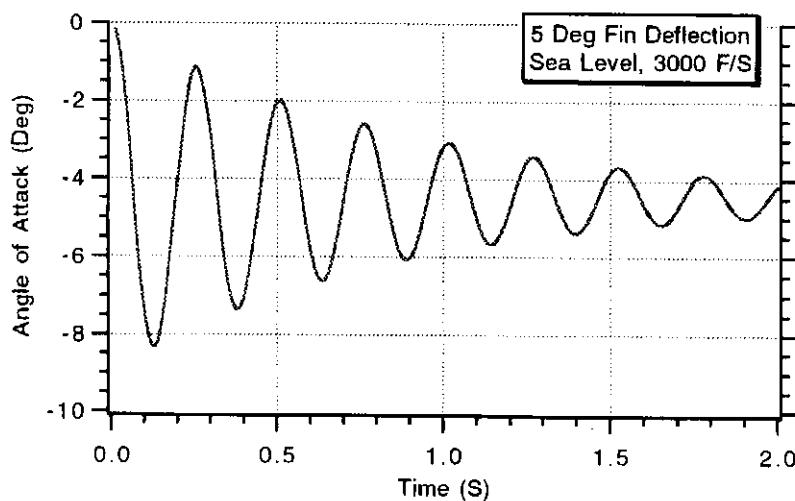


Fig. 22.4 A 5-deg fin deflection results in approximately -5 -deg angle of attack at sea level.

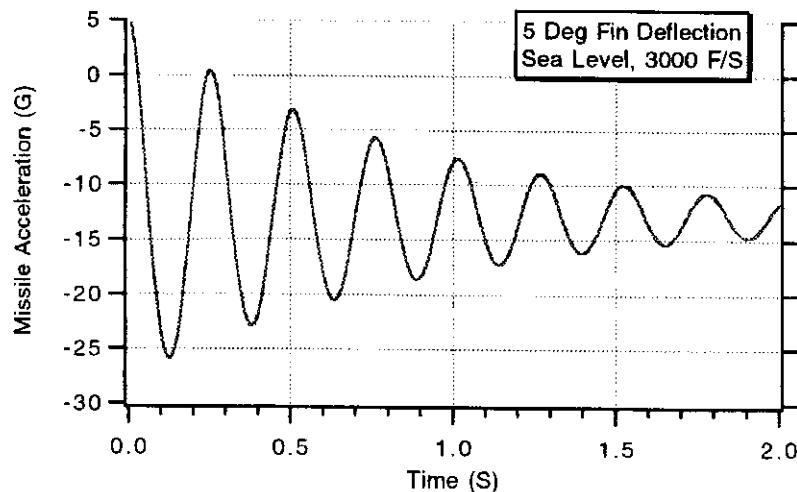


Fig. 22.5 A 5-deg fin deflection results in approximately 13 g of acceleration at sea level.

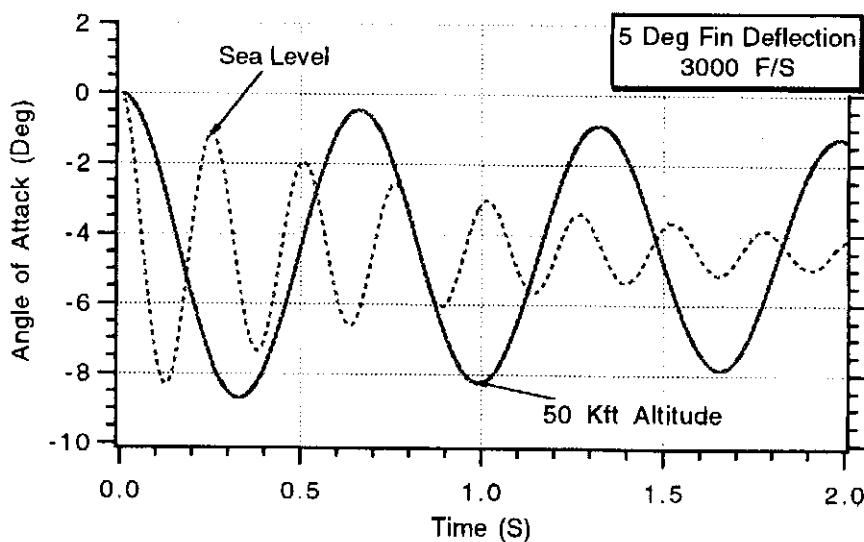


Fig. 22.6 More angle of attack required at higher altitudes for fixed fin deflection.

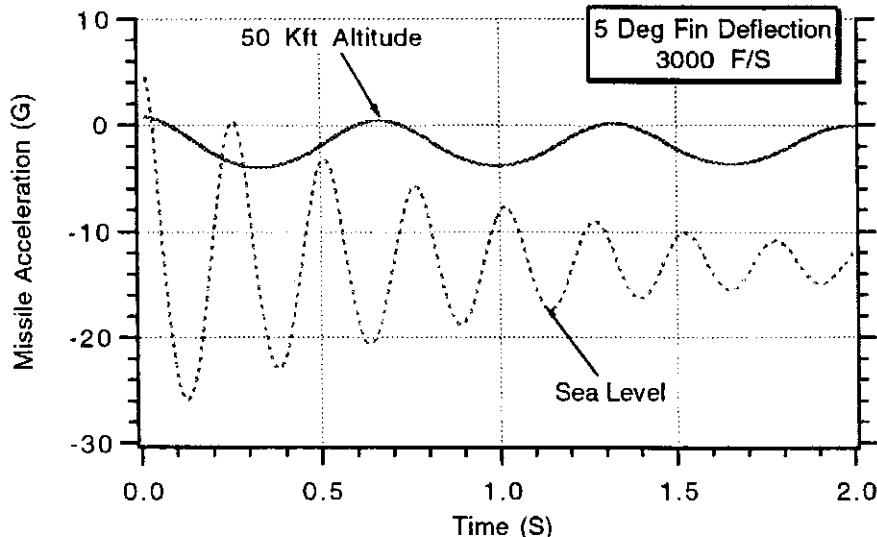


Fig. 22.7 Acceleration capability diminishes at higher altitudes for fixed fin deflection.

The simulation was also run when the altitude was increased to 50 kft (i.e., ALT = 50000). We can see from Figs. 22.6 and 22.7 that at higher altitudes the missile will pull more angle of attack and have less acceleration available for a given fin deflection. In addition, the airframe natural frequency decreases with increasing altitude. Because the amount of fin travel permitted is limited, the simulation demonstrates that an aerodynamic missile will have less acceleration available at the higher altitudes.

Linearization of the Airframe

An examination of the force and moment coefficients reveals that if we assume that the missile speed and altitude are constant, the equations are mostly linear except for the angle of attack squared term in each of the equations. For the constant speed, constant altitude condition we can linearize by assuming that each equation is linear in angle of attack and fin deflection. This means that the normal force coefficient is approximated as

$$C_N = f(\alpha, \delta) \approx C_{N\alpha}\alpha + C_{N\delta}\delta$$

One method of finding $C_{N\alpha}$ and $C_{N\delta}$ is to simply divide the angle of attack terms in C_N by α to get $C_{N\alpha}$ and then divide the fin deflection term by δ to get $C_{N\delta}$ yielding

$$C_{N\alpha} = 2 + \frac{1.5S_{PLAN}\alpha}{S_{ref}} + \frac{8S_W}{\beta S_{ref}} + \frac{8S_T}{\beta S_{ref}}$$

$$C_{N\delta} = \frac{8S_T}{\beta S_{ref}}$$

Note that $C_{N\alpha}$ depends on the angle of attack.

Because we are assuming that the acceleration normal to the body is nearly equal to the missile acceleration perpendicular to the velocity vector (i.e., angle of attack is small), we can express the missile turning rate in terms of $C_{N\alpha}$ and

$C_{N\delta}$ or

$$\dot{\gamma} \approx \frac{n_L}{V_M} = \frac{g Q S_{\text{ref}}}{W V_M} [C_{N\alpha}\alpha + C_{N\delta}\delta] = -Z_\alpha\alpha - Z_\delta\delta$$

where Z_α and Z_δ are defined as

$$Z_\alpha = \frac{-g Q S_{\text{ref}} C_{N\alpha}}{W V_M}$$

$$Z_\delta = \frac{-g Q S_{\text{ref}} C_{N\delta}}{W V_M}$$

Therefore we have expressed the missile turning rate or acceleration as a linear function of angle of attack or fin deflection. In a similar way the moment coefficient can be linearized as

$$C_M = f(\alpha, \delta) \approx C_{M\alpha}\alpha + C_{M\delta}\delta$$

As before, we can find $C_{M\alpha}$ and $C_{M\delta}$ by simply dividing the angle of attack terms of C_M first by α and then dividing the fin deflection term by δ yielding

$$\begin{aligned} C_{M\alpha} &= 2(X_{CG} - X_{CPN}) + \frac{1.5 S_{\text{PLAN}} \alpha}{S_{\text{ref}}} (X_{CG} - X_{CPB}) \\ &\quad + \frac{8 S_W}{\beta S_{\text{ref}}} (X_{CG} - X_{CPW}) + \frac{8 S_T}{\beta S_{\text{ref}}} (X_{CG} - X_{HL}) \\ C_{M\delta} &= \frac{8 S_T}{\beta S_{\text{ref}}} (X_{CG} - X_{HL}) \end{aligned}$$

Note that $C_{M\alpha}$ is not a constant for a given speed and altitude but depends on the angle of attack.

We can now express the linearized missile angular acceleration as

$$\ddot{\theta} = \frac{M}{I_{yy}} = \frac{Q S_{\text{ref}} d}{I_{yy}} [C_{M\alpha}\alpha + C_{M\delta}\delta] = M_\alpha\alpha + M_\delta\delta$$

where M_α and M_δ are defined as

$$M_\alpha = \frac{Q S_{\text{ref}} d C_{M\alpha}}{I_{yy}}$$

$$M_\delta = \frac{Q S_{\text{ref}} d C_{M\delta}}{I_{yy}}$$

Since the derivative of the angle of attack is given by

$$\dot{\alpha} = \dot{\theta} - \dot{\gamma}$$

we can say that

$$\dot{\alpha} = \dot{\theta} + Z_\alpha\alpha + Z_\delta\delta$$

As mentioned previously, M_α and Z_α are not constants in our linearized model but vary with angle of attack. These aerodynamic parameters are usually evaluated at a trim angle of attack. The vehicle is considered to be at trim when the moment

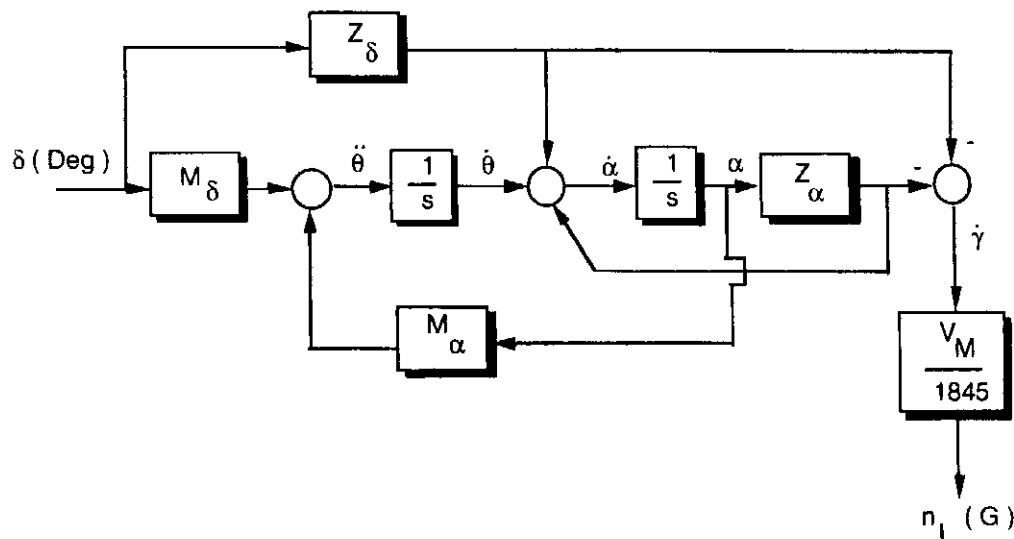


Fig. 22.8 Linearized airframe.

is zero (i.e., $C_M = 0$). At the trim condition one solves for the angle of attack and uses that value to evaluate M_α and Z_α .

The linearized airframe equations can also be represented in block diagram form as shown in Fig. 22.8. The two integrators shown in the block diagram indicate that the airframe can be considered to be a second-order system. It is important to note that this diagram assumes that the input fin deflection δ is in units of degrees and that the output acceleration n_L is in units of gees. All internal angles and rates are either in units of degrees or degrees per second.

Often it is convenient to have a transfer function representation of the airframe. Strictly speaking, the transfer function is only valid when the missile is at a fixed speed, altitude, and trim angle of attack. After some algebra we can find the transfer function relating the achieved missile acceleration to the fin deflection from Fig. 22.8 as

$$\frac{n_L}{\delta} = \frac{-V_M [M_\alpha Z_\delta - Z_\alpha M_\delta]}{1845 M_\alpha} \left[1 - \frac{Z_\delta s^2}{M_\alpha Z_\delta - Z_\alpha M_\delta} \right] / \left(1 + \frac{Z_\alpha}{M_\alpha} s - \frac{s^2}{M_\alpha} \right)$$

The preceding transfer function can be simplified to

$$\frac{n_L}{\delta} = K_1 \left(1 - \frac{s^2}{\omega_z^2} \right) / \left(1 + \frac{2\zeta_{AF}}{\omega_{AF}} s + \frac{s^2}{\omega_{AF}^2} \right)$$

where

$$K_1 = \frac{-V_M [M_\alpha Z_\delta - Z_\alpha M_\delta]}{1845 M_\alpha}$$

and

$$\omega_z = \frac{M_\alpha Z_\delta - Z_\alpha M_\delta}{Z_\delta}$$

$$\omega_{AF} = \sqrt{-M_\alpha}$$

$$\zeta_{AF} = \frac{Z_\alpha \omega_{AF}}{2M_\alpha}$$

Similarly, the transfer function from missile pitch rate to fin deflection can also be written from Fig. 22.8 as

$$\frac{\dot{\theta}}{\delta} = \frac{-[M_\alpha Z_\delta - Z_\alpha M_\delta]}{M_\alpha} \left[1 + \frac{M_\delta s}{M_\alpha Z_\delta - Z_\alpha M_\delta} \right] \Bigg/ \left(1 + \frac{Z_\alpha}{M_\alpha} s - \frac{s^2}{M_\alpha} \right)$$

which simplifies to

$$\frac{\dot{\theta}}{\delta} = K_3 (1 + T_\alpha s) \Bigg/ \left(1 + \frac{2\zeta_{AF}}{\omega_{AF}} s + \frac{s^2}{\omega_{AF}^2} \right)$$

where

$$K_3 = \frac{-[M_\alpha Z_\delta - Z_\alpha M_\delta]}{M_\alpha} = \frac{1845 K_1}{V_M}$$

$$T_\alpha = \frac{M_\delta}{M_\alpha Z_\delta - Z_\alpha M_\delta}$$

If an accelerometer is used in a flight-control system to measure the achieved missile acceleration, it will probably not be located at the center of gravity. Its measurement of the actual acceleration will be corrupted by the body angular acceleration according to

$$n_A = n_L + \frac{(X_{CG} - X_{ACC})\ddot{\theta}}{1845}$$

where X_{ACC} is the accelerometer location with respect to the nose, X_{CG} is the center of gravity of the missile, and n_A is acceleration measured by the accelerometer. However we shall neglect this effect and assume that the measured and achieved accelerations are identical in order to simplify the ensuing analysis.

Numerical Example

To test the accuracy of the airframe linearization, the example presented at the beginning of this chapter was repeated. However, before we begin, we must first have a method of calculating the trim angle of attack for a given fin deflection (i.e., fin deflection is 5 deg in this example). The moment coefficient can be written as

$$C_M = 2\alpha(X_{CG} - X_{CPN}) + \frac{1.5 S_{PLANA} \alpha^2}{S_{ref}} (X_{CG} - X_{CPB}) \\ + \frac{8 S_W \alpha}{\beta S_{ref}} (X_{CG} - X_{CPW}) + \frac{8 S_T (\alpha + \delta)}{\beta S_{ref}} (X_{CG} - X_{HL})$$

or in shorthand notation as

$$C_M = y_1 \alpha + y_2 \alpha^2 + y_3 \delta$$

where

$$\begin{aligned}y_1 &= 2(X_{CG} - X_{CPN}) + \frac{8S_W}{\beta S_{ref}}(X_{CG} - X_{CPW}) + \frac{8S_T}{\beta S_{ref}}(X_{CG} - X_{HL}) \\y_2 &= \frac{1.5S_{PLAN}\alpha}{S_{ref}}(X_{CG} - X_{CPB}) \\y_3 &= \frac{8S_T}{\beta S_{ref}}(X_{CG} - X_{HL})\end{aligned}$$

At trim the moment coefficient is zero. Therefore, for a given fin deflection δ_{NOM} we get the equation for the trim angle of attack α_{TR} to be

$$0 = y_1\alpha_{TR} + y_2\alpha_{TR}^2 + y_3\delta_{NOM}$$

We can use the quadratic formula to solve for the trim angle of attack. After eliminating the unrealistic root we get

$$\alpha_{TR} = \frac{-y_1 - \sqrt{y_1^2 - 4y_2y_3\delta_{NOM}}}{2y_2}$$

and can now evaluate $C_{N\alpha}$ and $C_{M\alpha}$ as

$$\begin{aligned}C_{N\alpha} &= 2 + \frac{1.5S_{PLAN}\alpha_{TR}}{S_{ref}} + \frac{8S_W}{\beta S_{ref}} + \frac{8S_T}{\beta S_{ref}} \\C_{M\alpha} &= 2(X_{CG} - X_{CPN}) + \frac{1.5S_{PLAN}\alpha_{TR}}{S_{ref}}(X_{CG} - X_{CPB}) \\&\quad + \frac{8S_W}{\beta S_{ref}}(X_{CG} - X_{CPW}) + \frac{8S_T}{\beta S_{ref}}(X_{CG} - X_{HL})\end{aligned}$$

To write a simulation involving the linearized airframe, we must convert the transfer functions to differential equations as was done in Chapter 1. The transfer function relating missile acceleration to fin deflection was already shown to be

$$\frac{n_L}{\delta} = K_1 \left(1 - \frac{s^2}{\omega_z^2} \right) / \left(1 + \frac{2\zeta_{AF}}{\omega_{AF}}s + \frac{s^2}{\omega_{AF}^2} \right)$$

Using the chain rule from calculus we can say that

$$\frac{n_L}{\delta} = \frac{e}{\delta} * \frac{n_L}{e}$$

Therefore, as was done in Chapter 1, we can split the missile acceleration transfer

function and get two equivalent transfer functions or

$$\frac{e}{\delta} = 1 / \left(1 + \frac{2\zeta_{AF}}{\omega_{AF}}s + \frac{s^2}{\omega_{AF}^2} \right) \quad \text{and} \quad \frac{n_L}{e} = K_1 \left(1 - \frac{s^2}{\omega_z^2} \right)$$

Cross multiplying the first transfer function and converting Laplace transform notation to the time domain yields

$$e + \frac{2\zeta_{AF}}{\omega_{AF}}\dot{e} + \frac{\ddot{e}}{\omega_{AF}^2} = \delta$$

If we solve the preceding equation for the highest derivative, we get

$$\ddot{e} = \omega_{AF}^2 \left(\delta - e - \frac{2\zeta_{AF}}{\omega_{AF}}\dot{e} \right)$$

Repeating the procedure and cross multiplying the second transfer function and converting to the time domain yields the equation for the missile acceleration or

$$n_L = K_1 \left(e - \ddot{e}/\omega_z^2 \right)$$

Similarly recall that the body rate transfer function is given by

$$\frac{\dot{\theta}}{\delta} = K_3(1 + T_\alpha s) / \left(1 + \frac{2\zeta_{AF}}{\omega_{AF}}s + \frac{s^2}{\omega_{AF}^2} \right)$$

Again we can use the chain rule to split the transfer function as

$$\frac{\dot{\theta}}{\delta} = \frac{e}{\delta} * \frac{\dot{\theta}}{e}$$

The second term on the right-hand side of the preceding equation is simply the numerator of the transfer function or

$$\frac{\dot{\theta}}{e} = K_3(1 + T_\alpha s)$$

Cross multiplying and converting Laplace transforms to the time domain yields the differential equation for the body rate as

$$\dot{\theta} = K_3(e + T_\alpha \dot{e})$$

We now have the two differential equations required to simulate the linearized airframe. A linear simulation of the airframe, based on the definitions of the previous section and the differential equations derived from the airframe transfer functions, appears in Listing 22.2. The linear airframe coefficients are evaluated using the trim value of the angle of attack. The inputs of the linear simulation are identical to those of the nonlinear airframe simulation of Listing 22.1. We can see that the linear differential equations appear after statement label 200.

The nominal case was run in which there was a 5-deg fin deflection when the missile was at sea level and traveling at 3000 ft/s. Figure 22.9 shows that the linear airframe acceleration response is a near perfect match to the nonlinear airframe results derived from Listing 22.1. This means that our linearized model is a good approximation to reality.

Listing 22.2 Linear airframe simulation

```

VM=3000.
Del=5./57.3
ALT=0.
A=1000.
DIAM=1.
FR=3.
XL=20.
CTW=0.
CRW=6.
HW=2.
CTT=0.
CRT=2.
HT=2.
XN=4.
XCG=10.
XHL=19.5
WGT=1000.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
IF(ALT.LE.30000.)THEN
    RHO=.002378*EXP(-ALT/30000.)
ELSE
    RHO=.0034*EXP(-ALT/22000.)
ENDIF
SWING=.5*HW*(CTW+CRW)
STAIL=.5*HT*(CTT+CRT)
SREF=3.1416*DIAM*DIAM/4.
XLP=FR*DIAM
SPLAN=(XL-XLP)*DIAM+1.33*XLP*DIAM/2.
XCPN=2*XLP/3
AN=.67*XLP*DIAM
AB=(XL-XLP)*DIAM
XCPB=.67*AN*XLP+AB*(XLP+.5*(XL-XLP))/(AN+AB)
XCPW=XLP+XN+.7*CRW-.2*CTW
XMACH=VM/A
XIYY=WGT*(3*((DIAM/2)**2)+XL*XL)/(12*32.2)
TMP1=(XCG-XCPW)/DIAM
TMP2=(XCG-XHL)/DIAM
TMP3=(XCG-XCPB)/DIAM
TMP4=(XCG-XCPN)/DIAM
B=SQRT(XMACH**2-1)
Q=.5*RHO*VM*VM
Y1=2*TMP4+8*SWING*TMP1/(B*SREF)+8*STAIL*TM P2/(B*SREF)
Y2=1.5*SPLAN*TMP3/SREF
Y3=8*STAIL*TMP2*DEL/(B*SREF)
ALFTR=(-Y1-SQRT(Y1*Y1-4.*Y2*Y3))/(2*Y2)
CNA=2+1.5*SPLAN*ALFTR/SREF+8*SWING/(B*SREF )+8*STAIL/(B*SREF)
CND=8*STAIL/(B*SREF)
CMAP=2*TMP4+1.5*SPLAN*ALFTR*TMP3/SREF+8*S WING*TMP1/(B*SREF)

```

(Contd.)

Listing 22.2 (Continued)

```

CMA=CMAP+8*STAIL*TMP2/(B*SREF)
CMD=8*STAIL*TMP2/(B*SREF)
XMA=Q*SREF*DIAM*CMA/XIYY
XMD=Q*SREF*DIAM*CMD/XIYY
ZA=-32.2*Q*SREF*CNA/(WGT*VM)
ZD=-32.2*Q*SREF*CND/(WGT*VM)
WZ=SQRT((XMA*ZD-ZA*XMD)/ZD)
WAF=SQRT(-XMA)
ZAF=.5*WAF*ZA/XMA
XK1=-VM*(XMA*ZD-XMD*ZA)/(1845*XMA)
XK2=XK1
TA=XMD/(XMA*ZD-XMD*ZA)
XK3=1845*XK1/VM
E=0.
ED=0.
T=0
H=.0025
S=0
10 IF (T>1.99999) GOTO 999
    S=S+H
    EOLD=E
    EDOLD=ED
    STEP=1
    GOTO 200
66 STEP=2
    E=E+H*ED
    ED=ED+H*EDD
    T=T+H
    GOTO 200
55 CONTINUE
    E=.5*(EOLD+E+H*ED)
    ED=.5*(EDOLD+ED+H*EDD)
    IF(S.LE..0099999)GOTO 10
    S=0.
    WRITE(1,97)T,XNL,THD
    WRITE(*,97)T,XNL,THD
    GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
    EDD=WAF*WAF*(DEL*57.3-E-2.*ZAF*ED/WAF)
    XNL=XK1*(E-EDD/WZ**2)
    THD=XK3*(E+TA*ED)
    IF(STEP-1)66,66,55
999 CONTINUE
    CLOSE(1)
    PAUSE
    END

```

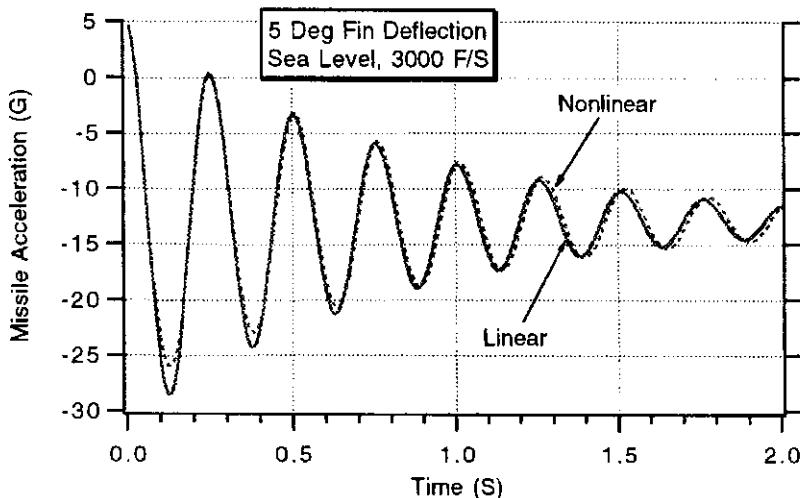


Fig. 22.9 Linear model accurately approximates actual missile acceleration.

The angle of attack squared terms in the force and moment equations prevent the linear model from being perfect. The linear model is a less accurate representation of reality when the angle of attack is large. To see if the linear model is less accurate when the angle of attack is larger, another case was run in which the fin deflection was increased from 5 deg to 10 deg. We can see from Fig. 22.10 that when the fin deflection is increased the linear approximation to the airframe starts to deteriorate.

Finally, another case was run in which the fin deflection was still 5 deg, but the altitude increased from sea level to 50,000 ft. A 5-deg fin deflection at 50 kft altitude will cause a larger angle of attack than a 5-deg fin deflection at sea level, and so we would expect our linear model to be less accurate because of the angle of attack squared term in the force and moment equations. We can see from Fig. 22.11 that the linear model still approximates reality very well at this high altitude flight condition.

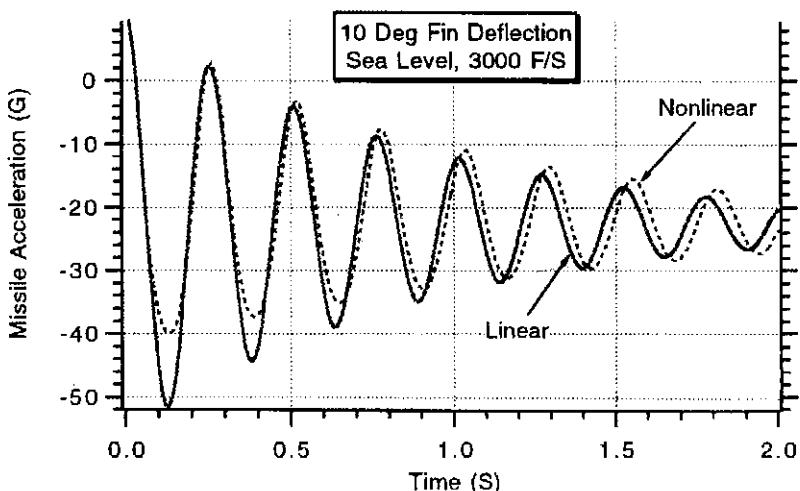


Fig. 22.10 Linear model is less accurate at larger fin deflections.

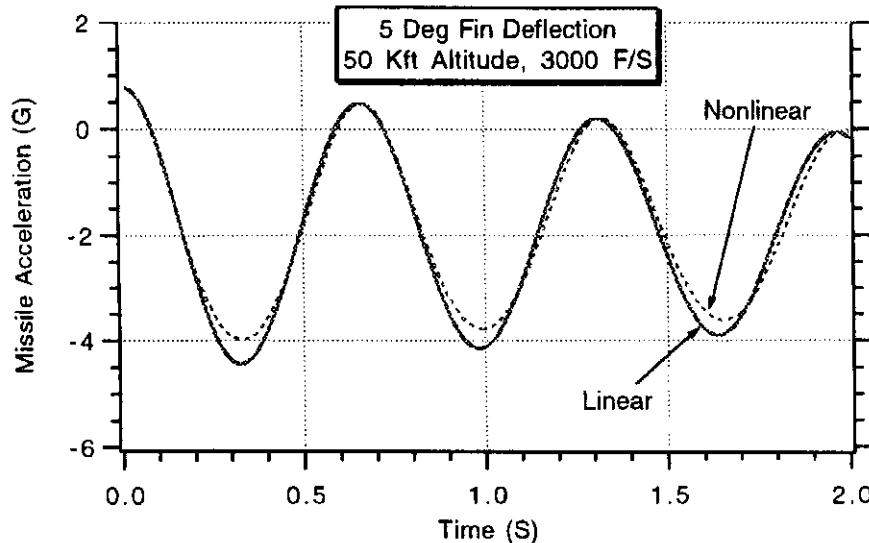


Fig. 22.11 Linear model is reasonable at higher altitudes.

Experiments

Using the linear transfer function approach, we can study the effect of flight condition on various important airframe parameters. We have seen that the airframe natural frequency is given by

$$\omega_{AF} = \sqrt{-M_a}$$

Figure 22.12 shows that for our hypothetical missile the airframe natural frequency decreases with increasing altitude and decreasing speed. For this example the airframe natural frequency varied between 10 rad/s and 30 rad/s. If we think of the airframe time constant being the inverse of the natural frequency, then the time constant variation is between 0.033 s and 0.1 s. In general the airframe time constant is fast, and an autopilot is usually not required to artificially speed up the airframe response.

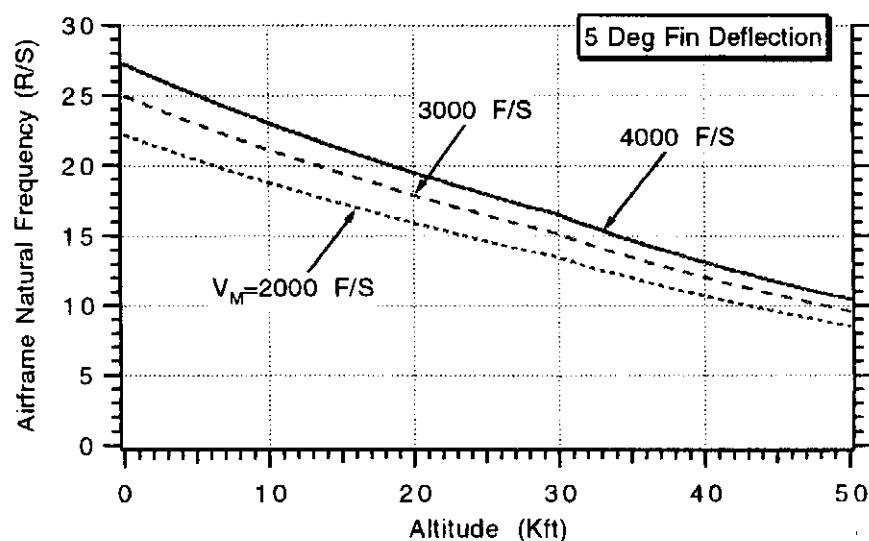


Fig. 22.12 Airframe natural frequency decreases with increasing altitude and decreasing speed.

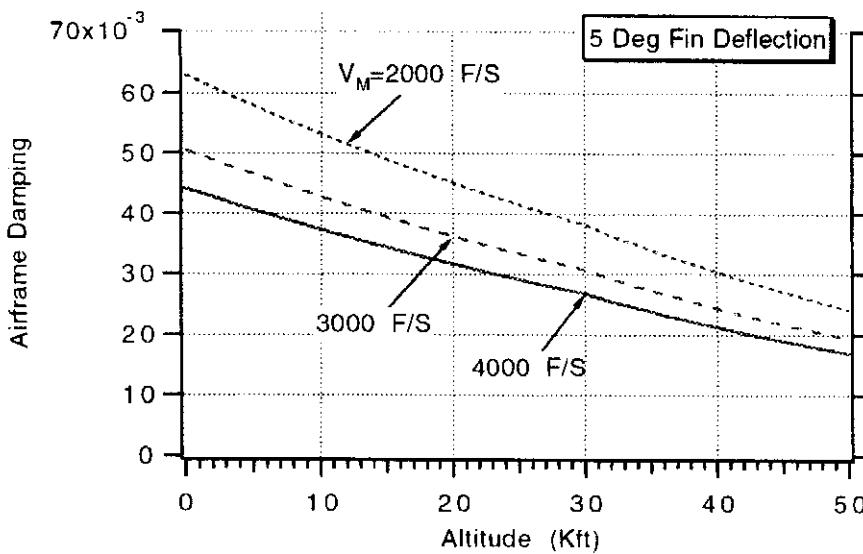


Fig. 22.13 Airframe damping is low and decreases with increasing altitude.

We have already shown that the airframe damping is given by

$$\zeta_{AF} = \frac{Z_\alpha \omega_{AF}}{2M_\alpha}$$

Figure 22.13 shows that the airframe damping decreases with increasing altitude and increasing missile speed. The airframe damping is quite low and in this example varies between 0.02 and 0.065. We shall soon see that this low damping is not satisfactory for overall system performance in a radar homing missile and that a flight-control system is required to artificially increase the low damping of the airframe.

M_α and M_δ are both important airframe parameters and are displayed in Figs. 22.14 and 22.15. A positive M_α indicates that the bare airframe is unstable. There are limits on how negative or positive M_α can be before the design of the flight-control system becomes impossible. During the normal design process it is natural for the baseline airframe to change due to either overly optimistic

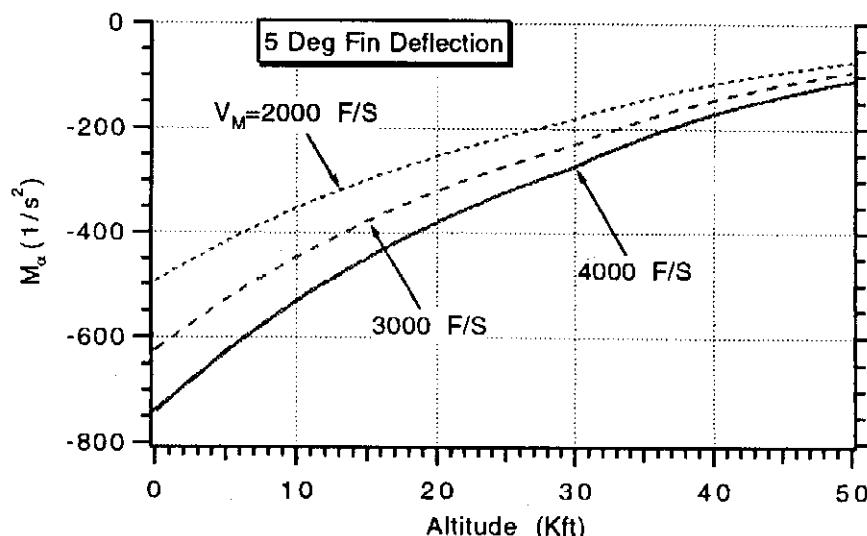


Fig. 22.14 M_α gets smaller with increasing altitude and decreasing speed.

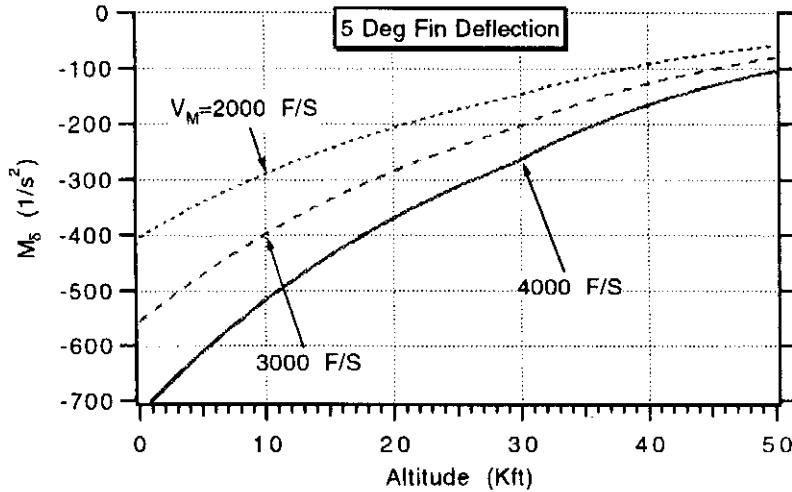


Fig. 22.15 M_δ gets smaller with increasing altitude and decreasing speed.

assumptions concerning weight and size or possibly due to new requirements. In both cases it is important that the flight-control system designer work closely with the aerodynamicist not only when an airframe is being selected but also as it is being modified. For example, large values of M_δ make it difficult to choose actuators that will work with the flight-control system. Therefore it is also important to limit the size of this key aerodynamic parameter.

Finally Fig. 22.16 shows that the missle turning rate time constant T_α increases with increasing altitude and increasing missle speed. At 50 kft altitude we can see that the turning rate time constant is approximately 4 s when the missle is traveling at 3000 ft/s. This is slightly smaller than the value used in Chapter 19. In Chapter 19 the turning rate time constant was calculated based on the nose and body only. We can see that the addition of the wing and tail decreased the turning rate time constant from 5 s to 4 s. Although all missles have tails, some do not have wings. Wingless missles will tend to have larger turning rate time constants. We have shown before that the turning rate time constant is related to the radome

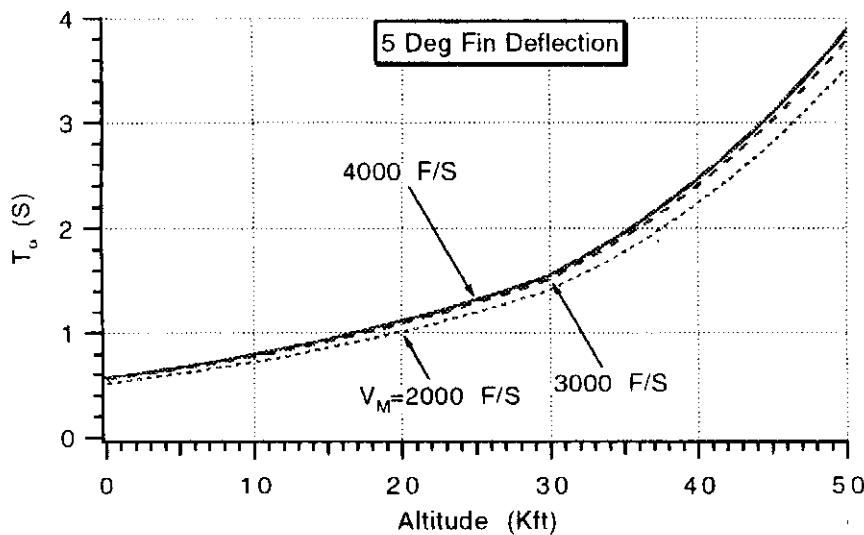


Fig. 22.16 Turning rate time constant increases with increasing altitude and increasing speed.

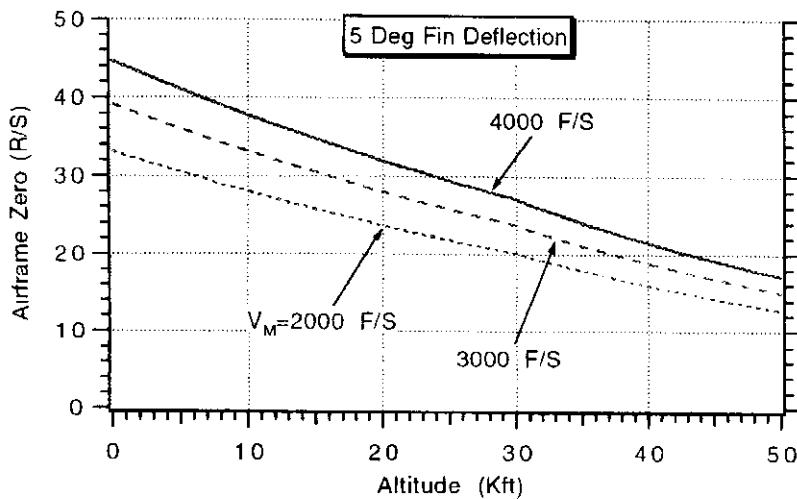


Fig. 22.17 Frequency of airframe zero decreases with increasing altitude.

stability problem. Large values of turning rate time constant require smaller values of radome slope for a given level of performance.

We have already shown that the airframe zero ω_z can be expressed in terms of the aerodynamic parameters as

$$\omega_z = \frac{M_\alpha Z_\delta - Z_\alpha M_\delta}{Z_\delta}$$

We can see from Fig. 22.17 that the airframe zero decreases with increasing altitude and decreasing missile velocity. Smaller values of the airframe zero will cause more wrong-way tail effect.

The acceleration aerodynamic gain K_1 is in units of gees per degree and tells how much steady-state acceleration there will be for a given fin deflection. Figure 22.18 shows that the magnitude of the aerodynamic gain gets smaller as the altitude increases and velocity decreases. For example, if the missile is traveling at 3000 ft/s the aerodynamic gain is approximately -2.7 at sea level and -0.4 at 50 kft altitude. Therefore a 5-deg fin deflection would result in $-13.5 g$ at sea level (i.e.,

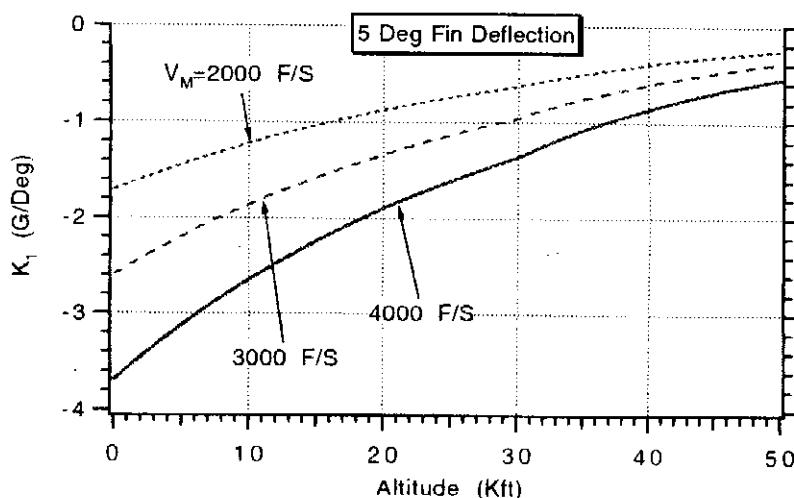


Fig. 22.18 Magnitude of acceleration aerodynamic gain gets smaller as altitude increases.

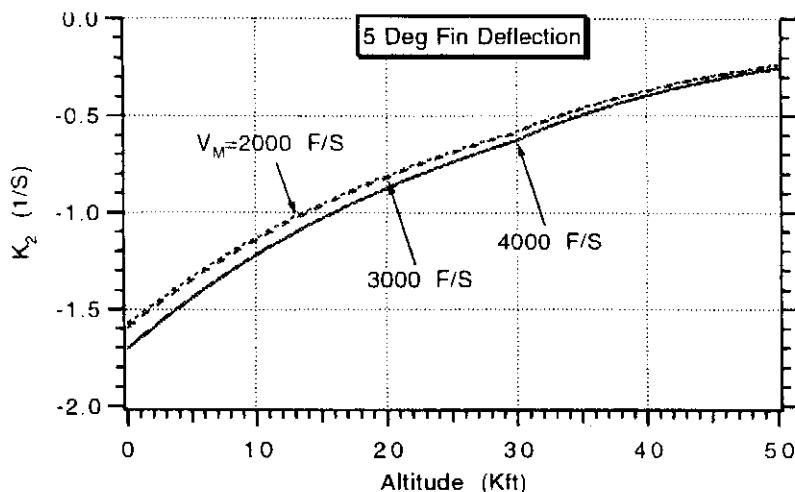


Fig. 22.19 Magnitude of body rate aerodynamic gain is independent of velocity and gets smaller as altitude increases.

$-2.7 * 5 = -13.5$) and only 2 g at 50 kft altitude (i.e., $-0.4 * 5 = -2$). The nonlinear results of Fig. 22.7 confirm these calculations. More fin travel will be required to achieve a given acceleration when the altitude is higher or when the missile velocity is smaller.

The body rate aerodynamic gain K_3 tells how much steady-state body rate there will be for a given fin deflection. Figure 22.19 shows that the magnitude of the aerodynamic gain gets smaller as the altitude increases and velocity decreases. For example, if the missile is traveling at 3000 ft/s the body rate aerodynamic gain is approximately -1.6 s^{-1} at sea level and -0.24 s^{-1} at 50 kft altitude. Therefore a 5-deg fin deflection would result in -8 deg/s at sea level (i.e., $-1.6 * 5 = -8$) and only -1.2 deg/s at 50 kft altitude (i.e., $-0.24 * 5 = -1.2$).

Summary

In this chapter we have seen how the nonlinear missile force and moment equations are related to the geometry of the missile airframe. A simple method for linearizing the force and moment equations was introduced so that transfer functions could be derived for the missile airframe. It was shown that the transfer function approximation to the airframe was a good approximation to the nonlinear force and moment equations at small angles of attack.

References

- ¹Chin, S. S., *Missile Configuration Design*, McGraw-Hill, New York, 1961.
- ²Jerger, J. J., *System Preliminary Design*, Van Nostrand, Princeton, NJ, 1960.
- ³Giragosian, P., "Aerodynamic Considerations in the Design of a Vertically Launched Advanced Interdiction Missile," *Proceedings of AIAA Atmospheric Flight Mechanics Conference*, AIAA, San Diego, CA, Aug. 9–11, 1982.
- ⁴Selby, S. M., *Standard Mathematical Tables, Twentieth Edition*, The Chemical Rubber Co., Cleveland, OH, 1972.

Introduction to Flight Control Design

Introduction

IN Chapter 22 we saw how to derive aerodynamic transfer functions and relate them to the missile airframe's force and moment equations. The missile's bare airframe response was shown to be highly oscillatory because of its low damping. The purpose of the flight-control system, shown in Fig. 23.1, is to convert the missile's acceleration command n_c generated by the guidance law to an achieved acceleration n_L . The flight-control system usually must improve the response characteristics of the bare missile airframe to ensure that the achieved acceleration closely follows the commanded acceleration.

As we can see from Fig. 23.1, the missile airframe is just one part of the flight-control system. Mathematically we can think of the airframe as a transfer function whose input is the tail fin deflection δ and whose output is the achieved missile acceleration n_L . The autopilot is another part of the flight-control system and is the mechanism for converting the acceleration command n_c to a fin deflection command δ_c . The actuator then takes the autopilot's electrical output and moves the missile control surfaces (i.e., canards, wings, or tails) through the appropriate angular deflection δ in response to the fin deflection command.

In this chapter we shall first see how the flight-control system interacts with the rest of the guidance system and how it influences system performance. Then we shall investigate a simple way in which the principles of feedback can be used in order to improve the flight-control system response so that homing guidance objectives can be met.

Open-Loop Flight-Control System

The simplest possible flight-control system is the open-loop system shown in Fig. 23.2. In this diagram, in which the airframe is treated as a transfer function and the actuator dynamics are neglected, the autopilot is simply a gain that attempts to cancel the aerodynamic gain of the airframe. In the steady state the missile-achieved acceleration will match the commanded acceleration provided that the autopilot gain can be changed with flight condition. The open-loop autopilot is the least expensive of all possible autopilots since it does not require a rate gyro or accelerometer.

In Chapter 22 we assumed a certain fin deflection and solved for the trim angle of attack along with the various aeroderivatives. In this chapter we shall specify the desired acceleration and solve for the trim fin deflection and angle of attack.

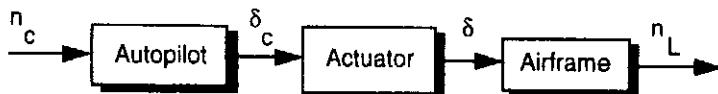


Fig. 23.1 Conceptual block diagram of a flight-control system.

Recall that the normal force coefficient is given by

$$C_N = 2\alpha + \frac{1.5 S_{PLAN} \alpha^2}{S_{ref}} + \frac{8 S_W \alpha}{\beta S_{ref}} + \frac{8 S_T (\alpha + \delta)}{\beta S_{ref}}$$

Since

$$F_N = ma = \frac{W n_L}{g} = Q S_{ref} C_N$$

we can solve for the normal force coefficient at trim according to

$$C_{NTRIM} = \frac{W n_{LTRIM}}{g Q S_{ref}}$$

Using shorthand notation we can now say that at trim we get

$$C_{NTRIM} = y_1 \alpha_{TRIM} + y_2 \alpha_{TRIM}^2 + y_3 \delta_{TRIM}$$

where

$$y_1 = 2 + \frac{8 S_W}{\beta S_{ref}} + \frac{8 S_T}{\beta S_{ref}}$$

$$y_2 = \frac{1.5 S_{PLAN}}{S_{ref}}$$

$$y_3 = \frac{8 S_T}{\beta S_{ref}}$$

Recall that the moment coefficient equation is given by

$$\begin{aligned} C_M &= 2\alpha(X_{CG} - X_{CPN}) + \frac{1.5 S_{PLAN} \alpha^2}{S_{ref}}(X_{CG} - X_{CPB}) \\ &\quad + \frac{8 S_W \alpha}{\beta S_{ref}}(X_{CG} - X_{CPW}) + \frac{8 S_T (\alpha + \delta)}{\beta S_{ref}}(X_{CG} - X_{HL}) \end{aligned}$$

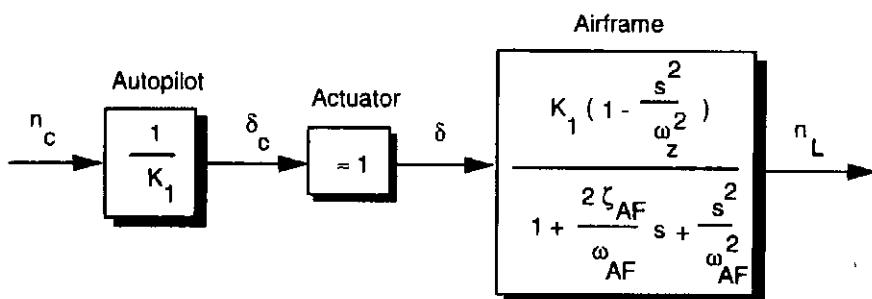


Fig. 23.2 Open-loop flight-control system.

At trim the moment coefficient is zero and we can rewrite the preceding equation in shorthand notation as

$$0 = y_4\alpha_{\text{TRIM}} + y_5\alpha_{\text{TRIM}}^2 + y_6\delta_{\text{TRIM}}$$

where

$$\begin{aligned}y_4 &= 2(X_{\text{CG}} - X_{\text{CPN}}) + \frac{8S_W(X_{\text{CG}} - X_{\text{CPW}})}{\beta S_{\text{ref}}} + \frac{8S_T(X_{\text{CG}} - X_{\text{HL}})}{\beta S_{\text{ref}}} \\y_5 &= \frac{1.5S_{\text{PLAN}}(X_{\text{CG}} - X_{\text{CPB}})}{S_{\text{ref}}} \\y_6 &= \frac{8S_T(X_{\text{CG}} - X_{\text{HL}})}{\beta S_{\text{ref}}}\end{aligned}$$

We now have two trim equations with two unknowns. The two equations can be reduced to one quadratic equation allowing us to solve for the trim angle of attack as

$$\alpha_{\text{TRIM}} = \frac{-p_3 + \sqrt{p_3^2 + 4p_2C_{N\text{TRIM}}}}{2p_2}$$

where

$$\begin{aligned}p_2 &= y_2 - \frac{y_3y_5}{y_6} \\p_3 &= y_1 - \frac{y_3y_4}{y_6}\end{aligned}$$

Substituting the trim angle of attack into the moment coefficient equation allows us to solve for the trim fin deflection as

$$\delta_{\text{TRIM}} = \frac{-y_4\alpha_{\text{TRIM}} - y_5\alpha_{\text{TRIM}}^2}{y_6}$$

We now have enough information to solve for the aerodynamic parameters at any flight condition given a desired acceleration level. Using the notional missile of Chapter 22, Table 23.1 numerically summarizes the values for the various missile transfer function parameters for the case in which the missile is traveling at 3000 ft/s at both sea level and 50 kft altitude and is trying to respond to a 10-g acceleration command.

Table 23.1 Missile transfer function parameters at two flight conditions

Airframe parameter	Definition	Sea level	50 kft
ω_{AF}	Airframe natural frequency	25.3 rad/s	10.0 rad/s
ζ_{AF}	Airframe damping	0.058	0.027
ω_z	Airframe zero	43.2 rad/s	18.9 rad/s
K_1	Aerodynamic acceleration gain	-3.07 g/deg	-0.559 g/deg

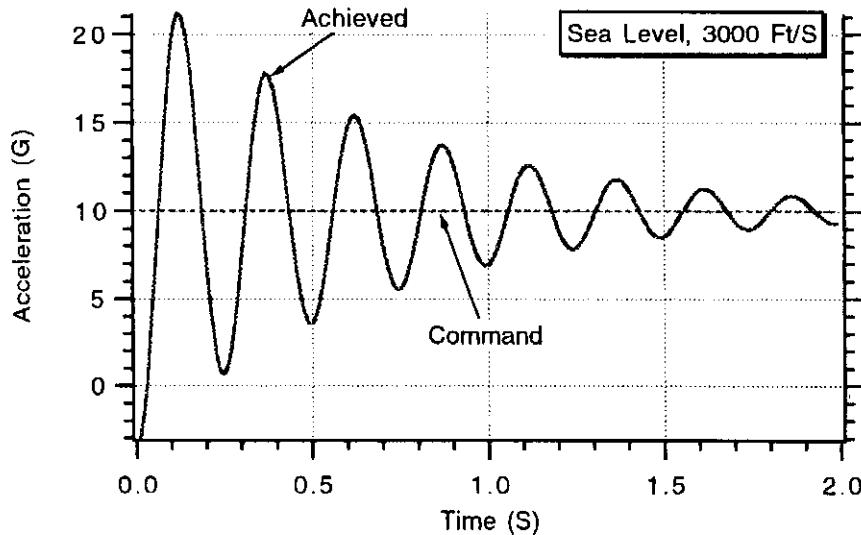


Fig. 23.3 Open-loop flight-control system is lightly damped.

The open-loop flight-control system of Fig. 23.2 was simulated using the linearized aerodynamics. We can see from Listing 23.1 the previously derived equations for the calculation of the trim angle of attack. We can also see after statement label 200 that the open-loop autopilot is simply a gain.

As expected, Fig. 23.3 shows that the flight-control system response due to a 10-g step at sea level is oscillatory due to the low damping of the airframe. The frequency of oscillation is also that of the bare airframe and is very high. In other words the flight-control system response is simply the response of the bare airframe. The sole purpose of the flight-control system is to ensure that the achieved acceleration looks like the commanded acceleration in the steady state.

Figure 23.4 shows that increasing the altitude decreases the damping and natural frequency of the open-loop flight-control system response. We can see that the flight-control system response is still that of the bare airframe.

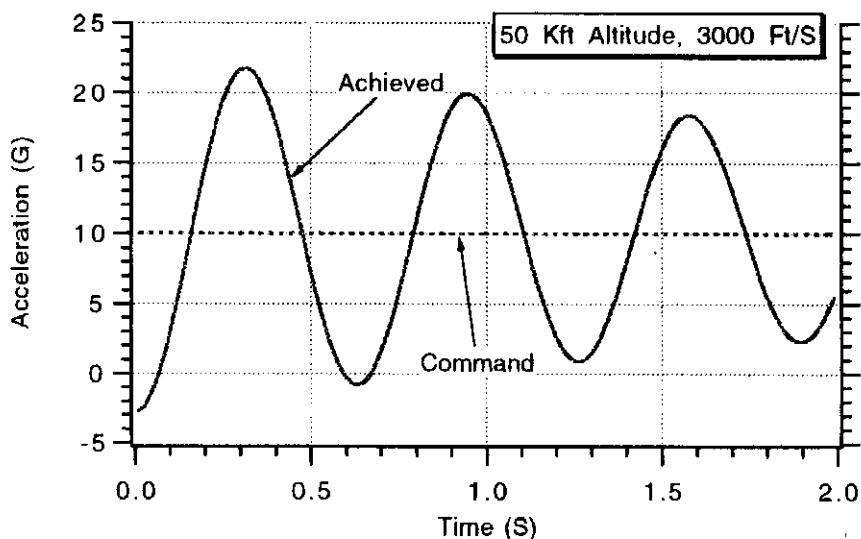


Fig. 23.4 Increasing the attitude decreases both the damping and natural frequency of the open-loop flight-control system.

Listing 23.1 Open-loop flight-control system

```

VM=3000.
XNCG=10.
ALT=0.
A=1000.
DIAM=1.
FR=3.
XL=20.
CTW=0.
CRW=6.
HW=2.
CTT=0.
CRT=2.
HT=2.
XN=4.
XCG=10.
XHL=19.5
WGT=1000.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
IF(ALT.LE.30000.)THEN
    RHO=.002378*EXP(-ALT/30000.)
ELSE
    RHO=.0034*EXP(-ALT/22000.)
ENDIF
SWING=.5*HW*(CTW+CRW)
STAIL=.5*HT*(CTT+CRT)
SREF=3.1416*DIAM*DIAM/4.
XLP=FR*DIAM
SPLAN=(XL-XLP)*DIAM+1.33*XLP*DIAM/2.
XCPN=2*XLP/3
AN=.67*XLP*DIAM
AB=(XL-XLP)*DIAM
XCPB=(.67*AN*XLP+AB*(XLP+.5*(XL-XLP)))/(AN+AB)
XCPW=XLP+XN+.7*CRW-.2*CTW
XMACH=VM/A
XIYY=WGT*(3*((DIAM/2)**2)+XL*XL)/(12*32.2)
TMP1=(XCG-XCPW)/DIAM
TMP2=(XCG-XHL)/DIAM
TMP3=(XCG-XCPB)/DIAM
TMP4=(XCG-XCPN)/DIAM
B=SQRT(XMACH**2-1)
Q=.5*RHO*VM*VM
P1=WGT*XNCG/(Q*SREF)
Y1=2.+8*SWING/(B*SREF)+8*STAIL/(B*SREF)
Y2=1.5*SPLAN/SREF
Y3=8*STAIL/(B*SREF)
Y4=2*TMP4+8*SWING*TMP1/(B*SREF)+8*STAIL*TMP2/(B*SREF)
Y5=1.5*SPLAN*TMP3/SREF
Y6=8*STAIL*TMP2/(B*SREF)
P2=Y2-Y3*Y5/Y6
P3=Y1-Y3*Y4/Y6
ALFTR=(-P3+SQRT(P3*P3+4.*P2*P1))/(2.*P2)

```

(Contd.)

Listing 23.1 (Continued)

```

DELTR=-Y4*ALFTR/Y6-Y5*ALFTR*ALFTR/Y6
CNA=2+1.5*SPLAN*ALFTR/SREF+8*SWING/(B*SREF)+8*STAIL/(B*SREF)
CND=8*STAIL/(B*SREF)
CMAP=2*TMP4+1.5*SPLAN*ALFTR*TMP3/SREF+8*SWING*TMP1/
(B*SREF)
CMA=CMAP+8*STAIL*TMP2/(B*SREF)
CMD=8*STAIL*TMP2/(B*SREF)
XMA=Q*SREF*DIAM*CMA/XIYY
XMD=Q*SREF*DIAM*CMD/XIYY
ZA=-32.2*Q*SREF*CNA/(WGT*VM)
ZD=-32.2*Q*SREF*CND/(WGT*VM)
WZ=SQRT((XMA*ZD-ZA*XMD)/ZD)
WAF=SQRT(-XMA)
ZAF=.5*WAF*ZA/XMA
XK1=-VM*(XMA*ZD-XMD*ZA)/(1845*XMA)
XK2=XK1
TA=XMD/(XMA*ZD-XMD*ZA)
XK3=1845*XK1/VM
XKDC=1./XK1
E=0.
ED=0.
T=0
H=.0001
S=0
10 IF (T>1.99999) GOTO 999
S=S+H
EOLD=E
EDOLD=ED
STEP=1
GOTO 200
66 STEP=2
E=E+H*ED
ED=ED+H*EDD
T=T+H
GOTO 200
55 CONTINUE
E=.5*(EOLD+E+H*ED)
ED=.5*(EDOLD+ED+H*EDD)
IF(S.LE.0099999)GOTO 10
S=0.
WRITE(1,97)T,XNL,XNCG
WRITE(*,97)T,XNL,XNCG
GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
DEL=XKDC*XNCG
EDD=WAF*WAF*(DEL-E-2.*ZAF*ED/WAF)
XNL=XK1*(E-EDD/WZ**2)
IF(STEP-1)66,66,55
999 CONTINUE
PAUSE
END

```

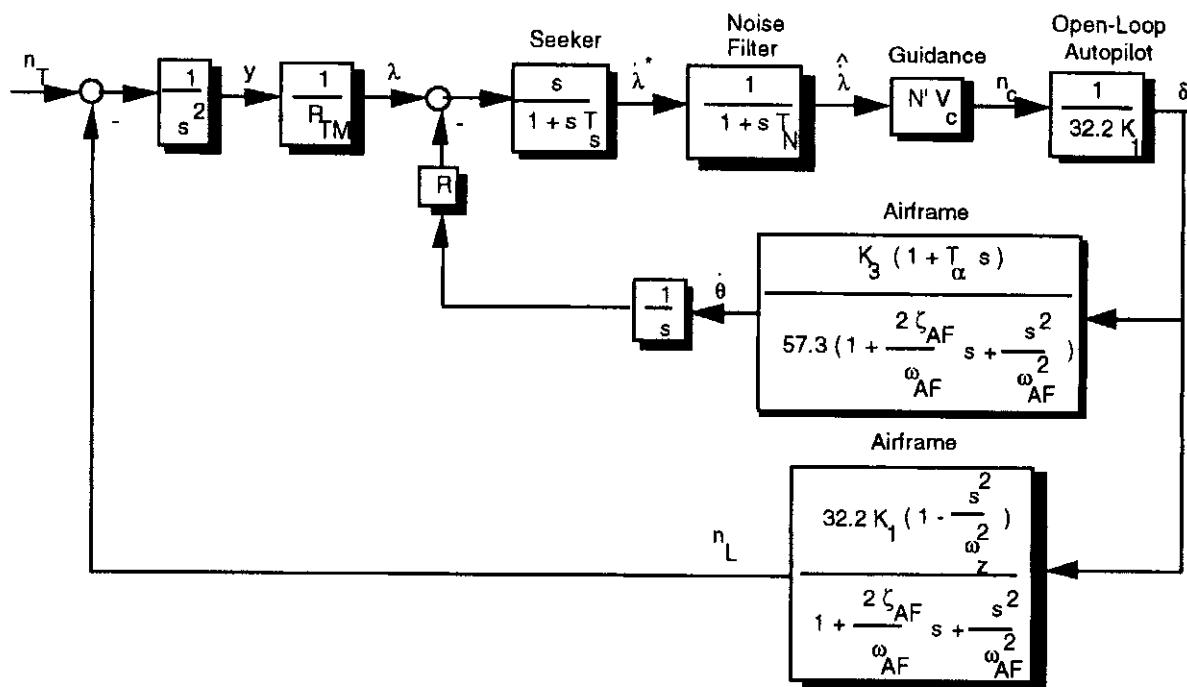


Fig. 23.5 Homing loop with open-loop flight-control system.

Guidance System Interactions

To get a clearer understanding of how the open-loop flight-control system (i.e., autopilot plus airframe) interacts with the missile guidance system, let us consider the homing loop block diagram of Fig. 23.5. In this diagram both the seeker and noise filter are represented by single lags while the autopilot is simply represented by a gain. Because actuator dynamics are neglected, the autopilot output is the achieved fin deflection. Airframe transfer functions are present in the block diagram to convert fin deflection to body rate and missile acceleration. The values of the additional parameters for the body rate transfer function along with the guidance system parameters appear in Table 23.2 for the two flight conditions of interest. Because the two airframe transfer functions assume angles are in units of degrees and acceleration is in units of gees, the constants 57.3 and 32.2 appear in the homing loop block diagram to convert quantities to the English system of units. Notice that radome effects are also included in the block diagram.

To investigate how the open-loop flight-control system interacts with the guidance system, an adjoint was constructed based on the model of Fig. 23.5. The rms

Table 23.2 Additional linearized aerodynamics at two flight conditions

Parameter	Definition	Sea level	50 kft
T_α	Turning rate time constant	0.457 s	2.40 s
K_3	Aerodynamic body rate gain	-1.89 1/s	-.344 1/s
T_s	Seeker time constant	0.1 s	0.1 s
T_N	Noise filter time constant	0.1 s	0.1 s
V_c	Closing velocity	4000 ft/s	4000 ft/s
N'	Effective navigation ratio	3	3
t_F	Flight time	10 s	10 s

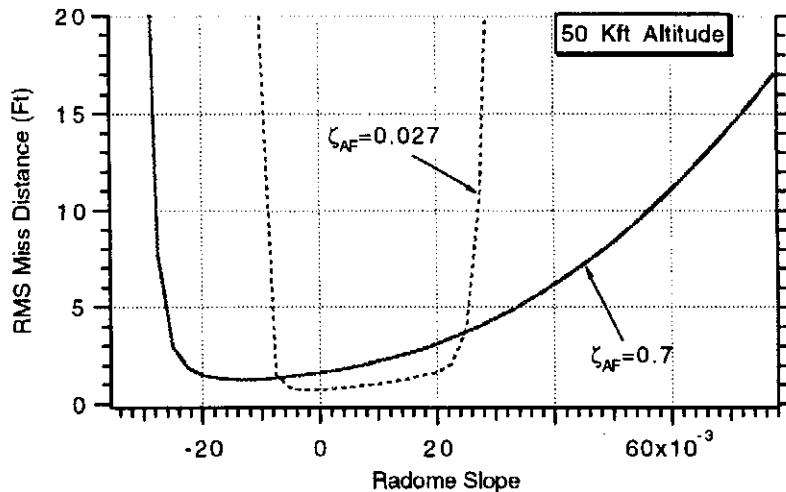


Fig. 23.6 Law airframe damping increases sensitivity to radome.

miss distance for a 10-s flight due to a 1-g uniformly distributed target maneuver (i.e., see Chapter 4 for more details) was evaluated for different radome slopes when the missile was traveling at 3000 ft/s and the target was traveling at 1000 ft/s at 50 kft altitude. The rms miss distance results are shown in Fig. 23.6. Here we can see that nominally the airframe damping is 0.027 (i.e., see Table 23.1) and that the rms miss distance can be enormous if the radome slope is more negative than -0.005 or more positive than 0.04 . However, if the airframe damping could somehow be increased to 0.7 we can also see from Fig. 23.6 that the system would be less sensitive to radome slope because the rms miss distance would only start to increase if the radome slope became more negative than -0.03 or more positive than 0.06 . In many radar homing applications the low damping of the airframe must artificially be increased so that a wider range of radome slopes can be tolerated. Therefore in these applications the low cost open-loop flight-control system would not be suitable.

Rate Gyro Flight-Control System

The rate gyro flight-control system artificially increases the low damping of the open-loop flight-control system by the use of a rate gyro sensor and the principles of feedback. In this flight-control system the rate gyro measures the missile body rate and feeds back this information to develop an error signal. The rate gyro flight-control system shown in Fig. 23.7 has two autopilot gains. The gain K_R will determine the response of the flight-control system while the gain K_{DC} controls the steady-state value of the response.

The transfer function from the achieved to the commanded acceleration can be obtained from Fig. 23.7. After some algebraic manipulations we obtain

$$\frac{n_L}{n_c} = \frac{K_{DC} K_1 K_R}{1 - K_R K_3} \left\{ \left[1 - \frac{s^2}{\omega_z^2} \right] \Big/ \left[1 + \frac{(2\zeta_{AF}/\omega_{AF}) - K_R K_3 T_\alpha}{1 - K_R K_3} s \right. \right. \\ \left. \left. + \frac{s^2}{\omega_{AF}^2 (1 - K_R K_3)} \right] \right\}$$

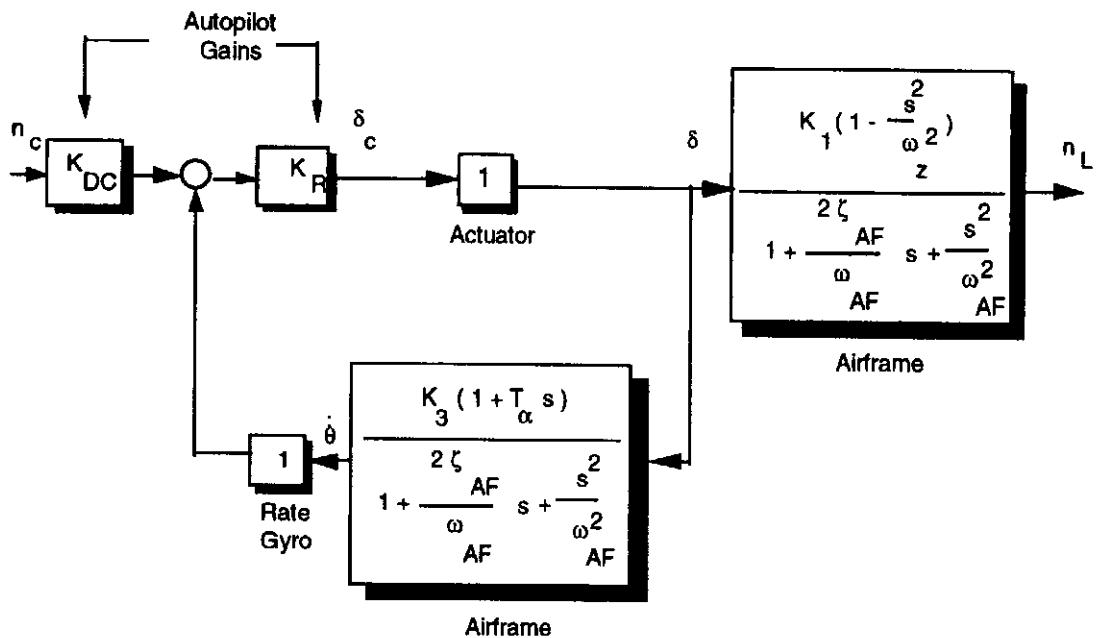


Fig. 23.7 Rate gyro flight-control system.

To get the achieved acceleration to match the commanded acceleration in the steady state, we can see from the preceding equation that the gain K_{DC} must be set to

$$K_{DC} = \frac{1 - K_R K_3}{K_1 K_R}$$

Because the denominator of the flight-control system is a quadratic, we can find the equivalent natural frequency ω and damping ζ of the rate gyro flight-control system by equating

$$1 + \frac{2\zeta}{\omega}s + \frac{s^2}{\omega^2} = 1 + \frac{(2\zeta_{AF}/\omega_{AF}) - K_R K_3 T_\alpha}{1 - K_R K_3} s + \frac{s^2}{\omega_{AF}^2(1 - K_R K_3)}$$

Solving for the frequency and damping of the rate gyro flight-control system yields

$$\omega = \omega_{AF} \sqrt{1 - K_R K_3}$$

$$\zeta = \frac{\omega}{2} \left[\frac{(2\zeta_{AF}/\omega_{AF}) - K_R K_3 T_\alpha}{1 - K_R K_3} \right]$$

Therefore we can see that the rate gyro autopilot gain K_R influences both the flight-control system frequency and damping.

Figures 23.8 and 23.9 show that increasing the autopilot gain K_R increases the damping of the flight-control system above that of the bare airframe both at low and high altitudes. At sea level an autopilot gain of approximately 0.1 is required to increase the damping of the rate gyro flight-control system to unity whereas at 50 kft altitude a gain of approximately 0.25 is required to get the same damping. In other words the autopilot gain must be changed with flight condition to ensure adequate damping in order to desensitize the guidance system to radome slope effects. Changing the autopilot gain or gains with flight condition is known as gain scheduling. It is important to note that for the rate gyro autopilot the autopilot gain

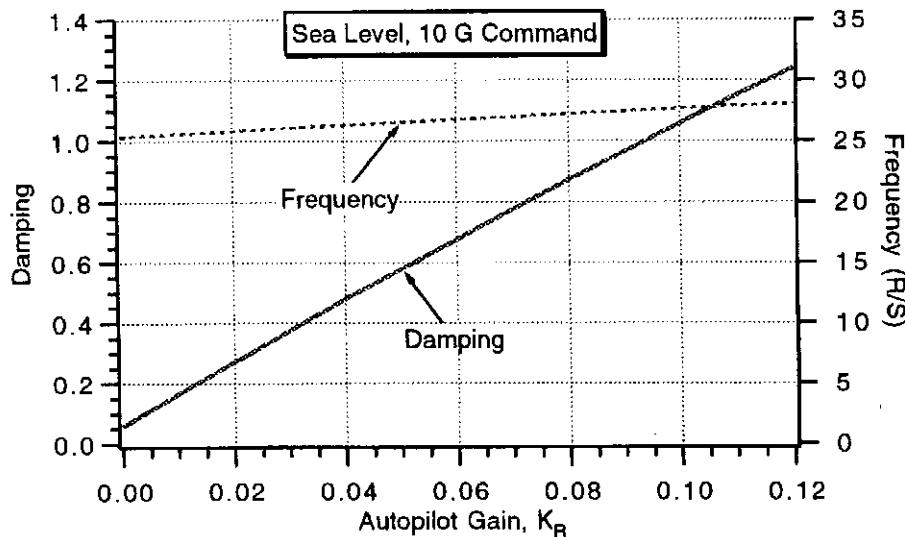


Fig. 23.8 Autopilot gain determines damping of flight-control system at sea level.

does not cause the natural frequency of the flight-control system to be significantly different from the airframe's natural frequency.

Listing 23.2 presents the simulation of the rate gyro flight-control system. In this simulation parameters are first expressed in terms of the geometry of the airframe and then the linearized aerodynamic parameters are derived. Note that the simulation integration interval is very small (i.e., $h = 0.0001$ s). If the integration interval were 10 times larger, the system would appear to go unstable when the autopilot gain is large (i.e., $K_R = 5$). However this instability is simply due to numerical integration and can be corrected by choosing a smaller integration interval. In this simplified model of the world, the autopilot gain can be made arbitrarily large without causing an instability in the guidance system. The specific equations for the rate gyro flight-control system appear after statement label 200.

The simulation of the linear rate gyro flight-control system of Listing 23.2 was run for the cases in which the missile was traveling at 3000 ft/s at both sea level and 50 kft altitude. An autopilot gain of 0.1 (i.e., $XKR = 0.1$) was chosen at sea

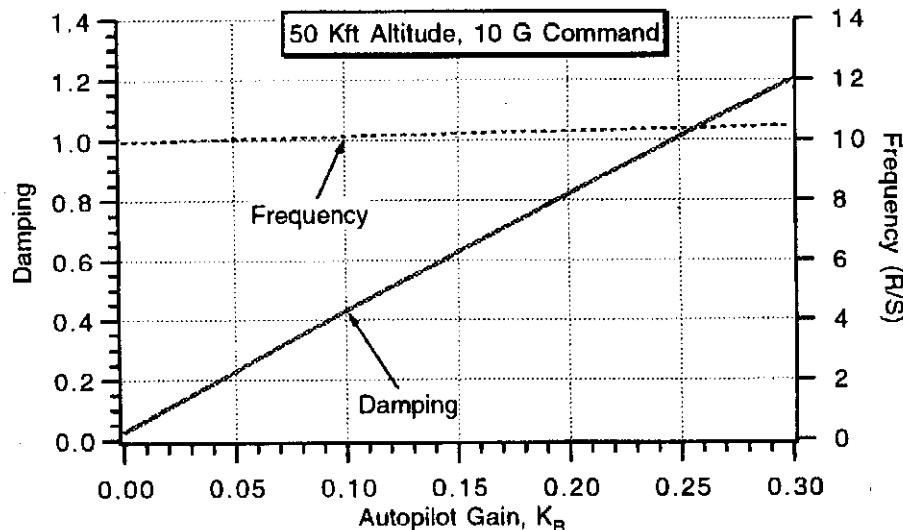


Fig. 23.9 Autopilot gain also determines damping of flight-control system at 50 Kft. altitude.

Listing 23.2 Rate gyro flight-control system in presence of linear airframe

```

VM=3000.
XNCG=10.
ALT=0.
A=1000.
DIAM=1.
FR=3.
XL=20.
CTW=0.
CRW=6.
HW=2.
CTT=0.
CRT=2.
HT=2.
XN=4.
XCG=10.
XHL=19.5
WGT=1000.
XKR=.1
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
IF(ALT.LE.30000.)THEN
    RHO=.002378*EXP(-ALT/30000.)
ELSE
    RHO=.0034*EXP(-ALT/22000.)
ENDIF
SWING=.5*HW*(CTW+CRW)
STAIL=.5*HT*(CTT+CRT)
SREF=3.1416*DIAM*DIAM/4.
XLP=FR*DIAM
SPLAN=(XL-XLP)*DIAM+1.33*XLP*DIAM/2.
XCPN=2*XLP/3
AN=.67*XLP*DIAM
AB=(XL-XLP)*DIAM
XCPB=(.67*AN*XLP+AB*(XLP+.5*(XL-XLP)))/(AN+AB)
XCPW=XLP+XN+.7*CRW-.2*CTW
XMACH=VM/A
XIYY=WGT*(3*((DIAM/2)**2)+XL*XL)/(12*32.2)
TMP1=(XCG-XCPW)/DIAM
TMP2=(XCG-XHL)/DIAM
TMP3=(XCG-XCPB)/DIAM
TMP4=(XCG-XCPN)/DIAM
B=SQRT(XMACH**2-1)
Q=.5*RHO*VM*VM
P1=WGT*XNCG/(Q*SREF)
Y1=2.+8*SWING/(B*SREF)+8*STAIL/(B*SREF)
Y2=1.5*SPLAN/SREF
Y3=8*STAIL/(B*SREF)
Y4=2*TMP4+8*SWING*TMP1/(B*SREF)+8*STAIL*TMP2/(B*SREF)
Y5=1.5*SPLAN*TMP3/SREF
Y6=8*STAIL*TMP2/(B*SREF)
P2=Y2-Y3*Y5/Y6
P3=Y1-Y3*Y4/Y6
ALFTR=(-P3+SQRT(P3*P3+4.*P2*P1))/(2.*P2)

```

Listing 23.2 (Continued)

```

DELTR=-Y4*ALFTR/Y6-Y5*ALFTR*ALFTR/Y6
CNA=2+1.5*SPLAN*ALFTR/SREF+8*SWING/(B*SREF)+8*STAIL/(B*SREF)
CND=8*STAIL/(B*SREF)
CMAP=2*TMP4+1.5*SPLAN*ALFTR*TMP3/SREF+8*SWING*TMP1/
(B*SREF)
CMA=CMAP+8*STAIL*TMP2/(B*SREF)
CMD=8*STAIL*TMP2/(B*SREF)
XMA=Q*SREF*DIAM*CMA/XIYY
XMD=Q*SREF*DIAM*CMD/XIYY
ZA=-32.2*Q*SREF*CNA/(WGT*VM)
ZD=-32.2*Q*SREF*CND/(WGT*VM)
WZ=SQRT((XMA*ZD-ZA*XMD)/ZD)
WAF=SQRT(-XMA)
ZAF=.5*WAF*ZA/XMA
XK1=-VM*(XMA*ZD-XMD*ZA)/(1845*XMA)
XK2=XK1
TA=XMD/(XMA*ZD-XMD*ZA)
XK3=1845*XK1/VM
XKDC=(1.-XKR*XK3)/(XK1*XKR)
E=0.
ED=0.
T=0
H=.0001
S=0
10 IF (T>.99999) GOTO 999
S=S+H
EOLD=E
EDOLD=ED
STEP=1
GOTO 200
66 STEP=2
E=E+H*ED
ED=ED+H*EDD
T=T+H
GOTO 200
55 CONTINUE
E=.5*(EOLD+E+H*ED)
ED=.5*(EDOLD+ED+H*EDD)
IF(S.LE..0099999)GOTO 10
S=0.
WRITE(1,97)T,XNL,XNCG
WRITE(*,97)T,XNL,XNCG
GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
DEL=XKR*(XKDC*XNCG+THD)
EDD=WAF*WAF*(DEL-E-2.*ZAF*ED/WAF)
XNL=XK1*(E-EDD/WZ**2)
THD=XK3*(E+TA*ED)
IF(STEP-1)66,66,55
999 CONTINUE
PAUSE
END

```

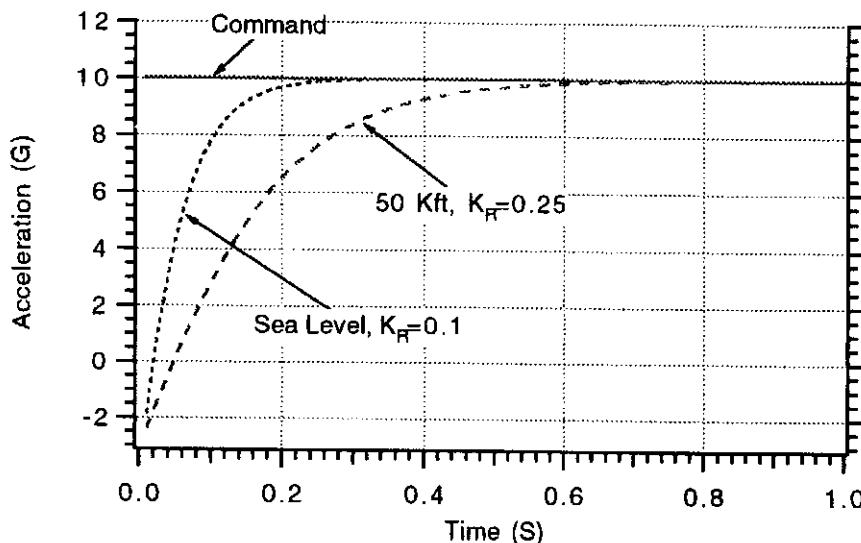


Fig. 23.10 Linear rate gyro flight-control system response is well damped when autopilot gain is a function of flight condition.

level and a gain of 0.25 (i.e., $XKR = 0.25$) was chosen at 50 kft in order to ensure that the flight-control system damping was approximately unity. We can see from Fig. 23.10 that although the desired damping is achieved at both flight conditions the response is more sluggish at the higher altitude. We shall see in Chapter 24 that a more advanced flight-control system will be required to control both the damping and the time constant at the same time.

To make the simulation of the flight-control system more realistic, a simplified model of the actuator was used. Nominally the second-order actuator, shown in Fig. 23.11, has a natural frequency of 150 rad/s and a damping of 0.7 (i.e., $\omega_{ACT} = 150$ rad/s, $\zeta_{ACT} = 0.7$).

Figure 23.12 shows then when the dynamics of the actuator are included in the simulation of the rate gyro flight-control system the step response only changes

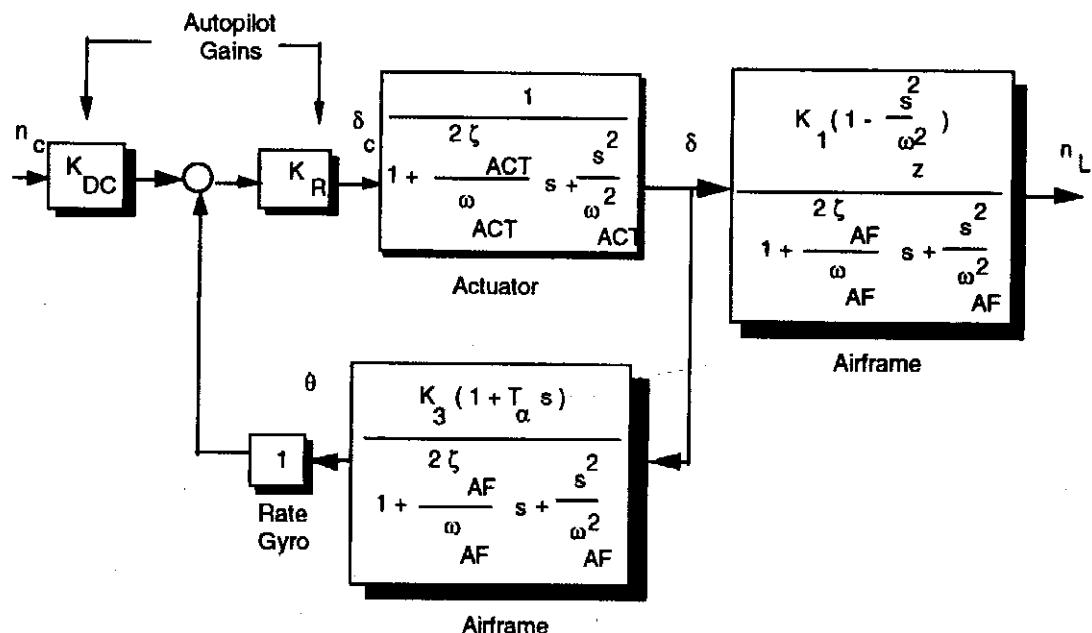


Fig. 23.11 Rate gyro flight-control system with actuator dynamics.

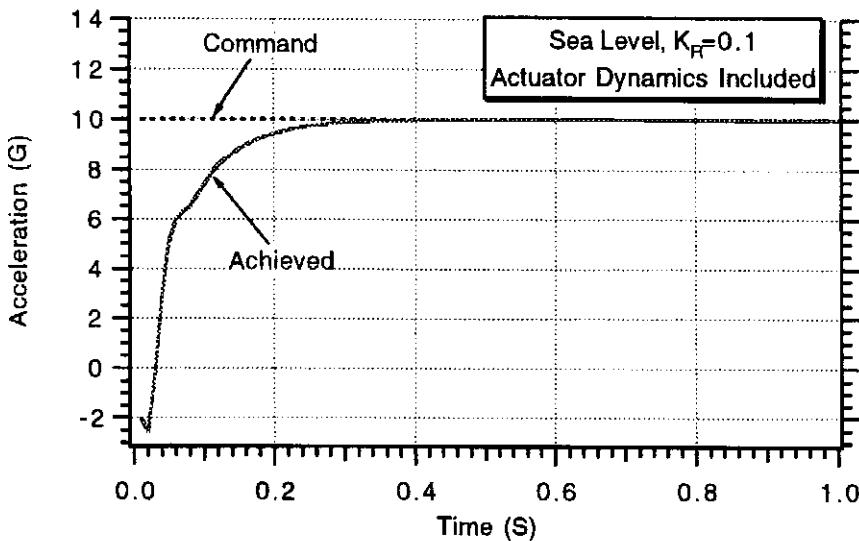


Fig. 23.12 Inclusion of actuator dynamics appears to have very little effect on flight-control system response.

slightly. It appears that the inclusion of the actuator dynamics has very little effect on the flight-control system response.

Experiments were conducted to see how large the autopilot gain could be made when actuator dynamics were included. If we increase the autopilot gain to 0.4, we can see from Fig. 23.13 that the flight-control system response goes unstable. Because there are 24 peaks in 1 s, the frequency of oscillation is 24 Hz or 150 rad/s. Therefore we can see empirically that actuator dynamics place an upper limit on the achievable autopilot gain with the rate gyro flight-control system. When the actuator dynamics were neglected, we could have made the autopilot gain arbitrarily large.

Open-Loop Transfer Function

We have seen through out this text that valuable information is available from the time domain simulation of the system differential equations. However, additional

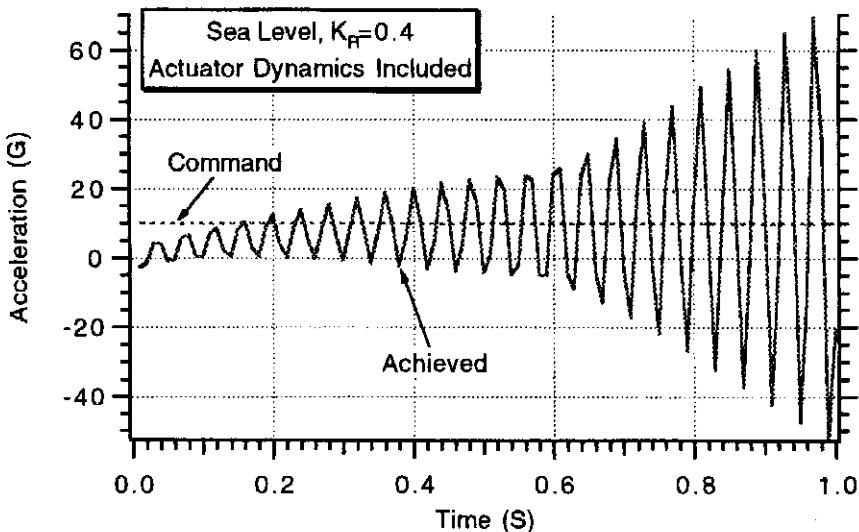


Fig. 23.13 Actuator dynamics have a destabilizing effect when autopilot gain is large.

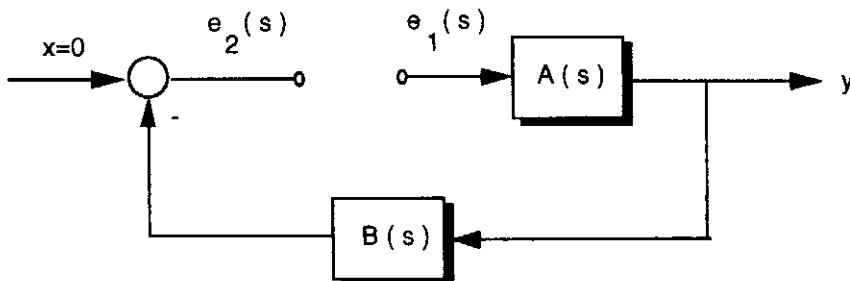


Fig. 23.14 Sample open-loop system.

information is also available from the systems open-loop transfer function, which is in the frequency domain. The concept of the open-loop transfer function is the basis of classical feedback control systems analysis.¹ Both relative stability and robustness can be determined from an analysis of the magnitude and phase of the open-loop frequency response, and, even more importantly, the designer can determine from it what changes to make in order to achieve design goals. The open-loop transfer function is the transfer function around the loop when the loop is broken at a point. Although the loop can be broken anywhere, it is usually broken in series with some parameter whose value the designer can control to achieve a desired characteristic. For example, we can break the loop of a single-loop feedback control system at the error signal as shown in Fig. 23.14. In this case the open-loop transfer is defined as

$$HG(s) = -\frac{e_2(s)}{e_1(s)} = A(s)B(s)$$

To fully understand open-loop concepts, it is first required to understand the mechanics of finding the magnitude and phase of an open-loop transfer function. This can be done by replacing the complex frequency s in the transfer function with

$$s = j\omega$$

where

$$j = \sqrt{-1}$$

Usually the magnitude of the open-loop transfer function is expressed in dB where

$$\text{dB} = 20 \log_{10}(\text{Magnitude})$$

and the phase is expressed in degrees.

With the open-loop transfer function other quantities are also important. For example, the gain margin (gm) is the value of additional gain required at the loop break (assuming the phase remains constant) to cause instability while the phase margin ϕ_{pm} is the amount of phase loss required at the loop break (assuming that the gain remains constant) to cause instability. In addition to these margins, crossover frequencies are also of interest. The gain crossover frequency ω_{CR} is the frequency at which the open-loop magnitude is unity or zero dB, while the phase crossover frequency ω_{180} is the frequency at which the open-loop phase is ~ 180 deg. Both these crossover frequencies indicate the frequency of the ensuing

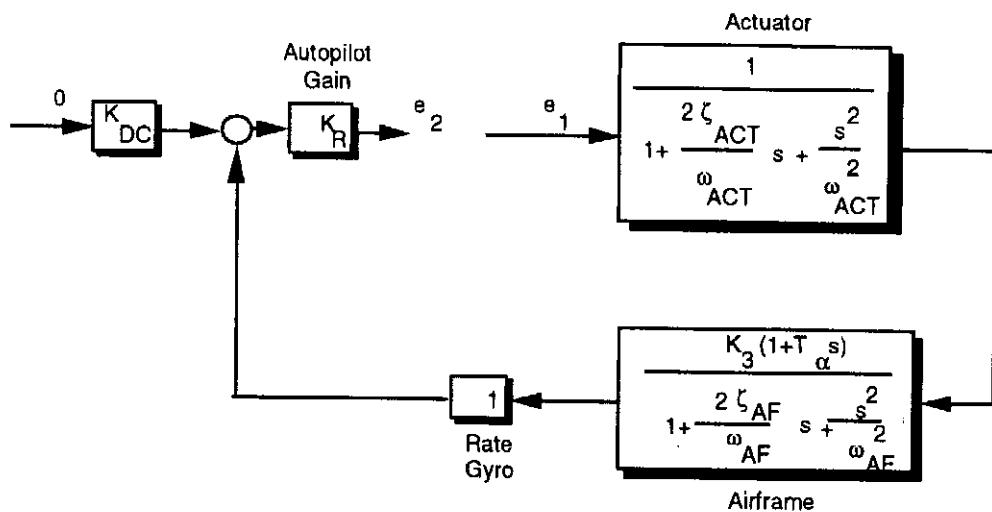


Fig. 23.15 Open-loop model of rate gyro flight-control system.

oscillation in the time domain should the system go unstable due to either an increase in gain or decrease in phase.

To demonstrate the utility of the open-loop transfer function, let us revisit the rate gyro flight-control system of Fig. 23.11. Figure 23.15 shows the same system, except this time the loop is broken at the input to the actuator. The loop is broken here because the designer can control the autopilot gain K_R . From the definition of open-loop transfer function, we can express $HG(s)$ as

$$HG(s) = -K_3 K_R (1 + T_\alpha s) \Big/ \left\{ \left[1 + \frac{2\zeta_{ACT}}{\omega_{ACT}} s + \frac{s^2}{\omega_{ACT}^2} \right] \left[1 + \frac{2\zeta_{AF}}{\omega_{AF}} s + \frac{s^2}{\omega_{AF}^2} \right] \right\}$$

By going to the complex frequency domain, we can rewrite the open-loop transfer function as

$$HG(j\omega) = -K_3 K_R (1 + j\omega T_\alpha) \Big/ \left\{ \left[1 - \frac{\omega^2}{\omega_{ACT}^2} + \frac{j2\zeta_{ACT}\omega}{\omega_{ACT}} \right] \left[1 - \frac{\omega^2}{\omega_{AF}^2} + \frac{j2\zeta_{AF}\omega}{\omega_{AF}} \right] \right\}$$

where care has been taken in the preceding equation to separate the real and imaginary parts. The magnitude and phase of the open-loop transfer function can now be expressed as

$$|HG(j\omega)| = -K_R K_3$$

$$\times \sqrt{1 + \omega^2 T_\alpha^2} \Big/ \left\{ \left[\left[1 - \frac{\omega^2}{\omega_{ACT}^2} \right]^2 + \left[\frac{2\zeta_{ACT}\omega}{\omega_{ACT}} \right]^2 \right] \left[\left[1 - \frac{\omega^2}{\omega_{AF}^2} \right]^2 + \left[\frac{2\zeta_{AF}\omega}{\omega_{AF}} \right]^2 \right] \right\}$$

$$\angle HG(j\omega) = \tan^{-1} \omega T_\alpha$$

$$- \tan^{-1} \left[\frac{2\zeta_{ACT}\omega}{\omega_{ACT}} \Big/ \left(1 - \frac{\omega^2}{\omega_{ACT}^2} \right) \right] - \tan^{-1} \left[\frac{2\zeta_{AF}\omega}{\omega_{AF}} \Big/ \left(1 - \frac{\omega^2}{\omega_{AF}^2} \right) \right]$$

Listing 23.3 Open-loop Bode response for rate gyro flight-control system

```

REAL K3,KR
DATA ZACT,WACT,K3,TA,ZAF,WAF,KR/.7,150.,-1.89,.457,.058,25.3,.1/
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
DO 10 I=2,160
W=10**(.025*I-1)
XMAG1=SQRT(1+(W*TA)**2)
XMAG2=SQRT((1-(W/WAF)**2)**2+(2*ZAF*W/WAF)**2)
XMAG3=SQRT((1-(W/WACT)**2)**2+(2*ZACT*W/WACT)**2)
GAIN=20*LOG10(-K3*KR*XMAG1/(XMAG2*XMAG3))
PHASE1=57.3*ATAN2(W*TA,1.)
PHASE2=57.3*ATAN2(2*ZAF*W/WAF,1-(W/WAF)**2)
PHASE3=57.3*ATAN2(2*ZACT*W/WACT,1-(W/WACT)**2)
PHASE=PHASE1-PHASE2-PHASE3
WRITE(*,97)W,GAIN,PHASE
WRITE(1,97)W,GAIN,PHASE
10 CONTINUE
97 FORMAT(3F10.3)
CLOSE(1)
PAUSE
END

```

Therefore the open-loop gain (magnitude) and phase can be expressed in conventional units as

$$\text{Gain} = 20 \log_{10} |HG(j\omega)| \quad (\text{dB})$$

$$\text{Phase} = 57.3 \angle HG(j\omega) \quad (\text{deg})$$

Designers have found several useful ways of displaying open-loop data. One of these ways is a Bode plot in which the magnitude, expressed in dB, and phase, expressed in degrees, are displayed versus frequency on a logarithmic scale.² The preceding equations for the magnitude and phase of the rate gyro flight-control system were programmed in order to generate a Bode plot and the resultant program appears in Listing 23.3. Note that in this program we are incrementally updating the frequency logarithmically and then solving for the magnitude and phase. Unlike most other programs in this text, this program runs extremely rapidly because numerical integration is not involved.

Figure 23.16 presents the resultant Bode plot, using the data generated by the FORTRAN program. Here we can see that the gain (or magnitude) peaks due to the low airframe damping ($\zeta_{AF} = 0.058$) and then is quickly attenuated due to the dynamics of the actuator. At the gain crossover frequency (i.e., frequency at which gain is zero dB) the phase is -125 deg. Because the phase margin represents the phase departure from -180 deg, the phase margin is 55 deg ($180 - 125 = 55$). At the phase crossover frequency (i.e., frequency at which phase is -180 deg) the gain is -11.5 dB. Because the gain margin represents the gain departure from 0 dB, the gain margin is 11.5 dB. The various margins and crossover frequencies have important practical interpretations. For example, if the system phase is decreased by the phase margin the system will go unstable and oscillate at the gain

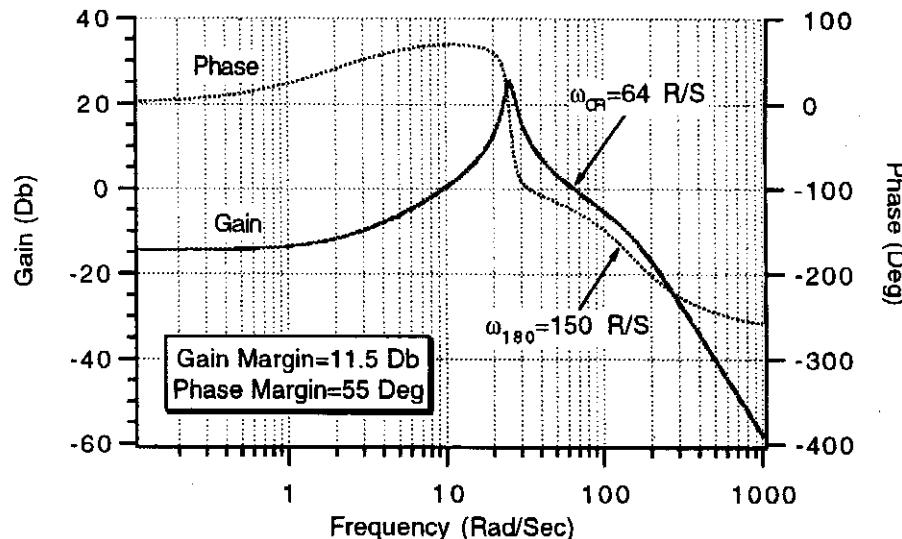


Fig. 23.16 Bode plot for rate gyro flight-control system.

crossover frequency. If the system gain is increased by the gain margin, the system will go unstable and oscillate at the phase crossover frequency. We can see from Fig. 23.16 that the gain and phase crossover frequencies are 64 rad/s and 150 rad/s respectively.

Time Domain Verification of Open-Loop Results³

The open-loop analysis of the previous section indicated that the system gain margin was 11.5 dB and the phase crossover frequency was 150 rad/s. This means that if the gain K_R was increased by 11.5 dB the system would go unstable and oscillate at 150 rad/s. A gain increase of 11.5 dB means that K_R must increase from 0.1 to 0.376 to destabilize the system. In other words,

$$20 \log_{10} \frac{K_{\text{UNSTABLE}}}{0.1} = 11.5$$

$$\log_{10} \frac{K_{\text{UNSTABLE}}}{0.1} = 0.575$$

$$\frac{K_{\text{UNSTABLE}}}{0.1} = 10^{0.575} = 3.76$$

$$K_{\text{UNSTABLE}} = .376$$

We have already seen from Fig. 23.13 that when the autopilot gain was increased to 0.4 the rate gyro flight-control system indeed oscillated at 150 rad/s. This means that the time domain and frequency domain results are in total agreement.

We can also illustrate the concept of phase margin by first observing that an ideal delay can be represented by the transfer function

$$\text{DELAY} = e^{-sT}$$

Converting this representation to the complex frequency domain yields

$$\text{DELAY}(j\omega) = e^{-j\omega T} = \cos \omega T - j \sin \omega T$$

The magnitude and phase of the ideal delay is therefore

$$|\text{DELAY}(j\omega)| = (\cos^2 \omega T + \sin^2 \omega T)^{\frac{1}{2}} = 1$$

$$\angle \text{DELAY}(j\omega) = \tan^{-1} \left[\frac{\sin \omega T}{\cos \omega T} \right] = -\omega T$$

In summary, an ideal delay can be represented in the frequency domain as a transfer function with unity magnitude and pure phase loss. The phase loss at 64 rad/s (open-loop gain crossover frequency ω_{CR}) in units of rad can be obtained from the preceding equation as

$$\text{DELAY PHASE LOSS} = -64T$$

Therefore a delay of 0.015 s in the time domain corresponds to a phase loss of 55 deg (i.e., $64 * 0.015 = 0.96 \text{ rad} = 55 \text{ deg} = \text{phase margin}$). Because the phase margin of the open-loop system (with the loop broken at K_R) is 55 deg, this means that if a pure delay of 0.015 s were inserted in series with K_R , the system would go unstable and oscillate at a frequency of 64 rad/s (i.e., open-loop gain crossover frequency). The rate gyro flight-control time domain simulation of Listing 23.2 was modified to include a pure time delay of 0.015 s, and the new simulation appears in Listing 23.4. Again note that a very small integration step size is used to avoid numerical difficulties.

The system step response is shown in Fig. 23.17. Here we can see that the system does go unstable when a delay of 0.015 s is inserted before the actuator. Because there are 10 peaks in the response, the frequency of oscillation is 10 Hz or 62.8 rad/s, which is approximately the gain crossover frequency. Thus again we can see that the time and frequency domain results are in total agreement.

We have just seen that there is a relationship between the time and frequency domains. The gain margin of the open-loop response tells us how much the gain of the flight-control system can be increased by in order for the system to go unstable. When the system oscillates in the time domain due to the increased gain, the frequency of oscillation will be the phase crossover frequency of the open-loop

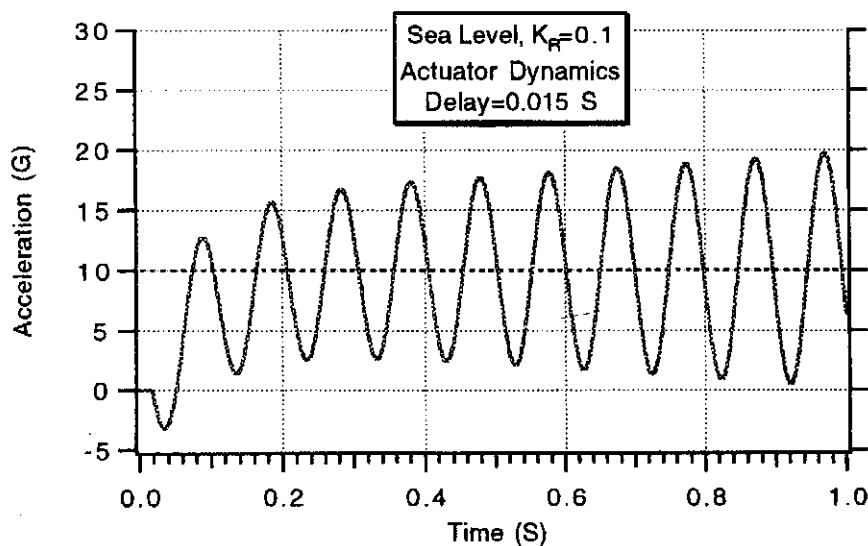


Fig. 23.17 Rate gyro flight-control system goes unstable when delay of 0.015 s is inserted.

Listing 23.4 Rate gyro flight-control system with pure delay

```

INTEGER STEP,DINT
DIMENSION Z(0:20002)
DELAY=.015
VM=3000.
XNCG=10.
ALT=0.
A=1000.
DIAM=1.
FR=3.
XL=20.
CTW=0.
CRW=6.
HW=2.
CTT=0.
CRT=2.
HT=2.
XN=4.
XCG=10.
XHL=19.5
WGT=1000.
XKR=.1
WACT=150.
ZACT=.7
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
IF(ALT.LE.30000.)THEN
    RHO=.002378*EXP(-ALT/30000.)
ELSE
    RHO=.0034*EXP(-ALT/22000.)
ENDIF
SWING=.5*HW*(CTW+CRW)
STAIL=.5*HT*(CTT+CRT)
SREF=3.1416*DIAM*DIAM/4.
XLP=FR*DIAM
SPLAN=(XL-XLP)*DIAM+1.33*XLP*DIAM/2.
XCPN=2*XLP/3
AN=.67*XLP*DIAM
AB=(XL-XLP)*DIAM
XCPB=(.67*AN*XLP+AB*(XLP+.5*(XL-XLP)))/(AN+AB)
XCPW=XLP+XN+.7*CRW-.2*CTW
XMACH=VM/A
XIYY=WGT*(3*((DIAM/2)**2)+XL*XL)/(12*32.2)
TMP1=(XCG-XCPW)/DIAM
TMP2=(XCG-XHL)/DIAM
TMP3=(XCG-XCPB)/DIAM
TMP4=(XCG-XCPN)/DIAM
B=SQRT(XMACH**2-1)
Q=.5*RHO*VM*VM
P1=WGT*XNCG/(Q*SREF)
Y1=2.+8*SWING/(B*SREF)+8*STAIL/(B*SREF)
Y2=1.5*SPLAN/SREF
Y3=8*STAIL/(B*SREF)

```

(Contd.)

Listing 23.4 (Continued)

```

Y4=2*TMP4+8*SWING*TMP1/(B*SREF)+8*STAIL*TMP2/(B*SREF)
Y5=1.5*SPLAN*TMP3/SREF
Y6=8*STAIL*TMP2/(B*SREF)
P2=Y2-Y3*Y5/Y6
P3=Y1-Y3*Y4/Y6
ALFTR=(-P3+SQRT(P3*P3+4.*P2*P1))/(2.*P2)
DELTR=-Y4*ALFTR/Y6-Y5*ALFTR*ALFTR/Y6
CNA=2+1.5*SPLAN*ALFTR/SREF+8*SWING/(B*SREF)+8*STAIL/(B*SREF)
CND=8*STAIL/(B*SREF)
CMAP=2*TMP4+1.5*SPLAN*ALFTR*TMP3/SREF+8*SWING*TMP1/
(B*SREF)
CMA=CMAP+8*STAIL*TMP2/(B*SREF)
CMD=8*STAIL*TMP2/(B*SREF)
XMA=Q*SREF*DIAM*CMA/XIYY
XMD=Q*SREF*DIAM*CMD/XIYY
ZA=-32.2*Q*SREF*CNA/(WGT*VM)
ZD=-32.2*Q*SREF*CND/(WGT*VM)
WZ=SQRT((XMA*ZD-ZA*XMD)/ZD)
WAF=SQRT(-XMA)
ZAF=.5*WAF*ZA/XMA
XK1=VM*(XMA*ZD-XMD*ZA)/(1845*XMA)
XK2=XK1
TA=XMD/(XMA*ZD-XMD*ZA)
XK3=1845*XK1/VM
XKDC=(1.-XKR*XK3)/(XK1*XKR)
E=0.
ED=0.
DELD=0.
DEL=0.
THD=0.
DELDC=0.
DELCP=0.
T=0
H=.0001
Z(1)=0.
I=1
DINT=INT(DELAY/H)
S=0
10 IF (T>.99999) GOTO 999
S=S+H
EOLD=E
EDOLD=ED
DELOLD=DEL
DELDOLD=DELD
STEP=1
GOTO 200
66 STEP=2
E=E+H*ED
ED=ED+H*EDD
DEL=DEL+H*DELD
DELD=DELD+H*DELDD

```

(Contd.)

Listing 23.4 (Continued)

```

T=T+H
GOTO 200
55  CONTINUE
E=.5*(EOLD+E+H*ED)
ED=.5*(EDOLD+ED+H*EDD)
DEL=.5*(DELOLD+DEL+H*DEL)
DELD=.5*(DELDOLD+DELD+H*DELDD)
Z(I+1)=DELCP
IF((I+1)<DINT)THEN
    DELC=Z(1)
ELSE
    DELC=Z(I+1-DINT)
ENDIF
I=I+1
IF(S.LE..00099999)GOTO 10
S=0.
WRITE(1,97)T,XNL,XNCG
WRITE(*,97)T,XNL,XNCG
GOTO 10
97  FORMAT(3F10.3)
200  CONTINUE
DELCP=XKR*(XKDC*XNCG+THD)
DELDD=WACT*WACT*(DELC-DEL-2.*ZACT*DELD/WACT)
EDD=WAF*WAF*(DEL-E-2.*ZAF*ED/WAF)
XNL=XK1*(E-EDD/WZ**2)
THD=XX3*(E+TA*ED)
IF(STEP-1)66,66,55
999 CONTINUE
CLOSE(1)
PAUSE
END

```

response. The phase margin of the open-loop response tells us the amount of phase loss (or pure time delay) that the system can tolerate before it goes unstable. When the system oscillates in the time domain due to the phase loss (i.e., time delay), the frequency of oscillation will be the open-loop gain crossover frequency.

Simplified Expression for Open-Loop Crossover Frequency

We have seen that if the autopilot gain K_R is made too large the rate gyro flight-control system can go unstable. To place realistic bounds on the autopilot gain, it is necessary to go back to the frequency domain. If we neglect actuator dynamics, the open-loop transfer function of the rate gyro flight-control system can be written by inspection of Fig. 23.7 as

$$HG = -K_R K_3 (1 + T_\alpha s) \left/ \left(1 + \frac{2\zeta_{AF}}{\omega_{AF}} s + \frac{s^2}{\omega_{AF}^2} \right) \right.$$

The magnitude of the open-loop transfer function can be found from the preceding expression. Recall that the open loop gain crossover frequency occurs when the magnitude of the open-loop transfer function is unity. Therefore if we assume that the open-loop crossover frequency is beyond the airframe dynamics we can say that

$$1 \approx \frac{-K_R K_3 T_\alpha \omega_{CR} \omega_{AF}^2}{\omega_{CR}^2}$$

Solving for the crossover frequency yields

$$\omega_{CR} \approx -K_R K_3 T_\alpha \omega_{AF}^2 = K_R K_3 T_\alpha M_\alpha$$

Therefore we can see that open-loop crossover frequency is proportional to the autopilot gain. Increasing the autopilot gain will increase the crossover frequency. However it is important to note that since we have assumed that the crossover frequency is far beyond the airframe dynamics the preceding expression is approximate. For our 3000 ft/s missile at sea level, the preceding formula indicates that the approximate crossover frequency is 55.2 rad/s or

$$\omega_{CR} = 0.1 * 1.89 * 0.457 * 25.3^2 = 55.2 \text{ rad/s}$$

From our open-loop Bode response we already know that the actual crossover frequency is 60 rad/s. Therefore in this case the approximate expression for the open-loop crossover frequency is in error by slightly less than 10%. The technique for deriving crossover frequency will prove to be very useful when we deal with the three-loop autopilot in the next chapter.

We can simplify the expression for the open-loop crossover frequency even further by recalling that

$$\begin{aligned} K_1 &= \frac{V_M(M_\alpha Z_\delta - Z_\alpha M_\delta)}{1845 M_\alpha} \approx \frac{V_M Z_\delta}{1845} \\ K_3 &= \frac{1845 K_1}{V_M} \approx Z_\delta \\ T_\alpha &= \frac{M_\delta}{(M_\alpha Z_\delta - Z_\alpha M_\delta)} \approx \frac{M_\delta}{M_\alpha Z_\delta} \end{aligned}$$

Therefore substitution yields an even simpler expression for the crossover frequency

$$\omega_{CR} \approx K_R K_3 T_\alpha M_\alpha = -K_R M_\delta$$

Therefore at a given flight condition the autopilot gain and linearized aerodynamic parameter M_δ determine the open-loop crossover frequency. According to the preceding equation, doubling the autopilot gain from 0.1 to 0.2 should double the crossover frequency from 52.6 rad/s to 105 rad/s. If the open-loop frequency response program of Listing 23.3 is run with actuator dynamics (neglected in approximate analysis), we can see from Fig. 23.18 that the actual crossover frequency increases from 60 rad/s to 103 rad/s when the autopilot gain increases from 0.1 to 0.2. Thus we can see that at the higher crossover frequency, simulation results are more in agreement with the approximate analysis. The agreement is better

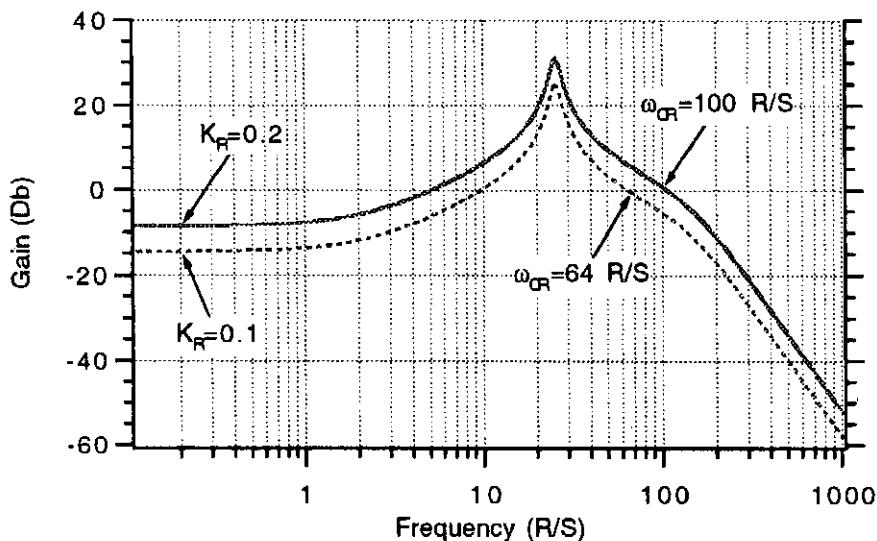


Fig. 23.18 Doubling autopilot gain approximately doubles crossover frequency.

because the crossover frequency is now far beyond the airframe dynamics. Later on we shall see that the crossover frequency is chosen to be no more than one third of the bandwidth of the actuator to ensure a well behaved flight-control system response.

Summary

In this chapter we have seen how the flight-control system interacts with the guidance system. The open-loop flight-control system has the dynamics of the bare airframe. This type of flight-control system is usually not acceptable in radar homing applications because of low damping. The rate gyro flight-control system improves the system damping by using a sensor and feedback. We have also seen that the autopilot gain in a rate gyro flight-control system also can be used to control the open-loop crossover frequency.

References

- ¹Del Toro, V., and Parker, S., *Principles of Control Systems Engineering*, McGraw-Hill, New York, 1960.
- ²D'Azzo, J. J., and Houpis, C. H., *Feedback Control System Analysis and Synthesis*, McGraw-Hill, New York, 1960.
- ³Zarchan, P., "Micro Based Technology—A New Tool for Missile Guidance System Design and Visualization," *AGARD Lecture Series 173*, Oct. 1990, pp 8.1–8.16.

Three-Loop Autopilot

Introduction

In this chapter we shall see that by adding an accelerometer to the flight-control system we can independently select the system damping, time constant, and open-loop crossover frequency. Controlling the system damping will ensure that the guidance system is not overly sensitive to radome slope effects at the high altitudes. Selecting the system time constant means that we can have adequate performance against maneuvering targets. Finally, controlling the open-loop crossover frequency means that we will have a robust design that is not overly sensitive to unmodeled high frequency dynamics.

Three-Loop Autopilot Configuration

The flight-control system with the three-loop autopilot appears in Fig. 24.1. In this system the rate gyro feeds body rate information into the autopilot while the accelerometer feeds back achieved acceleration information. For simplicity, it has been assumed that the accelerometer location is at the missile center of gravity so that the acceleration sensed is the true acceleration. The three autopilot gains K_A , ω_I , and K_R must be chosen to satisfy some designer-chosen criteria and the gain K_{DC} is computed from the other gains so that the achieved acceleration will match the commanded acceleration. An interesting discussion of the initial design considerations of the three-loop autopilot can be found in Ref. 1.

An example of a particularly useful methodology²⁻⁶ in gain selection is to choose the open-loop crossover frequency so that many stability problems can be avoided. In addition, the dominant flight-control system time constant can be selected so that rapid speed of response can be achieved in order to hit maneuvering targets. Finally, adequate damping can also be chosen by the designer to alleviate potential radome coupling problems.

Open-Loop Analysis

Because the open-loop crossover frequency has no meaning in the time domain, we must first shift to the frequency domain to see how the autopilot gains influence the crossover frequency. Figure 24.2 shows the three-loop autopilot with the loop broken right before the actuator as was done in Chapter 23. By inspection of Fig. 24.2, we can write an expression for the open loop transfer function $HG(s)$ as

$$HG(s) = -\frac{y}{x} = -K_R \left[G_3 + \frac{G_3 \omega_I}{s} + \frac{G_1 K_A \omega_I}{s} \right]$$

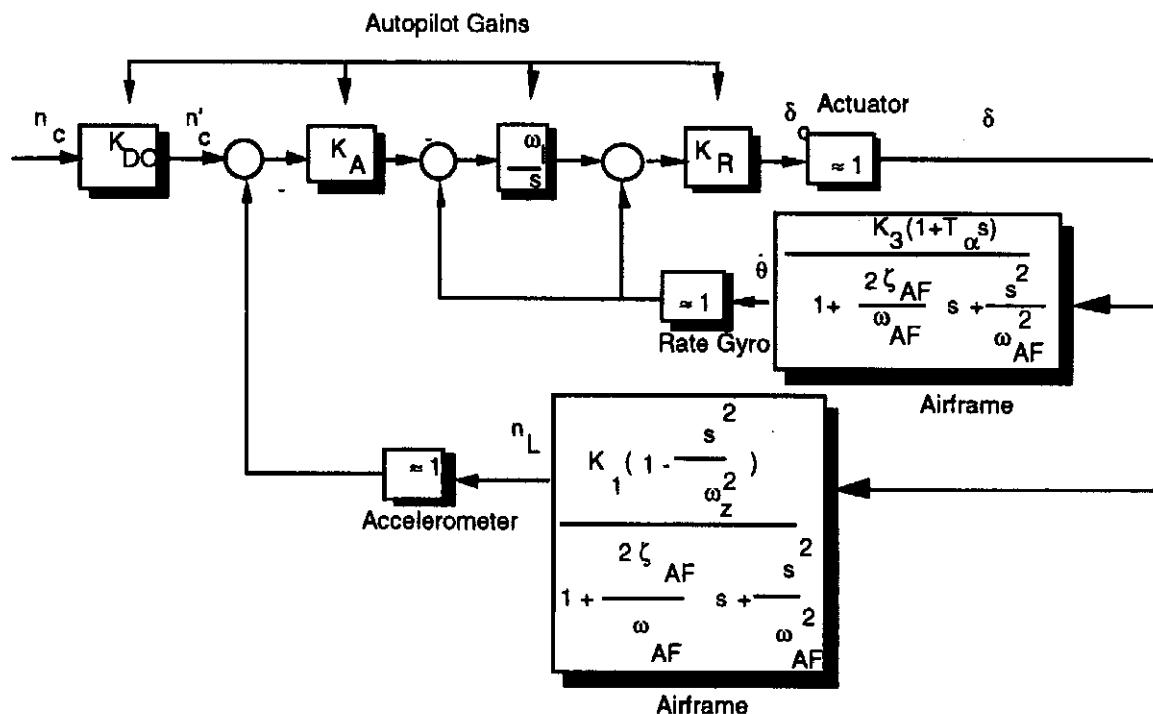


Fig. 24.1 Flight-control system with three-loop autopilot.

where G_1 and G_3 are shorthand notation for the airframe transfer functions that were derived in Chapter 22 and are given by

$$G_1 = \frac{n_L}{\delta} = K_1 \left(1 - \frac{s^2}{\omega_z^2} \right) / \left(1 + \frac{2\zeta_{AF}}{\omega_{AF}}s + \frac{s^2}{\omega_{AF}^2} \right)$$

$$G_3 = \frac{\dot{\theta}}{\delta} = K_3(1 + T_\alpha s) / \left(1 + \frac{2\zeta_{AF}}{\omega_{AF}}s + \frac{s^2}{\omega_{AF}^2} \right)$$

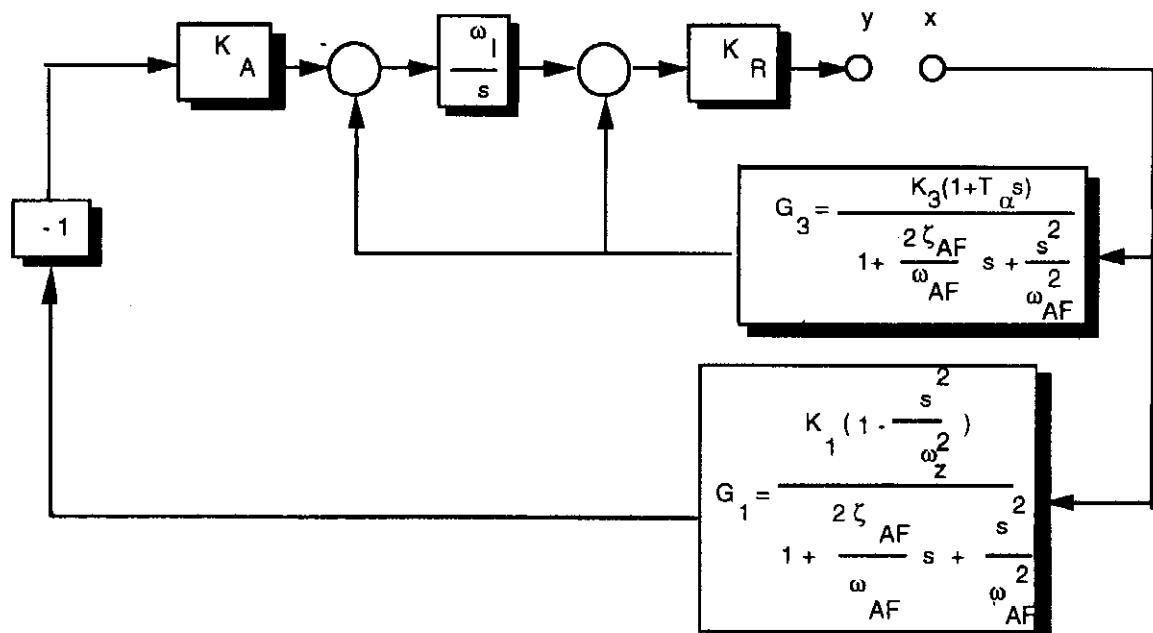


Fig. 24.2 Open-loop representation of three-loop autopilot.

After much algebra the open-loop transfer function can be rewritten as

$$HG = -K_R \omega_I K_A \left(\frac{K_3}{K_A} + K_1 \right) \left/ \left[s \left(1 + \frac{2\zeta_{AF}}{\omega_{AF}} s + \frac{s^2}{\omega_{AF}^2} \right) \right] \right. \\ \times \left[1 + \frac{s(K_3 + \omega_I K_3 T_\alpha)}{\omega_I K_3 + K_A \omega_I K_1} + \frac{s^2 [K_3 T_\alpha - (K_A \omega_I K_1 / \omega_z^2)]}{\omega_I K_3 + K_A \omega_I K_1} \right]$$

which simplifies to

$$HG = -K_0 \left(1 + \frac{2\zeta_0}{\omega_0} s + \frac{s^2}{\omega_0^2} \right) \left/ \left[s \left(1 + \frac{2\zeta_{AF}}{\omega_{AF}} s + \frac{s^2}{\omega_{AF}^2} \right) \right] \right.$$

where the gain K_0 is given by

$$K_0 = K_R \omega_I K_A \left(\frac{K_3}{K_A} + K_1 \right)$$

and the numerator coefficients can be expressed in terms of the autopilot gains and aerodynamic parameters according to

$$\frac{2\zeta_0}{\omega_0} = \frac{(K_3 + \omega_I K_3 T_\alpha)}{\omega_I K_3 + K_A \omega_I K_1}$$

$$\frac{1}{\omega_0^2} = \frac{[K_3 T_\alpha - (K_A \omega_I K_1 / \omega_z^2)]}{\omega_I K_3 + K_A \omega_I K_1}$$

If we define

$$K_c = \frac{K_3}{K_A K_1}$$

then

$$\frac{2\zeta_0}{\omega_0} = \frac{(K_c / \omega_I) + K_c T_\alpha}{1 + K_c}$$

$$\frac{1}{\omega_0^2} = \frac{-(1/\omega_z^2) + (K_c T_\alpha / \omega_I)}{1 + K_c}$$

If we define the intermediate gain K by

$$K = K_R \omega_I K_A$$

We can say that

$$K_0 = K_R \omega_I K_A [(K_3 / K_A) + K_1] = K(K_1 K_c + K_1) = K K_1 (1 + K_c)$$

If we assume that the crossover frequency is beyond the airframe dynamics, we can set the magnitude of the open-loop crossover frequency to unity as we did in the previous chapter in order to obtain

$$1 \approx \frac{-(K_0 \omega_{CR}^2 / \omega_0^2)}{(\omega_{CR} \omega_{CR}^2 / \omega_{AF}^2)} = \frac{-K_0 \omega_{AF}^2}{\omega_{CR} \omega_0^2}$$

Solving for the open-loop crossover frequency yields

$$\omega_{CR} = \frac{-K_0 \omega_{AF}^2}{\omega_0^2}$$

From the preceding equation we can see that the open-loop crossover frequency is a function of both the aerodynamics and the autopilot gains.

Closed-Loop Analysis

Now we can go back to the time domain to complete the autopilot design. By inspection of Fig. 24.1, we can write an expression for the relationship between the output control surface deflection and the input acceleration command as

$$\frac{\delta}{n'_c} = \frac{-K_A K_R \omega_I / s}{1 - K_R G_3 - (K_R \omega_I G_3 / s) - (K_R \omega_I K_A G_1 / s)}$$

Rewriting the preceding expression in terms of the open-loop transfer function yields

$$\frac{\delta}{n'_c} = \frac{-K_A K_R \omega_I / s}{1 + HG}$$

The relationship between the flight-control system output acceleration and the input command can then be obtained from the chain rule as

$$\begin{aligned} \frac{n_L}{n'_c} &= \frac{\delta}{n'_c} * \frac{n_L}{\delta} = \left[\frac{-K_A K_R \omega_I / s}{1 + HG} \right] \\ &\times \left[K_1 \left(1 - \frac{s^2}{\omega_z^2} \right) \middle/ \left(1 + \frac{2\zeta_{AF}}{\omega_{AF}} s + \frac{s^2}{\omega_{AF}^2} \right) \right] \end{aligned}$$

After much algebra one can show that the flight-control system transfer function becomes

$$\frac{n_L}{n'_c} = \frac{K_A K_R \omega_I K_1 \left(1 - \frac{s^2}{\omega_z^2} \right) \middle/ K_0}{1 + s \left[\frac{2\zeta_0}{\omega_0} - \frac{1}{K_0} \right] + s^2 \left[\frac{1}{\omega_0^2} - \frac{2\zeta_{AF}}{\omega_{AF} K_0} \right] - \frac{s^3}{\omega_{AF}^2 K_0}}$$

We are not interested in controlling the airframe zeros (i.e., numerator in flight-control system transfer function) but would like to have the preceding closed-loop transfer function to have the form

$$\frac{n_L}{n'_c} = \frac{K_A K_R \omega_I K_1 \left(1 - \frac{s^2}{\omega_z^2} \right) \middle/ K_0}{(1 + \tau s) \left[1 + \frac{2\zeta s}{\omega} + \frac{s^2}{\omega^2} \right]}$$

where we have made the denominator of the flight-control system transfer function a real pole times a quadratic. The two preceding flight-control system transfer

functions are equivalent if the denominators are the same or

$$\begin{aligned}\frac{2\zeta}{\omega} + \tau &= \frac{2\zeta_0}{\omega_0} - \frac{1}{K_0} \\ \frac{1}{\omega^2} + \frac{2\zeta\tau}{\omega} &= \frac{1}{\omega_0^2} - \frac{2\zeta_{AF}}{\omega_{AF}K_0} \\ \frac{\tau}{\omega^2} &= -\frac{1}{\omega_{AF}^2 K_0}\end{aligned}$$

In addition, we have already derived an expression for the open-loop crossover frequency to be

$$\omega_{CR} = \frac{-K_0 \omega_{AF}^2}{\omega_0^2}$$

The preceding four equations have four unknowns. At a given flight condition we know the aerodynamics or ω_{AF} and ζ_{AF} . If we specify the desired time constant τ , damping ζ , and open-loop crossover frequency ω_{CR} of the flight-control system, the remaining four unknowns are ω , ζ_0 , ω_0 , and K_0 . Because we have four equations, there is sufficient information to solve for these four unknowns.

If we solve the fourth equation for K_0 and substitute it into the second equation, we get

$$\frac{1}{\omega^2} + \frac{2\zeta\tau}{\omega} = \frac{1}{\omega_0^2} - \frac{2\zeta_{AF}}{\omega_{AF}K_0} = \frac{1}{\omega_0^2} + \frac{2\zeta_{AF}\omega_{AF}^2}{\omega_{AF}\omega_{CR}\omega_0^2} = \frac{1}{\omega_0^2} \left(1 + \frac{2\zeta_{AF}\omega_{AF}}{\omega_{CR}} \right)$$

We can also substitute K_0 into the third equation yielding

$$\frac{\tau}{\omega^2} = -\frac{1}{\omega_{AF}^2 K_0} = \frac{\omega_{AF}^2}{\omega_{AF}^2 \omega_{CR} \omega_0^2} = \frac{1}{\omega_{CR} \omega_0^2}$$

Substituting this result into the preceding equation yields

$$\frac{1}{\omega^2} + \frac{2\zeta\tau}{\omega} = \frac{\omega_{CR}\tau}{\omega^2} \left(1 + \frac{2\zeta_{AF}\omega_{AF}}{\omega_{CR}} \right)$$

Note that all of the terms in the preceding equation are known except for ω . We can solve the preceding equation for ω yielding

$$\boxed{\omega = \left[\tau \omega_{CR} \left(1 + \frac{2\zeta_{AF}\omega_{AF}}{\omega_{CR}} \right) - 1 \right] / (2\zeta\tau)}$$

Since we already know that

$$\frac{\tau}{\omega^2} = \frac{1}{\omega_{CR} \omega_0^2}$$

we can solve for ω_0 in terms of known quantities as

$$\boxed{\omega_0 = \frac{\omega}{\sqrt{\tau \omega_{CR}}}}$$

Substituting the solutions for ω , ω_0 , and K_0 into the first of the four equations with four unknowns allows us to solve for ζ_0 as

$$\boxed{\zeta_0 = .5\omega_0 \left(\frac{2\zeta}{\omega} + \tau + \frac{1}{K_0} \right) = .5\omega_0 \left[\frac{2\zeta}{\omega} + \tau - \frac{\omega_{AF}^2}{\omega_{CR}\omega_0^2} \right]}$$

Recalling that

$$\begin{aligned} \frac{2\zeta_0}{\omega_0} &= \left[\left(\frac{K_c}{\omega_I} + K_c T_\alpha \right) \middle/ (1 + K_c) \right] = a_1 \\ \frac{1}{\omega_0^2} &= - \left[\left(\frac{1}{\omega_z^2} + \frac{K_c T_\alpha}{\omega_I} \right) \middle/ (1 + K_c) \right] = a_2 \end{aligned}$$

we would like to solve for K_c and eliminate ω_I . Therefore

$$\begin{aligned} a_1(1 + K_c) &= \frac{K_c}{\omega_I} + K_c T_\alpha \\ a_2(1 + K_c) &= \frac{T_\alpha K_c}{\omega_I} - \frac{1}{\omega_z^2} \end{aligned}$$

Multiplying the first of the preceding two equations by T_α on both sides and then subtracting it from the second equation yields

$$a_2(1 + K_c) - a_1 T_\alpha (1 + K_c) = -(1/\omega_z^2) - T_\alpha^2 K_c$$

Therefore

$$K_c(a_2 - a_1 T_\alpha + T_\alpha^2) = -(1/\omega_z^2) - a_2 + a_1 T_\alpha$$

Solving for K_c yields

$$K_c = \frac{-(1/\omega_z^2) - a_2 + a_1 T_\alpha}{a_2 - a_1 T_\alpha + T_\alpha^2}$$

Substitution of the expressions for a_1 and a_2 into the preceding equation yields after some algebra

$$\boxed{K_c = \frac{-(\omega_0^2/\omega_z^2) - 1 + 2\zeta_0\omega_0 T_\alpha}{1 - 2\zeta_0\omega_0 T_\alpha + \omega_0^2 T_\alpha^2}}$$

Since we already know that

$$K_c = \frac{K_3}{K_A K_1}$$

we can solve for the autopilot gain K_A yielding

$$\boxed{K_A = \frac{K_3}{K_c K_1}}$$

Since we already know that

$$\frac{1}{\omega_0^2} = \left[\left(\frac{-1}{\omega_z^2} + \frac{K_c T_\alpha}{\omega_I} \right) \middle/ (1 + K_c) \right]$$

we can invert the preceding equation and solve for the autopilot gain ω_I or

$$\boxed{\omega_I = \frac{T_\alpha K_c \omega_0^2}{1 + K_c + \omega_0^2 / \omega_z^2}}$$

Recall that

$$\frac{\tau}{\omega^2} = -\frac{1}{\omega_{AF}^2 K_0}$$

We can now solve for K_0 yielding

$$\boxed{K_0 = -\frac{\omega^2}{\tau \omega_{AF}^2}}$$

Recall that

$$K_0 = K K_1 (1 + K_c)$$

We can now invert the preceding expression in order to solve for K or

$$\boxed{K = \frac{K_0}{K_1 (1 + K_c)}}$$

Since

$$K = K_R \omega_I K_A$$

we can solve for the final autopilot gain K_R yielding

$$\boxed{K_R = \frac{K}{K_A \omega_I}}$$

Finally, in order to get unity flight-control system gain we set the gain of the closed-loop transfer function to unity or

$$\frac{K_{DC} K_A K_R \omega_I K_1}{K_0} = 1$$

and get

$$K_{DC} = \frac{K_0}{K_A K_R \omega_I K_1} = \frac{K_0}{K K_1} = \frac{K K_1 (1 + K_c)}{K K_1} = 1 + K_c$$

which simplifies to

$$\boxed{K_{DC} = 1 + \frac{1}{K_A V_M}}$$

We now have enough information to simulate the three-loop flight-control system with the autopilot gain algorithm we just derived. Listing 24.1 presents a time domain step response simulation of the three-loop autopilot in the presence of the linear airframe dynamics. We can see from the listing that the required aerodynamic parameters are derived from the geometry of the airframe. For a given flight condition these airframe parameters are used to both describe the airframe and to determine the autopilot gains using the gain algorithm we just derived. Although the original gain algorithm derivation neglected the dynamics of the actuator, these dynamics are included in the step response simulation to test the robustness of the autopilot gains. The actuator is modeled as a second-order transfer function or

$$\frac{\delta}{\delta_c} = 1 / \left(1 + \frac{2\zeta_{ACT}}{\omega_{ACT}} s + \frac{s^2}{\omega_{ACT}^2} \right)$$

with a natural frequency ω_{ACT} of 150 rad/s and damping ζ_{ACT} of 0.7.

Using Listing 24.1 for the flight condition in which the missile was at sea level and traveling at 3000 ft/s, a 10-g command was issued to the autopilot. We can see from Listing 24.1 that the nominal design goals given the autopilot gain algorithm were to achieve a time constant of 0.3 s, an open-loop crossover frequency of 50 rad/s, and a damping of 0.7. The three autopilot gains for these requirements at this flight condition turn out to be $K_A = 1.15$ deg/g-s, $\omega_I = 12.9$ rad/s, and $K_R = 0.0928$ s. We can see from Fig. 24.3 that the overall time constant of the flight-control system is slightly in excess of 0.3 s. This is not in disagreement with theory since the overall time constant of the third-order flight-control system can be approximated as

$$\tau_{TOT} = \tau + \frac{2\zeta}{\omega}$$

Because the damping is 0.7 and the natural frequency is approximately 35 rad/s, the overall or total time constant is 0.34 s, which is consistent with Fig. 24.3. In other words the response reaches 63% of the steady-state value in 0.34 s.

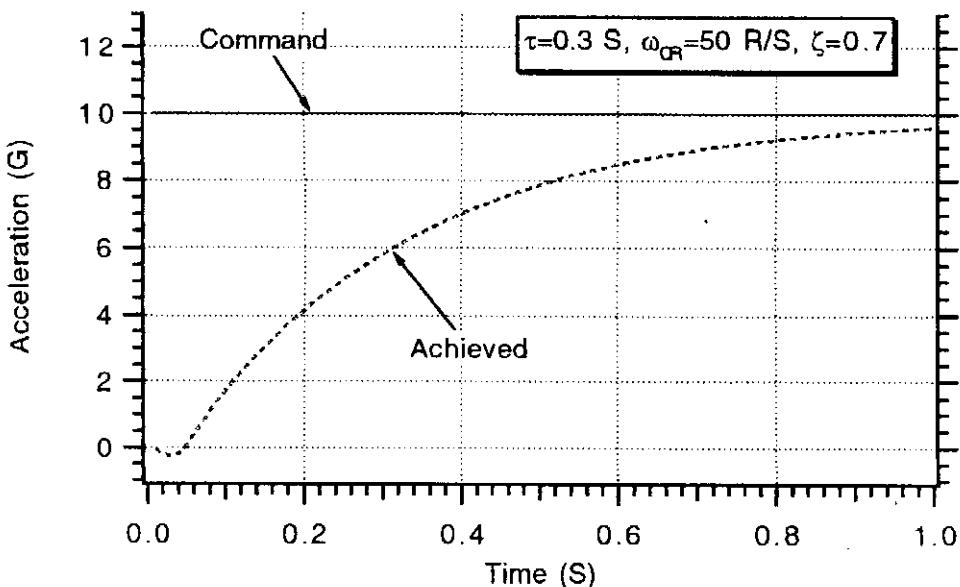


Fig. 24.3 Autopilot gain algorithm allows us to select time constant.

Listing 24.1 Three-loop autopilot step response simulation

```

FR=3.
DIAM=1.
XL=20.
CTW=0.
CRW=6.
HW=2.
CTT=0.
CRT=2.
HT=2.
XN=4.
XCG=10.
XHL=19.5
WACT=150.
ZACT=.7
TF=1.
VM=3000.
XNCG=10.
WCR=50.
ZETA=.7
TAU=.3
ALT=0.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
A=1000.
IF(ALT.LE.30000.)THEN
    RHO=.002378*EXP(-ALT/30000.)
ELSE
    RHO=.0034*EXP(-ALT/22000.)
ENDIF

WGT=1000.
XNLLIN=0.
XACC=XCG
SWING=.5*HW*(CTW+CRW)
STAIL=.5*HT*(CTT+CRT)
SREF=3.1416*DIAM*DIAM/4.
XLP=FR*DIAM
SPLAN=(XL-XLP)*DIAM+1.33*XLP*DIAM/2.
XCPN=2*XLP/3
AN=.67*XLP*DIAM
AB=(XL-XLP)*DIAM
XCPB=(.67*AN*XLP+AB*(XLP+.5*(XL-XLP)))/(AN+AB)
XCPW=XLP+XN+.7*CRW-.2*CTW
XMACH=VM/A
XIYY=WGT*(3*((DIAM/2)**2)+XL*XL)/(12*32.2)
TMP1=(XCG-XCPW)/DIAM
TMP2=(XCG-XHL)/DIAM
TMP3=(XCG-XCPB)/DIAM
TMP4=(XCG-XCPN)/DIAM
B=SQRT(XMACH**2-1)
Q=.5*RHO*VM*VM
P1=WGT*XNCG/(Q*SREF)

```

(Contd.)

Listing 24.1 (Continued)

```

Y1=2+8*SWING/(B*SREF)+8*STAIL/(B*SREF)
Y2=1.5*SPLAN/SREF
Y3=8*STAIL/(B*SREF)
Y4=2*TMP4+8*SWING*TMP1/(B*SREF)+8*STAIL*TMP2/(B*SREF)
Y5=1.5*SPLAN*TMP3/SREF
Y6=8*STAIL*TMP2/(B*SREF)
P2=Y2-Y3*Y5/Y6
P3=Y1-Y3*Y4/Y6
ALFTR=(-P3+SQRT(P3*P3+4.*P2*P1))/(2.*P2)
DELTR=-Y4*ALFTR/Y6-Y5*ALFTR*ALFTR/Y6
CNA=2+1.5*SPLAN*ALFTR/SREF+8*SWING/(B*SREF)+8*STAIL/(B*SREF)
CND=8*STAIL/(B*SREF)
CMAP=2*TMP4+1.5*SPLAN*ALFTR*TMP3/SREF+8*SWING*TMP1/
(B*SREF)
CMA=CMAP+8*STAIL*TMP2/(B*SREF)
CMD=8*STAIL*TMP2/(B*SREF)
XMA=Q*SREF*DIAM*CMA/XIYY
XMD=Q*SREF*DIAM*CMD/XIYY
ZA=-32.2*Q*SREF*CNA/(WGT*VM)
ZD=-32.2*Q*SREF*CND/(WGT*VM)
WZ=SQRT((XMA*ZD-ZA*XMD)/ZD)
WAF=SQRT(-XMA)
ZAF=.5*WAF*ZA/XMA
XK1=VM*(XMA*ZD-XMD*ZA)/(1845*XMA)
XK2=XK1
TA=XMD/(XMA*ZD-XMD*ZA)
XK3=1845*XK1/VM
W=(TAU*WCR*(1+2.*ZAF*WAF/WCR)-1)/(2*ZETA*TAU)
W0=W/SQRT(TAU*WCR)
Z0=.5*W0*(2*ZETA/W+TAU-WAF**2/(W0*W0*WCR))
XKC=(-W0**2/WZ**2-1.+2*Z0*W0*TA)/(1.-2.*Z0*W0*TA+W0*W0*TA*TA)
XKA=XK3/(XK1*XKC)
XK0=-W*W/(TAU*WAF*WAF)
XK=XK0/(XK1*(1+XKC))
WI=XKC*TA*W0*W0/(1+XKC+W0**2/WZ**2)
XKR=XK/(XKA*WI)
XKDC=1.+1845./(XKA*VM)
E=0.
ED=0.
DELD=0.
DEL=0.
X=0.
T=0
H=.0001
S=0
10 IF(T>(TF-.00001))GOTO 999
S=S+H
EOLD=E
EDOLD=ED
DELOLD=DEL
DELDOLD=DELD
XOLD=X

```

Listing 24.1 (Continued)

```

STEP=1
GOTO 200
66 STEP=2
E=E+H*ED
ED=ED+H*EDD
DEL=DEL+H*DELD
DELD=DELD+H*DELDD
X=X+H*XD
T=T+H
GOTO 200
55 CONTINUE
E=.5*(EOLD+E+H*ED)
ED=.5*(EDOLD+ED+H*EDD)
DEL=.5*(DELOUD+DEL+H*DELD)
DELD=.5*(DELDOLD+DELD+H*DELDD)
X=.5*(XOLD+X+H*XD)
IF(S.LE..0099999)GOTO 10
S=0.
WRITE(1,97)T,XNL,XNCG
WRITE(*,97)T,XNL,XNCG
GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
THD=XK3*(E+TA*ED)
DELC=XKR*(X+THD)
DELDD=WACT*WACT*(DELC-DEL-2.*ZACT*DELD/WACT)
EDD=WAF*WAF*(DEL-E-2.*ZAF*ED/WAF)
XNL=XK1*(E-EDD/WZ**2)
XD=WI*(THD+XKA*(XNL-XNCG*XKDC))
IF(STEP-1)66,66,55
999 CONTINUE
CLOSE(1)
PAUSE
END

```

The open-loop transfer function for the three-loop flight-control system neglecting actuator dynamics has already been derived as

$$HG_{\text{NO ACTUATOR}} = -K_0 \left(1 + \frac{2\zeta_0}{\omega_0} s + \frac{s^2}{\omega_0^2} \right) / \left[s \left(1 + \frac{2\zeta_{\text{AF}}}{\omega_{\text{AF}}} s + \frac{s^2}{\omega_{\text{AF}}^2} \right) \right]$$

If the dynamics of the actuator are now included, it is easy to show that the open-loop transfer function is simply a multiplication of the previous open-loop transfer function with the actuator transfer function or

$$HG_{\text{ACTUATOR}} = -K_0 \left(1 + \frac{2\zeta_0}{\omega_0} s + \frac{s^2}{\omega_0^2} \right) / \left[s \left(1 + \frac{2\zeta_{\text{AF}}}{\omega_{\text{AF}}} s + \frac{s^2}{\omega_{\text{AF}}^2} \right) \right] \\ \times \left(1 + \frac{2\zeta_{\text{ACT}}}{\omega_{\text{ACT}}} s + \frac{s}{\omega_{\text{ACT}}^2} \right)$$

Therefore the magnitude and phase of the open-loop transfer function can be written by inspection of the previous expression as

$$|HG_{\text{ACTUATOR}}| = \frac{-K_0}{\omega} \sqrt{\frac{\left(1 - \frac{\omega^2}{\omega_0^2}\right)^2 + \left(\frac{2\zeta_0\omega}{\omega_0}\right)^2}{\left[\left(1 - \frac{\omega^2}{\omega_{\text{AF}}^2}\right)^2 + \left(\frac{2\zeta_{\text{AF}}\omega}{\omega_{\text{AF}}}\right)^2\right]\left[\left(1 - \frac{\omega^2}{\omega_{\text{ACT}}^2}\right)^2 + \left(\frac{2\zeta_{\text{ACT}}\omega}{\omega_{\text{ACT}}}\right)^2\right]}}$$

$$\phi = -90 + 57.3 \left[\tan^{-1} \left[\frac{\frac{2\zeta_0\omega}{\omega_0}}{1 - \frac{\omega^2}{\omega_0^2}} \right] - \tan^{-1} \left[\frac{\frac{2\zeta_{\text{AF}}\omega}{\omega_{\text{AF}}}}{1 - \frac{\omega^2}{\omega_{\text{AF}}^2}} \right] - \tan^{-1} \left[\frac{\frac{2\zeta_{\text{ACT}}\omega}{\omega_{\text{ACT}}}}{1 - \frac{\omega^2}{\omega_{\text{ACT}}^2}} \right] \right]$$

Listing 24.2 is a program that evaluates the magnitude and phase of the open-loop transfer function (i.e., preceding two equations) for different frequencies. We can see from the listing that the magnitude is expressed in units of dB whereas the phase is expressed in units of deg. The program runs quickly because numerical integration is not involved.

The output of Listing 24.2 is a Bode plot as shown in Fig. 24.4. We can see that the achieved gain crossover frequency of 60 rad/s is close to the desired goal of 50 rad/s. We did not meet the exact design goal because in the derivation of the formula for the open-loop gain crossover frequency it was assumed that the crossover frequency was far beyond the dynamics of the airframe, which is not quite true. Figure 24.4 also indicates that the flight-control system stability margins are adequate since the gain margin (gm) is 11 dB and the phase margin (ϕ_{PM}) is 45 deg.

Figure 24.5 indicates that the autopilot gain algorithm is successful in controlling the flight-control system time constant. When the desired time constant is reduced to 0.1 s, the new gains become $K_A = 3.72$ deg/g-s, $\omega_I = 11.2$ rad/s, and $K_R = 0.098$ s. Recall that when the desired autopilot time constant was 0.3 s the autopilot gains were $K_A = 3.72$ deg/g-s, $\omega_I = 11.2$ rad/s, and $K_R = 0.098$ s.

To check if the open-loop crossover frequency remains the same when one reduces the desired flight-control system time constant from 0.3 s to 0.1 s, it is necessary to go back to the frequency domain. Figure 24.6 shows that although the

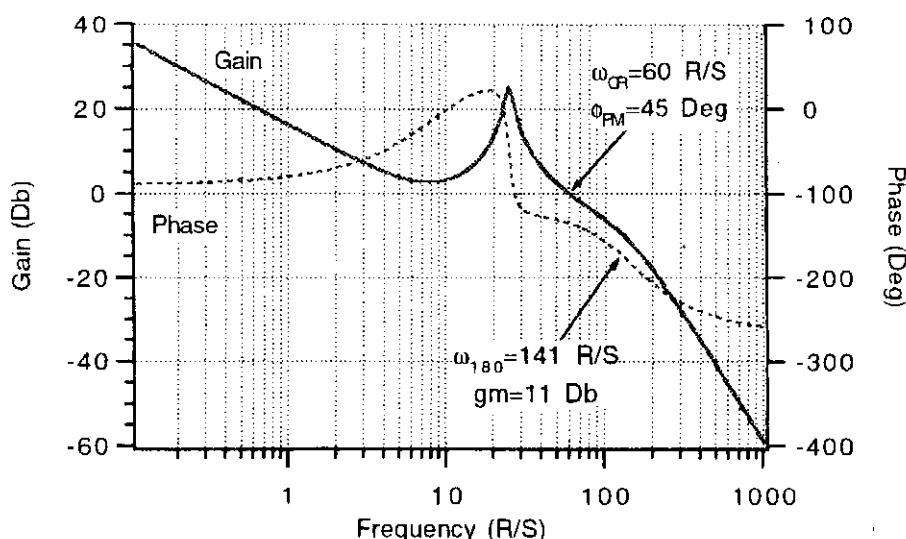


Fig. 24.4 Desired crossover frequency is achieved—even in presence of actuator dynamics.

Listing 24.2 Open-loop response of three-loop autopilot

```

FR=3.
DIAM=1.
XL=20.
CTW=0.
CRW=6.
HW=2.
CTT=0.
CRT=2.
HT=2.
XN=4.
XCG=10.
XHL=19.5
WACT=150.
ZACT=.7
VM=3000.
XNCG=10.
WCR=50.
ZETA=.7
TAU=.3
ALT=0.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
A=1000.
IF(ALT.LE.30000.)THEN
    RHO=.002378*EXP(-ALT/30000.)
ELSE
    RHO=.0034*EXP(-ALT/22000.)
ENDIF

WGT=1000.
XNLLIN=0.
XACC=XCG
SWING=.5*HW*(CTW+CRW)
STAIL=.5*HT*(CTT+CRT)
SREF=3.1416*DIAM*DIAM/4.
XLP=FR*DIAM
SPLAN=(XL-XLP)*DIAM+1.33*XLP*DIAM/2.
XCPN=2*XLP/3
AN=.67*XLP*DIAM
AB=(XL-XLP)*DIAM
XCPB=(.67*AN*XLP+AB*(XLP+.5*(XL-XLP)))/(AN+AB)
XCPW=XLP+XN+.7*CRW-.2*CTW
XMACH=VM/A
XIYY=WGT*(3*((DIAM/2)**2)+XL*XL)/(12*32.2)
TMP1=(XCG-XCPW)/DIAM
TMP2=(XCG-XHL)/DIAM
TMP3=(XCG-XCPB)/DIAM
TMP4=(XCG-XCPN)/DIAM
B=SQRT(XMACH**2-1)
Q=.5*RHO*VM*VM
P1=WGT*XNCG/(Q*SREF)
Y1=2+8*SWING/(B*SREF)+8*STAIL/(B*SREF)

```

(Contd.)

Listing 24.2 (Continued)

```

Y2=1.5*SPLAN/SREF
Y3=8*STAIL/(B*SREF)
Y4=2*TMP4+8*SWING*TMP1/(B*SREF)+8*STAIL*TMP2/(B*SREF)
Y5=1.5*SPLAN*TMP3/SREF
Y6=8*STAIL*TMP2/(B*SREF)
P2=Y2-Y3*Y5/Y6
P3=Y1-Y3*Y4/Y6
ALFTR=(-P3+SQRT(P3*P3+4.*P2*P1))/(2.*P2)
DELTR=-Y4*ALFTR/Y6-Y5*ALFTR*ALFTR/Y6
CNA=2+1.5*SPLAN*ALFTR/SREF+8*SWING/(B*SREF)+8*STAIL/(B*SREF)
CND=8*STAIL/(B*SREF)
CMAP=2*TMP4+1.5*SPLAN*ALFTR*TMP3/SREF+8*SWING*TMP1/
(B*SREF)
CMA=CMAP+8*STAIL*TMP2/(B*SREF)
CMD=8*STAIL*TMP2/(B*SREF)
XMA=Q*SREF*DIAM*CMA/XIYY
XMD=Q*SREF*DIAM*CMD/XIYY
ZA=-32.2*Q*SREF*CNA/(WGT*VM)
ZD=-32.2*Q*SREF*CND/(WGT*VM)
WZ=SQRT((XMA*ZD-ZA*XMD)/ZD)
WAF=SQRT(-XMA)
ZAF=.5*WAF*ZA/XMA
XK1=-VM*(XMA*ZD-XMD*ZA)/(1845*XMA)
XK2=XK1
TA=XMD/(XMA*ZD-XMD*ZA)
XK3=1845*XK1/VM
W=(TAU*WCR*(1+2.*ZAF*WAF/WCR)-1)/(2*ZETA*TAU)
W0=W/SQRT(TAU*WCR)
Z0=.5*W0*(2*ZETA/W+TAU-WAF**2/(W0*W0*WCR))
XKC=(-W0**2/WZ**2-1.+2.*Z0*W0*TA)/(1.-2.*Z0*W0*TA+W0*W0*TA*TA)
XKA=XK3/(XK1*XKC)
XK0=-W*W/(TAU*WAF*WAF)
XK=XK0/(XK1*(1+XKC))
WI=XKC*TA*W0*W0/(1+XKC+W0**2/WZ**2)
XKR=XK/(XKA*WI)
XKDC=1.+1845./(XKA*VM)
DO 10 I=2,160
W=10**(.025*I-1)
XMAGTOP=-XK0*SQRT((1.-(W/W0)**2)**2+(2.*Z0*W/W0)**2)
XMAGBOT=W*SQRT((1.-(W/WAF)**2)**2+(2.*ZAF*W/WAF)**2)
XMAG=XMAGTOP/XMAGBOT
XMAGACT=1./SQRT((1.-W*W/(WACT*WACT))**2+(2.*ZACT*W/WACT)**2)
PHASETOP=ATAN2(2.*Z0*W/W0,1.-(W/W0)**2)
PHASEBOT=ATAN2(2.*ZAF*W/WAF,1.-(W/WAF)**2)
PHASEACT=ATAN2(2.*ZACT*W/WACT,1.-W*W/(WACT*WACT))
GAIN=20.*LOG10(XMAG*XMAGACT)
PHASE=-90.+57.3*(PHASETOP-PHASEBOT-PHASEACT)
WRITE(*,97)W,GAIN,PHASE
WRITE(1,97)W,GAIN,PHASE
10  CONTINUE
97  FORMAT(3F10.3)
PAUSE
END

```

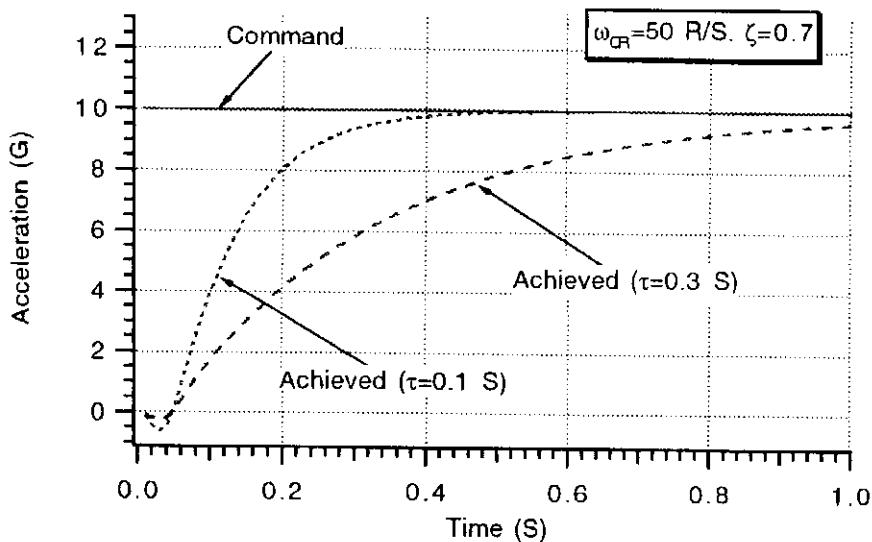


Fig. 24.5 Time constant control can be achieved with autopilot gain algorithm.

open-loop gain changes at some frequencies when the time constant is reduced, the gain crossover frequency remains unchanged thus showing that the open-loop gain crossover frequency is independent of the flight-control system time constant.

We can see from Fig. 24.7 that if the desired time constant in the flight-control algorithm is reset to 0.3 s and if we increase the desired crossover frequency from 50 rad/s to 100 rad/s the achieved crossover frequency is 104 rad/s, which is close to the desired goal. In this case the open-loop crossover frequency is far beyond the airframe dynamics thus justifying the approximation made in the derivation of the formula for the crossover frequency. However we can also see from Fig. 24.7 that the inclusion of the actuator dynamics tends to reduce the stability margins when the crossover frequency is increased. In this case the gain margin has been reduced from 11 dB to 1.5 dB while the phase margin has been reduced from 45 deg to 5 deg.

Figure 24.8 confirms from a time domain point of view that the higher crossover frequency flight-control system (which has lower stability margins) has an

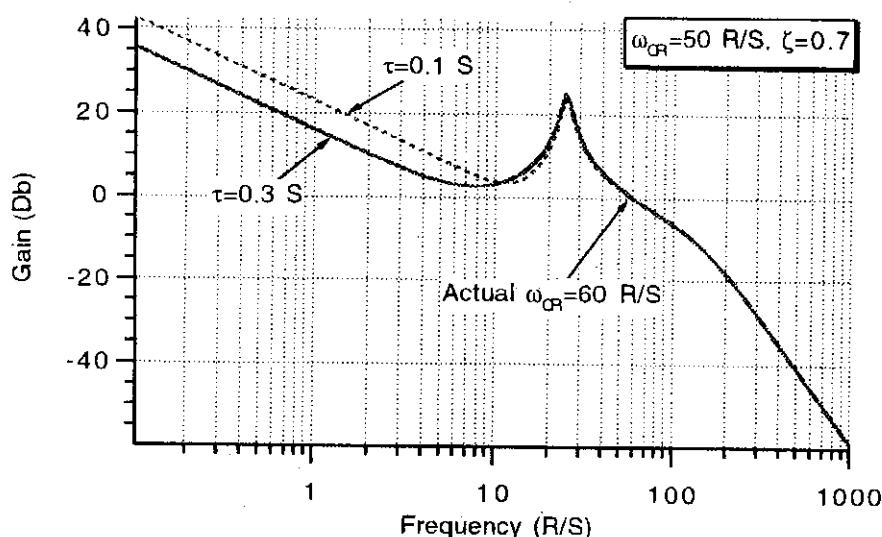


Fig. 24.6 Actual crossover frequency is independent of time constant.

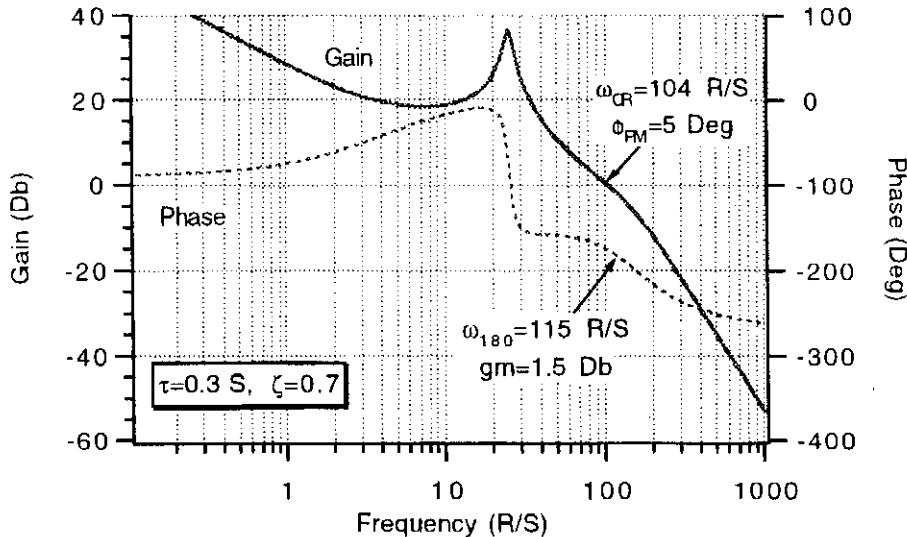


Fig. 24.7 Crossover frequency can be controlled but higher crossover frequency reduces stability margins.

oscillatory response. In practice the selection of the open-loop crossover frequency is limited by the dynamics of the actuator.

Figure 24.9 shows that when the actuator natural frequency is increased to 300 rad/s the step response stabilizes and is well behaved when the open-loop crossover frequency is 100 rad/s. In practice the maximum open-loop crossover frequency should be approximately one third the bandwidth of the actuator.

If the flight-control system damping is reduced from 0.7, system performance will suffer. Figure 24.10 shows that for the nominal case in which it is desired that the flight-control system time constant be 0.3 s and the open-loop crossover frequency be 50 rad/s, reducing the damping to 0.4 reduces the stability margins. In this case the phase margin reduces from 43 deg to 6 deg while the gain margin reduces from 11 dB to 3.8 dB.

Finally, Fig. 24.11 confirms from a time domain point of view that decreasing the system damping reduces the stability margins. We can see that when the damping is decreased from 0.7 to 0.4 the system step response becomes oscillatory.

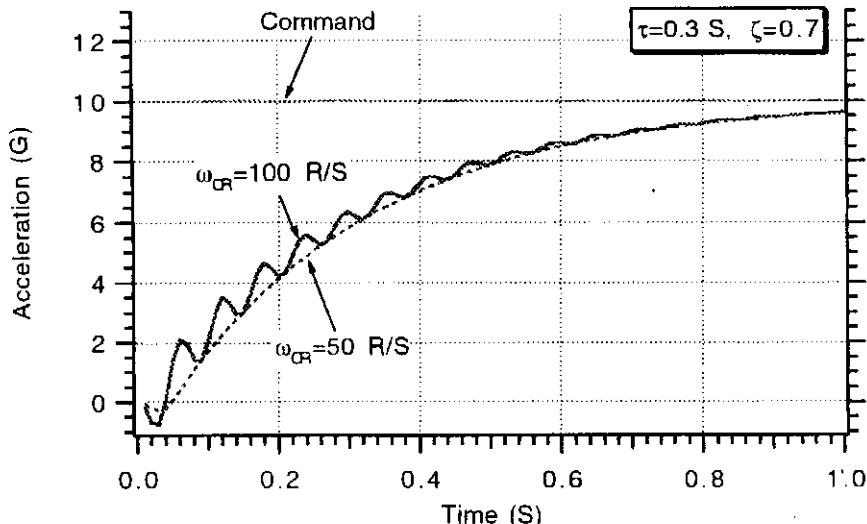


Fig. 24.8 High crossover frequencies are dangerous when actuator dynamics are considered.

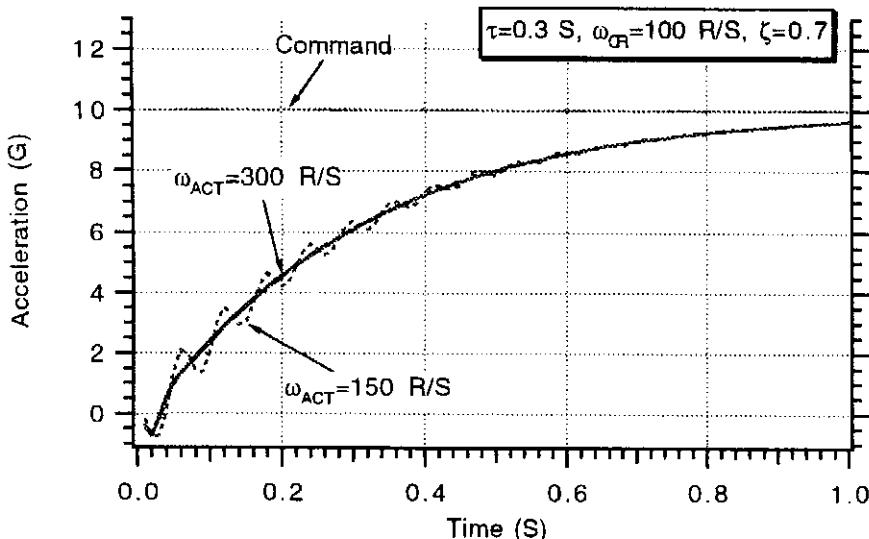


Fig. 24.9 Increasing actuator bandwidth allows operation at high crossover frequencies.

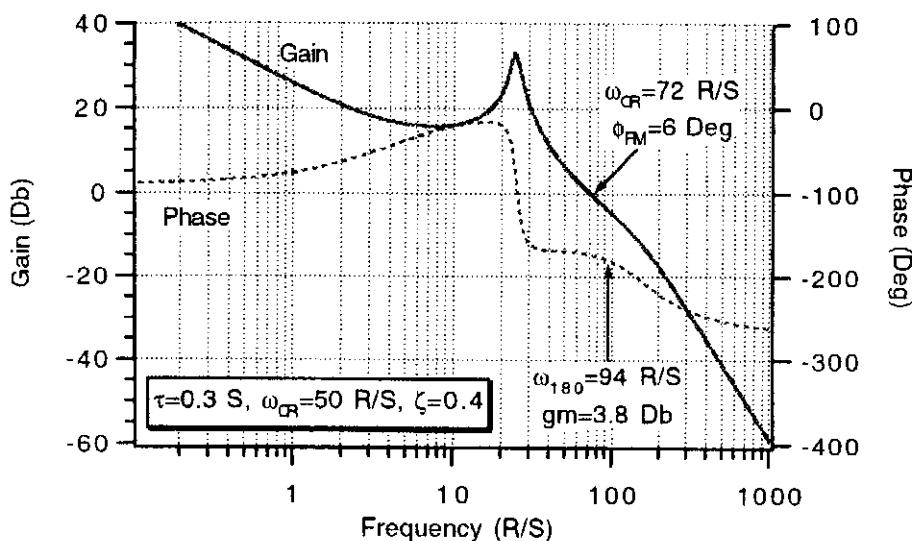


Fig. 24.10 Decreasing damping decreases stability margins.

Experiments with Flight Condition

The autopilot gain algorithm was used on the hypothetical missile airframe of Chapter 22 for the flight condition in which there was a 10-*g* step acceleration command when the missile was traveling at 3000 ft/s. Three-loop autopilot gains were determined at sea level, 30 kft altitude, and 50 kft altitude. The gain algorithm was told that the desired flight-control system time constant was 0.3 s, the damping was 0.7, and the open-loop crossover frequency was 50 rad/s. The resultant autopilot gains for each of the three flight conditions are summarized below in Table 24.1. We can see from the table that the autopilot gain K_A increases by a factor of 4 in going from sea level to 50 kft altitude. The gain K_R increases by a factor of nearly 10 whereas the gain ω_I remains approximately constant.

It is apparent from Fig. 24.12 that the autopilot gains enable the flight-control system to maintain the same step response for the three different flight

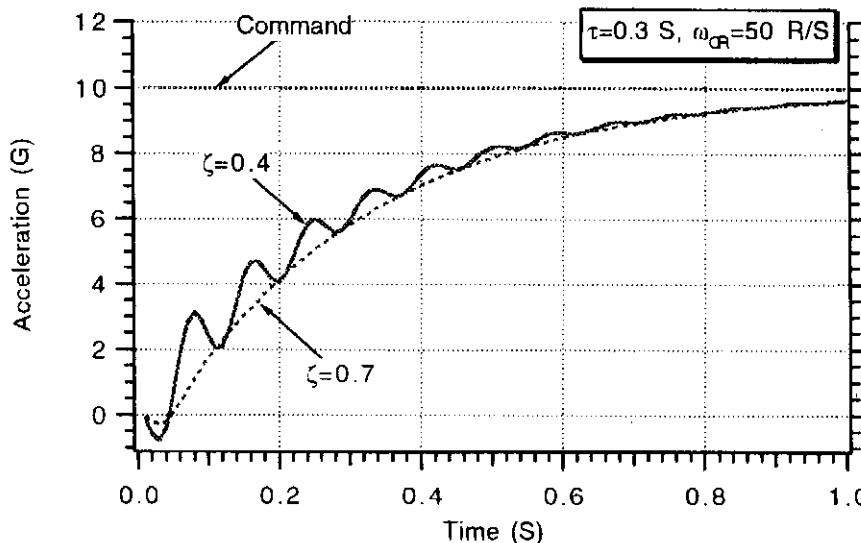


Fig. 24.11 Damping can be controlled, but low damping is dangerous when actuator dynamics are considered.

conditions. The only difference between each of the step responses is that the wrong-way effect increases as the altitude increases. The autopilot gain algorithm does not attempt to control the airframe zeroes and therefore the wrong-way effect will get worse as the altitude increases.

To understand why the wrong-way tail effect gets worse at the higher altitudes, it is necessary to examine the flight-control system transfer functions in more detail. For example, the flight-control system transfer function from commanded to achieved acceleration is given by

$$\frac{n_L}{n_c} = \left(1 - \frac{s^2}{\omega_z^2}\right) / (1 + s\tau) \left[1 + \frac{2\zeta s}{\omega} + \frac{s^2}{\omega^2}\right]$$

where the negative sign in the numerator of the preceding equation indicates that the airframe zero is in the right-half plane. We recall from Chapter 22 that the airframe zero ω_z is related to the airframe parameters according to

$$\omega_z = \sqrt{\frac{M_\alpha Z_\delta - M_\delta Z_\alpha}{Z_\delta}}$$

For the three different flight conditions the linearized airframe parameters were calculated and are summarized in Table 24.2. Using the preceding linearized aerodynamic parameters, the airframe zero can be evaluated at the three different

Table 24.1 Autopilot gains vary with flight condition

Altitude	K_A	K_R	ω_I
0 kft	1.15 deg/g	0.0928 s	12.9 rad/s
30 kft	1.98 deg/g	0.264 s	19.3 rad/s
50 kft	4.16 deg/g	0.726 s	19.7 rad/s

Table 24.2 Linearized aerodynamic parameters for different flight conditions

Altitude	M_α	M_δ	Z_α	Z_δ
0 kft	-642 s^{-2}	-555 s^{-2}	-2.94 s^{-1}	-0.65 s^{-1}
30 kft	-240 s^{-2}	-204 s^{-2}	-1.17 s^{-1}	-0.239 s^{-1}
50 kft	-99.1 s^{-2}	-81.7 s^{-2}	-0.533 s^{-1}	-0.0957 s^{-1}

altitudes yielding

$$\omega_z|_{0 \text{ kft}} = \sqrt{\frac{642 * 0.65 - 555 * 2.94}{-0.65}} = 43.2 \text{ rad/s}$$

$$\omega_z|_{30 \text{ kft}} = \sqrt{\frac{240 * 0.239 - 204 * 1.17}{-0.239}} = 27.6 \text{ rad/s}$$

$$\omega_z|_{50 \text{ kft}} = \sqrt{\frac{99.1 * 0.0957 - 81.7 * 0.533}{-0.0957}} = 18.8 \text{ rad/s}$$

We can see from the preceding calculations that the right-half plane airframe zero decreases as the altitude increases thus causing more wrong-way effect.

For the same 10-g step acceleration input command, Fig. 24.13 shows how the missile fin rate response changes with time. We can see that the maximum fin rate increases with increasing altitude. At sea level only a few deg/s of fin rate are required whereas at 50 kft altitude nearly 600 deg/s are required to ensure the well-behaved step response of Fig. 24.12. In practice, the actuator must be sized to handle the expected maximum fin rate or else saturation occurs and flight catastrophe can result. It is apparent from Fig. 24.13 that the requirements for actuator fin rate sizing will take place at the highest expected altitude in the flight envelope.

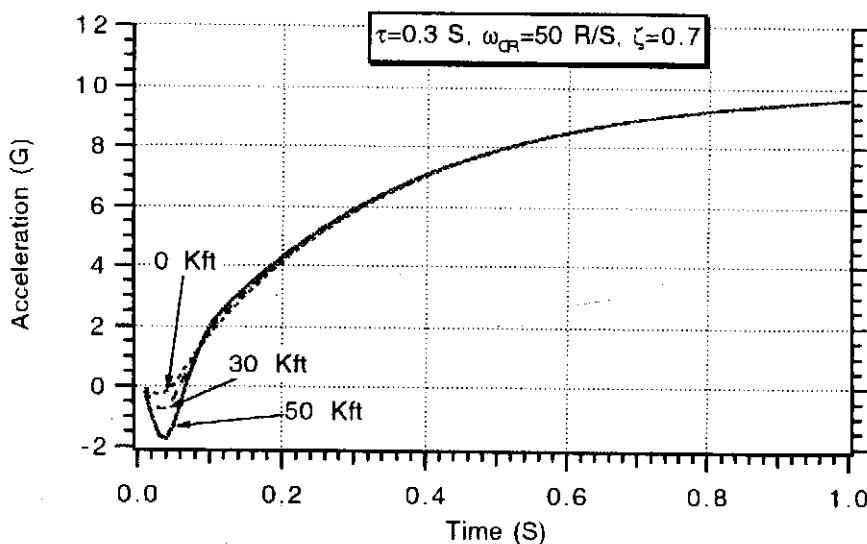


Fig. 24.12 With exception of wrong-way effect, acceleration response of autopilot is independent of flight condition.

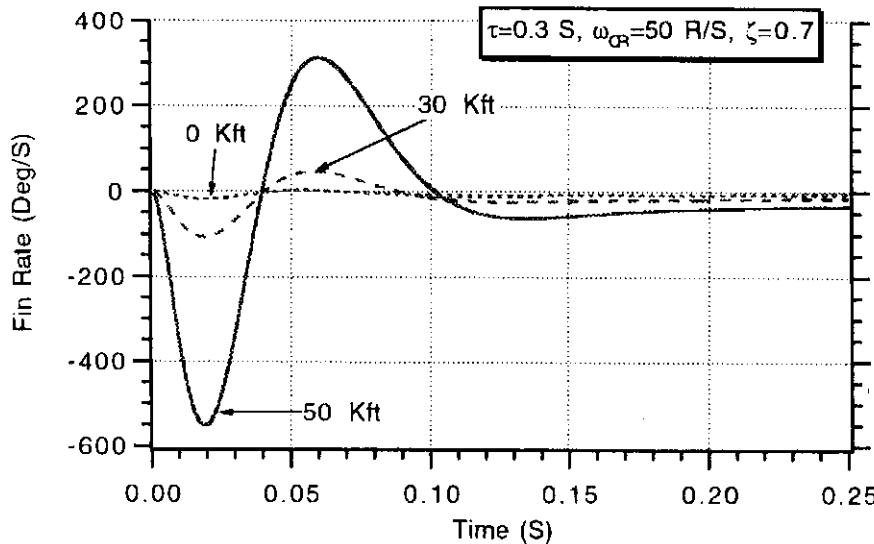


Fig. 24.13 Fin rate increases with increasing altitude.

Using a transfer function type of analysis, we can try and understand why more fin rate is required at the higher altitudes. We have shown in Chapter 22 that the airframe transfer function from fin rate to achieved acceleration is given by

$$\frac{n_L}{\delta} = \left[K_1 \left(1 - \frac{s^2}{\omega_z^2} \right) \right] / \left(1 + \frac{2\zeta_{AF}s}{\omega_{AF}} + \frac{s^2}{\omega_{AF}^2} \right)$$

Using the chain rule from calculus, we can find the transfer function of fin rate due to commanded acceleration by combining the previous airframe transfer function with the flight-control system transfer function yielding

$$\frac{\dot{\delta}}{n_c} = \frac{s\delta}{n_L} * \frac{n_L}{n_c} = s \left[1 + \frac{2\zeta_{AF}s}{\omega_{AF}} + \frac{s^2}{\omega_{AF}^2} \right] / \left\{ K_1(1+s\tau) \left[1 + \frac{2\zeta s}{\omega} + \frac{s^2}{\omega^2} \right] \right\}$$

We can see from the preceding transfer function that the magnitude of the fin rate is inversely proportional to the aerodynamic gain K_1 . We have shown in Chapter 22 that the aerodynamic gain is given by

$$K_1 = \frac{-V_M(M_\alpha Z_\delta - Z_\alpha M_\delta)}{1845M_\alpha}$$

Evaluating the aerodynamic gain at the three different flight conditions using the values of Table 24.2 for the various aerodynamic parameters reveals that the magnitude of the aerodynamic gain decreases as the altitude increases or

$$K_1|_{0\text{ kft}} = \frac{-3000(642 * 0.65 - 2.94 * 555)}{-1845 * 642} = -3.07 \text{ g/deg}$$

$$K_1|_{30\text{ kft}} = \frac{-3000(240 * 0.239 - 1.17 * 204)}{-1845 * 240} = -1.23 \text{ g/deg}$$

$$K_1|_{50\text{ kft}} = \frac{-3000(99.1 * 0.0957 - 0.533 * 81.7)}{-1845 * 99.1} = -0.558 \text{ g/deg}$$

Because the fin rate transfer function is inversely proportional to the aerodynamic gain, the fin rate magnitude will increase with increasing altitude as was already demonstrated in Fig. 24.13.

Guidance System Analysis

The guidance system with the flight-control system modeled explicitly appears in Fig. 24.14. In this model the airframe is represented by the second-order transfer function derived in Chapter 22, and the three-loop autopilot is modeled in conjunction with a second-order actuator model. In this homing loop model we can see that there are four random error sources that will cause miss distance: white semiactive noise, white glint noise, white range independent noise, and a uniformly distributed target maneuver. The airframe parameters and the autopilot gains in the flight-control system are a function of the flight-condition. In this model we have assumed that the missile speed and altitude are constant so that the aerodynamic parameters and flight control gains will be constant for a given engagement. The various homing loop parameters used in the study of this section appear in Table 24.3.

The adjoint of the homing loop was taken and appears in Fig. 24.15. Shown in the adjoint diagram are the miss distance outputs due to the four random error sources. The values of the error sources used in the ensuing study appear in Table 24.4. The spectral densities for each of the error sources can be calculated from the preceding table according to

$$\Phi_{RN} = 2T_{RN}\sigma_{RN}^2 = 2 * 0.01 * 0.01^2 = 2 * 10^{-6} \frac{\text{rad}^2}{\text{hz}}$$

$$\Phi_{GL} = 2T_{GL}\sigma_{GL}^2 = 2 * 0.1 * 10^2 = 20 \frac{\text{ft}^2}{\text{hz}}$$

$$\Phi_{FN} = 2T_{FN}\sigma_{FN}^2 = 2 * 0.01 * 0.002^2 = 8 * 10^{-8} \frac{\text{rad}^2}{\text{hz}}$$

$$\Phi_{n_T} = \frac{n_T^2}{t_F} = \frac{64.4^2}{5} = 829 \frac{\text{ft/s}^2}{\text{hz}}$$

The adjoint simulation, derived from Fig. 24.15, appears in Listing 24.3. We can see from the listing that the adjoint can yield miss distance results (MISS = 1,

Table 24.3 Nominal guidance system parameters

Name	Definition	Value
V_c	Closing velocity	4000 ft/s
T_1	Seeker track loop time constant	0.1 s
T_2	Noise filter time constant	0.15 s
N'	Effective navigation ratio	3
ω_{ACT}	Actuator natural frequency	150 rad/s
ζ_{ACT}	Actuator damping	0.7

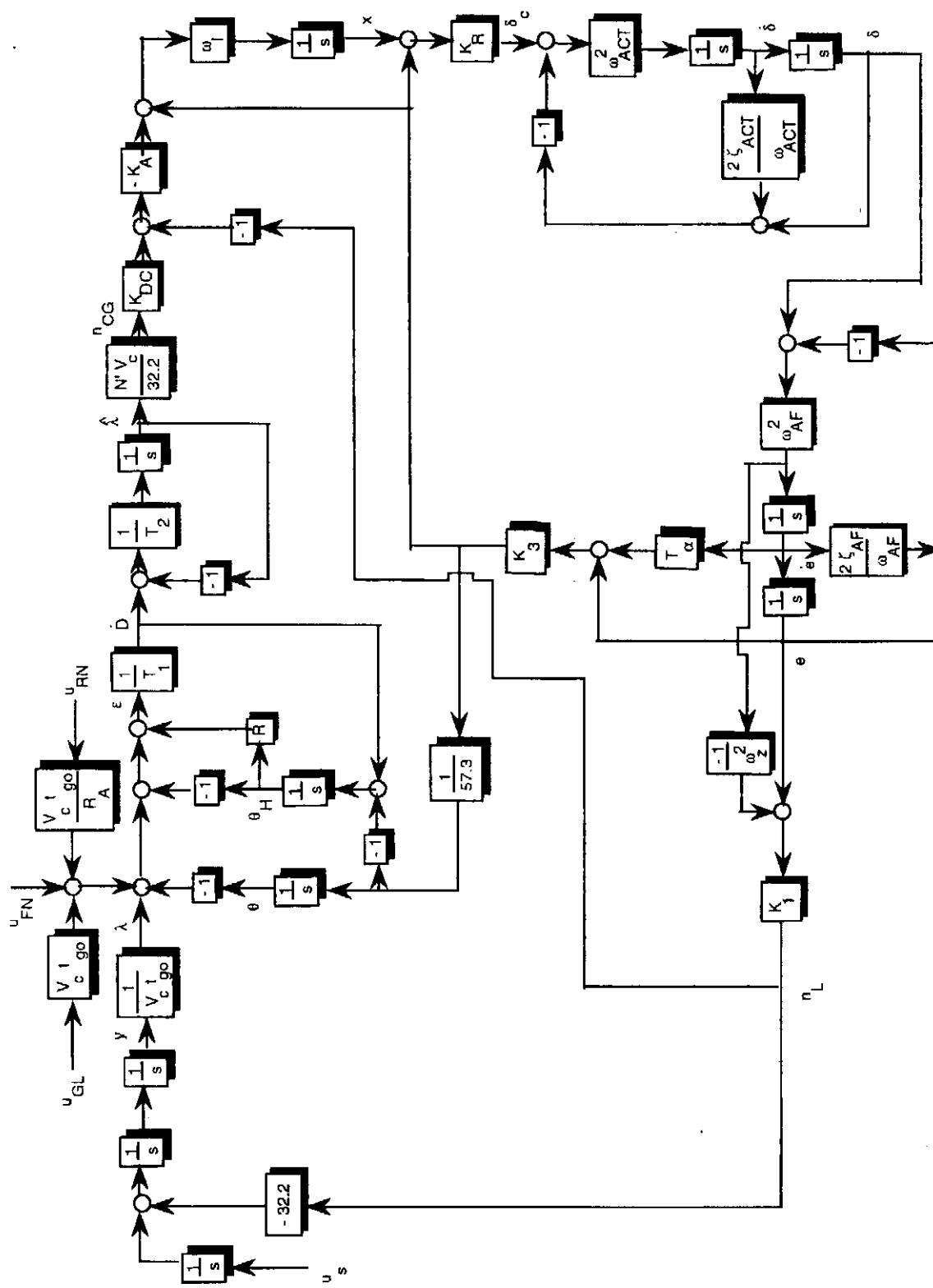


Fig. 24.14 Homing loop with flight-control system.

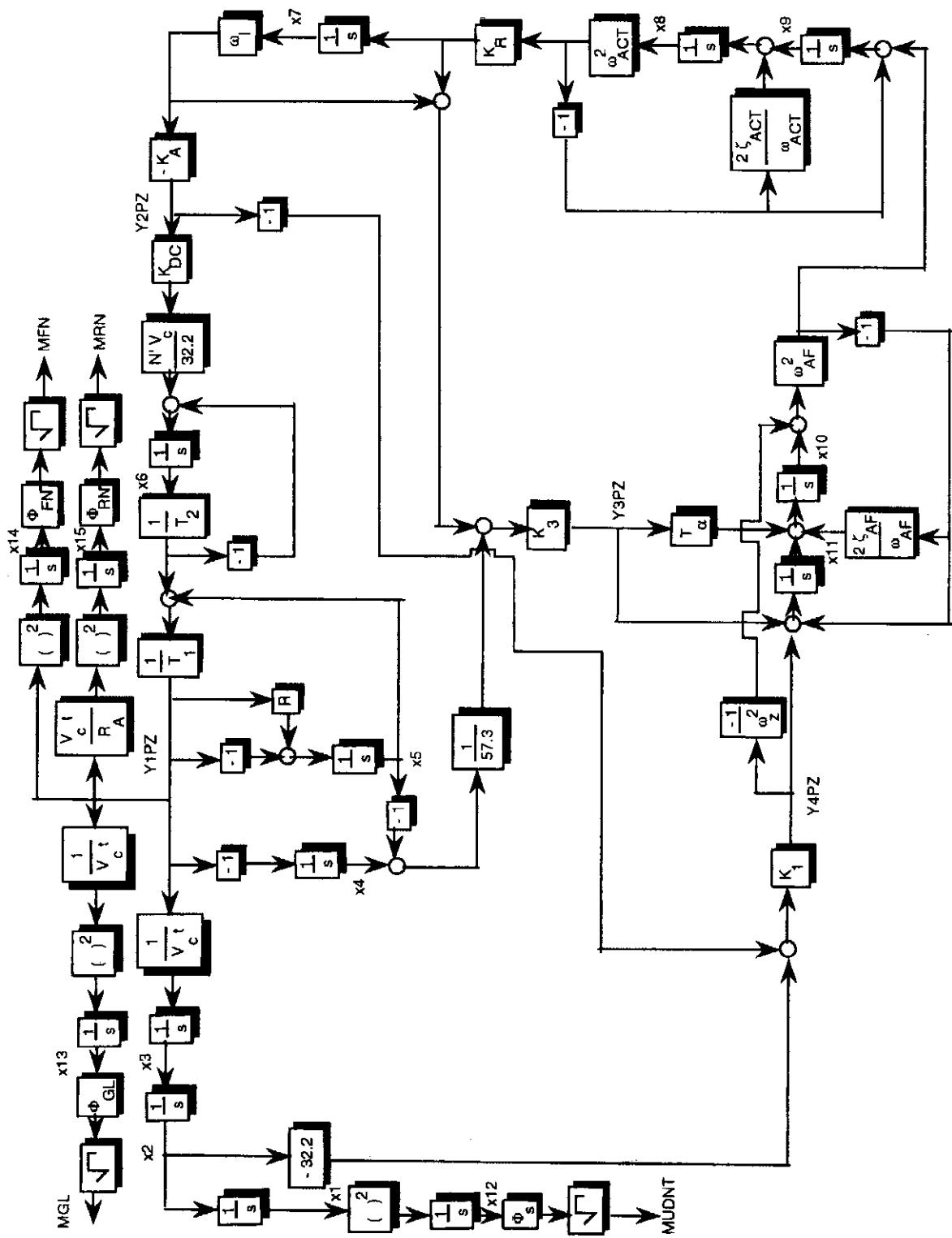


Fig. 24.15 Adjoint model of homing loop.

Table 24.4 Nominal error sources

Error source	Parameter values
Semiactive noise	.01 rad @ 30 kft in 0.01 s
Glint noise	10 ft in 0.1 s
Range independent noise	0.002 rad in 0.01 s
Random target maneuver	2 g over 5 s

$TINT = 0$), fin rate results at a particular time to go ($\text{MISS} = 2$, $TINT = t_{go}$), or commanded acceleration results at a particular time to go ($\text{MISS} = 3$, $TINT = t_{go}$) by simply changing the initial conditions on the adjoint differential equations. We can also see from Listing 24.3 that the aerodynamic parameters and autopilot gains are computed automatically for a given flight condition using the formulas derived in Chapter 22 and this chapter.

Running the adjoint simulation to calculate miss distances for the three different flight conditions yields the miss distance error budget of Table 24.5. Here we can see that the total rms miss distance is independent of flight condition because the autopilot gain algorithm has maintained the same flight-control system response at the three different flight conditions. We can also see from Table 24.5, that for the values of the error sources selected, the major contributors to the miss distance are target maneuver and glint noise. In this particular case, improving the sensor (i.e., reducing range independent noise and semiactive noise) will not significantly reduce the total rms miss distance.

Miss distance results were generated as a function of the radome slope for each of the three different flight conditions. We can see from Fig. 24.16 that at the lower altitudes there is very little miss distance variation with radome slope. However, at 50 kft altitude we can see that the miss distance can be very large due to either large negative or positive slopes. The specification on the allowable radome slope swing will be set at the highest operating altitude.⁶

We can examine the miss distance results at high altitude in more detail. Figure 24.17 shows the miss distance error budget at 50 kft altitude as a function of radome slope. For large negative slopes all the noise error sources cause the miss to be big whereas for large positive slopes the dominant contributor to the miss is random target maneuver. If the maximum rms miss that can be tolerated at

Table 24.5 Miss distance performance appears to be independent of altitude

Error source	0 kft	30 kft	50 kft
Random maneuver	12.4 ft	12.7 ft	13.2 ft
Range independent noise	2.01 ft	2.06 ft	2.17 ft
Semiactive noise	2.27 ft	2.33 ft	2.44 ft
Glint noise	9.03 ft	9.21 ft	9.61 ft
RMS miss	15.6 ft	16.0 ft	16.6 ft

Listing 24.3 Adjoint simulation of homing loop with detailed flight-control system

```

FR=3.
DIAM=1.
XL=20.
CTW=0.
CRW=6.
HW=2.
CTT=0.
CRT=2.
HT=2.
XN=4.
XCG=10.
XHL=19.5
WACT=150.
ZACT=.7
TF=5.
VM=3000.
XNCG=10.
WCR=50.
ZETA=.7
TAU=.3
ALT=0.
XNT=64.4
XNP=3.
VC=4000.
T1=.1
T2=.15
PHIRN=.000002
PHIGL=20.
PHIFN=.00000008
RA=30000.
R=0.
MISS=1
TINT=0.
106 OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
      WRITE(*,*) 
      WRITE(*,*)"ALTITUDE AND AUTOPILOT TIME CONSTANT"
      WRITE(*,*)"ALT=",ALT,'FT','TAU=',TAU,'SEC'
      WRITE(*,*) 
      READ(*,*)ALT,TAU
      A=1000.
      IF(ALT.LE.30000.)THEN
          RHO=.002378*EXP(-ALT/30000.)
      ELSE
          RHO=.0034*EXP(-ALT/22000.)
      ENDIF

      WGT=1000.
      XNLLIN=0.
      XACC=XCG

```

(Contd.)

Listing 24.3 (Continued)

```

SWING=.5*HW*(CTW+CRW)
STAIL=.5*HT*(CTT+CRT)
SREF=3.1416*DIAM*DIAM/4.
XLP=FR*DIAM
SPLAN=(XL-XLP)*DIAM+1.33*XLP*DIAM/2.
XCPN=2*XLP/3
AN=.67*XLP*DIAM
AB=(XL-XLP)*DIAM
XCPB=(.67*AN*XLP+AB*(XLP+.5*(XL-XLP)))/(AN+AB)
XCPW=XLP+XN+.7*CRW-.2*CTW
XMACH=VM/A
XIYY=WGT*(3*((DIAM/2)**2)+XL*XL)/(12*32.2)
TMP1=(XCG-XCPW)/DIAM
TMP2=(XCG-XHL)/DIAM
TMP3=(XCG-XCPB)/DIAM
TMP4=(XCG-XCPN)/DIAM
B=SQRT(XMACH**2-1)
Q=.5*RHO*VM*VM
P1=WGT*XNCG/(Q*SREF)
Y1=2+8*SWING/(B*SREF)+8*STAIL/(B*SREF)
Y2=1.5*SPLAN/SREF
Y3=8*STAIL/(B*SREF)
Y4=2*TMP4 + 8*SWING*TMP1/(B*SREF)+8*STAIL*TMP2/(B*SREF)
Y5=1.5*SPLAN*TMP3/SREF
Y6=8*STAIL*TMP2/(B*SREF)
P2=Y2-Y3*Y5/Y6
P3=Y1-Y3*Y4/Y6
ALFTR=(-P3+SQRT(P3*P3+4.*P2*P1))/(2.*P2)
DELTR=-Y4*ALFTR/Y6-Y5*ALFTR*ALFTR/Y6
CNA=2+1.5*SPLAN*ALFTR/SREF+8*SWING/(B*SREF)+8*STAIL/(B*SREF)
CND=8*STAIL/(B*SREF)
CMAP=2*TMP4 + 1.5*SPLAN*ALFTR*TMP3/SREF+8*SWING*TMP1/(B*SREF)
CMA=CMAP+8*STAIL*TMP2/(B*SREF)
CMD=8*STAIL*TMP2/(B*SREF)
XMA=Q*SREF*DIAM*CMA/XIYY
XMD=Q*SREF*DIAM*CMD/XIYY
ZA=-32.2*Q*SREF*CNA/(WGT*VM)
ZD=-32.2*Q*SREF*CND/(WGT*VM)
WZ=SQRT((XMA*ZD-ZA*XMD)/ZD)
WAF=SQRT(-XMA)
ZAF=.5*WAF*ZA/XMA
XK1=-VM*(XMA*ZD-XMD*ZA)/(1845*XMA)
XK2=XK1
TA=XMD/(XMA*ZD-XMD*ZA)
XK3=1845*XK1/VM
W=(TAU*WCR*(1+2.*ZAF*WAF/WCR)-1)/(2*ZETA*TAU)
W0=W/SQRT(TAU*WCR)
Z0=.5*W0*(2*ZETA/W+TAU-WAF**2/(W0*W0*WCR))
XKC=(-W0**2-1.+2.*Z0*W0*TA)/(1.-2.*Z0*W0*TA+W0*W0*TA*TA)
XKA=XK3/(XK1*XKC)
XK0=-W*W/(TAU*WAF*WAF)

```

(Contd.)

Listing 24.3 (Continued)

```

XK=XK0/(XK1*(1+XKC))
WI=XKC*TA*W0*W0/(1+XKC+W0**2/WZ**2)
XKR=XK/(XKA*WI)
XKDC=1.+1845./(XKA*VM)
DO 33 R=-.06,.06,.005
X1=0.
X2=0.
X3=0.
X4=0.
X5=0.
X6=0.
X7=0.
X8=0.
X9=0.
X10=0.
X11=0.
X12=0.
X13=0.
X14=0.
X15=0.
IF(MISS.EQ.1)THEN
    X3=1.
ELSEIF(MISS.EQ.2)THEN
    X8=1.
ELSEIF(MISS.EQ.3)THEN
    X6=XNP*VC/32.2
ENDIF
T=0
H=.0001
S=0
TP=T+.00001+TINT
10 IF(TP>(TF-.00001)) GOTO 999
S=S+H
X1OLD=X1
X2OLD=X2
X3OLD=X3
X4OLD=X4
X5OLD=X5
X6OLD=X6
X7OLD=X7
X8OLD=X8
X9OLD=X9
X10OLD=X10
X11OLD=X11
X12OLD=X12
X13OLD=X13
X14OLD=X14
X15OLD=X15
STEP=1
GOTO 200
66 STEP=2

```

(Contd.)

Listing 24.3 (Continued)

```

X1=X1+H*X1D
X2=X2+H*X2D
X3=X3+H*X3D
X4=X4+H*X4D
X5=X5+H*X5D
X6=X6+H*X6D
X7=X7+H*X7D
X8=X8+H*X8D
X9=X9+H*X9D
X10=X10+H*X10D
X11=X11+H*X11D
X12=X12+H*X12D
X13=X13+H*X13D
X14=X14+H*X14D
X15=X15+H*X15D
TP=TP+H
GOTO 200
55  CONTINUE
X1=.5*(X1OLD+X1+H*X1D)
X2=.5*(X2OLD+X2+H*X2D)
X3=.5*(X3OLD+X3+H*X3D)
X4=.5*(X4OLD+X4+H*X4D)
X5=.5*(X5OLD+X5+H*X5D)
X6=.5*(X6OLD+X6+H*X6D)
X7=.5*(X7OLD+X7+H*X7D)
X8=.5*(X8OLD+X8+H*X8D)
X9=.5*(X9OLD+X9+H*X9D)
X10=.5*(X10OLD+X10+H*X10D)
X11=.5*(X11OLD+X11+H*X11D)
X12=.5*(X12OLD+X12+H*X12D)
X13=.5*(X13OLD+X13+H*X13D)
X14=.5*(X14OLD+X14+H*X14D)
X15=.5*(X15OLD+X15+H*X15D)
GOTO 10
200  CONTINUE
TGO=TP+.00001
X1D=X2
X2D=X3
Y1PZ=(X6/T2+X5)/T1
X3D=Y1PZ/(VC*TGO)
X4D=-Y1PZ
X5D=-Y1PZ+R*Y1PZ
Y2PZ=-XKA*WI*X7
X6D=-X6/T2+XNP*VC*XKDC*Y2PZ/32.2
X7D=XKR*WACT*WACT*X8
X8D=X9-2.*ZACT*WACT*X8
Y4PZ=XK1*(-32.2*X2-Y2PZ)
X9D=WACT*WACT*X8+X10*WAF*WAF-WAF*WAF*Y4PZ/WZ**2
Y3PZ=XK3*(X7D+WI*X7+(X4-X5)/57.3)
X10D=-2.*ZAF*WAF*(X10-Y4PZ/WZ**2)+X11+TA*Y3PZ
X11D=-WAF*WAF*(X10-Y4PZ/WZ**2)+Y4PZ+Y3PZ

```

(Contd.)

Listing 24.3 (Continued)

```

X12D=X1*X1
X13D=(Y1PZ/(VC*TGO))**2
X14D=(Y1PZ*VC*TGO/RA)**2
X15D=Y1PZ**2
IF(STEP~1)66,66,55
999 CONTINUE
XMNT=XNT*X1
XMFN=SQRT(X15*PHIFN)
XMRN=SQRT(X14*PHIRN)
XMGL=SQRT(X13*PHIGL)
XMUDNT=XNT*SQRT(X12/TGO)
RMS=SQRT(XMFN**2+XMRN**2+XMGL**2+XMUDNT**2)
WRITE(1,97)R,XMFN,XMRN,XMGL,XMUDNT,RMS
WRITE(*,97)R,XMFN,XMRN,XMGL,XMUDNT,RMS
33 CONTINUE
97 FORMAT(6F10.3)
CLOSE(1)
PAUSE
GOTO 106
END

```

50 kft altitude is 25 ft, then the radome slope swing would have to be confined to range from $R = -0.03$ to $R = 0.03$ for a total radome swing of 0.06.

Increasing the flight-control system time constant at high altitude is sometimes effective in reducing system sensitivity to radome slope. Figure 24.18 shows that when the flight-control system time constant is increased from 0.3 s to 0.5 s the guidance system performance appears to be less sensitive to radome slope. However a closer examination of Fig. 24.18 reveals that if the allowable rms miss is 25 ft then the maximum negative slope that could be tolerated is -0.05 and the maximum positive slope which could be tolerated is 0.01 for a total radome swing

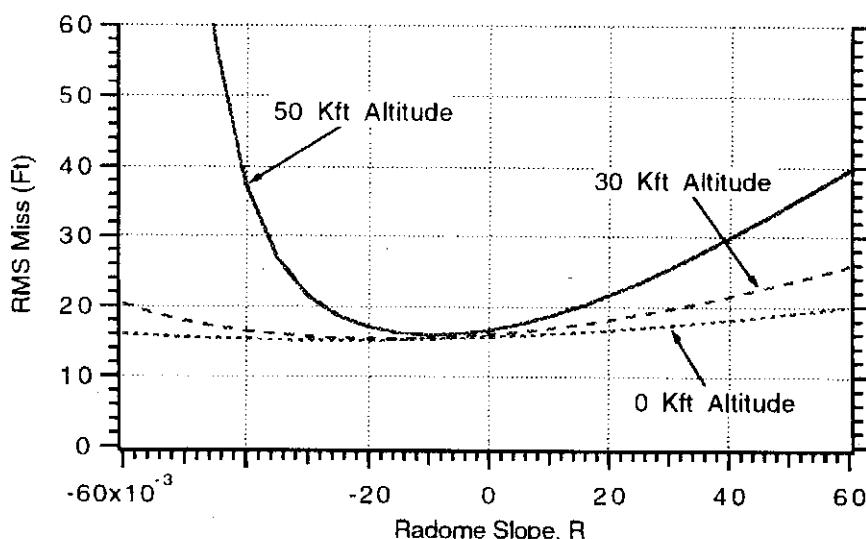


Fig. 24.16 High-altitude performance is more sensitive to radome slope.

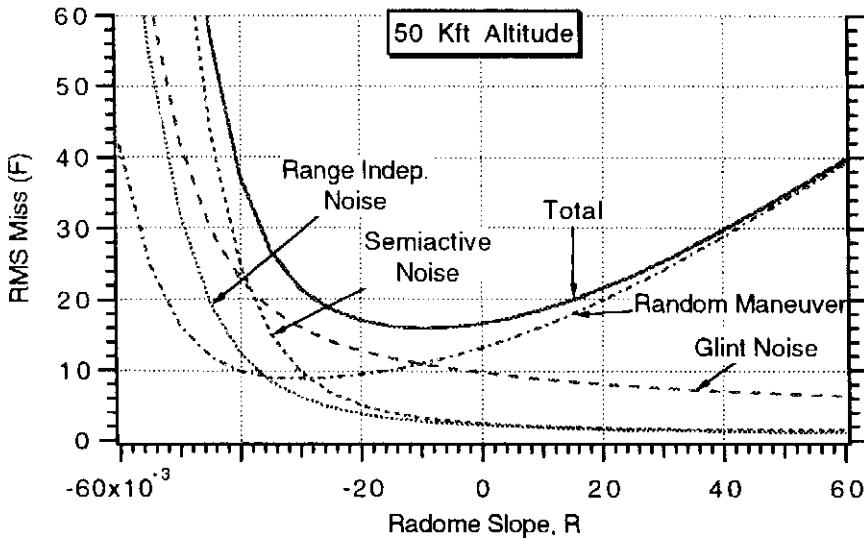


Fig. 24.17 Miss distance error budget at 50 kft altitude.

of 0.06, which is the same radome swing as before. Soon we will find another reason why increasing the time constant may be beneficial at high altitudes.

We can also run the adjoint program to generate fin rate results at a specific time to go before intercept. Therefore a number of adjoint runs will yield fin rate results for all times to go. These results can be inverted for a particular flight time to yield rms fin rate as a function of time. Figure 24.19 displays rms fin rate as a function of time, obtained from the adjoint program of Listing 24.3, for a 5-s flight assuming zero radome slope for the three different flight conditions. We can see that the largest fin rates occur at the higher altitudes and at the end of the flight. Figure 24.19 indicates that if intercepts are to be supported at 50 kft altitude for this example, the actuator must be able to support rms fin rates in excess of 500 deg/s.

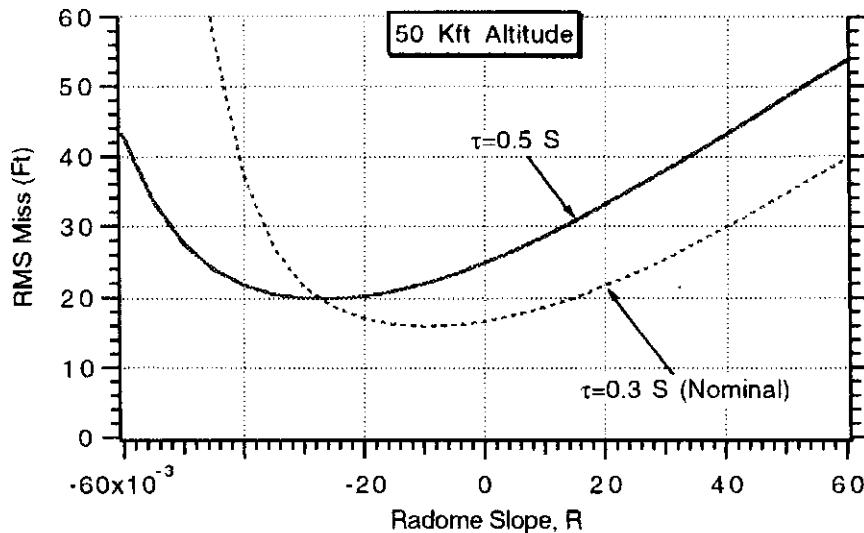


Fig. 24.18 Sometimes increasing flight-control system time constant reduces sensitivity to radome slope.

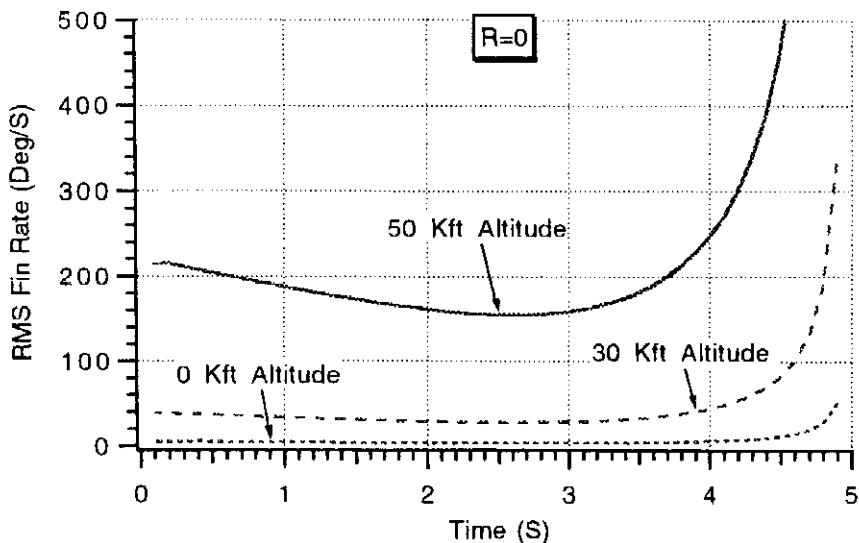


Fig. 24.19 Fin rate increases with increasing altitude.

The fin rate results can be examined in more detail at 50 kft altitude. Figure 24.20 presents the adjoint fin rate error budget. We can see that at the beginning of flight the large fin rates are due to semiactive noise. An improved sensor will relax these fin rate requirements. However at the end of the flight the large fin rates are mainly due to glint noise. Because the glint noise is usually a function of the target and not the sensor, it would be difficult to relax the fin rate requirement at this altitude.

Figure 24.21 shows that the fin rate can be significantly reduced by increasing the flight-control system time constant from 0.3 s to 0.5 s. Therefore increasing the flight-control system time constant at the higher altitudes might be considered to be a viable option for a given level of actuator performance.

The adjoint program was run again to yield commanded acceleration information. We can see from Fig. 24.22 that the rms commanded acceleration is independent of flight condition. For this example 25 g of rms missile acceleration capability would be required to prevent the missile from acceleration saturation. Avoiding saturation will validate the rms miss distance results of the

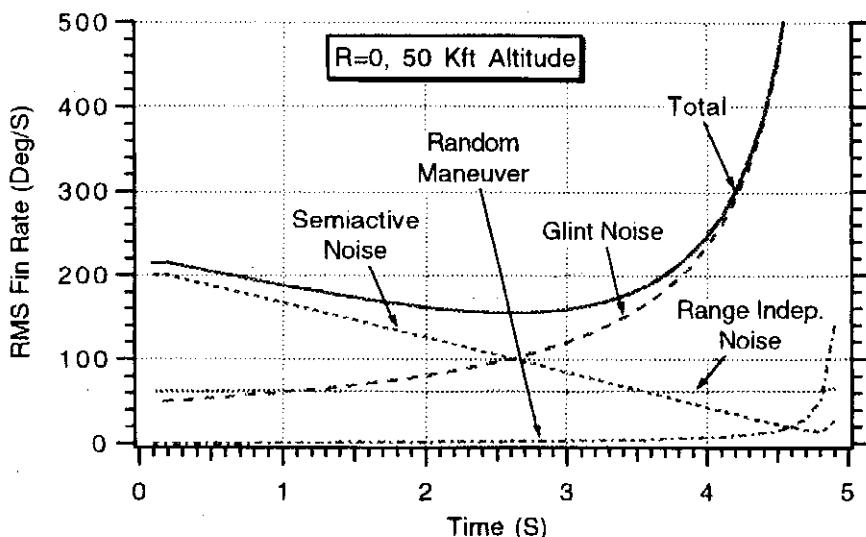


Fig. 24.20 Fin rate error budget at 50 kft altitude.

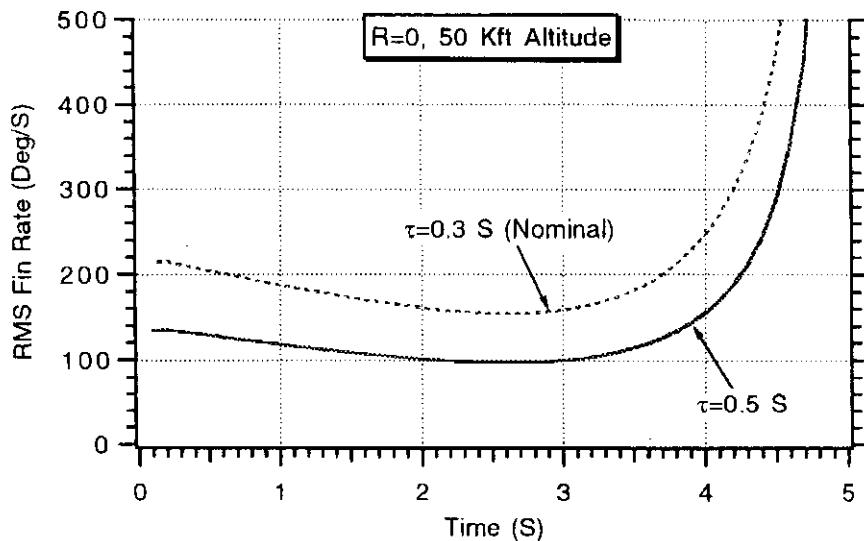


Fig. 24.21 Fin rate can be reduced by increasing flight-control system time constant.

previous figures. If saturation occurs, the rms miss distance will be considerably higher.

A typical rms commanded acceleration error budget, obtained from the adjoint simulation of Listing 24.3, appears in Fig. 24.23. Here we can see that near the end of the flight most of the rms commanded acceleration is due to both random target maneuver and glint noise. Therefore an improved sensor will not relax the acceleration requirements since only the acceleration due to semiactive and range independent noise will be reduced. The guidance system must be sized to handle the commanded acceleration at the end of the flight due to the random target maneuver and glint noise.

Finally Fig. 24.24 shows that increasing the flight-control system time constant does not reduce the rms commanded acceleration requirements. Therefore increasing the time constant is mainly an option for reducing the fin rate requirements and possibly also to desensitize the system to radome effects but not for reducing the acceleration requirements.

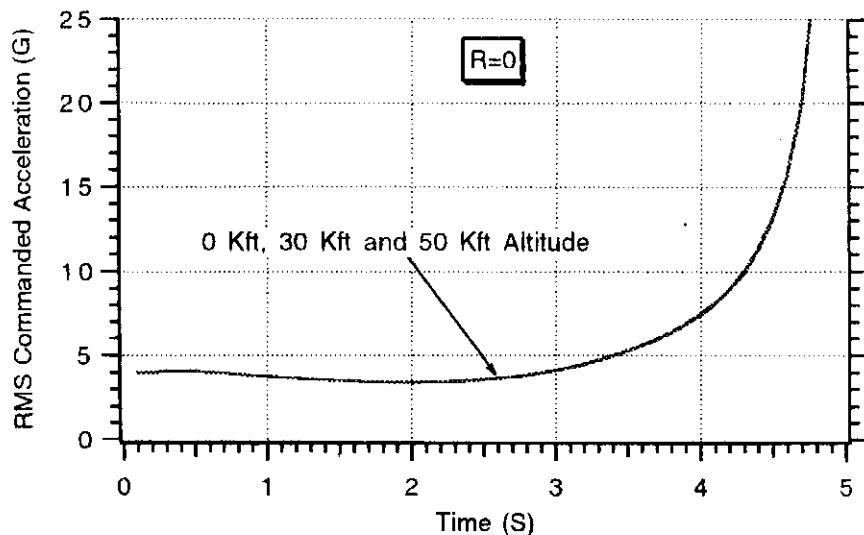


Fig. 24.22 Commanded acceleration profile is independent of altitude.

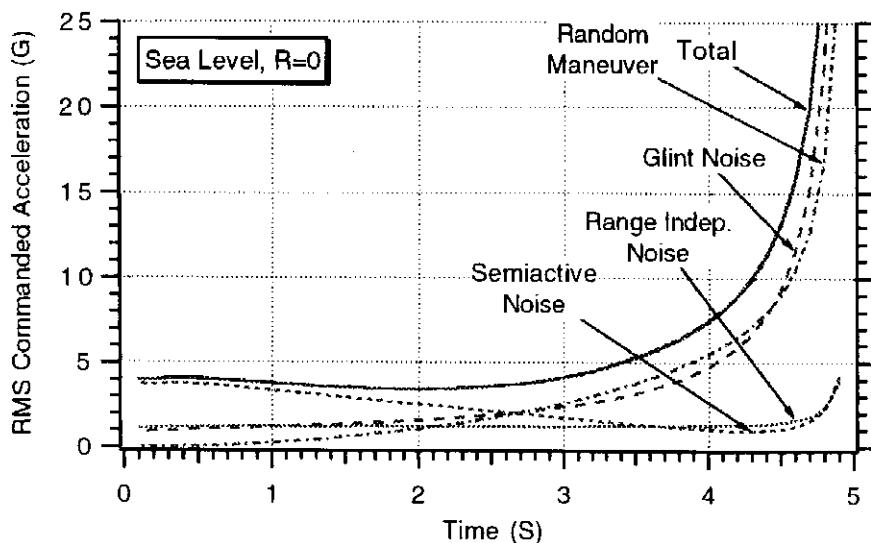


Fig. 24.23 Acceleration error budget at sea level.

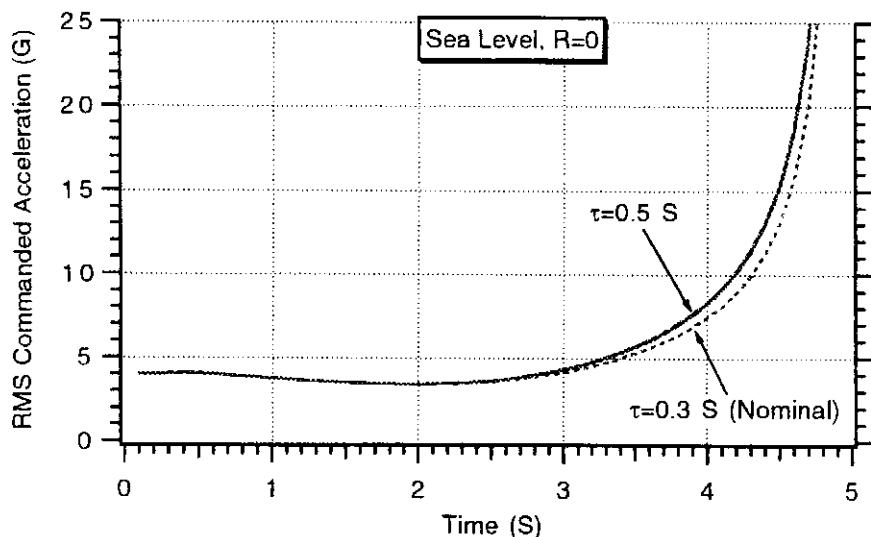


Fig. 24.24 Increasing flight-control system time constant does not reduce acceleration requirements.

Summary

In this chapter we have shown how the three-loop autopilot and associated gain selection algorithm can be used to independently specify the time constant, damping, and crossover frequency. Typical performance studies were conducted using the method of adjoints to show how miss distance results could be generated, actuator requirements set, and missile acceleration requirements derived.

References

¹Fossier, M. W., "The Development of Radar Homing Missiles," *Journal of Guidance, Control, and Dynamics*, Vol. 7, Nov.–Dec. 1984, pp. 641–651.

²Stallard, D. V., "An Approach to Autopilot Design for Homing Interceptor Missiles," *Proceedings of the 1991 AIAA Guidance and Control Conference*, AIAA, Washington, DC, 1991.

³Nesline, F. W., and Nabbeleld, N. C., "Design of Digital Autopilots for Homing Missiles," *Proceedings of AGARD Flight Mechanics Panel Symposium*, London, May 1979.

⁴Wells, B. H., "Tactical Missile Structural Testing and Model Verification for Autopilot Design," *Proceedings of the 1991 AIAA Guidance and Control Conference*, AIAA, Washington, DC, 1991.

⁵Gratt, H. J., and Zarchan, P., "A Practical Approach to Augmenting Missile Autopilot Design (The 3-Loop Autopilot on Steroids)," *Proceedings of AIAA/BMDO Interceptor Technology Conference*, San Diego, CA, July 1994.

⁶Horton, M. P., "A Study of Autopilots for the Adaptive Control of Tactical Guided Missiles," MSc Thesis, Univ. of Bath, Bath, UK, 1992.

⁷Nesline, F. W., and Zarchan, P., "Radome Induced Miss Distance in Aerodynamically Controlled Homing Missiles," *Proceedings of AIAA Guidance and Control Conference*, AIAA, New York, 1984.

Trajectory Shaping Guidance

Introduction

In all of the guidance work done thus far, the goal has been to hit the target using the least amount of energy. In some applications, in addition to hitting the target, it may also be desirable to shape the missile trajectory near impact. For example, in antitank or antiballistic missile applications we may want to have the missile approach the target at certain strike angles to improve lethality. In this chapter we will show how the guidance problem can be reformulated so that a new guidance law can be developed that both hits the target using minimum energy and, in addition, travels on the desired trajectory. It will be shown that the new guidance law is actually the same one used to land the Apollo spacecraft on the moon. We will then evaluate the trajectory shaping guidance law and see how it performs in a more realistic nonlinear environment.

Problem Setup

Before we derive the new guidance law, we must first express mathematically what we desire to do. Let us first revisit our homing loop model for a zero-time constant guidance system as shown in Fig. 25.1

As was the case in Chapter 8 for deriving augmented proportional navigation, we are still assuming a constant target maneuver, which means that the derivative of n_T must be zero. Therefore we can express the model of Fig. 25.1 in matrix form as

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{n}_T \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ n_T \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} n_c$$

This is the same form as the state space matrix differential equation

$$\dot{x} = Fx + Gu$$

As with our other guidance problems, we still desire to minimize the integral of the commanded acceleration squared. In addition, we want the miss distance to be zero. We will soon see that selecting the relative velocity at the end of flight to be some specified value is the same as shaping the missile trajectory. Mathematically we have just stated that our goals are

$$y(t_F) = 0 \quad \text{and} \quad \dot{y}(t_F) = \dot{y}_F \quad \text{subject to minimizing} \int_0^{t_F} n_c^2(t) dt$$

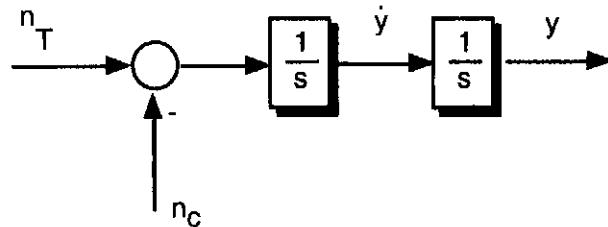


Fig. 25.1 Zero-time constant homing loop model for guidance law development.

Recall from Chapter 8 that the general solution to the state space differential equation describing the homing loop model is given by the vector relationship

$$\mathbf{x}(t_F) = \Phi(t_F - t)\mathbf{x}(t) + \int_t^{t_F} \Phi(t_F - \lambda) \mathbf{G}(\lambda) \mathbf{u}(\lambda) d\lambda$$

where Φ is the fundamental matrix. By comparing the state space equation to our matrix equation representing the homing loop model of Fig. 25.1 we can see that \mathbf{F} , \mathbf{G} and \mathbf{u} are given by

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$$

$$\mathbf{u} = n_c$$

We have already shown in Chapter 8 that for the systems dynamics matrix \mathbf{F} under consideration the continuous fundamental matrix is given by

$$\Phi(t) = \begin{bmatrix} 1 & t & .5t^2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}$$

Substitution of the preceding matrices into the general solution of the state space equation yields

$$\begin{bmatrix} y(t_F) \\ \dot{y}(t_F) \\ n_T(t_F) \end{bmatrix} = \begin{bmatrix} 1 & t_F - t & .5(t_F - t)^2 \\ 0 & 1 & t_F - t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y(t) \\ \dot{y}(t) \\ n_T(t) \end{bmatrix} + \int_t^{t_F} \begin{bmatrix} 1 & t_F - \lambda & .5(t_F - \lambda)^2 \\ 0 & 1 & t_F - \lambda \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} n_c(\lambda) d\lambda$$

We can multiply out the preceding matrix equation and end up with three scalar equations. Because we are only interested in controlling the relative position and

velocity at the end of flight we can ignore the last scalar equation of the three equations and obtain

$$y(t_F) = y(t) + (t_F - t)\dot{y}(t) + .5(t_F - t)^2 n_T(t) - \int_t^{t_F} (t_F - \lambda)n_c(\lambda) d\lambda$$

$$\dot{y}(t_F) = \dot{y}(t) + (t_F - t)n_T(t) - \int_t^{t_F} n_c(\lambda) d\lambda$$

We will see in the next section how the Schwartz inequality can be used on the preceding two scalar equations to obtain the trajectory shaping guidance law.

Using the Schwartz Inequality for Trajectory Shaping Guidance

We will first simplify the guidance problem by using shorthand notation to represent the two scalar equations that were just derived in the preceding section. Let us first define

$$f_1 = y(t) + (t_F - t)\dot{y}(t) + .5(t_F - t)^2 n_T(t)$$

$$f_2^* = \dot{y}(t) + (t_F - t)n_T(t)$$

$$h_1(\lambda) = t_F - \lambda$$

$$h_2(\lambda) = 1$$

We can now rewrite the two scalar equations at the end of the preceding section in shorthand notation as

$$y(t_F) = f_1 - \int_t^{t_F} h_1(\lambda)n_c(\lambda) d\lambda$$

$$\dot{y}(t_F) = f_2^* - \int_t^{t_F} h_2(\lambda)n_c(\lambda) d\lambda$$

Recall that we want the miss distance to be zero [i.e., $y(t_F) = 0$] and we also want to specify the value of the relative velocity at the terminal time. Therefore, the two preceding equations simplify to

$$f_1 = \int_t^{t_F} h_1(\lambda)n_c(\lambda) d\lambda$$

$$f_2^* - \dot{y}(t_F) = \int_t^{t_F} h_2(\lambda)n_c(\lambda) d\lambda = f_2$$

Let us now combine the two preceding scalar equations into one equation by introducing a new variable δ or

$$f_1 - \delta f_2 = \int_t^{t_F} [h_1(\lambda) - \delta h_2(\lambda)]n_c(\lambda) d\lambda$$

If we apply the Schwartz inequality to the preceding expression we obtain

$$(f_1 - \delta f_2)^2 \leq \int_t^{t_F} [h_1(\lambda) - \delta h_2(\lambda)]^2 d\lambda \int_t^{t_F} n_c^2(\lambda) d\lambda$$

Inverting the preceding equation and solving for the integral of the square of the commanded acceleration yields

$$\int_t^{t_F} n_c^2(\lambda) d\lambda \geq \frac{(f_1 - \delta f_2)^2}{\int_t^{t_F} [h_1(\lambda) - \delta h_2(\lambda)]^2 d\lambda}$$

The integral of the square of the acceleration will be minimized when the equality sign of the preceding inequality holds. According to the Schwartz inequality, the equality sign holds when

$$n_c(\lambda) = K[h_1(\lambda) - \delta h_2(\lambda)]$$

where K is a constant. We will soon use the preceding equation. When the equality sign holds, the integral of the square of the commanded acceleration can be expanded to

$$\begin{aligned} z &= \int_t^{t_F} n_c^2(\lambda) d\lambda = \frac{(f_1 - \delta f_2)^2}{\int_t^{t_F} [h_1(\lambda) - \delta h_2(\lambda)]^2 d\lambda} \\ &= \frac{(f_1 - \delta f_2)^2}{\int_t^{t_F} h_1^2(\lambda) d\lambda - 2\delta \int_t^{t_F} h_1(\lambda)h_2(\lambda) d\lambda + \delta^2 \int_t^{t_F} h_2^2(\lambda) d\lambda} \end{aligned}$$

In the work that follows we will be performing many algebraic manipulations. To simplify our task, we will again use shorthand notation. We can define

$$\begin{aligned} \|h_1^2\| &= \int_t^{t_F} h_1^2(\lambda) d\lambda \\ \|h_2^2\| &= \int_t^{t_F} h_2^2(\lambda) d\lambda \\ \|h_1 h_2\| &= \int_t^{t_F} h_1(\lambda)h_2(\lambda) d\lambda \end{aligned}$$

Therefore, our expression for the integral of the square of the commanded acceleration simplifies to

$$z = \frac{(f_1 - \delta f_2)^2}{\|h_1^2\| - 2\delta \|h_1 h_2\| + \delta^2 \|h_2^2\|}$$

Now we have to choose a value for the variable δ . The best value of δ is one that minimizes the preceding expression. We know from calculus that the value of δ that minimizes the preceding expression can be found by taking the derivative of

the preceding expression with respect to δ and setting the result to zero. Using the quotient rule from calculus to take the derivative yields

$$\frac{dz}{d\delta} = 0 = \frac{\left[\|h_1^2\| - 2\delta\|h_1h_2\| + \delta^2\|h_2^2\| \right] 2(f_1 - \delta f_2)(-f_2) - (f_1 - \delta f_2)^2 \left[-2\|h_1h_2\| + 2\delta\|h_2^2\| \right]}{\left[\|h_1^2\| - 2\delta\|h_1h_2\| + \delta^2\|h_2^2\| \right]^2}$$

We can now solve the preceding expression for δ . After some algebra, we obtain

$$\delta = \frac{f_1\|h_1h_2\| - f_2\|h_1^2\|}{f_1\|h_2^2\| - f_2\|h_1h_2\|}$$

Because we know that

$$f_1 = \int_t^{t_F} h_1(\lambda) n_c(\lambda) d\lambda$$

and

$$n_c(\lambda) = K[h_1(\lambda) - \delta h_2(\lambda)]$$

Substitution yields

$$f_1 = \int_t^{t_F} h_1(\lambda) K[h_1(\lambda) - \delta h_2(\lambda)] d\lambda$$

If we solve for the constant K we obtain

$$K = \frac{f_1}{\int_t^{t_F} h_1(\lambda)[h_1(\lambda) - \delta h_2(\lambda)] d\lambda} = \frac{f_1}{\|h_1^2\| - \delta\|h_1h_2\|}$$

Therefore, the optimal acceleration command can be rewritten as

$$n_c(\lambda) = K[h_1(\lambda) - \delta h_2(\lambda)] = \frac{f_1[h_1(\lambda) - \delta h_2(\lambda)]}{\|h_1^2\| - \delta\|h_1h_2\|}$$

Substitution of the optimal value of δ in the preceding expression yields

$$n_c(\lambda) = \frac{f_1 \left[h_1(\lambda) - h_2(\lambda) \left[\frac{f_1\|h_1h_2\| - f_2\|h_1^2\|}{f_1\|h_2^2\| - f_2\|h_1h_2\|} \right] \right]}{\|h_1^2\| - \|h_1h_2\| \left[\frac{f_1\|h_1h_2\| - f_2\|h_1^2\|}{f_1\|h_2^2\| - f_2\|h_1h_2\|} \right]}$$

After some algebra, we obtain

$$n_c(\lambda) = \frac{f_1 h_1(\lambda) \|h_2^2\| - \|h_1h_2\| [f_2 h_1(\lambda) + f_1 h_2(\lambda)] + f_2 h_2(\lambda) \|h_1^2\|}{\|h_1^2\| \|h_2^2\| - \|h_1h_2\|^2}$$

or in the time domain

$$n_c(t) = \frac{f_1 h_1(t) \|h_2^2\| - \|h_1 h_2\| [f_2 h_1(t) + f_1 h_2(t)] + f_2 h_2(t) \|h_1^2\|}{\|h_1^2\| \|h_2^2\| - \|h_1 h_2\|^2}$$

Now we have enough information to evaluate the numerator and denominator of the preceding expression to find the new guidance law. Recall that for this problem

$$f_1 = y(t) + (t_F - t)\dot{y}(t) + .5(t_F - t)^2 n_T(t) = y + t_{go}\dot{y} + .5t_{go}^2 n_T$$

$$f_2 = f_2^* - \dot{y}(t_F) = \dot{y}(t) + (t_F - t)n_T(t) - \dot{y}(t_F) = \dot{y} + t_{go} n_T - \dot{y}(t_F)$$

$$h_1(t) = t_F - t = t_{go}$$

$$h_2(t) = 1$$

Therefore, we can easily solve the necessary integrals as

$$\|h_1^2\| = \int_t^{t_F} h_1^2(\lambda) d\lambda = \int_t^{t_F} (t_F - \lambda)^2 d\lambda = \frac{t_{go}^3}{3}$$

$$\|h_2^2\| = \int_t^{t_F} h_2^2(\lambda) d\lambda = \int_t^{t_F} d\lambda = t_{go}$$

$$\|h_1 h_2\| = \int_t^{t_F} h_1(\lambda) h_2(\lambda) d\lambda = \int_t^{t_F} (t_F - \lambda) d\lambda = \frac{t_{go}^2}{2}$$

where the time to go until intercept is given by

$$t_{go} = t_F - t$$

Substituting the preceding expressions into the formula for the acceleration command yields

$$n_c(t) = \left\{ (y + t_{go}\dot{y} + .5t_{go}^2 n_T) t_{go} - .5t_{go}^2 \left[(\dot{y} + t_{go} n_T - \dot{y}_F) t_{go} + (y + t_{go}\dot{y} + .5t_{go}^2 n_T)(1) \right] + (\dot{y} + t_{go} n_T - \dot{y}_F)(1) \frac{t_{go}^3}{3} \right\} \Bigg/ \left[\frac{t_{go}^3}{3} t_{go} - \left(\frac{t_{go}^2}{2} \right)^2 \right]$$

After some algebra, we see that the new trajectory shaping guidance law simplifies to

$$n_c(t) = \frac{6y + 4\dot{y}t_{go} + n_T t_{go}^2 + 2\dot{y}(t_F)t_{go}}{t_{go}^2}$$

The guidance law that landed the Apollo spacecraft on the moon in 1969 used the preceding guidance law. In the Apollo case, there was no target acceleration and the relative velocity at intercept was chosen to be zero (that is, this special case is also known as a rendezvous). Therefore, the Apollo guidance law is simply^{1, 2, 3, 4}

$$n_c(t)_{\text{Apollo}} = \frac{6y + 4\dot{y}t_{go}}{t_{go}^2}$$

Alternate Form of Trajectory Shaping Guidance Law

We can rewrite the trajectory shaping guidance law of the preceding section as

$$n_c(t) = \frac{4y + 4\dot{y}t_{go} + n_T t_{go}^2 + 2y + 2\dot{y}(t_F)t_{go}}{t_{go}^2} = \frac{4(y + \dot{y}t_{go})}{t_{go}^2}$$

$$+ \frac{2[y + \dot{y}(t_F)t_{go}] + n_T t_{go}^2}{t_{go}^2}$$

Recall that the formula for the line of sight angle is given by

$$\lambda = \frac{y}{R_{TM}} = \frac{y}{V_c t_{go}} = \frac{y}{V_c(t_F - t)}$$

Therefore, the line of sight rate can be found by differentiating the preceding expression using the quotient rule from calculus. After some algebra, we obtain

$$\dot{\lambda} = \frac{y + \dot{y}t_{go}}{V_c t_{go}^2}$$

Therefore, the trajectory shaping guidance law simplifies to

$$n_c(t) = 4V_c \dot{\lambda} + \frac{2[\lambda V_c + \dot{y}(t_F)]}{t_{go}} + n_T$$

To express the guidance law in terms of a final angle rather than a final relative velocity, we can invert the expression for the line of sight rate and solve for the relative velocity or

$$\dot{y} = \frac{\dot{\lambda} V_c t_{go}^2 - y}{t_{go}} = \frac{\dot{\lambda} V_c t_{go}^2 - \dot{\lambda} V_c t_{go}}{t_{go}} = \dot{\lambda} V_c t_{go}^2 - \lambda V_c$$

We can evaluate the preceding expression at intercept. At the end of the flight, time to go is zero and the final line of sight angle is λ_F . Therefore, at the end of the flight we can see that the relative velocity is simply

$$\dot{y}(t_F) = -\lambda(t_F)V_c = -\lambda_F V_c$$

and the trajectory shaping guidance law simplifies to

$$n_c(t) = 4V_c \dot{\lambda} + \frac{2V_c[\lambda - \lambda_F]}{t_{go}} + n_T$$

Thus, we can see that we can think of the trajectory shaping guidance law as one that minimizes the integral of the square of the commanded acceleration, makes the miss zero, and drives the final line of sight angle to the designer-chosen value λ_F . The trajectory shaping guidance law appears to be a form of augmented proportional navigation (with an effective navigation ratio of 4 and a different multiplier for the target acceleration term) plus an extra term that is proportional to the difference between the true line of sight angle and the desired line of sight angle at the end of the flight.

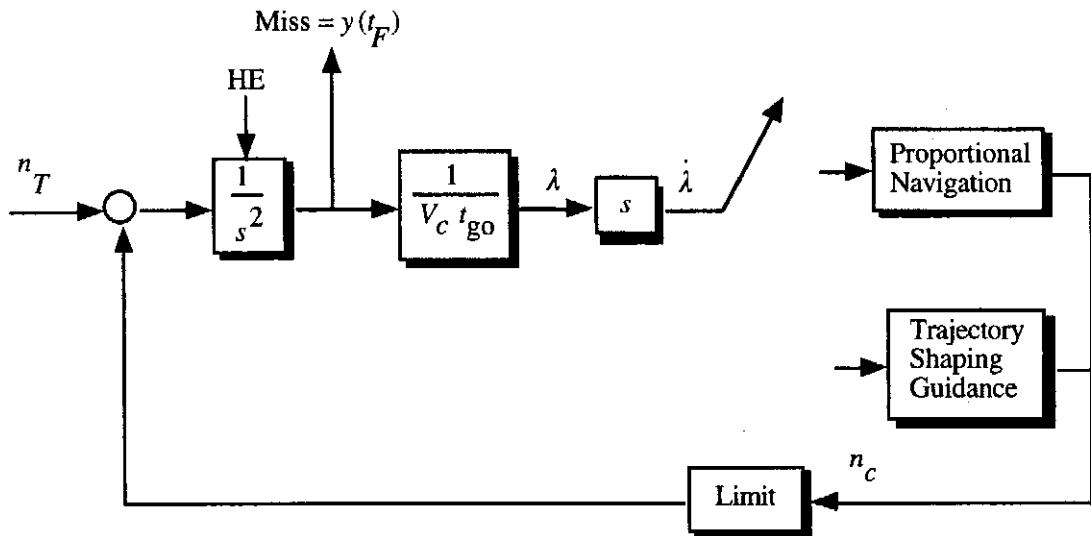


Fig. 25.2 Homing loop model for guidance law comparison.

Testing Trajectory Shaping Guidance in the Linear World

Now that the trajectory shaping guidance law has been derived, it is important to first test the new guidance law in the linear world to see if it works as anticipated. In addition, we would like to compare trajectory shaping guidance with proportional navigation in terms of both accuracy and acceleration requirements. Figure 25.2 presents a block diagram of a zero-lag homing loop to be used in evaluating both proportional navigation and trajectory shaping guidance. We can see from the homing loop that the two sources of error considered are target maneuver, n_T , and heading error, HE. We can see from Fig. 25.2 that the two measures of performance will be the miss distance, $y(t_F)$, and the final line of sight angle, λ_F .

The homing loop model of Fig. 25.2 was programmed, and the resultant engagement simulation appears in Listing 25.1. We can see from the listing that the parameter PN determines the type of guidance law to be used. If $PN = 1$ then proportional navigation is used, whereas if $PN = 0$ the trajectory shaping guidance law is used. When trajectory shaping guidance is used the final specified line of sight angle is denoted XLAMFDEG and is in units of degrees. The trajectory shaping guidance law assumes that time-to-go information and line of sight angle and rate information are available. It is important to note that proportional navigation does not require time-to-go information.

The nominal case of Listing 25.1 was run in which there was a 10-s flight and -20 deg of heading error. When the trajectory shaping guidance law is used it is specified that the final line of sight angle should be -30 deg. We can see from Fig. 25.3 that both guidance laws enable the missile to take out the heading error and hit the target because $y(10) = 0$ in both cases. However, we can see that the relative trajectories are totally different for both guidance laws.

The acceleration requirements for both guidance laws are displayed in Fig. 25.4. We can see that significantly more commanded acceleration is required for trajectory shaping guidance to take out 20 deg of heading error (25 g at the beginning of the flight and approximately -30 g near the end of the flight) than is required with proportional navigation (10 g at the beginning of the flight and 0 g near the end of the flight).

Listing 25.1 Linear engagement simulation for guidance law comparison

```

IMPLICIT REAL*8(A-H,O-Z)
INTEGER STEP,PN
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
XNT=0.
HEDEG=-20.
XNCLIM=999999.
PN=0
XLAMFDEG=-30.
VC=4000.
VM=3000.
TF=10.
XNP=3.
XLAMF=XLAMFDEG/57.3
Y=0.
YD=- VM*HEDEG/57.3
T=0.
H=.001
S=0.
10 IF(T>(TF-.0001))GOTO 999
YOLD=Y
YDOLD=YD
STEP=1
GOTO 200
66 STEP=2
Y=Y+H*YD
YD=YD+H*YDD
T=T+H
GOTO 200
55 CONTINUE
Y=.5*(YOLD+Y+H*YD)
YD=.5*(YDOLD+YD+H*YDD)
S=S+H
IF(S<.09999)GOTO 10
S=0.
XLAMDEG=XLAM*57.3
WRITE(9,*)T,Y,XNC/32.2,XLAMDEG
WRITE(1,*)T,Y,XNC/32.2,XLAMDEG
GOTO 10
200 CONTINUE
TGO=TF-T+.00001
XLAM=Y/(VC*TGO)
XLAMD=(Y+YD*TGO)/(VC*TGO*TGO)
IF(PN.EQ.1)THEN
    XNC=XNP*VC*XLAMD
ELSE
    XNC=4.*VC*XLAMD+XNT+2.*VC*(XLAM-XLAMF)/TGO
ENDIF
IF(XNC>XNCLIM)XNC=XNCLIM
IF(XNC<-XNCLIM)XNC=-XNCLIM
YDD=XNT-XNC

```

(Contd.)

Listing 25.1 (Continued)

```

IF(STEP-1)66,66,55
999  CONTINUE
PAUSE
CLOSE(1)
END

```

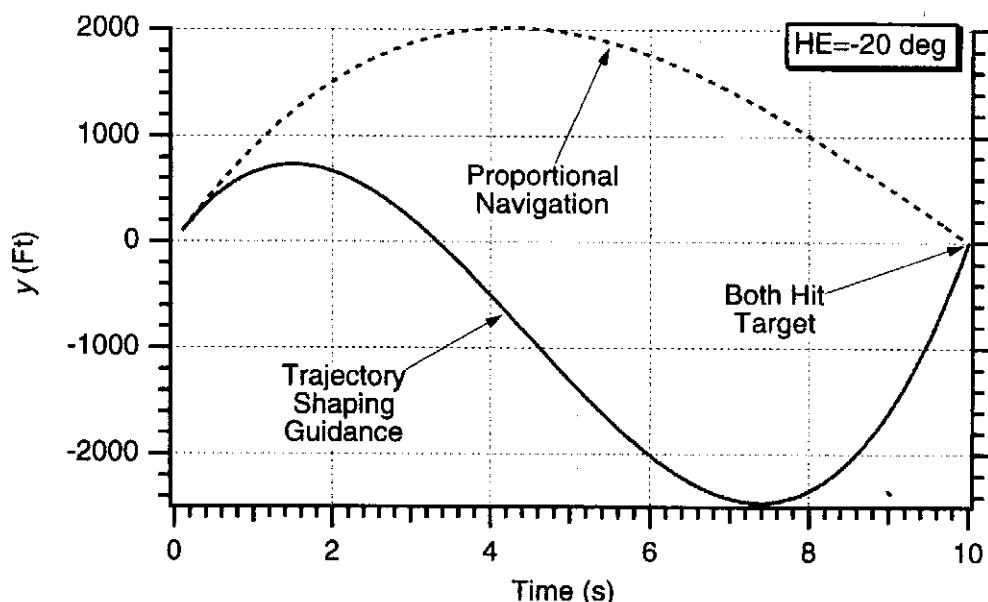


Fig. 25.3 Both guidance laws enable missile to take out the heading error disturbance and hit the target using different relative trajectories.

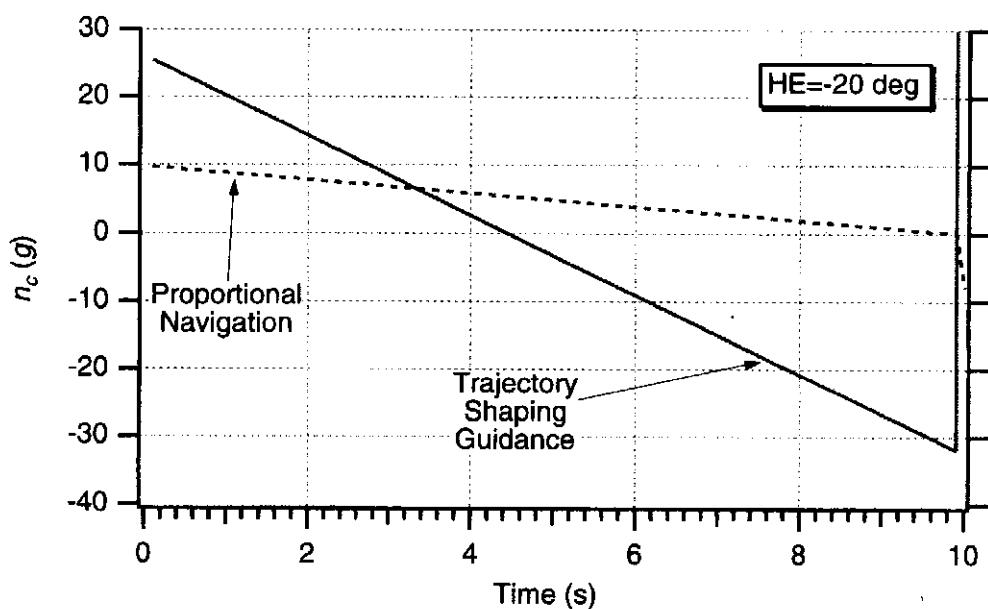


Fig. 25.4 Trajectory shaping guidance requires more acceleration than proportional navigation to take out heading error disturbance.

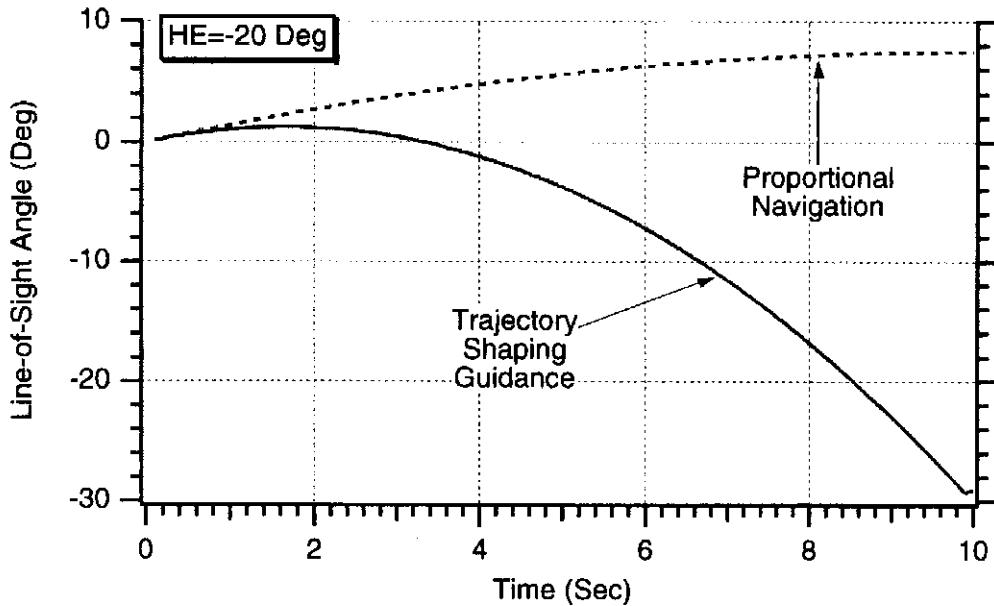


Fig. 25.5 Trajectory shaping guidance law can control final line of sight angle.

Finally, we can see from Fig. 25.5 that with trajectory shaping guidance the line of sight angle matches the design goal of -30 deg at the end of the flight. With proportional navigation the final line of sight angle is not controlled and it is really a matter of luck on what that angle will be (that is, approximately 8 deg in this example). Thus, we can say that simulation results indicate that trajectory shaping guidance appears to be working correctly against the heading error disturbance.

Next, both guidance laws were compared in terms of their response to a $6 g$ target maneuver. Again, we can see from Fig. 25.6 that both guidance laws enable the missile to hit the maneuvering target because $y(10) = 0$ in both cases. As was the case before, both guidance laws result in relative trajectories that are significantly different.

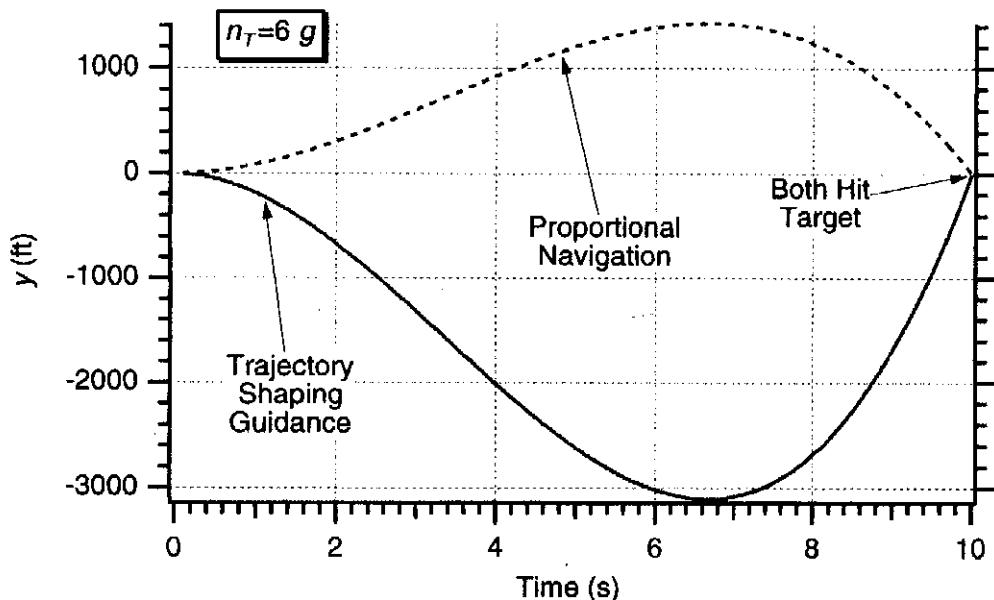


Fig. 25.6 Trajectory shaping guidance law can also hit maneuvering target.

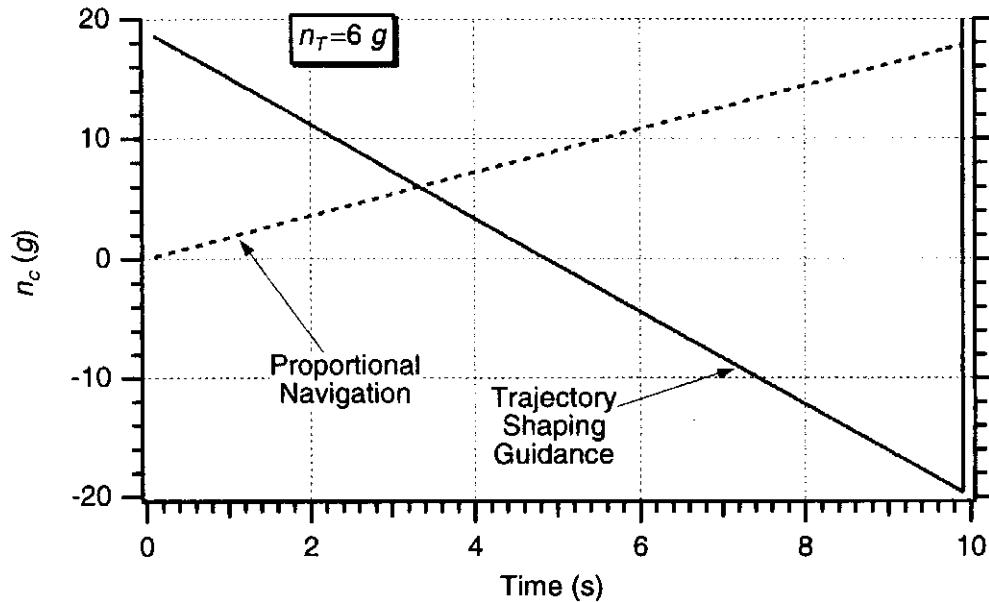


Fig. 25.7 Trajectory shaping guidance law requires more acceleration than proportional navigation against maneuvering target.

Again, we can see from Fig. 25.7 that trajectory shaping guidance requires more acceleration than proportional navigation to hit the maneuvering target. For the case of the 6 g maneuvering target and desired final line of sight angle of -30 deg , trajectory shaping guidance required 20 g at the beginning of flight and -20 g at the end of flight. On the other hand, proportional navigation, which did not reach the final line of sight angle goal, required 0 g at the beginning of flight and nearly 20 g at the end of flight.

From Fig. 25.8 we can see that with trajectory shaping guidance we achieved the goal of the line of sight angle becoming -30 deg at the end of the flight. We can also see that for this example proportional navigation ended up with a final line of sight angle of 14 deg . Thus, we can conclude that simulation results indicate

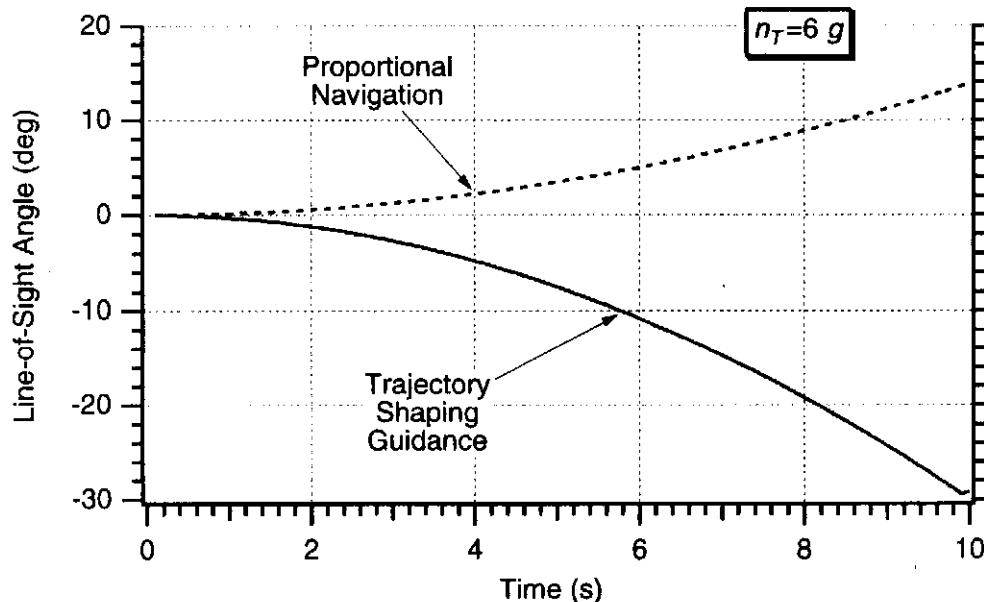


Fig. 25.8 Trajectory shaping guidance law can still control final line of sight angle—even in presence of maneuvering target.

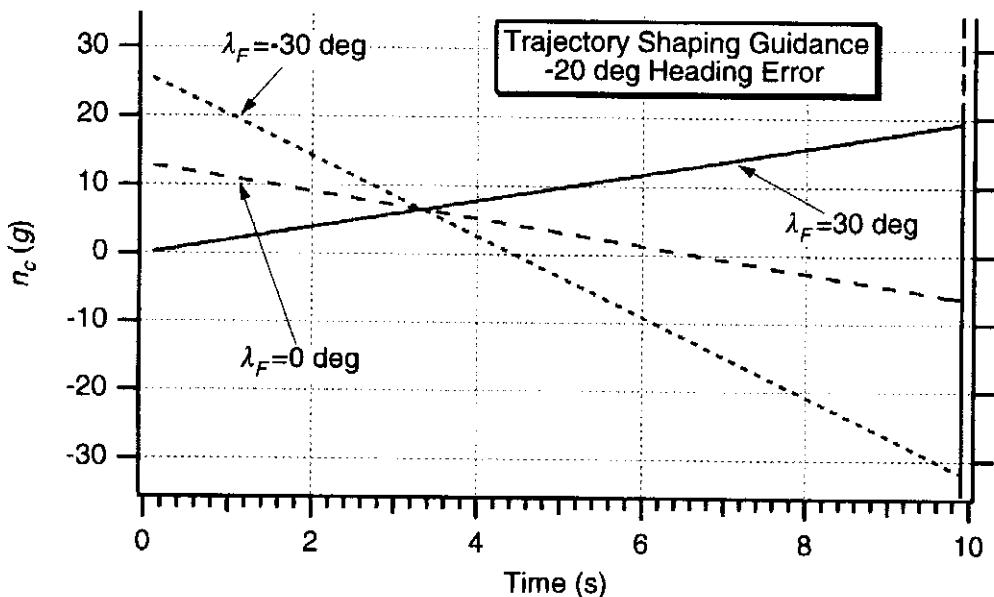


Fig. 25.9 Acceleration requirements depend on final line of sight angle specification when disturbance is heading error.

that trajectory shaping guidance appears to be working correctly against the target maneuver disturbance.

Other cases were run with the trajectory shaping guidance law for the case in which there was -20 deg of heading error and the final line of sight angle is made a parameter. Figure 25.9 shows that the acceleration requirements for the trajectory shaping guidance law are dependent on the final line of sight angle. Figure 25.10 shows that the various design goals for the final line of sight angle are all met with trajectory shaping guidance in the presence of the heading error disturbance, provided adequate acceleration is available.

Finally, even more cases were run with the trajectory shaping guidance law for the situation in which there was a 6 g maneuvering target and the final

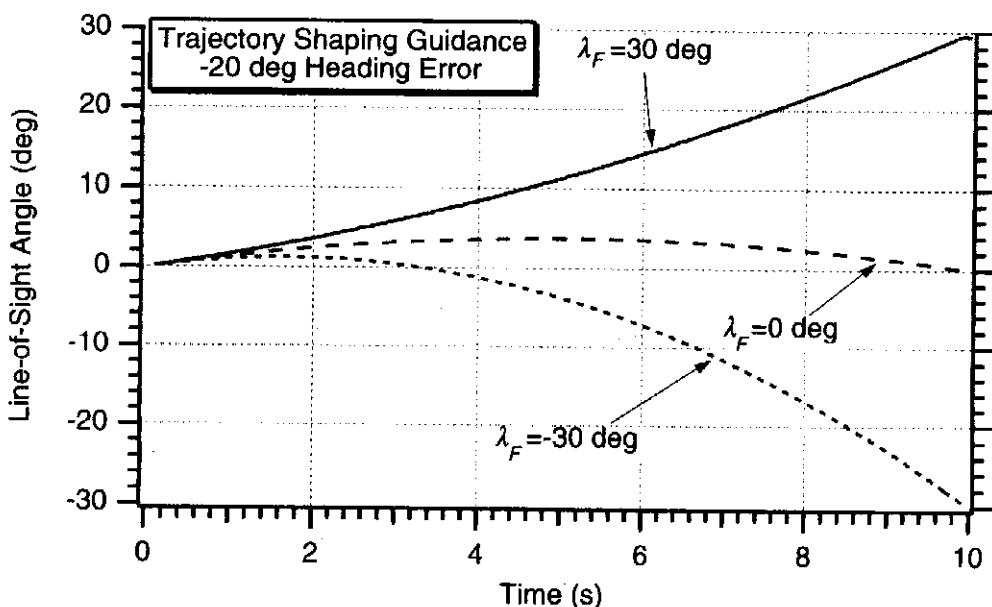


Fig. 25.10 Final line of sight angle goals are met in presence of heading error with trajectory shaping guidance.

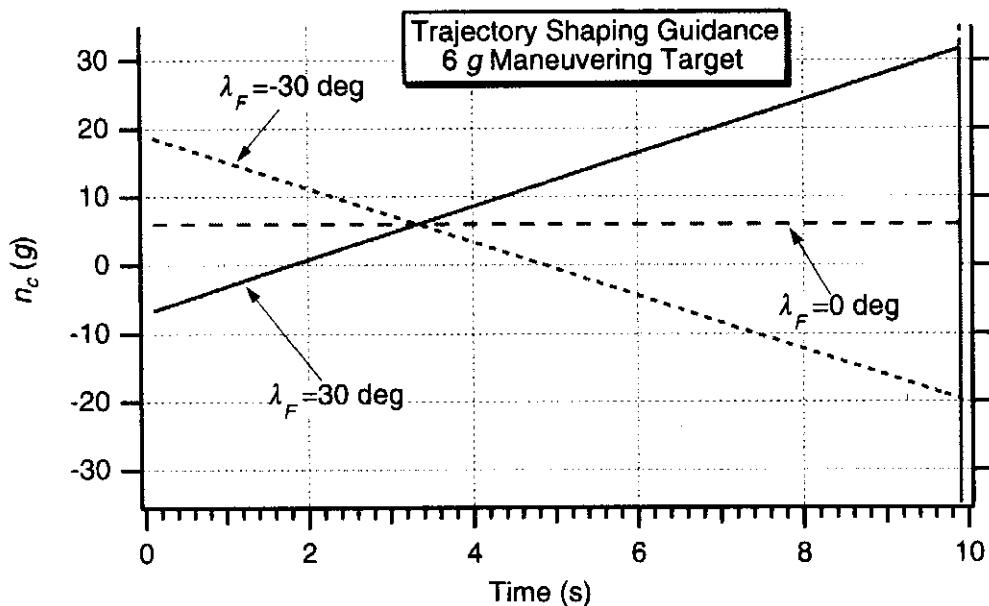


Fig. 25.11 Acceleration requirements still depend on final line of sight angle specification when disturbance is target maneuver.

line of sight angle was made a parameter. Figure 25.11 shows that the acceleration requirements for the trajectory shaping guidance law are again dependent on the final line of sight angle. Figure 25.12 shows that the various design goals for the final line of sight angle are all met with trajectory shaping guidance in the presence of the target maneuver disturbance.

Closed-Form Solutions

With the proportional and augmented proportional navigation guidance laws, we were able to derive closed-form solutions for the missile acceleration due to a step in target maneuver, heading error, and a step in target displacement for a zero-time constant missile guidance system. The solutions were obtained by solving a

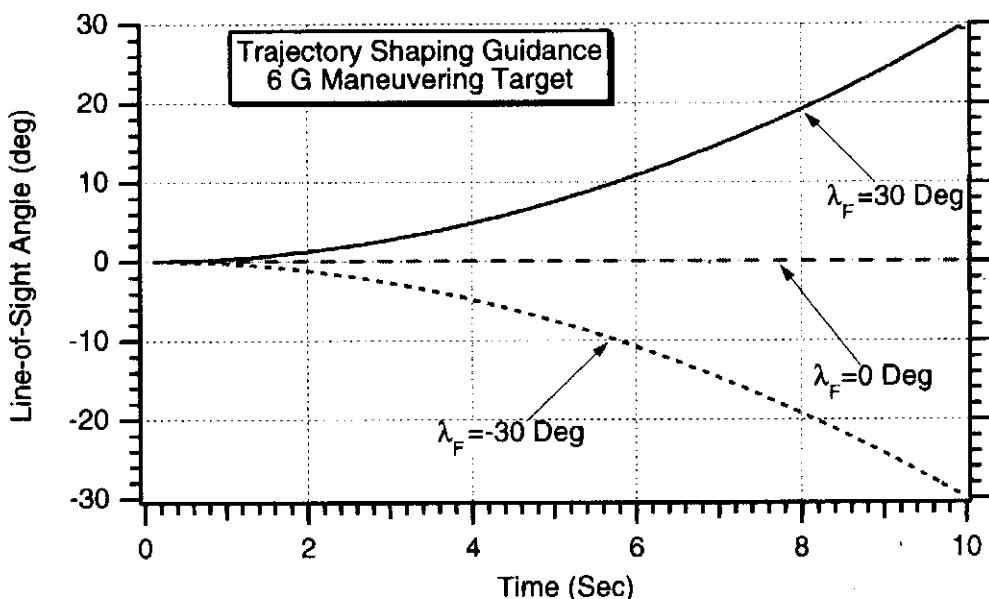


Fig. 25.12 Final line of sight angle goals are met in presence of target maneuver with trajectory shaping guidance.

linear, first-order, time-varying differential equation as was demonstrated in Chapters 2 and 20. Let us see if we can use the same techniques to derive acceleration formulas for the trajectory shaping guidance law.

Consider the case in which the only disturbance to the guidance system is target maneuver. From Fig. 25.2 we can see that the relative acceleration is simply target acceleration minus missile acceleration, or

$$\ddot{y} = n_T - n_c$$

Substituting the original expression for the trajectory shaping guidance law into the preceding equation yields

$$\begin{aligned}\ddot{y} &= n_T - n_c(t) = n_T - \left[\frac{6y + 4\dot{y}t_{go} + n_T t_{go}^2 + 2\dot{y}(t_F)t_{go}}{t_{go}^2} \right] \\ &= \frac{-6y - 4\dot{y}t_{go} - 2\dot{y}(t_F)t_{go}}{t_{go}^2}\end{aligned}$$

or more simply

$$\ddot{y} + \frac{4\dot{y}}{t_F - t} + \frac{6y}{(t_F - t)^2} = \frac{-2\dot{y}(t_F)}{t_F - t}$$

with initial conditions

$$y(0) = 0 \quad \text{and} \quad \dot{y}(0) = 0$$

Recall that in Chapters 2 and 20 we obtained closed-form solutions by solving a first-order linear differential equation with time-varying coefficients. However, now we now have a second-order linear differential equation with time-varying coefficients. The solution to such an equation is extremely difficult at best. Let us see if we can take another, less conventional, approach to the problem.

Recall that the general solution to the state space equation at the final time

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u}$$

was given by

$$\mathbf{x}(t_F) = \Phi(t_F - t)\mathbf{x}(t) + \int_t^{t_F} \Phi(t_F - \lambda)\mathbf{G}(\lambda)\mathbf{u}(\lambda) d\lambda$$

We can also say that the general solution to the state space equation can be expressed in terms of the initial conditions at time zero as

$$\mathbf{x}(t_F) = \Phi(t_F)\mathbf{x}(0) + \int_0^{t_F} \Phi(t_F - \lambda)\mathbf{G}(\lambda)\mathbf{u}(\lambda) d\lambda$$

Therefore, for the homing-loop problem under consideration we have

$$\begin{bmatrix} y(t_F) \\ \dot{y}(t_F) \\ n_T(t_F) \end{bmatrix} = \begin{bmatrix} 1 & t_F & .5t_F^2 \\ 0 & 1 & t_F \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y(0) \\ \dot{y}(0) \\ n_T(0) \end{bmatrix} + \int_0^{t_F} \begin{bmatrix} 1 & t_F - \lambda & .5(t_F - \lambda)^2 \\ 0 & 1 & t_F - \lambda \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} n_c(\lambda) d\lambda$$

Multiplying out the preceding matrix equation and leaving out the third scalar equation yields

$$y(t_F) = y(0) + t_F \dot{y}(0) + .5t_F^2 n_T(0) - \int_0^{t_F} (t_F - \lambda) n_c(\lambda) d\lambda$$

$$\dot{y}(t_F) = \dot{y}(0) + t_F n_T(0) - \int_0^{t_F} n_c(\lambda) d\lambda$$

We still want to minimize the integral of the square of the commanded acceleration subject to the miss being zero and the relative velocity at the end of flight being specified or

$$y(t_F) = 0 \quad \text{and} \quad \dot{y}(t_F) = \dot{y}_F \quad \text{subject to minimizing } \int_0^{t_F} n_c^2(t) dt$$

We get the same solution for the acceleration command as before or

$$n_c(t) = \frac{f_1 h_1(t) \|h_2^2\| - \|h_1 h_2\| [f_2 h_1(t) + f_1 h_2(t)] + f_2 h_2(t) \|h_1^2\|}{\|h_1^2\| \|h_2^2\| - \|h_1 h_2\|^2}$$

except that this time the shorthand notation definitions have changed to

$$f_1 = y(0) + t_F \dot{y}(0) + .5t_F^2 n_T(0)$$

$$f_2 = \dot{y}(0) + t_F n_T(0) - \dot{y}(t_F)$$

$$h_1(t) = t_F - t = t_{go}$$

$$h_2(t) = 1$$

$$\|h_1^2\| = \int_0^{t_F} h_1^2(t) dt = \int_0^{t_F} (t_F - t)^2 d\lambda = \frac{t_F^3}{3}$$

$$\|h_2^2\| = \int_0^{t_F} h_2^2(t) dt = \int_0^{t_F} dt = t_F$$

$$\|h_1 h_2\| = \int_0^{t_F} h_1(t) h_2(t) dt = \int_0^{t_F} (t_F - t) dt = \frac{t_F^2}{2}$$

Substitution of the preceding definitions into the formula for the acceleration command yields

$$n_c(t) = \left\{ [y(0) + t_F \dot{y}(0) + .5t_F^2 n_T(0)] t_{go} t_F - .5t_F^2 [(\dot{y}(0) + t_F n_T(0) - \dot{y}(t_F)) t_{go} + [y(0) + t_F \dot{y}(0) + .5t_F^2 n_T(0)](1)] + [(\dot{y}(0) + t_F n_T(0) - \dot{y}(t_F))(1) \frac{t_F^3}{3}] \right\} \Bigg/ \left[\frac{t_F^3}{3} t_F - \left[\frac{t_F^2}{2} \right]^2 \right]$$

After much algebra, we obtain

$$n_c(t) = \frac{12y(0)\left(t_{go} - \frac{t_F}{2}\right) + t_F \dot{y}(0)(6t_{go} - 2t_F) + t_F^3 n_T(0) + t_F \dot{y}(t_F)(6t_{go} - 4t_F)}{t_F^3}$$

The preceding expression is the closed-form solution for the total missile acceleration due to the various initial conditions or error sources when using the trajectory shaping guidance law. Therefore, the missile acceleration due to an initial condition in relative velocity can be written by inspection from the preceding formula as

$$n_c(t)|_{\dot{y}_0} = \frac{t_F \dot{y}(0)(6t_{go} - 2t_F)}{t_F^3} = \frac{2\dot{y}(0)}{t_F} \left[2 - \frac{3t}{t_F} \right]$$

Because the initial relative velocity and heading error are related by

$$\dot{y}(0) = -V_M \text{HE}$$

we can say that the acceleration due to heading error is given by

$$n_c(t)|_{\text{HE}} = \frac{-2V_M \text{HE}}{t_F} \left[2 - \frac{3t}{t_F} \right]$$

Therefore, the acceleration required to take out the heading error is proportional to the amount of heading error and inversely proportional to the amount of homing time. More heading error and less homing time both work in the direction of increasing the missile acceleration requirements.

From the general closed-form acceleration formula, we can see that the acceleration due to a target maneuver is given by

$$n_c(t)|_{n_T} = \frac{t_F^3 n_T(0)}{t_F^3} = n_T(0)$$

We can see from the preceding expression that in this case the missile is simply matching the target acceleration. Therefore, as expected, larger target maneuvers will require more acceleration capability from the missile.

Finally, we can see that the acceleration due to shaping the trajectory to match a desired final relative velocity can also be written by inspection of the total acceleration formula as

$$n_c(t)|_{\dot{y}_F} = \frac{t_F \dot{y}(t_F)(6t_{go} - 4t_F)}{t_F^3} = \frac{2\dot{y}(t_F)}{t_F} \left[1 - \frac{3t}{t_F} \right]$$

Recall that the final relative velocity can also be expressed in terms of the final line of sight angle as

$$\dot{y}(t_F) = -V_c \lambda(t_F)$$

Therefore, the acceleration due to shaping the final line of sight angle can be rewritten as

$$n_c(t)|_{\lambda_F} = \frac{-2V_c \lambda(t_F)}{t_F} \left[1 - \frac{3t}{t_F} \right]$$

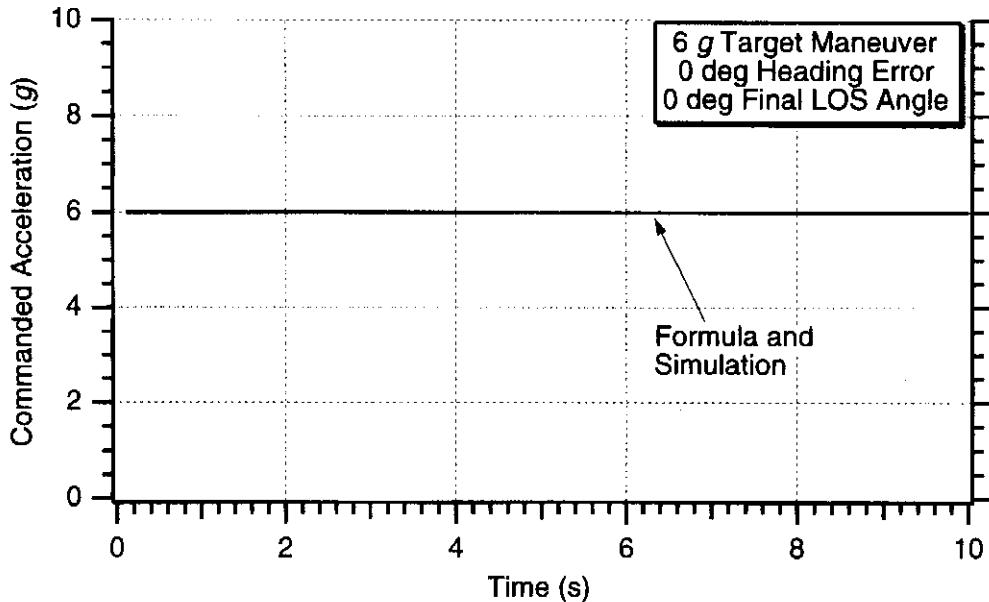


Fig. 25.13 Formula for acceleration because target maneuver is accurate.

The acceleration requirements are proportional to the amount of shaping we want to do and inversely proportional to the amount of homing time. Larger desired final line of sight angles (that is, more shaping) will require more missile acceleration.

To check the formulas derived in this section, cases were run with the linear engagement simulation of Listing 25.1. First, a case was run with trajectory shaping guidance in which the desired final line of sight angle was zero and there was a 6 g target maneuver for a 10-s flight. Recall that the formula for the commanded missile acceleration due to a maneuvering target is given by

$$n_c(t)|_{n_T} = n_T(0)$$

We can see from Fig. 25.14 that the simulation results of Listing 25.1 (namely, XNT=193.2, HEDEG=0, PN=0, XLAMFDEG=0) and the preceding formula are in exact agreement, thus demonstrating that the acceleration due to target maneuver formula is correct.

Listing 25.1 was again run with trajectory shaping guidance in which the desired final line of sight was zero and there was a -20 deg heading error for a 10-s flight. Recall that the formula for the commanded missile acceleration due to heading error is given by

$$n_c(t)|_{HE} = \frac{-2V_M HE}{t_F} \left[2 - \frac{3t}{t_F} \right]$$

We can see from Fig. 25.14 that the simulation results of Listing 25.1 (namely, XNT=0, HEDEG=-20, PN=0, XLAMFDEG=0) and the preceding formula are in exact agreement, thus demonstrating that the acceleration due to heading error formula is also correct.

Finally, Listing 25.1 was run again with trajectory shaping guidance in which the desired final line of sight angle is set to -30 deg. In this case there is no heading

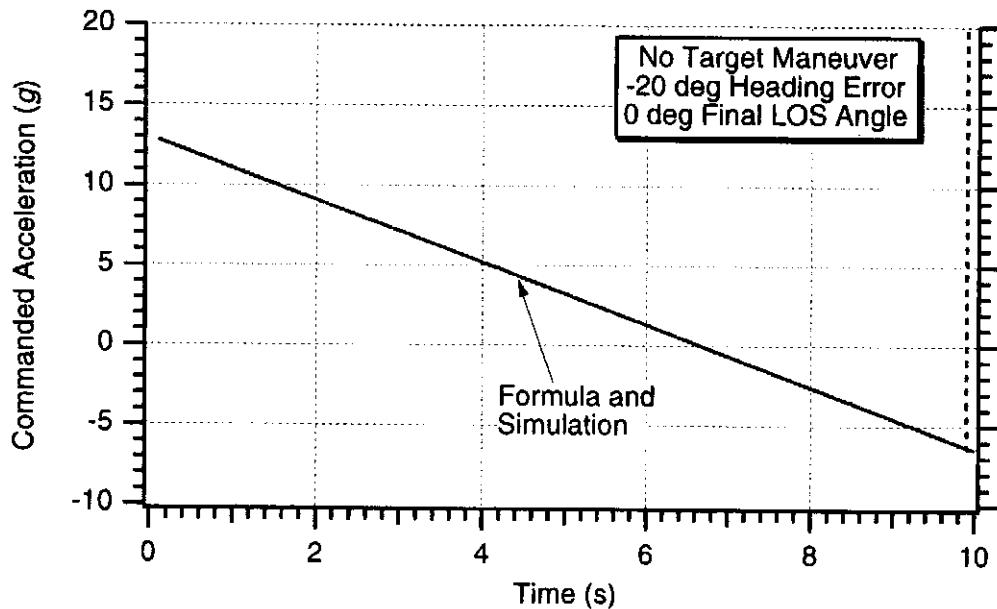


Fig. 25.14 Formula for acceleration heading error is accurate.

error or target maneuver. However, the flight time is still 10-s. Recall that the formula for the commanded missile acceleration due to specifying the final line of sight angle is given by

$$n_c(t)|_{\lambda_F} = \frac{-2V_c \lambda(t_F)}{t_F} \left[1 - \frac{3t}{t_F} \right]$$

We can see from Fig. 25.15 that again the simulation results of Listing 25.1 (namely, XNT=0, HEDEG=-0, PN=0, XLAMFDEG=-30) and the preceding formula are in exact agreement, thus demonstrating that the acceleration due to specifying the final line of sight angle formula is also correct.

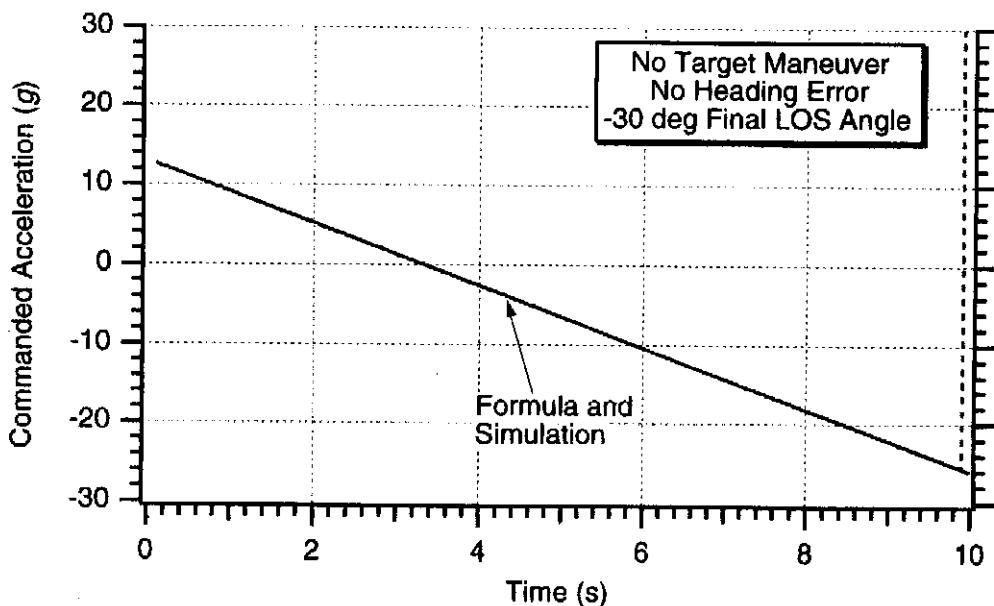


Fig. 25.15 Formula for acceleration because of controlling final line of sight angle is accurate.

Nonlinear Results

The trajectory shaping guidance law was really derived for operation in the world in which the geometry was linear. This implies small angle approximations. It is now of interest to see how the new guidance law works in the two-dimensional world in which the equations of motion are nonlinear. Listing 25.2 is a slight modification of the original zero-time constant nonlinear missile-target engagement simulation of Listing 2.1. An option has been included so that the target can either be stationary (target flight path rate has been modified so there is no division by zero) or moving. It is important to note that the trajectory shaping guidance law requires knowledge of the target acceleration. In the nonlinear engagement simulation, the target acceleration perpendicular to the line of sight is used in the guidance law. The components of the target acceleration in the downrange and altitude direction can be expressed in terms of the target flight path angle as

$$n_{T1} = n_T \sin \beta$$

$$n_{T2} = n_T \cos \beta$$

Therefore, the target acceleration that appears perpendicular to the line of sight can be obtained from trigonometry and can be expressed as

$$n_{T_{\text{LOS}}} = -n_{T1} \sin \lambda + n_{T2} \cos \lambda$$

Now the trajectory shaping guidance law for the nonlinear world can be written as

$$n_c(t) = 4V_c \dot{\lambda} + \frac{2V_c[\lambda - \lambda_F]}{t_{\text{go}}} + n_{T_{\text{LOS}}}$$

where the direction of the commanded acceleration is perpendicular to the line of sight. The new nonlinear engagement simulation appears in Listing 25.2. We can see that the simulation can also be run using proportional navigation by simply setting APN=0.

The nominal case of Listing 25.2 was run for the example in which the target is considered to be stationary (namely, VT=0) and is located 30 kft downrange from the missile (namely, RT1IC=30000). The missile is traveling at 3000 ft/s and is initially at 10 kft altitude (namely, VM=3000, RM1IC=10000). The geometry is such that the missile is on a collision path with the target (that is, zero heading error). We can see from Fig. 25.16 that when proportional navigation is used the missile essentially travels in a straight line to the target because it is already on a collision triangle with the target. However, in this application we would like to hit the target vertically for lethality reasons (such as an antitank application). This means that for the trajectory shaping guidance law we would like the final line of sight angle to be -90 deg (namely, XLAMFDEG=-90.). We can see from Fig. 25.16 that it indeed appears that the trajectory shaping guidance law is enabling the missile to hit the target near vertically. In addition, we can see that the trajectory shaping guidance law trajectory is entirely different than the proportional navigation guidance trajectory. Of course, trajectory shaping guidance requires time to go information, whereas proportional navigation does not.

Figure 25.17 shows that the price paid for shaping the trajectory is that considerable acceleration is required by the missile to hit the target. Unlike proportional navigation, which does not require any acceleration to hit the target in this scenario because it is already on a collision triangle, trajectory shaping guidance requires

Listing 25.2 Nonlinear engagement simulation to test trajectory shaping guidance law

```

IMPLICIT REAL*8(A-H,O-Z)
INTEGER STEP,APN
XNTG=0.
HEDEG=0.
XNP=3.
RM1IC=0.
RM2IC=10000.
RT1IC=30000.
RT2IC=0.
VM=3000.
VT=0.
XNCLIMG=9999999.
APN=1
XLAMFDEG=-90.
H=.0001
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
XNCLIM=32.2*XNCLIMG
XLAMF=XLAMFDEG/57.3
XNT=32.2*XNTG
RM1=RM1IC
RM2=RM2IC
RT1=RT1IC
RT2=RT2IC
BETA=0.
VT1=-VT*COS(BETA)
VT2=VT*SIN(BETA)
HE=HEDEG/57.3
T=0.
S=0.
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
XLAM=ATAN2(RTM2,RTM1)
XLEAD=ASIN(VT*SIN(BETA+XLAM)/VM)
THET=XLAM+XLEAD
VM1=VM*COS(THET+HE)
VM2=VM*SIN(THET+HE)
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
10 IF(VC<0.)GOTO 999
IF(RTM<1000.)THEN
    H=.00001
ELSE
    H=.0001
ENDIF
BETAOLD=BETA
RT1OLD=RT1
RT2OLD=RT2

```

(Contd.)

Listing 25.2 (Continued)

```

RM1OLD=RM1
RM2OLD=RM2
VM1OLD=VM1
VM2OLD=VM2
STEP=1
GOTO 200
66 STEP=2
BETA=BETA+H*BETAD
RT1=RT1+H*VT1
RT2=RT2+H*VT2
RM1=RM1+H*VM1
RM2=RM2+H*VM2
VM1=VM1+H*AM1
VM2=VM2+H*AM2
T=T+H
GOTO 200
55 CONTINUE
BETA=.5*(BETAOLD+BETA+H*BETAD)
RT1=.5*(RT1OLD+RT1+H*VT1)
RT2=.5*(RT2OLD+RT2+H*VT2)
RM1=.5*(RM1OLD+RM1+H*VM1)
RM2=.5*(RM2OLD+RM2+H*VM2)
VM1=.5*(VM1OLD+VM1+H*AM1)
VM2=.5*(VM2OLD+VM2+H*AM2)
S=S+H
IF(S<.09999)GOTO 10
S=0.
RT1K=RT1/1000.
RT2K=RT2/1000.
RM1K=RM1/1000.
RM2K=RM2/1000.
XNCG=XNC/32.2
WRITE(9,*)T, RM1K, RM2K, RT1K, RT2K, XNC/32.2, XLAM*57.3
WRITE(1,*)T, RM1K, RM2K, RT1K, RT2K, XNC/32.2, XLAM*57.3
GOTO 10
200 CONTINUE
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
XLAM=ATAN2(RTM2,RTM1)
XLAMD=(RTM1*VTM2-RTM2*VTM1)/(RTM*RTM)
TGO=RTM/VC
IF(APN.EQ.0)THEN
    XNC=XNP*VC*XLAMD
ELSE
    XNT1=XNT*SIN(BETA)
    XNT2=XNT*COS(BETA)

```

(Contd.)

Listing 25.2 (Continued)

```

XNTPLOS=-XNT1*SIN(XLAM)+XNT2*COS(XLAM)
XNC=4.*VC*XLAMD+XNTPLOS+2.*VC*(XLAM-XLAMF)/TGO
ENDIF
IF(XNC>XNCLIM)XNC=XNCLIM
IF(XNC<-XNCLIM)XNC=-XNCLIM
AM1=-XNC*SIN(XLAM)
AM2=XNC*COS(XLAM)
VT1=-VT*COS(BETA)
VT2=VT*SIN(BETA)
IF(VT.EQ.0.)THEN
    BETAD=0.
ELSE
    BETAD=XNT/VT
ENDIF
IF(STEP>1)66,66,55
999  CONTINUE
WRITE(9,*)T,RTM
CLOSE(1)
PAUSE
END

```

more than 20 g of acceleration at the beginning of the flight and nearly -10 g at the end of the flight.

Finally, we can see from Fig. 25.18 that the line of sight angle for trajectory shaping guidance achieved the design goal by reaching -90 deg at the end of the flight. Figure 25.18 also shows that proportional navigation, which does not shape the trajectory, ended up with a final line of sight angle of -18.4 deg. We can also see from Fig. 25.18 that because proportional navigation enabled the missile to fly directly to the target, the flight time was nearly 4 s shorter than when the trajectory shaping guidance law was used.

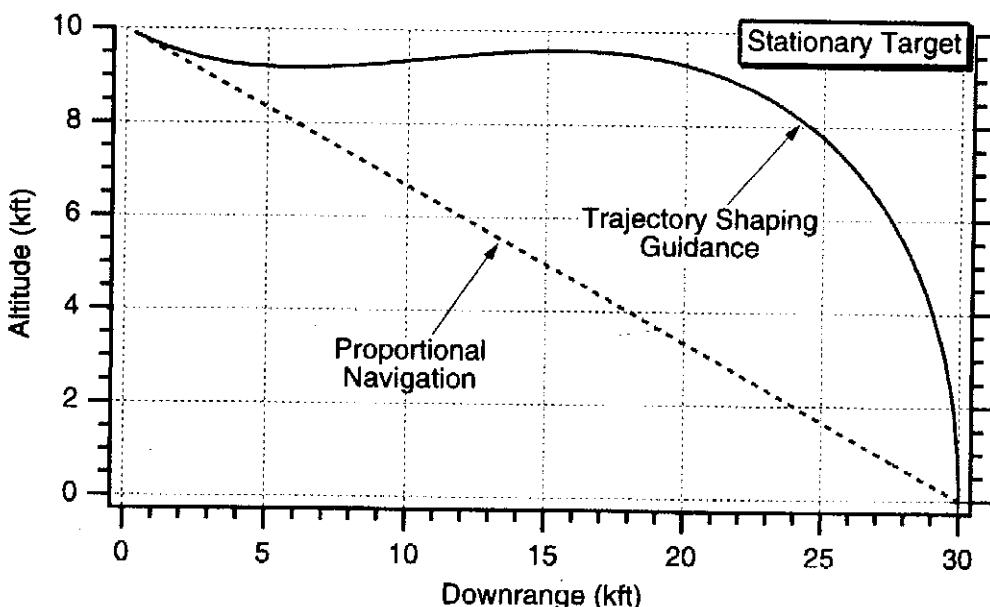


Fig. 25.16 Trajectory shaping also works in the nonlinear world.

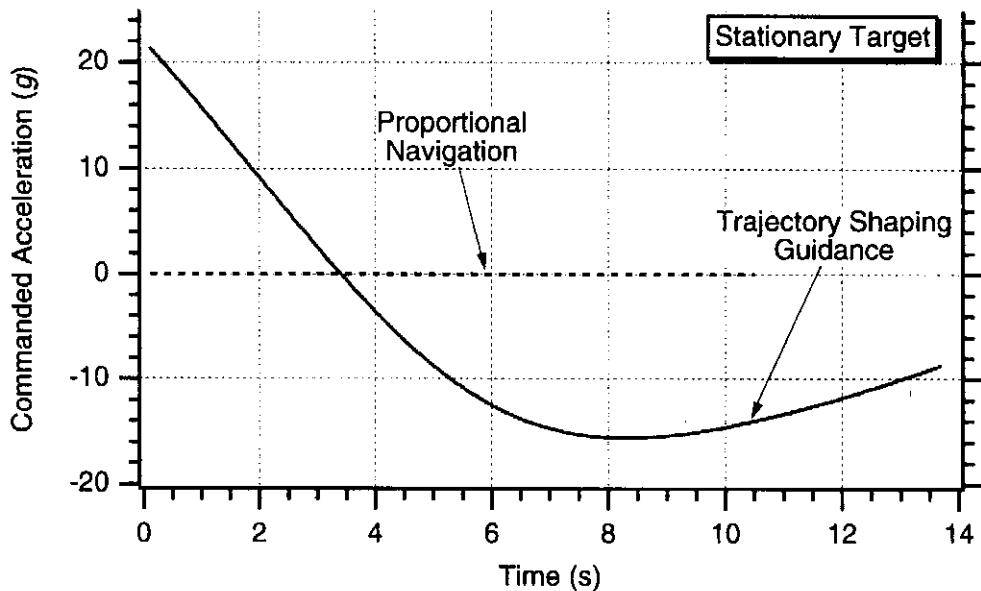


Fig. 25.17 A great deal of acceleration may be required to shape trajectory to get a final line of sight angle of -90 deg.

It is of considerable interest to see if the formulas we derived in the previous section for the commanded acceleration are useful in predicting the nonlinear results. Recall that in the previous section we showed that the acceleration in units of ft/s^2 required to turn the missile through an angle of λ_F in units of radians is given by

$$n_c(t)|_{\lambda_F} = \frac{-2V_c\lambda_F}{t_F} \left[1 - \frac{3t}{t_F} \right]$$

where V_c is the closing velocity in ft/s , t_F is the amount of flight or guidance time, and t is instantaneous time both in units of seconds. For the problem depicted in Fig. 25.16, the final line of sight angle that would have been achieved without trajectory shaping guidance is -18.4 deg (see Fig. 25.18). Therefore, trajectory

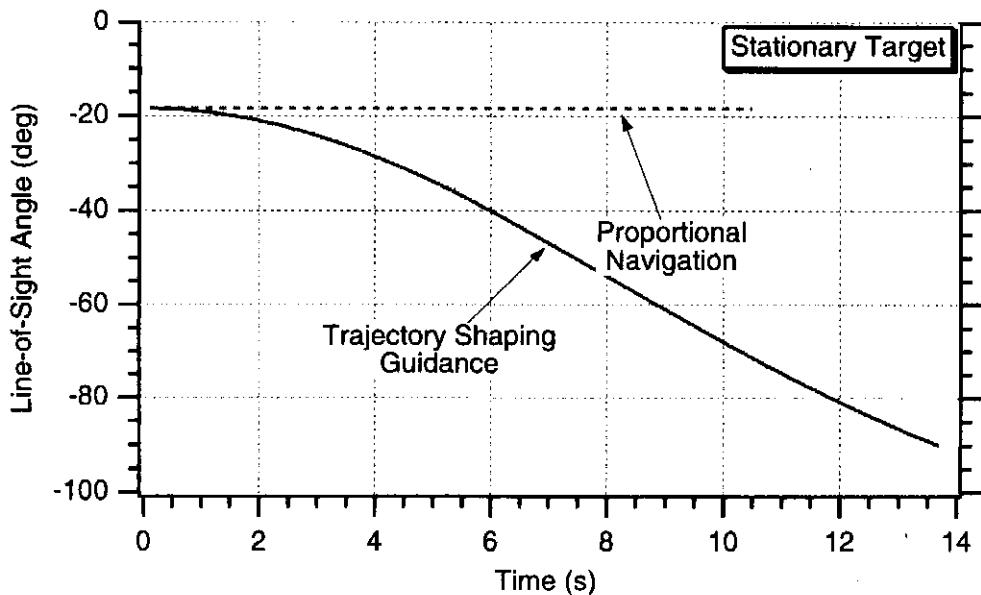


Fig. 25.18 Trajectory shaping enables line of sight angle to reach its goal.

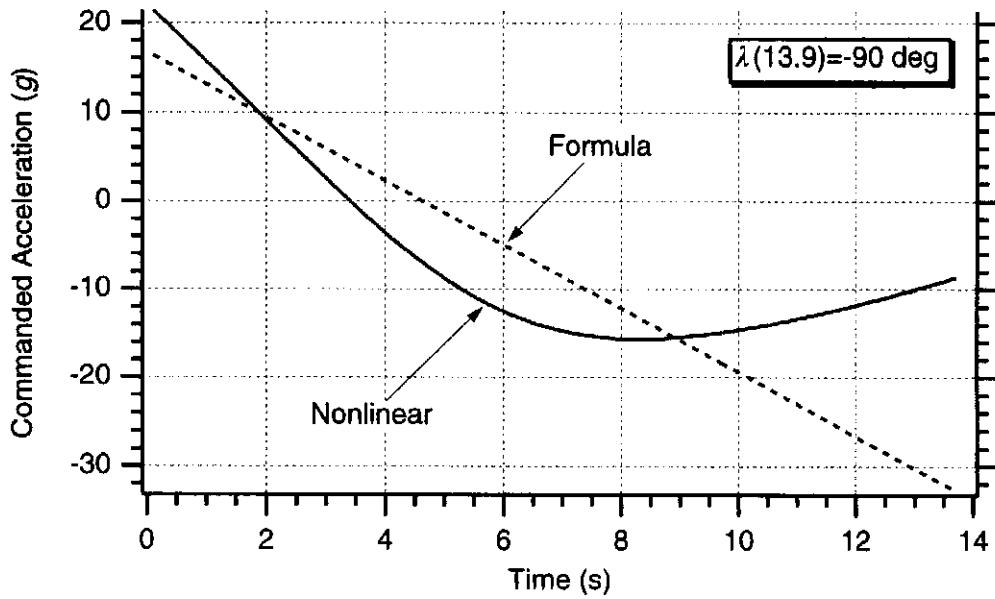


Fig. 25.19 Linear formula is not a great match to nonlinear results.

shaping guidance is attempting to change the angle from -18.4 deg to -90 deg, or a change of -71.6 deg. In addition, we can tell from Fig. 25.18 that the amount of guidance time for trajectory shaping guidance is 13.9 s.

Figure 25.19 indicates that the linear formula for missile acceleration is not very accurate in this example for predicting the nonlinear commanded missile acceleration. It is hypothesized that perhaps the angular change in the line of sight angle is too great for linear theory to hold. Another case was run with the nonlinear simulation in which the desired final line of sight angle was -30 deg (a change of only 11.6 deg from -18.4 deg that could be obtained with proportional navigation). Because there is less trajectory shaping, the flight time reduces to 10.6 s. We can see from Fig. 25.20 that the formula now matches the nonlinear results quite accurately.

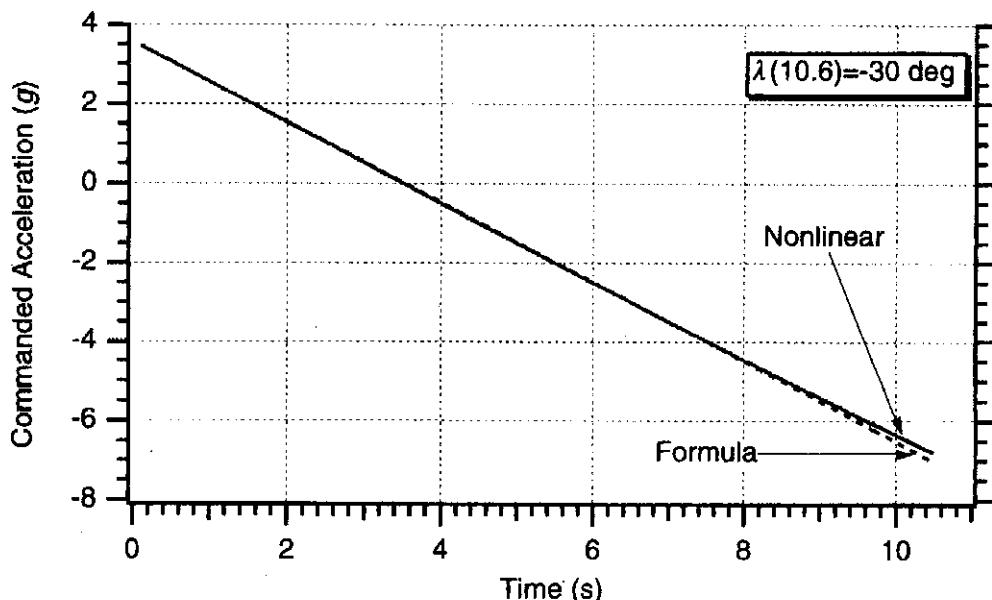


Fig. 25.20 Linear formula is nearly perfect when angular turn is smaller.

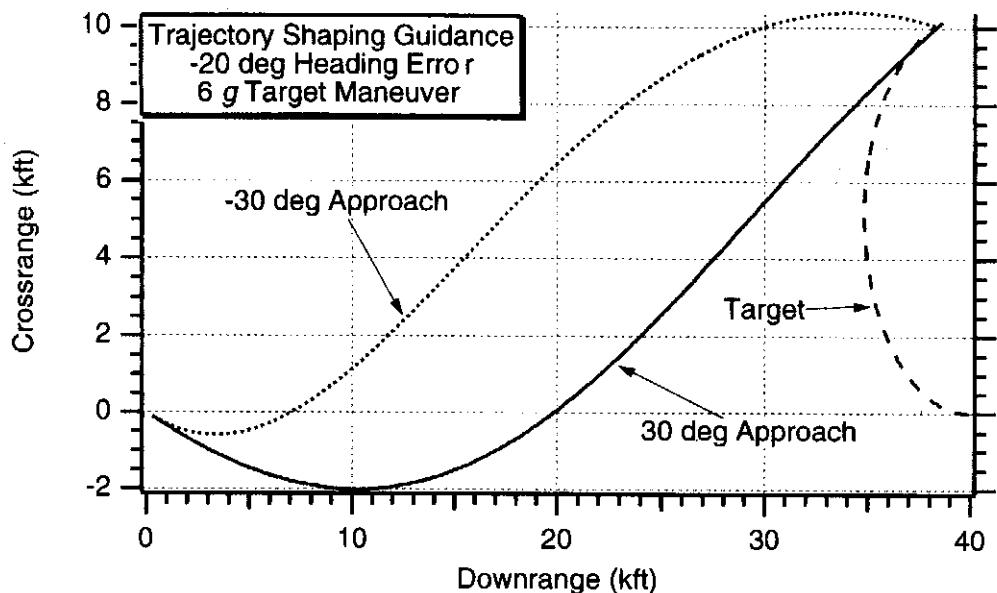


Fig. 25.21 Trajectory shaping guidance works against maneuvering target for different approach angles.

Another, more stressing case was considered in which the target was both moving and maneuvering. In this example, the missile had a -20 deg heading error, while the target was executing a 6 g maneuver. Two cases were considered—one in which the desired final line of sight angle was -30 deg and the other in which the desired final line of sight angle was 30 deg. Figure 25.21 shows that when trajectory shaping guidance was used intercepts were achieved in both cases. It is too difficult to tell from Fig. 25.21 if the final line of sight angle design goals have actually been met. Figure 25.22 displays the commanded acceleration profiles that were required in both cases for successful intercepts. We can see that for the

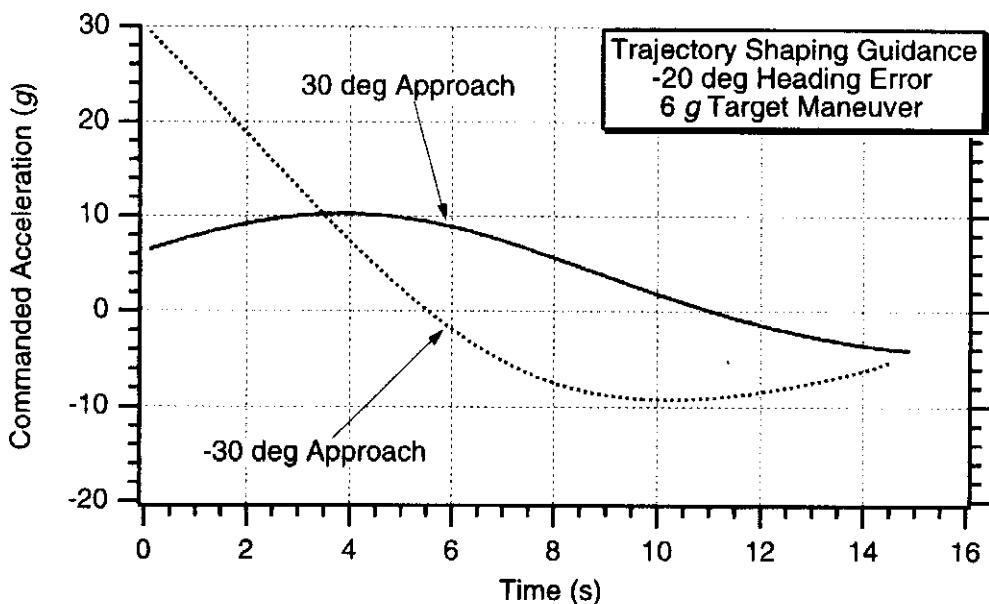


Fig. 25.22 Acceleration requirements are larger with maneuvering target than with stationary target.

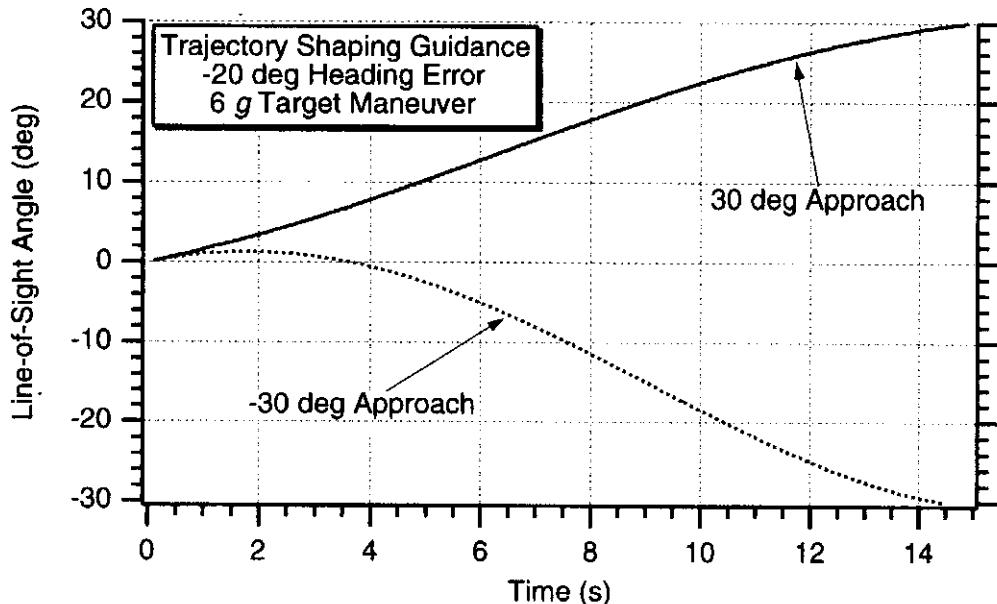


Fig. 25.23 Trajectory shaping guidance meets design goals against maneuvering target.

–30 deg intercept the maximum positive acceleration was 30 g while the maximum negative acceleration was –10 g. We can also see that for the 30-deg intercept the maximum positive acceleration was only 10 g while the maximum negative acceleration was –5 g. Finally, we can see from Fig. 25.23 that the design goals for the final line of sight angles for both cases were met.

Again, it is of considerable interest to see if the formulas we derived in the previous section for the commanded acceleration are useful in predicting the nonlinear results. Recall that the total acceleration in units of ft/s^2 required to turn the missile through an angle of λ_F in units of radians in the presence of target maneuver n_T in units of ft/s^2 and heading error HE in units of radians is given by

$$n_c(t)|_{\text{Total}} = \frac{-2V_c\lambda_F}{t_F} \left[1 - \frac{3t}{t_F} \right] + n_T + \frac{-2V_M\text{HE}}{t_F} \left[2 - \frac{3t}{t_F} \right]$$

For the case of interest the closing velocity is approximately 4000 ft/s while the flight time turns out to be 14.5 s. We can see from Fig. 25.24 that the match between the formula and simulation results is not very accurate.

It is hypothesized that the reason for the inaccurate comparison of Fig. 25.24 is because of the highly maneuvering target. As the target maneuvers, the portion of the maneuver perpendicular to the line of sight diminishes, making the formula less accurate. To test the hypothesis another case was run in which the maneuver level decreased to 3 g. The heading error remained at –20 deg, and the desired final line of sight angle remained at –30 deg. For this case, the time of flight reduced to 11.4 s. We can see from Fig. 25.25 that now the formula is an excellent approximation to the nonlinear results.

Thus, we can see that the trajectory shaping guidance law also works in the nonlinear world. We have demonstrated that under many circumstances we also have formulas that can be used to predict or explain the resultant acceleration requirements on the missile when trajectory shaping guidance is used.

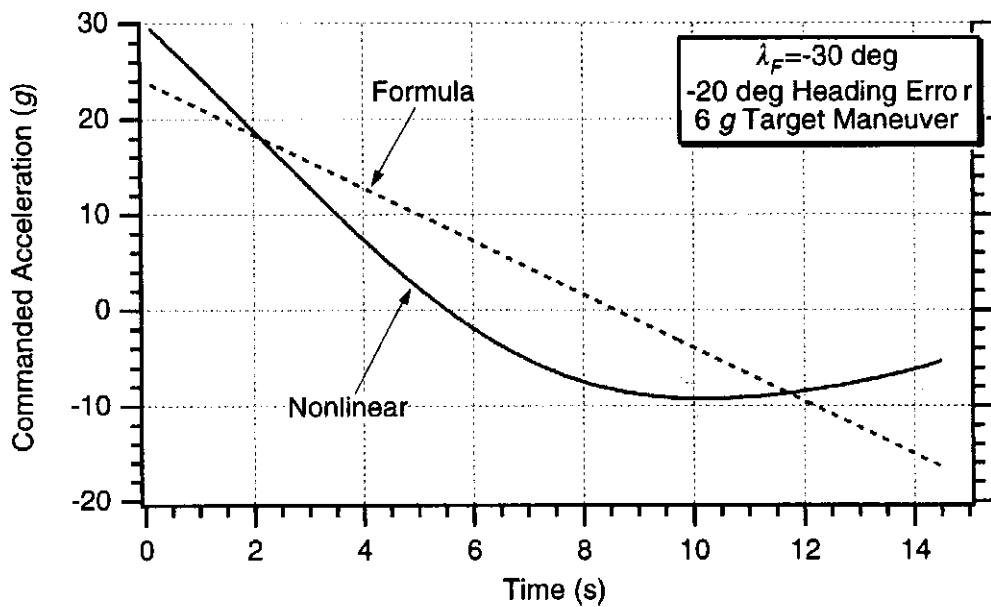


Fig. 25.24 Linear formula not very accurate for highly maneuvering target.

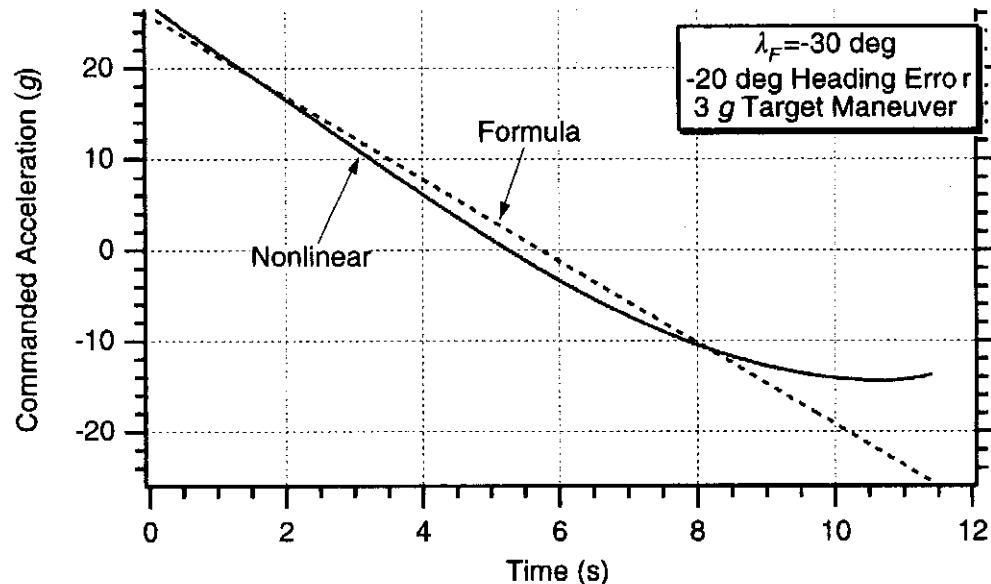


Fig. 25.25 Linear formula is much more accurate when target maneuver level is lower.

Summary

In this chapter the trajectory shaping guidance law has been derived. It was demonstrated that with this new guidance law we could not only hit the target but could also control the final line of sight angle. The price paid for the trajectory shaping was that more acceleration was required to hit the target. Formulas were also derived that could be used to predict the missile acceleration requirements for the new guidance law under a variety of circumstances. It was demonstrated that the formulas were also an accurate indicator of performance in the nonlinear world.

References

- ¹Battin, R., *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA, New York, 1987, p. 561.
- ²Lin, C. F., *Modern Navigation, Guidance, and Control Processing*, Prentice Hall, New Jersey, 1991, p. 605.
- ³Ben-Asher, J., and Yaesh, I., *Advances in Missile Guidance Theory*, AIAA, Reston, VA, 1998, pp. 25–88.
- ⁴Bryson, A. E., and Ho, Y. C., *Applied Optimal Control*, Chapter 5, Hemisphere, New York, 1975.



Filtering and Weaving Targets

Introduction

SO far we have shown that we could considerably improve our performance against weaving targets if we could either use a special purpose guidance law or somehow achieve a smaller guidance system time constant. If we choose to use the guidance law that is optimal against weaving targets, we then have to estimate the target acceleration, target jerk, and target weave frequency.

In this chapter we will explore the various filtering options that can be used against a weaving target by using a step-by-step approach. First we shall see how our original linear three-state Kalman filter from Chapter 9 is able to function in the presence of a weaving target. Although with this filter we cannot use the weave guidance law that was derived in Chapter 21, we can use either proportional navigation, augmented proportional navigation, or optimal guidance. Next we will assume that the target weave frequency is known (that is, estimated or derived using other sensors or phenomenology) and proceed to derive an optimal linear four-state weave Kalman filter that estimates both target acceleration and jerk. This filter can be used with either the weave guidance law or compensated weave guidance law that were both derived in Chapter 21. Finally, we will assume that the target weave frequency is not known in advance but must also be estimated. In this case an extended five-state Kalman filter that estimates the relative position, relative velocity, target acceleration, jerk, and weave frequency will be derived. All three Kalman filters and appropriate guidance laws will be compared in terms of both performance and robustness.

Review of Original Three-State Linear Kalman Filter

The original three-state linear Kalman filter from Chapter 9 was derived based on the homing loop model of Fig. 26.1. Recall that in this guidance system model we measured noisy relative position y^* and were attempting to estimate relative position, relative velocity, and target acceleration. As was the case in Chapter 9, the achieved missile acceleration n_L was assumed to be known, and the target acceleration was considered to be modeled as white noise through an integrator. It is important to note that we have already shown in Chapter 4 that the shaping filter equivalent of a target maneuver with constant amplitude and random starting time is mathematically equivalent (that is, in terms of second-order statistics) to white noise through an integrator.

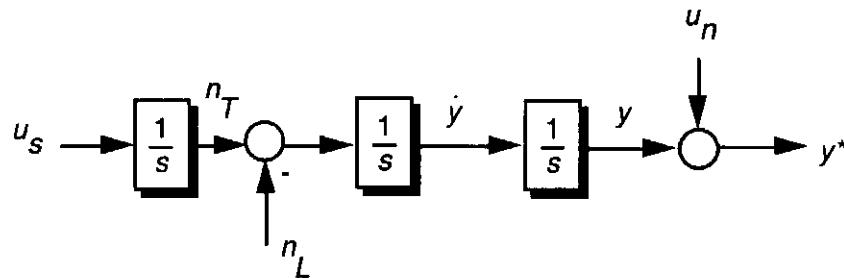


Fig. 26.1 Homing loop model for Kalman filter development.

According to the results of Chapter 4, the spectral density of the white noise source u_s , depicted in Fig. 26.1 was shown to be

$$\Phi_s = \frac{n_{T_{\text{MAX}}^2}}{t_F}$$

where $n_{T_{\text{MAX}}}$ is the assumed maximum target maneuver level magnitude and t_F is the flight time. The model of Fig. 26.1 can be expressed in state space form as

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{n}_T \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ n_T \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} n_L + \begin{bmatrix} 0 \\ 0 \\ u_s \end{bmatrix}$$

Because the systems dynamics matrix of the preceding equation is given by

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

the continuous fundamental matrix can easily be derived (as was the case in Chapter 9) as

$$\Phi(t) = \begin{bmatrix} 1 & t & .5t^2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}$$

By replacing t with the sampling time T_s we can obtain the discrete form of the fundamental matrix as

$$\Phi_k = \begin{bmatrix} 1 & T_s & .5T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}$$

The discrete measurement equation can be written by inspection of Fig. 26.1 as

$$y_k^* = [1 \ 0 \ 0] \begin{bmatrix} y_k \\ \dot{y}_k \\ n_{T_k} \end{bmatrix} + v_k$$

Therefore, the discrete measurement matrix can be written by inspection of the preceding equation as

$$\mathbf{H}_k = [1 \ 0 \ 0]$$

The continuous control matrix \mathbf{G} can also be written by inspection of the original state space equation as

$$\mathbf{G} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$$

Therefore the discrete control matrix \mathbf{G}_k becomes

$$\mathbf{G}_k = \int_0^{T_s} \Phi(\tau) \mathbf{G}(\tau) d\tau = \int_0^{T_s} \begin{bmatrix} 1 & \tau & .5\tau^2 \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} d\tau = \begin{bmatrix} -.5T_s^2 \\ -T_s \\ 0 \end{bmatrix}$$

Recall the discrete Kalman filtering equation is given by

$$\hat{\mathbf{x}}_k = \Phi_k \hat{\mathbf{x}}_{k-1} + \mathbf{G}_k \mathbf{u}_{k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \Phi_k \hat{\mathbf{x}}_{k-1} - \mathbf{H} \mathbf{G}_k \mathbf{u}_{k-1})$$

Substitution of the appropriate matrices into the preceding matrix difference equation yields

$$\begin{bmatrix} \hat{y}_k \\ \hat{\dot{y}}_k \\ \hat{n}_{T_k} \end{bmatrix} = \begin{bmatrix} 1 & T_s & .5T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{y}_{k-1} \\ \hat{\dot{y}}_{k-1} \\ \hat{n}_{T_{k-1}} \end{bmatrix} + \begin{bmatrix} -.5T_s^2 \\ -T_s \\ 0 \end{bmatrix} n_{L_{k-1}} + \begin{bmatrix} K_{1k} \\ K_{2k} \\ K_{3k} \end{bmatrix} \\ \left(y_k^* - [1 \ 0 \ 0] \begin{bmatrix} 1 & T_s & .5T_s^2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{y}_{k-1} \\ \hat{\dot{y}}_{k-1} \\ \hat{n}_{T_{k-1}} \end{bmatrix} - [1 \ 0 \ 0] \begin{bmatrix} -.5T_s^2 \\ -T_s \\ 0 \end{bmatrix} n_{L_{k-1}} \right)$$

We can multiply out the terms of the preceding matrix equation to yield the linear three-state Kalman filter scalar equations as

$$\begin{aligned} \text{RES}_k &= y_k^* - \hat{y}_{k-1} - T_s \hat{\dot{y}}_{k-1} - .5T_s^2(n_{T_{k-1}} - n_{L_{k-1}}) \\ \hat{y}_k &= \hat{y}_{k-1} + T_s \hat{\dot{y}}_{k-1} + .5T_s^2(\hat{n}_{T_{k-1}} - n_{L_{k-1}}) + K_{1k} \text{RES}_k \\ \hat{\dot{y}}_k &= \hat{\dot{y}}_{k-1} + T_s(\hat{n}_{T_{k-1}} - n_{L_{k-1}}) + K_{2k} \text{RES}_k \\ \hat{n}_{T_k} &= \hat{n}_{T_{k-1}} + K_{3k} \text{RES}_k \end{aligned}$$

A simulation of the linear three-state linear Kalman filter as part of a missile guidance system is a modified form of Listing 9.2 and appears in Listing 26.1. The simulation now has a single time constant representation of the flight control system plus a weaving target rather than a constant target maneuver. We can see from the listing that there is 3-g weaving target with a weave frequency of 2 rad/s. Nominally there is still 1 mr of measurement noise, but the closing velocity has been increased to 9000 ft/s to reflect a ballistic target engagement. We can see from the listing that the guidance law options for this filter are either proportional navigation, augmented proportional navigation, or optimal guidance (namely, APN=0, 1 or 2, respectively). Because there is a single time constant representation of the flight control system, the achieved missile acceleration rather than the commanded acceleration enters the filtering equations.

Listing 26.1 Original three-state linear Kalman filter and weaving target

```

GLOBAL DEFINE
    INCLUDE 'quickdraw.inc'
END
IMPLICIT REAL*8(A-H,O-Z)
REAL*8 M11,M12,M13,M22,M23,M33,K1,K2,K3
INTEGER STEP,APN
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
TAU=.5
APN=0
VC=9000.
XNT=96.6
XNTREAL=96.6
XNTMAX=96.6
W=2.
YIC=0.
VM=3000.
HEDEG=0.
HEDEGFIL=20.
XNP=3.
SIGRIN=.001
TS=.01
TF=10.
Y=YIC
YD=-VM*HEDEG/57.3
YDIC=YD
TS2=TS*TS
TS3=TS2*TS
TS4=TS3*TS
TS5=TS4*TS
PHIN=XNTMAX*XNTMAX/TF
RTM=VC*TF
SIGNOISE=SIGRIN
SIGPOS=RTM*SIGNOISE
SIGN2=SIGPOS**2
P11=SIGN2
P12=0.
P13=0.
P22=(VM*HEDEGFIL/57.3)**2
P23=0.
P33=XNTMAX*XNTMAX
T=0.
H=.001
S=0.
YH=0.
YDH=0.
XNTH=0.
XNC=0.
XNL=0.
10 IF(T>(TF-.0001))GOTO 999
    YOLD=Y
    YDOLD=YD

```

(Contd.)

Listing 26.1 (Continued)

```

XNLOLD=XNL
STEP=1
GOTO 200
66 STEP=2
Y=Y+H*YD
YD=YD+H*YDD
XNL=XNL+H*XNLD
T=T+H
GOTO 200
55 CONTINUE
Y=.5*(YOLD+Y+H*YD)
YD=.5*(YDOLD+YD+H*YDD)
XNL=.5*(XNLOLD+XNL+H*XNLD)
S=S+H
IF(S<(TS-.0001))GOTO 10
S=0.
TGO=TF-T+.000001
RTM=VC*TGO
SIGNOISE=SIGRIN
SIGPOS=RTM*SIGNOISE
SIGN2=SIGPOS**2
M11=P11+TS*P12+.5*TS2*P13+TS*(P12+TS*P22+.5*TS2*P23)
M11=M11+.5*TS2*(P13+TS*P23+.5*TS2*P33)+TS5*PHIN/20.
M12=P12+TS*P22+.5*TS2*P23+TS*(P13+TS*P23+.5*TS2*P33)+TS4*PHIN/8.
M13=P13+TS*P23+.5*TS2*P33+PHIN*TS3/6.
M22=P22+TS*P23+TS*(P23+TS*P33)+PHIN*TS3/3.
M23=P23+TS*P33+.5*TS2*PHIN
M33=P33+PHIN*TS
K1=M11/(M11+SIGN2)
K2=M12/(M11+SIGN2)
K3=M13/(M11+SIGN2)
P11=(1.-K1)*M11
P12=(1.-K1)*M12
P13=(1.-K1)*M13
P22=-K2*M12+M22
P23=-K2*M13+M23
P33=-K3*M13+M33
CALL GAUSS(XLAMNOISE,SIGNOISE)
YSTAR=RTM*(XLAM+XLAMNOISE)
RES=YSTAR-YH-TS*YDH-.5*TS*TS*(XNTH-XNL)
YH=K1*RES+YH+TS*YDH+.5*TS*TS*(XNTH-XNL)
YDH=K2*RES+YDH+TS*(XNTH-XNL)
XNTH=K3*RES+XNTH
XLAMDH=(YH+YDH*TGO)/(VC*TGO*TGO)
IF(APN.EQ.0)THEN
    XNC=XNP*(YH+YDH*TGO)/(TGO*TGO)
ELSEIF(APN.EQ.1)THEN
    XNC=XNP*(YH+YDH*TGO+.5*XNTH*TGO*TGO)/(TGO*TGO)
ELSE
    XS=TGO/TAU
    TOP=6.*XS*XS*(EXP(-XS)-1.+XS)

```

(Contd.)

Listing 26.1 (Continued)

```

BOT1=2.*XS*XS*XS+3.+6.*XS-6.*XS*XS
BOT2=-12.*XS*EXP(-XS)-3.*EXP(-2.*XS)
XNPP=TOP/(.0001+BOT1+BOT2)
C1=XNPP/(TGO*TGO)
C2=XNPP/TGO
C3=.5*XNPP
C4=-XNPP*(EXP(-XS)+XS-1.)/(XS*XS)
XNC=C1*YH+C2*YDH+C3*XNTH+C4*XNL

ENDIF
ERRNT=XNT-XNTH
SP33=SQRT(P33)
SP33P=-SP33
THEORY=SQRT(M11+SIGN2)
WRITE(9,*)T,XNT/32.2,XNTH/32.2,Y,YSTAR
WRITE(1,*)T,XNT/32.2,XNTH/32.2,Y,YSTAR
GOTO 10
200 CONTINUE
XNT=XNTREAL*SIN(W*T)
TGO=TF-T+.00001
RTM=VC*TGO
XLAM=Y/(VC*TGO)
XLAMD=(RTM*YD+Y*VC)/(RTM**2)
XNLD=(XNC-XNL)/TAU
YDD=XNT-XNL
IF(STEP-1)66,66,55
999 CONTINUE
WRITE(9,*)Y
PAUSE
CLOSE(1)
END

SUBROUTINE GAUSS(X,SIG)
IMPLICIT REAL*8(A-H,O-Z)
INTEGER SUM
SUM=0
DO 14 J=1,6
IRAN=Random()
SUM=SUM+IRAN
14 CONTINUE
X=SUM/65536.
X=1.414*X*SIG
RETURN
END

```

Before we see how well the filter can estimate the weaving target maneuver, it is important to see how much effective noise there is on the filter measurement in this high closing velocity engagement. Figure 26.2 shows that although there is only 1 mr of measurement noise on the line of sight angle, the high closing velocity causes there to be significant noise on the effective measured relative position y^* . Essentially, the filter will have to take two derivatives of this noisy measurement to estimate target acceleration.

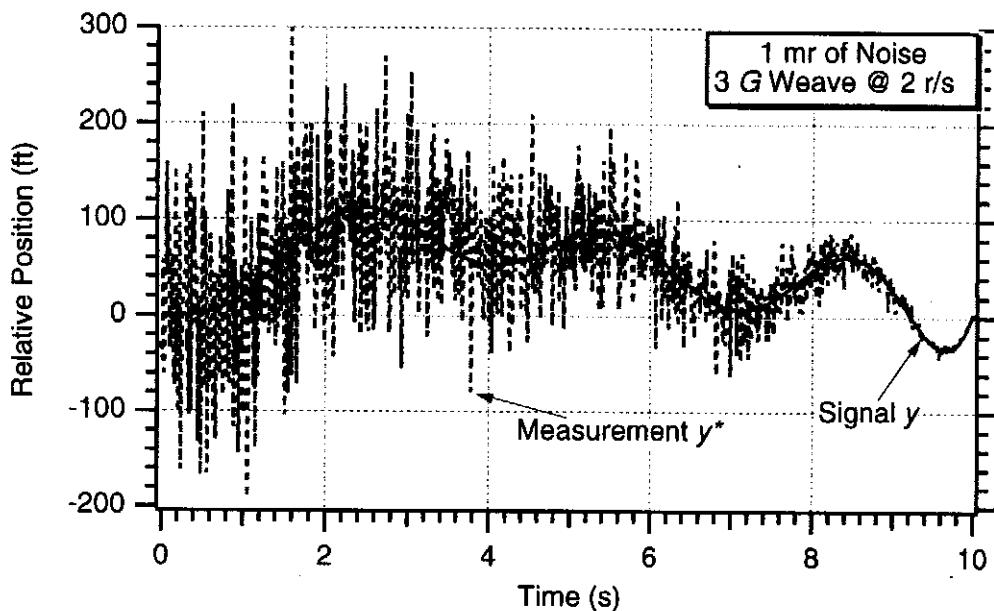


Fig. 26.2 With high closing velocity 1 mr of measurement noise translates into a great deal of noise on relative position.

The nominal case of Listing 26.1 was run and Fig. 26.3 shows that the three-state linear Kalman filter's estimate of target acceleration is not very good. The effective high-noise environment prevents the filter from accurately estimating the sinusoidal motion of the target maneuver. Near the end of the flight, where the effective measurement noise on relative position is diminished, the filter estimate improves but lags the actual target maneuver. It is important to note that the Kalman filter is really optimized for a constant target maneuver and is therefore suboptimal in this example because it is mismatched to the real world. However, the presence of process noise in the filter enables the Kalman filter to track all types of target maneuvers. Process noise lets the filter know that its model of the real world may be in error.

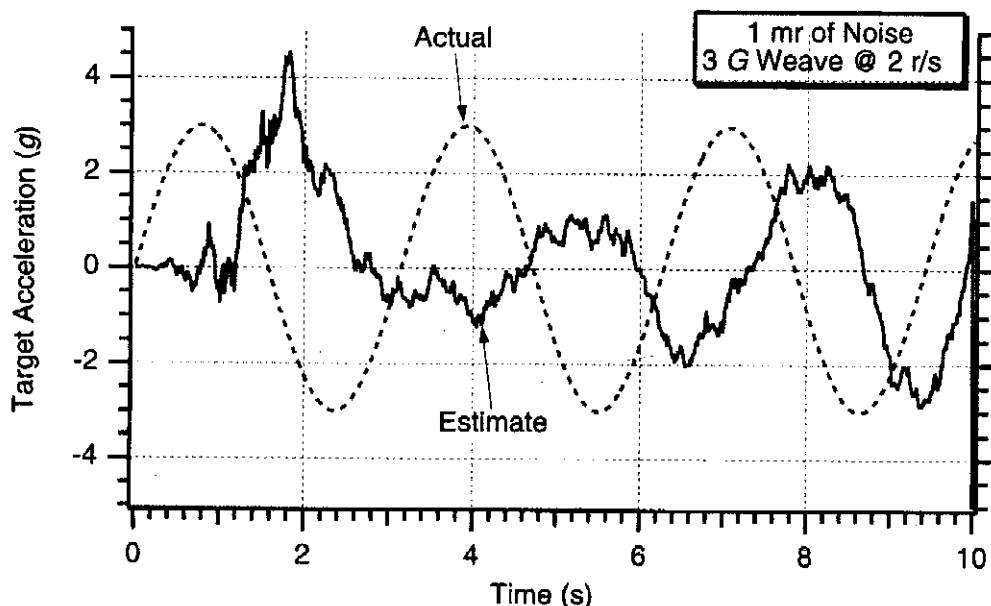


Fig. 26.3 With 1 mr of measurement noise the three-state Kalman filter has difficulty in estimating the sinusoidal nature of the weave maneuver.

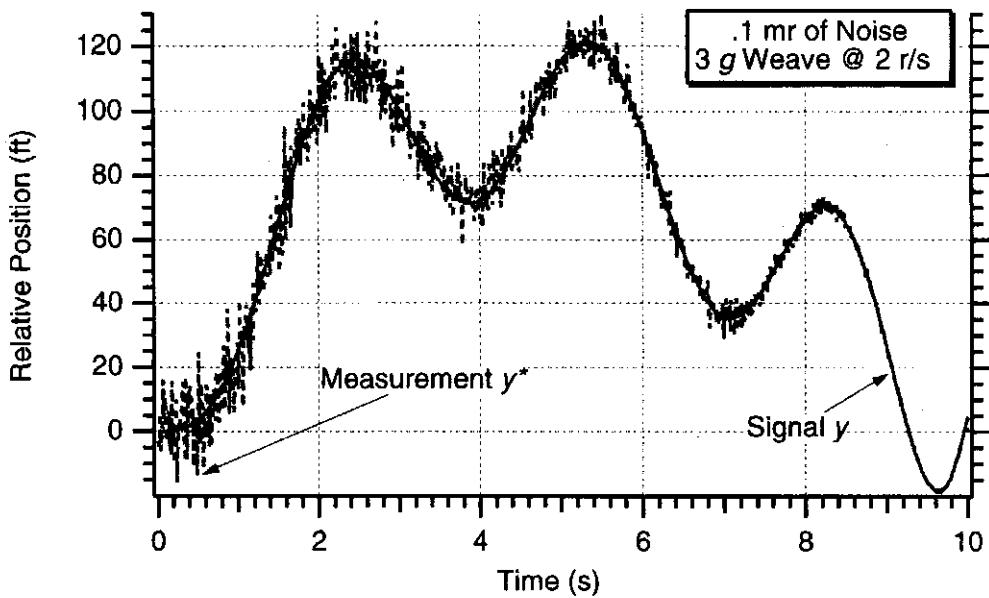


Fig. 26.4 Reducing measurement noise by order of magnitude is beneficial.

Listing 26.1 was modified so that the measurement noise was decreased by an order of magnitude to .1 mr (namely, SIGRIN=.0001). We can see from Fig. 26.4 that the effective measurement of relative position now more closely resembles the actual relative position. As a consequence of the reduced measurement noise we can see from Fig. 26.5 that the filter's estimate of the target maneuver now better approximates the sinusoidal nature of the maneuver. The estimate of the target maneuver is nearly perfect, except there is approximately a half-second lag between the actual maneuver and the estimate.

Thus, we can conclude that the original three-state linear Kalman filter of Chapter 9 can track a weaving target quite effectively if the measurement noise can be made small.

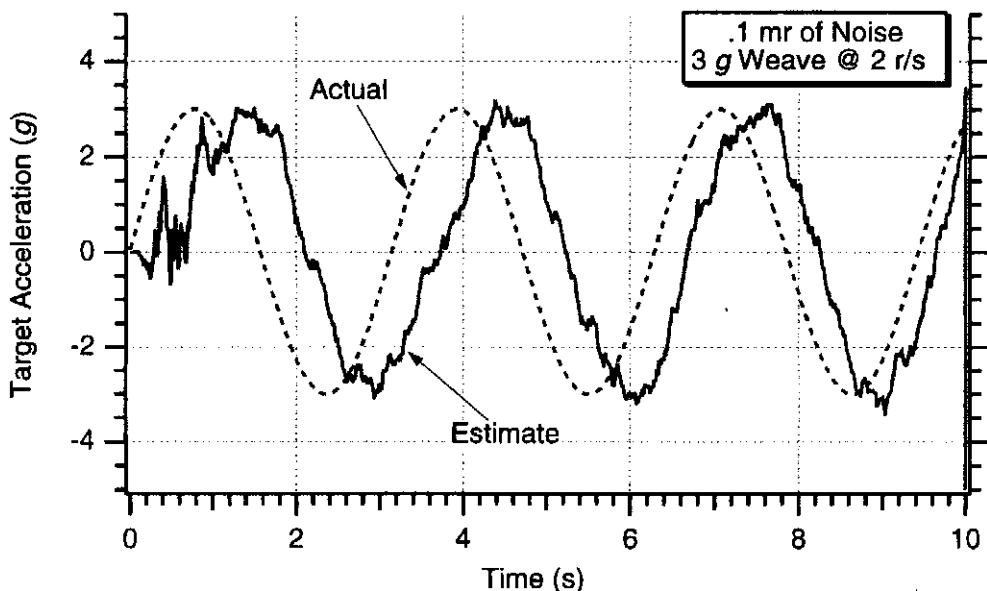


Fig. 26.5 Reducing measurement noise by order of magnitude improves target acceleration estimate.

Four-State Weave Kalman Filter

If we had a priori information that the target maneuver was sinusoidal in nature, one would think that a better Kalman filter could be designed. To design a Kalman filter optimized to estimate the states of a weaving target, we must first express the sinusoidal target motion in some statistical fashion. First recall from Chapter 1 that the Laplace transform of a sinusoidal signal is given by

$$\mathcal{L}(\sin \omega t) = \frac{\omega}{s^2 + \omega^2}$$

Therefore, if we assume that the target maneuver is sinusoidal in shape and that the starting time is still uniformly distributed over the flight time, we get the model of Fig. 26.6. Here the input to the sinusoidal transfer function is white noise u_s with spectral density

$$\Phi_s = \frac{n_{T\text{MAX}}^2}{t_F}$$

where again $n_{T\text{MAX}}$ is the peak of the sinusoidal maneuver and t_F the flight time. It was shown in Chapter 4 that mathematically this is the shaping filter equivalent of a target maneuver with sinusoidal amplitude but random starting time (where the starting time is uniformly distributed over the flight time).

In this homing system model we also effectively measure noisy relative position y^* . If the range from the interceptor to the target is known, it is easy to show that measuring relative position is equivalent to measuring the line of sight angle. The linear four-state weave Kalman filter will estimate relative position, relative velocity, target acceleration, and target jerk. The homing loop model of Fig. 26.2 assumes that the achieved missile acceleration n_L and the target weave frequency ω are both known and do not have to be estimated.

The model of Fig. 26.6 can be expressed in state space form as

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \ddot{y}_T \\ \ddot{y}_{T'} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \ddot{y}_T \\ \ddot{y}_{T'} \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} n_L + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega u_s \end{bmatrix}$$

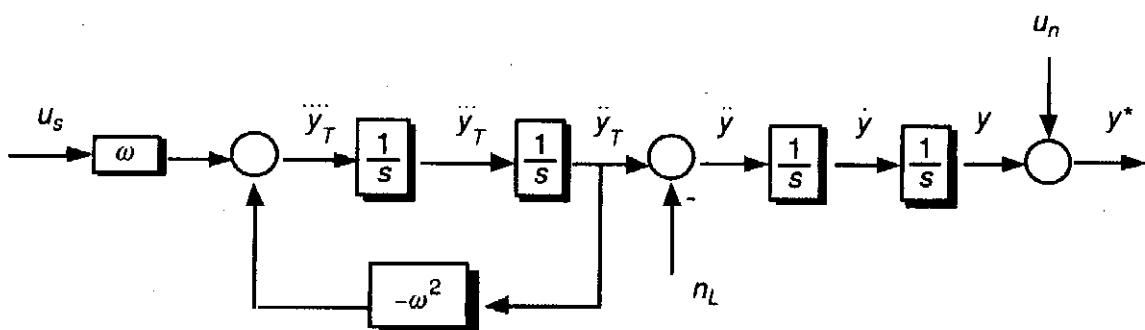


Fig. 26.6 Homing loop model for Kalman filter to be designed for sinusoidal target maneuver.

The systems dynamics matrix of the preceding equation can be written by inspection and is given by

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega^2 & 0 \end{bmatrix}$$

The fundamental matrix can be derived from the systems dynamics matrix according to

$$\Phi(t) = \mathcal{L}^{-1}[(sI - F)^{-1}]$$

Therefore, the fundamental matrix in the Laplace transform domain can be expressed as

$$\Phi(s) = (sI - F)^{-1} = \begin{bmatrix} s & -1 & 0 & 0 \\ 0 & s & -1 & 0 \\ 0 & 0 & s & -1 \\ 0 & 0 & s + \omega^2 & 0 \end{bmatrix}^{-1}$$

From the preceding equation we can see that first we must take the inverse of a four-by-four matrix and then take its inverse Laplace transform to find the fundamental matrix in the time domain. After considerable algebra, the continuous fundamental matrix turns out to be

$$\Phi(t) = \begin{bmatrix} 1 & t & \frac{1 - \cos \omega t}{\omega^2} & \frac{\omega t - \sin \omega t}{\omega^3} \\ 0 & 1 & \frac{\sin \omega t}{\omega} & \frac{1 - \cos \omega t}{\omega^2} \\ 0 & 0 & \cos \omega t & \frac{\sin \omega t}{\omega} \\ 0 & 0 & -\omega \sin \omega t & \cos \omega t \end{bmatrix}$$

By replacing time t with the sampling time T_s we obtain the discrete form of the fundamental matrix as

$$\Phi_k = \begin{bmatrix} 1 & T_s & \frac{1 - \cos x}{\omega^2} & \frac{x - \sin x}{\omega^3} \\ 0 & 1 & \frac{\sin x}{\omega} & \frac{1 - \cos x}{\omega^2} \\ 0 & 0 & \cos x & \frac{\sin x}{\omega} \\ 0 & 0 & -\omega \sin x & \cos x \end{bmatrix}$$

where

$$x = \omega T_s$$

The discrete measurement equation can be written by inspection of Fig. 26.6 as

$$y_k^* = [1 \ 0 \ 0 \ 0] \begin{bmatrix} y_k \\ \dot{y}_k \\ \ddot{y}_{T_k} \\ \ddot{y}_{T_k} \end{bmatrix} + v_k$$

which means that the discrete measurement matrix is given by

$$\mathbf{H}_k = [1 \ 0 \ 0 \ 0]$$

The continuous control matrix \mathbf{G} can be written by inspection of the original state space equation as

$$\mathbf{G} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}$$

After some algebra the discrete control matrix \mathbf{G}_k becomes

$$\mathbf{G}_k = \int_0^{T_s} \Phi(\tau) \mathbf{G}(\tau) d\tau = \begin{bmatrix} -.5T_s^2 \\ -T_s \\ 0 \\ 0 \end{bmatrix}$$

Finally, the continuous process noise matrix can be written from the system state space equation by inspection as

$$\mathbf{Q} = \mathbf{E} \left\{ [0 \ 0 \ 0 \ \omega u_s] \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega u_s \end{bmatrix} \right\} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega^2 \Phi_s \end{bmatrix}$$

After some algebra the discrete process noise matrix can be derived from the continuous process noise matrix according to

$$\mathbf{Q}_k = \int_0^{T_s} \Phi(\tau) \mathbf{Q} \Phi^T(\tau) d\tau = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} & Q_{14} \\ Q_{12} & Q_{22} & Q_{23} & Q_{24} \\ Q_{13} & Q_{23} & Q_{33} & Q_{34} \\ Q_{14} & Q_{24} & Q_{34} & Q_{44} \end{bmatrix}$$

where

$$Q_{11} = \frac{\Phi_s}{\omega^5} [.333x^3 - 2 \sin x + 2x \cos x + .5x - .25 \sin 2x]$$

$$Q_{12} = \frac{\Phi_s}{\omega^4} [.5x^2 - x \sin x + .5 \sin^2 x]$$

$$Q_{13} = \frac{\Phi_s}{\omega^3} [\sin x - x \cos x - .5x + .25 \sin 2x]$$

$$Q_{14} = \frac{\Phi_s}{\omega^2} [\cos x + x \sin x - .5 \sin^2 x - 1]$$

$$Q_{22} = \frac{\Phi_s}{\omega^3} [1.5x - 2 \sin x + .25 \sin 2x]$$

$$Q_{23} = \frac{\Phi_s}{\omega^2} [1 - \cos x - .5 \sin^2 x]$$

$$Q_{24} = \frac{\Phi_s}{\omega} [\sin x - .5x - .25 \sin 2x]$$

$$Q_{33} = \frac{\Phi_s}{\omega} [.5x - .25 \sin 2x]$$

$$Q_{34} = .5\Phi_s \sin^2 x$$

$$Q_{44} = \omega\Phi_s [.5x + .25 \sin 2x]$$

Recall that in the preceding set of expressions the process noise and normalized weave frequency have been defined as

$$\Phi_s = \frac{\omega^2 n_{T_{MAX}}^2}{t_F}$$

$$x = \omega T_s$$

Recall that the discrete Kalman filtering equation is given by

$$\hat{x}_k = \Phi_k \hat{x}_{k-1} + G_k u_{k-1} + K_k (z_k - H \Phi_k \hat{x}_{k-1} - H G_k u_{k-1})$$

Substitution of the appropriate matrices into the preceding matrix difference equation yields

$$\begin{aligned} \begin{bmatrix} \hat{y}_k \\ \hat{\dot{y}}_k \\ \hat{y}_{T_k} \\ \hat{\dot{y}}_{T_k} \end{bmatrix} &= \begin{bmatrix} 1 & T_s & \frac{1 - \cos x}{\omega^2} & \frac{x - \sin x}{\omega^3} \\ 0 & 1 & \frac{\sin x}{\omega} & \frac{1 - \cos x}{\omega^2} \\ 0 & 0 & \cos x & \frac{\sin x}{\omega} \\ 0 & 0 & -\omega \sin x & \cos x \end{bmatrix} \begin{bmatrix} \hat{y}_{k-1} \\ \hat{\dot{y}}_{k-1} \\ \hat{y}_{T_{k-1}} \\ \hat{\dot{y}}_{T_{k-1}} \end{bmatrix} + \begin{bmatrix} -.5T_s^2 \\ -T_s \\ 0 \\ 0 \end{bmatrix} n_{L_{k-1}} \\ &+ \begin{bmatrix} K_1 \\ K_2 \\ K_3 \\ K_4 \end{bmatrix} \left[y_k^* - [1 \ 0 \ 0 \ 0] \begin{bmatrix} 1 & T_s & \frac{1 - \cos x}{\omega^2} & \frac{x - \sin x}{\omega^3} \\ 0 & 1 & \frac{\sin x}{\omega} & \frac{1 - \cos x}{\omega^2} \\ 0 & 0 & \cos x & \frac{\sin x}{\omega} \\ 0 & 0 & -\omega \sin x & \cos x \end{bmatrix} \begin{bmatrix} \hat{y}_{k-1} \\ \hat{\dot{y}}_{k-1} \\ \hat{y}_{T_{k-1}} \\ \hat{\dot{y}}_{T_{k-1}} \end{bmatrix} \right. \\ &\quad \left. - [1 \ 0 \ 0 \ 0] \begin{bmatrix} -.5T_s^2 \\ -T_s \\ 0 \\ 0 \end{bmatrix} n_{L_{k-1}} \right] \end{aligned}$$

We can multiply out the terms of the preceding matrix equation to yield the Kalman filter scalar equations

$$\begin{aligned} \text{RES}_k &= y_k^* - \hat{y}_{k-1} - T_s \hat{y}_{k-1} - \frac{(1 - \cos x)}{\omega^2} \hat{y}_{T_{k-1}} - \frac{(x - \sin x)}{\omega^3} \hat{y}_{\dot{T}_{k-1}} + .5T_s^2 n_{L_{k-1}} \\ \hat{y}_k &= \hat{y}_{k-1} + T_s \hat{y}_{k-1} + \frac{(1 - \cos x)}{\omega^2} \hat{y}_{T_{k-1}} + \frac{(x - \sin x)}{\omega^3} \hat{y}_{\dot{T}_{k-1}} \\ &\quad - .5T_s^2 n_{L_{k-1}} + K_{1k} \text{RES}_k \\ \hat{y}_k &= \hat{y}_{k-1} + \frac{\sin x}{\omega} \hat{y}_{T_{k-1}} + \frac{(1 - \cos x)}{\omega^2} \hat{y}_{\dot{T}_{k-1}} - T_S n_{L_{k-1}} + K_{2k} \text{RES}_k \\ \hat{y}_{T_k} &= \cos x \hat{y}_{T_{k-1}} + \frac{\sin x}{\omega} \hat{y}_{\dot{T}_{k-1}} + K_{3k} \text{RES}_k \\ \hat{y}_{\dot{T}_k} &= -\omega \sin x \hat{y}_{T_{k-1}} + \cos x \hat{y}_{\dot{T}_{k-1}} + K_{4k} \text{RES}_k \end{aligned}$$

Both the Riccati equations and Kalman filtering equations for the linear four-state weave Kalman filter were programmed as part of the homing loop, and the resultant linearized missile-target engagement simulation appears in Listing 26.2. We can see that the simulation has a single time constant representation of the flight control system plus a 3 g weaving target with a weave frequency of 2 rad/s. Nominally there is 1 mr of measurement noise on the line of sight angle and the closing velocity is 9000 ft/s to reflect a ballistic target engagement. We can see from Listing 26.2 that the guidance law options for this filter are either proportional navigation, augmented proportional navigation, optimal guidance, weave guidance, or compensated weave guidance (namely, APN=0, 1, 2, 3, or 4, respectively).

The nominal case of Listing 26.2 was run, and Fig. 26.7 shows that the linear four-state weave Kalman filter's estimate of target acceleration is much better than the general purpose linear three-state Kalman filter when the measurement noise is 1 mr (see Fig. 26.3). Compared with the previous section, when the amount of measurement noise is large, it is now easier to see from the state estimates of the weave Kalman filter that the target maneuver is indeed sinusoidal. In addition, the weave Kalman filter also yields estimates of the target jerk. We can see from Fig. 26.8 that the weave Kalman filter provides fairly good estimates of the target jerk.

To see if the four-state weave Kalman filter is truly working properly, it is necessary to examine the errors in the state estimates. Figures 26.9 and 26.10 show that the errors in the estimates of target acceleration and jerk appear to lie within the theoretical bounds (that is, \pm square root of P_{33} and P_{44} , respectively) approximately 68% of the time, indicating that the filter is working properly.

Figures 26.11 and 26.12 indicate that when the measurement noise on the line of sight angle is reduced by an order of magnitude to .1 mr, the four-state weave Kalman filter estimates of target acceleration and jerk improve significantly. In fact, we can see that the estimates of these states are nearly perfect in the low-noise environment.

Thus, we can conclude that if the target is maneuvering in a sinusoidal fashion, and we have knowledge of the target wave frequency, superior estimates of the target acceleration and jerk can be obtained with the linear four-state weave Kalman filter.

Listing 26.2 Weave Kalman filter and weaving target

```

GLOBAL DEFINE
    INCLUDE 'quickdraw.inc'
END
IMPLICIT REAL*8(A-H,O-Z)
REAL*8 PHI(4,4),P(4,4),M(4,4),PHIP(4,4),PHIPPHIT(4,4),GAIN(4,1)
REAL*8 Q(4,4),HMAT(1,4),HM(1,4),MHT(4,1)
REAL*8 PHIT(4,4)
REAL*8 HMHT(1,1),HT(4,1),KH(4,4),IDN(4,4),IKH(4,4)
LOGICAL QPERFECT
INTEGER APN,ORDER
TAU=.5
APN=0
ORDER=4
MVR=1
VC=9000.
W=2.
WREAL=2.
WH=W
XNT=96.6
XNTREAL=96.6
TS=.01
YIC=0.
VM=3000.
HEDEG=0.
HEDEGFIL=20.
XNP=3.
SIGRIN=.001
SIGGL=0.
RA=21000.
SRN=0.
TF=10.
QPERFECT=.FALSE.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
OPEN(2,STATUS='UNKNOWN',FILE='COVFIL')
PHASE=0./57.3
X=WH*TS
Y=YIC
YD=-VM*HEDEG/57.3
PHIS=WH*WH*XNT*XNT/TF
RTM=VC*TF
SIGNOISE=SQRT(SIGRIN**2+(SIGGL/RTM)**2+
    (SRN*RTM*RTM/(RA*RA))**2)
SIGPOS=RTM*SIGNOISE
SIGN2=SIGPOS**2
DO 1000 I=1,ORDER
DO 1000 J=1,ORDER
    PHI(I,J)=0.
    P(I,J)=0.
    Q(I,J)=0.
    IDN(I,J)=0.

```

(Contd.)

Listing 26.2 (Continued)

```

1000 CONTINUE
    PHI(1,1)=1
    PHI(1,2)=TS
    PHI(1,3)=(1-COS(X))/(WH*WH)
    PHI(1,4)=(X-SIN(X))/(WH*WH*WH)
    PHI(2,2)=1
    PHI(2,3)=SIN(X)/WH
    PHI(2,4)=(1-COS(X))/(WH*WH)
    PHI(3,3)=COS(X)
    PHI(3,4)=SIN(X)/WH
    PHI(4,3)=-WH*SIN(X)
    PHI(4,4)=COS(X)
    Q(1,1)=PHIS*(.333*X**3-2*SIN(X)+2*X*COS(X)+.5*X-
1           .25*SIN(2*X))/(WH**5)
    Q(1,2)=PHIS*(.5*X*X-X*SIN(X)+.5*SIN(X)*SIN(X))/(WH**4)
    Q(2,1)=Q(1,2)
    Q(1,3)=PHIS*(SIN(X)-X*COS(X)-.5*X+.25*SIN(2*X))/(WH**3)
    Q(3,1)=Q(1,3)
    Q(1,4)=PHIS*(COS(X)+X*SIN(X)-.5*SIN(X)*SIN(X)-1)/(WH*WH)
    Q(4,1)=Q(1,4)
    Q(2,2)=PHIS*(1.5*X-2*SIN(X)+.25*SIN(2*X))/(WH**3)
    Q(2,3)=PHIS*(1-COS(X)-.5*SIN(X)*SIN(X))/(WH*WH)
    Q(3,2)=Q(2,3)
    Q(2,4)=PHIS*(SIN(X)-.5*X-.25*SIN(2*X))/WH
    Q(4,2)=Q(2,4)
    Q(3,3)=PHIS*(.5*X-.25*SIN(2*X))/WH
    Q(3,4)=.5*PHIS*SIN(X)*SIN(X)
    Q(4,3)=Q(3,4)
    Q(4,4)=WH*PHIS*(.5*X+.25*SIN(2*X))
    IDN(1,1)=1.
    IDN(2,2)=1.
    IDN(3,3)=1.
    IDN(4,4)=1.
    P(1,1)=SIGN2
    P(2,2)=(VM*HEDEGFIL/57.3)**2
    P(3,3)=XNT*XNT
    P(4,4)=WH*WH*XNT*XNT
    DO 1100 I=1,ORDER
        HMAT(1,I)=0.
        HT(I,1)=0.
1100 CONTINUE
    HMAT(1,1)=1.
    HT(1,1)=1.
    CALL MATTRN(PHI,ORDER,ORDER,PHIT)

    T=0.
    H=.001
    S=0.
    XNC=0.
    XNL=0.

```

(Contd.)

Listing 26.2 (Continued)

```

XLAM=Y/RTM
IF(MVR.EQ.0)THEN
    YTDD=XNTREAL
    YTDDD=0.
ELSE
    YTDD=XNTREAL*SIN(WREAL*T)
    YTDDD=XNTREAL*WREAL*COS(WREAL*T)
ENDIF
IF(QPERFECT)THEN
    YH=Y
    YDH=YD
    YTDDH=YTDD
    YTDDDH=YTDDD
ELSE
    YH=0.
    YDH=0.
    YTDDH=0.
    YTDDDH=0.
ENDIF
10 IF(T>(TF-.0001))GOTO 999
    YOLD=Y
    YDOLD=YD
    XNLOLD=XNL
    STEP=1
    GOTO 200
66 STEP=2
    Y=Y+H*YD
    YD=YD+H*YDD
    XNL=XNL+H*XNLD
    T=T+H
    GOTO 200
55 CONTINUE
    Y=.5*(YOLD+Y+H*YD)
    YD=.5*(YDOLD+YD+H*YDD)
    XNL=.5*(XNLOLD+XNL+H*XNLD)
    S=S+H
    IF(S<(TS-.00001))GOTO 10
    S=0.
    TGO=TF-T+.000001
    RTM=VC*TGO
    SIGNOISE=SQR(SIGRIN**2+(SIGGL/RTM)**2+
                  (SRN*RTM*RTM/(RA*RA))**2)
    SIGPOS=RTM*SIGNOISE
    SIGN2=SIGPOS**2
    CALL MATMUL(PHI,ORDER,ORDER,P,ORDER,ORDER,PHIP)
    CALL MATMUL(PHIP,ORDER,ORDER,PHIT,ORDER,ORDER,PHIPPHIT)
    CALL MATADD(PHIPPHIT,ORDER,ORDER,Q,M)
    CALL MATMUL(HMAT,1,ORDER,M,ORDER,ORDER,HM)
    CALL MATMUL(HM,1,ORDER,HT,ORDER,1,HMHT)
    HMHTR=HMHT(1,1)+SIGN2

```

(Contd.)

Listing 26.2 (Continued)

```

HMHTRINV=1./HMHTR
CALL MATMUL(M,ORDER,ORDER,HT,ORDER,1,MHT)
CALL MATMUL(MHT,ORDER,1,HMHTRINV,1,1,GAIN)
CALL MATMUL(GAIN,ORDER,1,HMAT,1,ORDER,KH)
CALL MATSUB(IDN,ORDER,ORDER,KH,IKH)
CALL MATMUL(IKH,ORDER,ORDER,M,ORDER,ORDER,P)
IF(MVR.EQ.0)THEN
    YTDD=XNTREAL
    YTDDD=0.
ELSE
    YTDD=XNTREAL*SIN(WREAL*T)
    YTDDD=XNTREAL*WREAL*COS(WREAL*T)
ENDIF
CALL GAUSS(XLAMNOISE,SIGNOISE)
YSTAR=RTM*(XLAM+XLAMNOISE)

RES=YSTAR-YH-TS*YDH-(1-COS(X))*YTDDH/(WH*WH)-
1      (X-SIN(X))*YTDDDH/(WH*WH*WH)+.5*TS*TS*XNL
1      YH=YH+TS*YDH+(1-COS(X))*YTDDH/(WH*WH)+(X-SIN(X))
1          *YTDDDH/(WH*WH*WH)+GAIN(1,1)*RES-.5*TS*TS*XNL
1      YDH=YDH+SIN(X)*YTDDH/WH+(1-COS(X))*YTDDDH/(WH*WH)+_
1          GAIN(2,1)
1          *RES-TS*XNL
1      YTDDHNEW=COS(X)*YTDDH+SIN(X)*YTDDDH/WH+GAIN(3,1)*RES
1      YTDDDH=-WH*SIN(X)*YTDDH+COS(X)*YTDDDH+GAIN(4,1)*RES
1      YTDDH=YTDDHNEW
IF(APN.EQ.0)THEN
    XNC=XNP*(YH+YDH*TGO)/(TGO*TGO)
ELSEIF(APN.EQ.1)THEN
    XNC=XNP*(YH+YDH*TGO+.5*YTDDH*TGO*TGO)/(TGO*TGO)
ELSEIF(APN.EQ.2)THEN
    XS=TGO/TAU
    TOP=6.*XS*XS*(EXP(-XS)-1.+XS)
    BOT1=2*XS*XS*XS+3.+6.*XS-6.*XS*XS
    BOT2=-12.*XS*EXP(-XS)-3.*EXP(-2.*XS)
    XNPP=TOP/(.0001+BOT1+BOT2)
    C1=XNPP/(TGO*TGO)
    C2=XNPP/TGO
    C3=.5*XNPP
    C4=-XNPP*(EXP(-XS)+XS-1.)/(XS*XS)
    XNC=C1*YH+C2*YDH+C3*YTDDH+C4*XNL
ELSEIF(APN.EQ.3)THEN
    XP=WH*TGO
    XNC=XNP*(YH+YDH*TGO)/(TGO*TGO)+_
1        XNP*YTDDH*(1.-COS(XP))/XP**2+_
1        XNP*YTDDDH*(XP-SIN(XP))/(XP*XP*WH)
ELSE
    XS=TGO/TAU
    TOP=6.*XS*XS*(EXP(-XS)-1.+XS)
    BOT1=2*XS*XS*XS+3.+6.*XS-6.*XS*XS

```

(Contd.)

Listing 26.2 (Continued)

```

BOT2=-12.*XS*EXP(-XS)-3.*EXP(-2.*XS)
XNPP=TOP/(.0001+BOT1+BOT2)
C1=XNPP/(TGO*TGO)
C2=XNPP/TGO
C3=XNPP*(1.-COS(WH*TGO))/(WH*WH*TGO*TGO)
C4=-XNPP*(EXP(-XS)+XS-1.)/(XS*XS)
C5=XNPP*(WH*TGO-SIN(WH*TGO))/(WH*WH*WH*TGO*TGO)
XNC=C1*YH+C2*YDH+C3*YTDDH+C4*XNL+C5*YTDDDH
ENDIF
YTDDG=YTDD/32.2
YTDDHG=YTDDH/32.2
ERRY=Y-YH
SP11=SQRT(P(1,1))
SP11P=-SP11
ERRYD=YD-YDH
SP22=SQRT(P(2,2))
SP22P=-SP22
ERRYTDDG=(YTDD-YTDDH)/32.2
SP33G=SQRT(P(3,3))/32.2
ERRYTDDDG=(YTDDD-YTDDDH)/32.2
SP44G=SQRT(P(4,4))/32.2
WRITE(9,*)T,YTDD/32.2,YTDDH/32.2,YTDDD/32.2,YTDDDH/32.2
WRITE(1,*)T,YTDD/32.2,YTDDH/32.2,YTDDD/32.2,YTDDDH/32.2
WRITE(2,*)T,ERRYTDDG,SP33G,-SP33G,ERRYTDDDG,SP44G,-SP44G
GOTO 10
200 CONTINUE
TGO=TF-T+.000001
RTM=VC*TGO
XLAM=Y/(VC*TGO)
IF(MVR.EQ.0)THEN
    YTDD=XNTREAL
ELSE
    YTDD=XNTREAL*SIN(WREAL*T)
ENDIF
XNLD=(XNC-XNL)/TAU
YDD=YTDD-XNL
IF(STEP-1)66,66,55
999 CONTINUE
WRITE(9,*)Y
PAUSE
CLOSE(1)
END

SUBROUTINE GAUSS(X,SIG)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER*2 Random
INTEGER SUM
SUM=0
DO 14 J=1,6
IRAN=Random()

```

(Contd.)

Listing 26.2 (Continued)

```

14    SUM=SUM+IRAN
      CONTINUE
      X=SUM/65536.
      X=1.414*X*SIG
      RETURN
      END

      SUBROUTINE MATMUL(A,IROW,ICOL,B,JROW,JCOL,C)
      IMPLICIT REAL*8 (A-H)
      IMPLICIT REAL*8 (O-Z)
      REAL*8 A(IROW,ICOL),B(JROW,JCOL),C(IROW,JCOL)
      DO 110 I=1,IROW
      DO 110 J=1,JCOL
          C(I,J)=0.
          DO 110 K=1,ICOL
              C(I,J)=C(I,J)+A(I,K)*B(K,J)
110   CONTINUE
      RETURN
      END

      SUBROUTINE MATADD(A,IROW,ICOL,B,C)
      IMPLICIT REAL*8 (A-H)
      IMPLICIT REAL*8 (O-Z)
      REAL*8 A(IROW,ICOL),B(IROW,ICOL),C(IROW,ICOL)
      DO 120 I=1,IROW
      DO 120 J=1,ICOL
          C(I,J)=A(I,J)+B(I,J)
120   CONTINUE
      RETURN
      END

      SUBROUTINE MATTRN(A,IROW,ICOL,AT)
      IMPLICIT REAL*8 (A-H)
      IMPLICIT REAL*8 (O-Z)
      REAL*8 A(IROW,ICOL),AT(ICOL,IROW)
      DO 105 I=1,IROW
      DO 105 J=1,ICOL
          AT(J,I)=A(I,J)
105   CONTINUE
      RETURN
      END

      SUBROUTINE MATSUB(A,IROW,ICOL,B,C)
      IMPLICIT REAL*8 (A-H)
      IMPLICIT REAL*8 (O-Z)
      REAL*8 A(IROW,ICOL),B(IROW,ICOL),C(IROW,ICOL)
      DO 120 I=1,IROW
      DO 120 J=1,ICOL
          C(I,J)=A(I,J)-B(I,J)
120   CONTINUE
      RETURN
      END

```

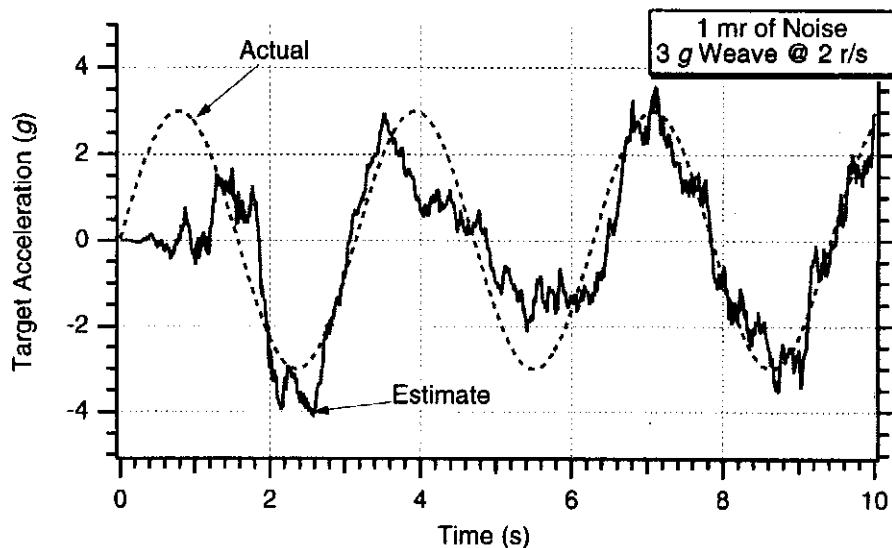


Fig. 26.7 Four-state weave Kalman filter yields better estimates than three-state Kalman filter when measurement noise is large.

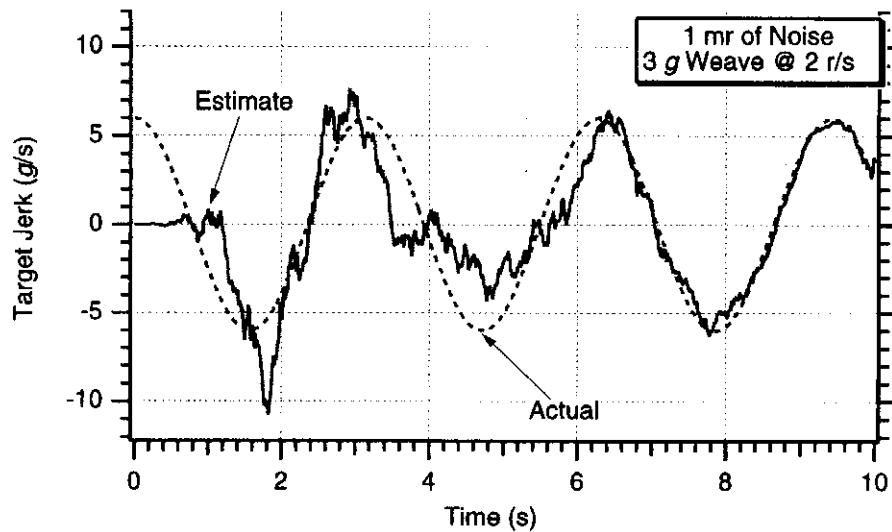


Fig. 26.8 Weave Kalman filter also provides a fairly good estimate of target jerk when measurement noise is large.

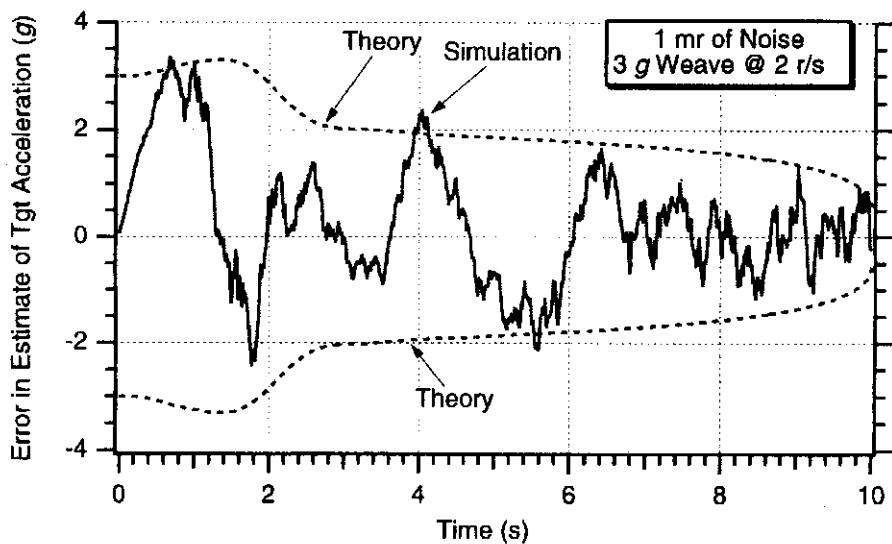


Fig. 26.9 Weave Kalman filter estimation errors for target acceleration are within theoretical bounds.

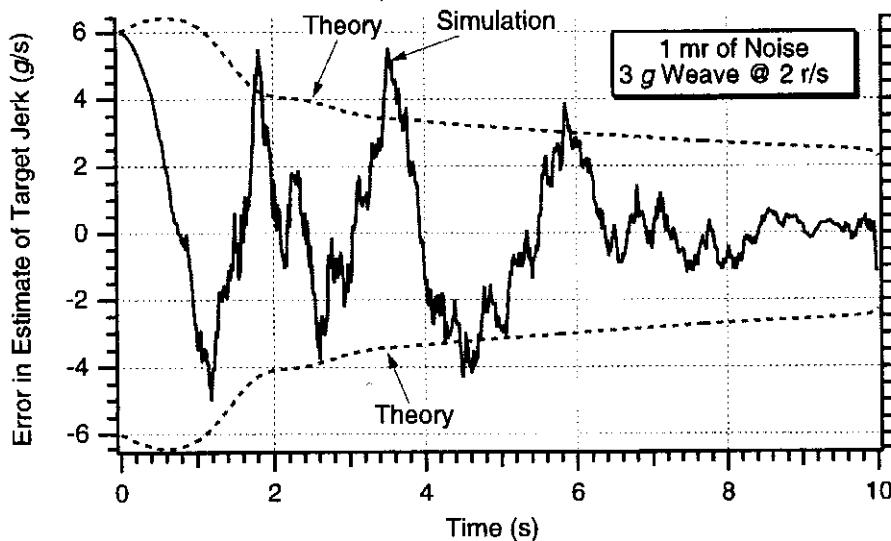


Fig. 26.10 Weave Kalman filter estimation errors for target jerk are within theoretical bounds.

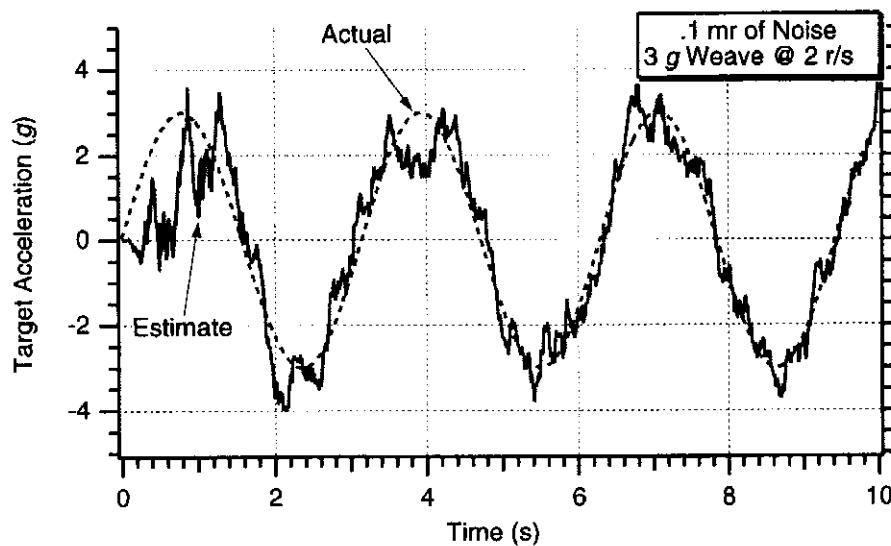


Fig. 26.11 Reducing measurement noise improves four-state weave Kalman filter's target acceleration estimate.

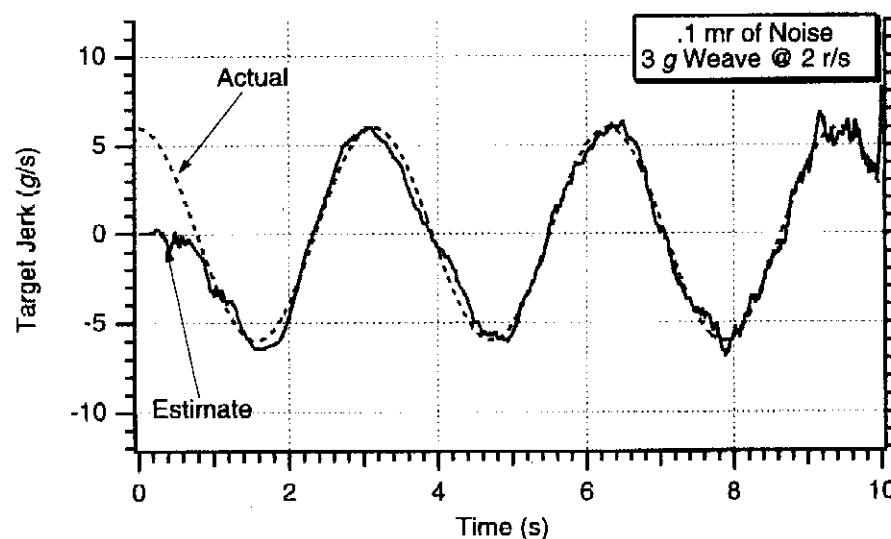


Fig. 26.12 Reducing measurement noise improves weave Kalman filter's target jerk estimate.

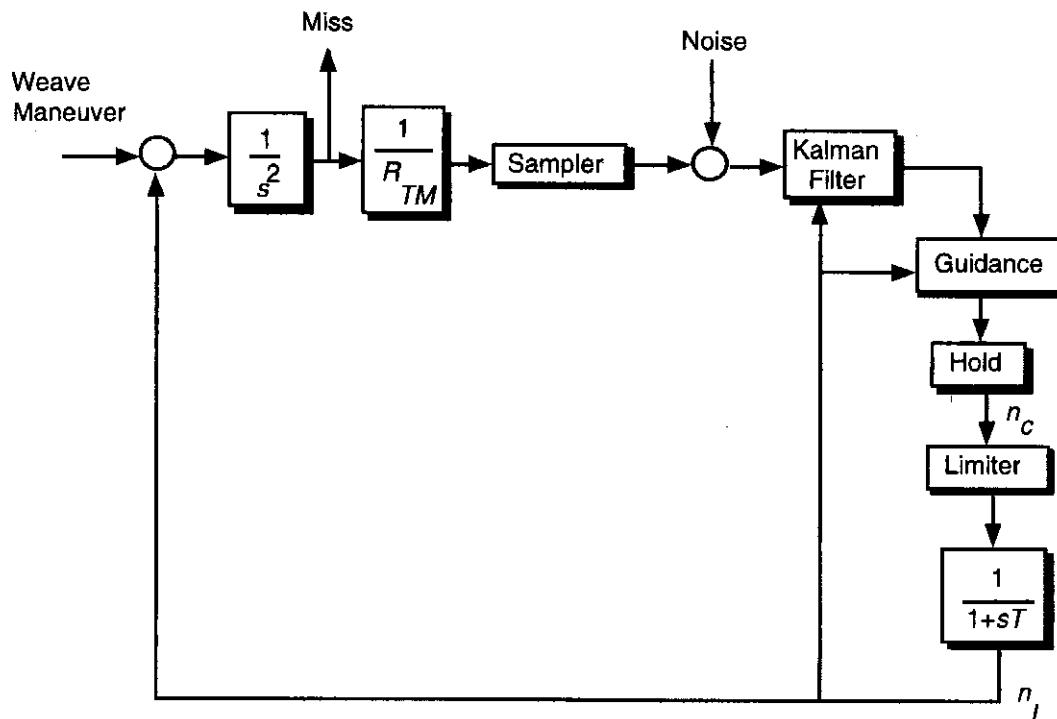


Fig. 26.13 Guidance system model for miss distance analysis.

Miss Distance Analysis

To see how the various filtering and guidance law options perform in terms of the RMS miss distance in the presence of a weaving target, the homing loop model of Fig. 26.13 is considered. With this guidance system model the Kalman filter can either be the general-purpose, linear, three-state filter or the special-purpose, four-state, weave filter. When the linear, three-state Kalman filter is used, the possible guidance laws that can be used are either proportional navigation or optimal guidance. Recall that these two guidance laws can be expressed as

$$n_{c_{PN}} = \frac{N'}{t_{go}^2}(y + \dot{y}t_{go})$$

$$n_{c_{Optimal}} = \frac{N'}{t_{go}^2} \left[y + \dot{y}t_{go} + \frac{t_{go}^2}{2} \ddot{y}_T - n_L T^2 (e^{-x} + x - 1) \right]$$

With proportional navigation, the effective navigation ratio is usually chosen to be a constant in the range of 3 to 5. With optimal guidance the effective navigation ratio is not constant but can be computed from

$$N' = \frac{6x^2(e^{-x} - 1 + x)}{2x^3 + 3 + 6x - 6x^2 - 12xe^{-x} - 3e^{-2x}}$$

We showed in Chapter 8 that when the missile is very far away from the target (that is, t_{go} is large) the effective navigation ratio approaches 3. As the missile gets closer to the target, the navigation ratio increases to a larger number. Both guidance laws make use of the state estimates from the linear three-state Kalman filter (that is, estimates of relative position, relative velocity, and target acceleration). The compensated weave guidance law can only be used with the four-state weave

Table 26.1 Nominal values for experiment

Parameter	Value
Autopilot time constant	.5 s
Missile velocity	3000 ft/s
Closing velocity	9000 ft/s
Target acceleration level	3 g
Target weave frequency	2 r/s

Kalman filter given by

$$n_{c\text{Weave Lag}} = \frac{N'}{t_{go}^2} \left[y + \dot{y}t_{go} + \frac{1 - \cos \omega t_{go}}{\omega^2} \ddot{y}_T + \frac{\omega t_{go} - \sin \omega t_{go}}{\omega^3} \ddot{\ddot{y}}_T - n_L T^2 (e^{-x} + x - 1) \right]$$

where the effective navigation ratio is the same as it was with the optimal guidance law. Again, this guidance law makes use of the state estimates from the linear four-state weave Kalman filter (that is, estimates of relative position, relative velocity, target acceleration, and target jerk). It is important to note that when this guidance law is used in conjunction with the weave Kalman filter it is assumed that the target weave frequency is known.

The parameters used for the guidance system analysis appear in Table 26.1. We can see from the table that the weaving target has a 3 g maneuver amplitude and a weave frequency of 2 r/s. The flight control system time constant is set at .5 s. Notice the high closing velocity in Table 26.1 is representative of a ballistic target engagement.

Experiments were run with the different guidance system configurations. Twenty-five Monte Carlo sets were run for flight times ranging from .5 s to 10 s in steps of .5 s. We can see from Fig. 26.14 that for the case in which there is a

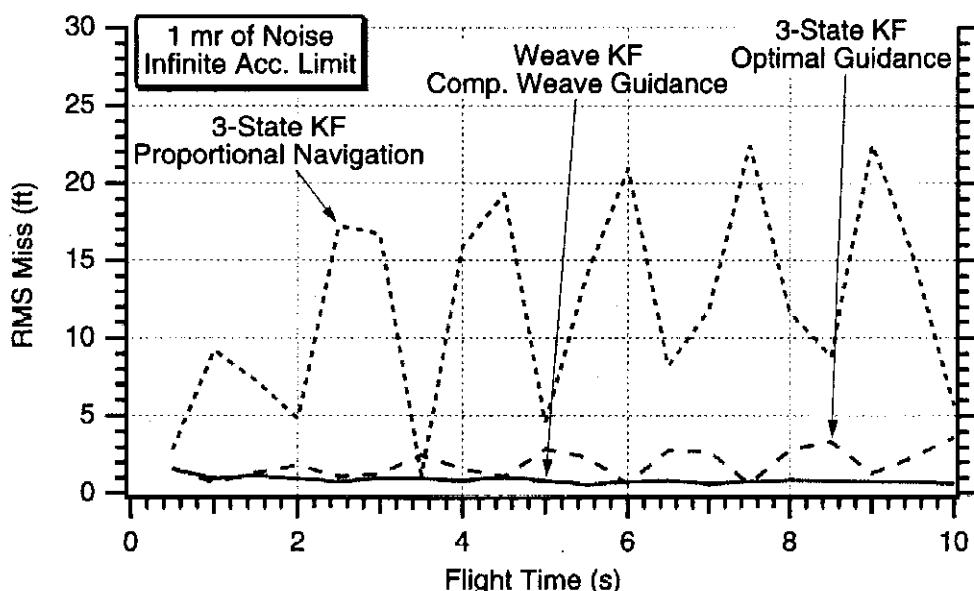


Fig. 26.14 Both optimal guidance and compensated weave guidance have similar performance.

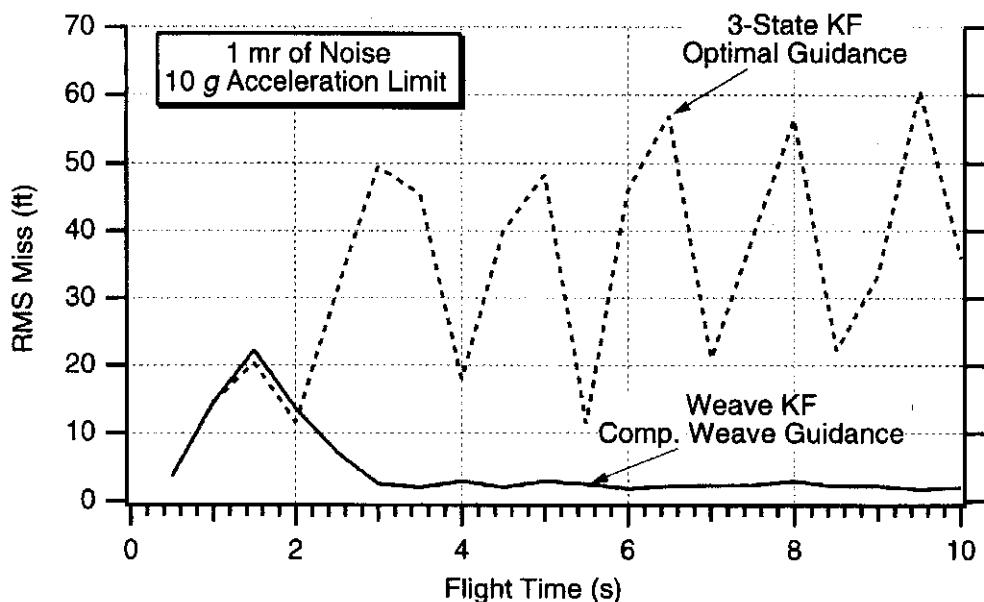


Fig. 26.15 Weave guidance can be superior to optimal guidance when acceleration saturation effects are considered.

infinite missile acceleration capability and 1 mr of measurement noise, both the three-state Kalman filter using optimal guidance and the four-state weave Kalman filter using the compensated weave guidance law yield approximately the same results. Both guidance laws yield significantly smaller RMS miss distances than proportional navigation.

The previous case assumed that the missile had infinite acceleration capability. If the acceleration limit is set to 10 g (more than a 3 to 1 advantage over the target), we can see from Fig. 26.15 that the RMS miss distance performance of optimal guidance deteriorates significantly. In this particular case we can see that the compensated weave guidance law in combination with the four-state weave Kalman filter yields dramatic performance advantages. Thus, we can conclude that it is only advantageous to use compensated weave guidance rather than optimal guidance if the missile has a small missile to target acceleration advantage and knows the target weave frequency.

If we compare Figs. 26.16 and 26.15, we can see that reducing the measurement noise by an order of magnitude to .1 mr improves the performance of both optimal guidance and compensated weave guidance. However, because in this example there is still a small missile to target acceleration advantage, it is still better to use compensated weave guidance if the target weave frequency is known.

Recall that the four-state weave Kalman filter and compensated weave guidance law both required knowledge of the target weave frequency. So far it has been assumed that the target weave frequency has been known perfectly. Errors in the knowledge of the target weave frequency will degrade both the performance of the four-state weave Kalman filter and compensated weave guidance law. Figure 26.17 shows that when the estimated weave frequency is either twice as large or half as small as the actual target weave frequency, significant performance degradation may occur. By comparing Figs. 26.17 and 26.16 we can see that when the target weave frequency is in error we can do just as well and sometimes better by using optimal guidance and the three-state Kalman filter because this combination does not require knowledge of the target weave frequency. Thus, we can see that if the target weave frequency is not known in advance it must somehow be estimated in real time if we wish to derive the benefits of the compensated weave guidance law.

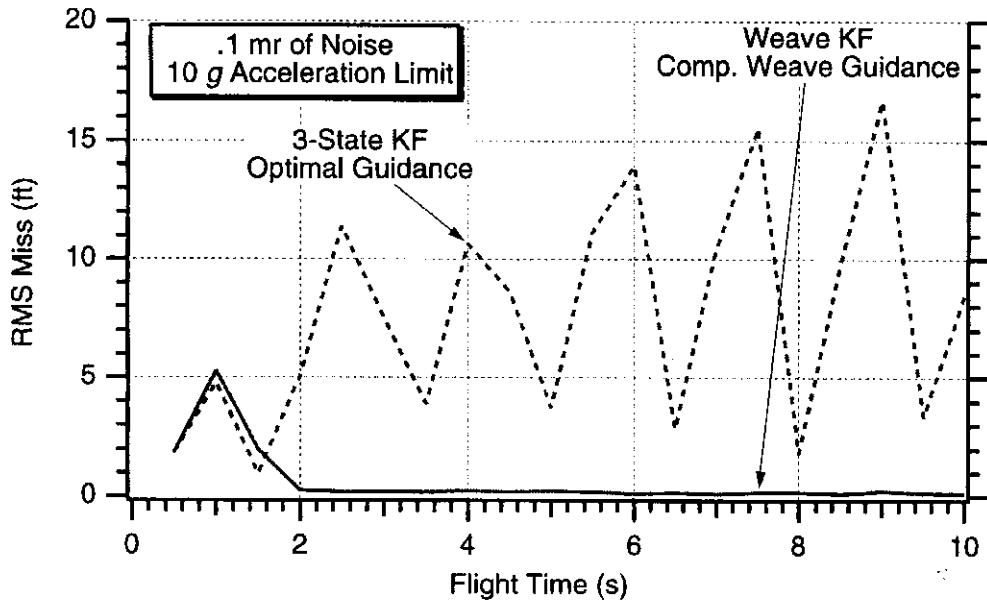


Fig. 26.16 Reducing measurement noise improves performance of both systems but compensated weave guidance is still better.

Extended Kalman Filter¹

To build a Kalman filter that can estimate the target weave frequency, it is first necessary to write the state equations representing our model of the real world. From Fig. 26.6 we can say that the equations for the homing loop with a weave target maneuver are still given by

$$\begin{aligned}\dot{y} &= \dot{y} \\ \ddot{y} &= \ddot{y}_T - n_L \\ \ddot{y}_T &= \ddot{y}_T \\ \ddot{\ddot{y}}_T &= -\omega^2 \ddot{y}_T + \omega u_{s1}\end{aligned}$$

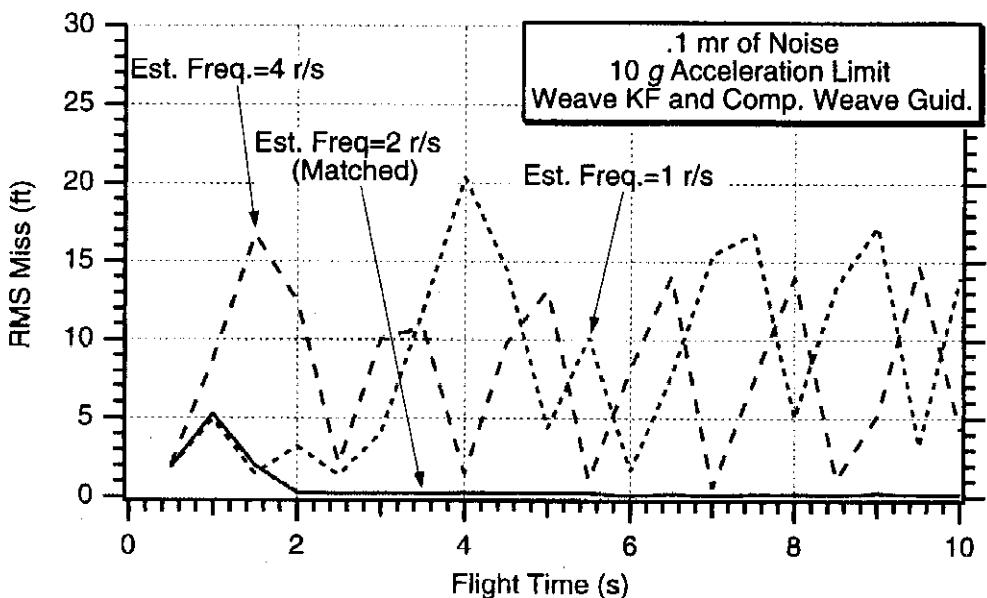


Fig. 26.17 Compensated weave guidance performance can be worse than optimal guidance if estimated weave frequency is in error.

We now need an additional equation that says something about the target weave frequency. If the target weave frequency is constant its derivative must be zero. However, for protection we can say that the derivative of the frequency is simply white noise or

$$\dot{\omega} = u_{s2}$$

In the preceding five scalar differential equations, u_{s1} and u_{s2} are white process noise sources. Uncertainty in when the target starts to maneuver is reflected in u_{s1} , while uncertainty in the fact that the weave frequency may not be a constant is reflected in u_{s2} . In the previous section we used for the spectral density of the first white process noise

$$\Phi_{s1} = \frac{\omega_{\text{EXP}}^2 n_{T\text{MAX}}^2}{t_F}$$

where ω_{EXP} can be interpreted as the maximum expected target weave frequency, $n_{T\text{MAX}}$ is the maximum target maneuver level, and t_F is the amount of homing time. For now we will simply treat the spectral density of the second white process noise Φ_{s2} as a fudge factor whose value will be determined by experiment.

Because the target weave frequency is a state, the preceding differential equations describing our model of the real world are nonlinear and the resultant filter will be an extended Kalman filter rather than a linear Kalman filter. However, the measurement equation for this model is still linear and turns out to be

$$y^* = [1 \ 0 \ 0 \ 0 \ 0] \begin{bmatrix} y \\ \dot{y} \\ \ddot{y}_T \\ \ddot{y}_T \\ \omega \end{bmatrix} + v$$

The systems dynamics matrix can be determined from the system state equations as a matrix of partial derivatives given by

$$F = \begin{bmatrix} \frac{\partial \dot{y}}{\partial y} & \frac{\partial \dot{y}}{\partial \dot{y}} & \frac{\partial \dot{y}}{\partial \ddot{y}_T} & \frac{\partial \dot{y}}{\partial \ddot{y}_T} & \frac{\partial \dot{y}}{\partial \omega} \\ \frac{\partial \ddot{y}}{\partial y} & \frac{\partial \ddot{y}}{\partial \dot{y}} & \frac{\partial \ddot{y}}{\partial \ddot{y}_T} & \frac{\partial \ddot{y}}{\partial \ddot{y}_T} & \frac{\partial \ddot{y}}{\partial \omega} \\ \frac{\partial \ddot{y}_T}{\partial y} & \frac{\partial \ddot{y}_T}{\partial \dot{y}} & \frac{\partial \ddot{y}_T}{\partial \ddot{y}_T} & \frac{\partial \ddot{y}_T}{\partial \ddot{y}_T} & \frac{\partial \ddot{y}_T}{\partial \omega} \\ \frac{\partial \ddot{y}_T}{\partial y} & \frac{\partial \ddot{y}_T}{\partial \dot{y}} & \frac{\partial \ddot{y}_T}{\partial \ddot{y}_T} & \frac{\partial \ddot{y}_T}{\partial \ddot{y}_T} & \frac{\partial \ddot{y}_T}{\partial \omega} \\ \frac{\partial \omega}{\partial y} & \frac{\partial \omega}{\partial \dot{y}} & \frac{\partial \omega}{\partial \ddot{y}_T} & \frac{\partial \omega}{\partial \ddot{y}_T} & \frac{\partial \omega}{\partial \omega} \end{bmatrix}$$

where the partial derivatives are evaluated at the current state estimates. After taking the appropriate partial derivatives, the systems dynamics matrix turns out

to be

$$\mathbf{F} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\hat{\omega}^2 & 0 & -2\hat{\omega}\hat{y}_T \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We will use a two-term Taylor series expansion to obtain the fundamental matrix. The number of terms used in the series approach is not critical because the fundamental matrix will only be used in the Riccati equations.² The approximate fundamental matrix turns out to be

$$\Phi_k \approx I + \mathbf{FT}_s = \begin{bmatrix} 1 & T_s & 0 & 0 & 0 \\ 0 & 1 & T_s & 0 & 0 \\ 0 & 0 & 1 & T_s & 0 \\ 0 & 0 & -\hat{\omega}^2 T_s & 1 & -2\hat{\omega}\hat{y}_T T_s \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

From our model of the real world, the continuous process noise matrix can be found from

$$\mathbf{Q} = E(\mathbf{ww}^T) = E \left[\begin{bmatrix} 0 \\ 0 \\ 0 \\ u_{s1} \\ u_{s2} \end{bmatrix} [0 \ 0 \ 0 \ u_{s1} \ u_{s2}] \right] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Phi_{s1} & 0 \\ 0 & 0 & 0 & 0 & \Phi_{s2} \end{bmatrix}$$

where the spectral densities Φ_{s1} and Φ_{s2} have been previously defined. The discrete process noise matrix can be obtained from the continuous process noise matrix according to

$$Q_k = \int_0^{T_s} \Phi(\tau) \mathbf{Q} \Phi^T(\tau) dt$$

After some algebra we obtain

$$Q_k = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Phi_{s1} \frac{T_s^3}{3} & \Phi_{s1} \frac{T_s^2}{2} & 0 \\ 0 & 0 & \Phi_{s1} \frac{T_s^2}{2} & \Phi_{s1} T_s + 4 \frac{T_s^3}{3} \hat{\omega}^2 \hat{y}_T \Phi_{s2} & -\hat{\omega} T_s^2 \hat{y}_T \Phi_{s2} \\ 0 & 0 & 0 & -\hat{\omega} T_s^2 \hat{y}_T \Phi_{s2} & T_s \Phi_{s2} \end{bmatrix}$$

Note that the elements of the discrete process noise matrix are also evaluated at the current state estimates. Finally, the equations for the extended Kalman filter

are simply

$$\begin{aligned}\hat{y}_k &= \bar{y}_k + K_{1_k}(y_k^* - \bar{y}_k) \\ \hat{\bar{y}}_k &= \bar{y}_k + K_{2_k}(y_k^* - \bar{y}_k) \\ \hat{\bar{y}}_{T_k} &= \bar{y}_{T_k} + K_{3_k}(y_k^* - \bar{y}_k) \\ \hat{\bar{y}}_{T_k} &= \bar{y}_{T_k} + K_{4_k}(y_k^* - \bar{y}_k) \\ \hat{\omega}_k &= \hat{\omega}_{k-1} + K_{5_k}(x_k^* - \bar{x}_k)\end{aligned}$$

where the barred quantities represent projections of the previous state estimates to the current time. Normally, the barred quantities would be obtained by multiplying the previous state estimates by the fundamental matrix to project the states ahead one sampling interval. However, because the fundamental matrix in this example is not exact (because it was obtained by using a two-term Taylor series approximation to a linearized systems dynamics matrix), it is better to use brute force to obtain the necessary projections of all the states. In this case we actually numerically integrate the nonlinear equations of motion forward one sampling interval. Euler integration is used with an integration step size that is much smaller than the sampling interval to accurately numerically integrate the state equations forward.

Listing 26.3 presents a simulation of the extended Kalman filter as part of the homing loop. We can see that subroutine PROJECT is used to propagate ahead the state estimates one sampling interval. It is important to note that the extended Kalman filter's estimate of the target weave frequency is intentionally initialized wrong to -1 r/s (i.e., WHIC=-1) rather than to 2 r/s (W=2). The incorrect initialization is used to ensure that the extended Kalman filter is robust to initialization errors.

The nominal case of Listing 26.3 was run in which there was 1 mr of measurement noise and the second process noise spectral density Φ_{s2} was set to zero. At first, it appears from Fig. 26.18 that the extended Kalman filter is unable to estimate the target weave frequency. However, a closer examination of Fig. 26.18 reveals that the estimated magnitude of the target weave frequency in the steady state is nearly correct (that is, 2.5 r/s rather than 2 r/s), but the sign is wrong. If the filter was initialized with a positive frequency, the sign would have been correct. Figures 26.19 and 26.20 reveal that the large amount of measurement noise causes the estimate of the target acceleration and jerk to be fairly bad, except at the end of the flight.

We can see from Fig. 26.21 that reducing the measurement noise by an order of magnitude to $.1$ mr improves the estimate of the magnitude of the target weave frequency. However, we are still unable to estimate the sign of the target weave frequency. Figures 26.22 and 26.23 now show that the estimates of the target acceleration and jerk are very good for most of the flight with the reduced measurement noise, even though the sign of the target weave frequency is in error. However, these estimates are not quite as good as those estimates that were obtained with the four-state linear weave Kalman filter when the target weave frequency was known precisely (see Figs. 26.11 and 26.12 for comparison).

Figure 26.24 shows that if we add a small amount of process noise (namely, $\Phi_{s2} = .1$) to the frequency state, our estimate of the target weave frequency magnitude improves. However, if we add too much process noise (namely, $\Phi_{s2} = 1$)

Listing 26.3 Extended Kalman filter

```

GLOBAL DEFINE
    INCLUDE 'quickdraw.inc'
END
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
REAL*8 PHI(5,5),P(5,5),M(5,5),PHIP(5,5),PHIPPHIT(5,5),GAIN(5,1)
REAL*8 Q(5,5),HMAT(1,5),HM(1,5),MHT(5,1)
REAL*8 PHIT(5,5),RMAT(1,1),HMHTR(1,1),HMHTRINV(1,1)
REAL*8 HMHT(1,1),HT(5,1),KH(5,5),IDN(5,5),IKH(5,5)
INTEGER ORDER,STEP,APN
LOGICAL QPERFECT
PHIS2=0.
XNT=96.6
W=2.
PHASEDEG=0.
SIGRIN=.0001
SIGGL=0.
SRN=0.
RA=21000.
WHIC=-1.
TS=.01
TF=10.
PHIS1=W*W*XNT*XNT/TF
QPERFECT=.FALSE.
VC=9000.
XNP=3.
XNCLIM=9999999.
APN=4
TAU=.5
HEDEG=0.
VM=3000.
QEKF=0
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
OPEN(2,STATUS='UNKNOWN',FILE='COVFIL')
PHASE=PHASEDEG/57.3
ORDER=5
X=W*TS
TGO=TF
T=0.
S=0.
Y=0.
YD=-XNT/W-VM*HEDEG/57.3
YTDD=XNT*SIN(W*T)
YTDDD=XNT*W*COS(W*T)

XNC=0.
XNL=0.
H=.001
HP=.001
TS2=TS*TS
TS3=TS2*TS

```

Listing 26.3 (Continued)

```

TS4=TS3*TS
TS5=TS4*TS
TS6=TS5*TS
TS7=TS6*TS
WH=WHIC
IF(QPERFECT)THEN
    YH=Y
    YDH=YD
    YTDDH=YTDD
    YTDDD=YTDDD
    WH=W
ELSE
    YH=0.
    YDH=0.
    YTDDH=0.
    YTDDD=0.
ENDIF
DO 1000 I=1,ORDER
DO 1000 J=1,ORDER
    PHI(I,J)=0.
    P(I,J)=0.
    Q(I,J)=0.
    IDN(I,J)=0.
1000 CONTINUE
    IDN(1,1)=1.
    IDN(2,2)=1.
    IDN(3,3)=1.
    IDN(4,4)=1.
    IDN(5,5)=1.
    RTM=VC*TF
    SIGNOISE=SQRT(SIGRIN**2+(SIGGL/RTM)**2+
        (SRN*RTM*RTM/(RA*RA))**2)
    YNOISE=SIGNOISE*RTM
    P(1,1)=YNOISE*YNOISE
    P(2,2)=(VM*20./57.3)**2
    P(3,3)=XNT*XNT
    P(4,4)=(W*XNT)**2
    P(5,5)=W**2
    DO 1100 I=1,ORDER
        HMAT(I,I)=0.
        HT(I,1)=0.
1100 CONTINUE
    HMAT(1,1)=1.
    HT(1,1)=1.
10 IF(T>(TF-.0001))GOTO 999
    YOLD=Y
    YDOLD=YD
    XNLOLD=XNL
    STEP=1
    GOTO 200
66 STEP=2

```

(Contd.)

Listing 26.3 (Continued)

```

Y=Y+H*YD
YD=YD+H*YDD
XNL=XNL+H*XNLD
T=T+H
GOTO 200
55 CONTINUE
Y=.5*(YOLD+Y+H*YD)
YD=.5*(YDOLD+YD+H*YDD)
XNL=.5*(XNLOLD+XNL+H*XNLD)
S=S+H
IF(S<(TS-.0001))GOTO 10
S=0.
YTDD=XNT*SIN(W*T)
YTDDD=XNT*W*COS(W*T)
PHI(1,1)=1.
PHI(1,2)=TS
PHI(2,2)=1.
PHI(2,3)=TS
PHI(3,3)=1.
PHI(3,4)=TS
PHI(4,3)=-WH*WH*TS
PHI(4,4)=1.
PHI(4,5)=-2.*WH*YTDDH*TS
PHI(5,5)=1.
Q(3,3)=PHIS1*TS*TS*TS/3.
Q(3,4)=PHIS1*TS*TS/2.
Q(4,3)=Q(3,4)
Q(4,4)=4.*WH*WH*YTDDH*YTDDH*PHIS2*TS*TS*TS/3.+PHIS1*TS
Q(4,5)=-WH*YTDDH*TS*TS*PHIS2
Q(5,4)=Q(4,5)
Q(5,5)=PHIS2*TS
CALL MATTRN(PHI,ORDER,ORDER,PHIT)
CALL MATMUL(PHI,ORDER,ORDER,P,ORDER,ORDER,PHIP)
CALL MATMUL(PHIP,ORDER,ORDER,PHIT,ORDER,ORDER,PHIPPHIT)
CALL MATADD(PHIPPHIT,ORDER,ORDER,Q,M)
CALL MATMUL(HMAT,1,ORDER,M,ORDER,ORDER,HM)
CALL MATMUL(HM,1,ORDER,HT,ORDER,1,HMHT)
RTM=VC*TGO
SIGNOISE=SQRT(SIGRIN**2+(SIGGL/RTM)**2+
(SRN*RTM*RTM/(RA*RA))**2)
YNOISE=SIGNOISE*RTM
RMAT(1,1)=YNOISE**2
CALL MATADD(HMHT,1,1,RMAT,HMHTR)
HMHTRINV(1,1)=1./HMHTR(1,1)
CALL MATMUL(M,ORDER,ORDER,HT,ORDER,1,MHT)
CALL MATMUL(MHT,ORDER,1,HMHTRINV,1,1,GAIN)
CALL MATMUL(GAIN,ORDER,1,HMAT,1,ORDER,KH)
CALL MATSUB(IDN,ORDER,ORDER,KH,IKH)
CALL MATMUL(IKH,ORDER,ORDER,M,ORDER,ORDER,P)
RTM=VC*TGO
XLAM=Y/RTM

```

(Contd.)

Listing 26.3 (Continued)

```

CALL GAUSS(XNOISE,SIGNOISE)
XLAMS=XLAM+XNOISE
CALL PROJECT(T,TS,YH,YDH,YTDDH,YTDDDH,YB,YDB,YTDB,
1           YTDDDB,HP,XNL,WH)
RES=RTM*XLAMS-YB
YH=YB+GAIN(1,1)*RES
YDH=YDB+GAIN(2,1)*RES
YTDDH=YTDB+GAIN(3,1)*RES
YTDDDH=YTDDDB+GAIN(4,1)*RES
WH=WH+GAIN(5,1)*RES
IF(APN.EQ.0)THEN
    XNC=XNP*(YH+YDH*TGO)/(TGO*TGO)
ELSEIF(APN.EQ.1)THEN
    XNC=XNP*(YH+YDH*TGO+.5*YTDDH*TGO*TGO)/(TGO*TGO)
ELSEIF(APN.EQ.2)THEN
    XS=TGO/TAU
    TOP=6.*XS*XS*(EXP(-XS)-1.+XS)
    BOT1=2.*XS*XS*XS+3.+6.*XS-6.*XS*XS
    BOT2=-12.*XS*EXP(-XS)-3.*EXP(-2.*XS)
    XNPP=TOP/(.0001+BOT1+BOT2)
    C1=XNPP/(TGO*TGO)
    C2=XNPP/TGO
    C3=.5*XNPP
    C4=-XNPP*(EXP(-XS)+XS-1.)/(XS*XS)
    XNC=C1*YH+C2*YDH+C3*YTDDH+C4*XNL
ELSEIF(APN.EQ.3)THEN
    XP=WH*TGO
    XNC=XNP*(YH+YDH*TGO)/(TGO*TGO)+  

1      XNP*YTDDH*(1.-COS(XP))/XP**2+  

1      XNP*YTDDDH*(XP-SIN(XP))/(XP*XP*WH)
ELSE
    XS=TGO/TAU
    TOP=6.*XS*XS*(EXP(-XS)-1.+XS)
    BOT1=2.*XS*XS*XS+3.+6.*XS-6.*XS*XS
    BOT2=-12.*XS*EXP(-XS)-3.*EXP(-2.*XS)
    XNPP=TOP/(.0001+BOT1+BOT2)
    C1=XNPP/(TGO*TGO)
    C2=XNPP/TGO
    C3=XNPP*(1.-COS(WH*TGO))/(WH*WH*TGO*TGO)
    C4=-XNPP*(EXP(-XS)+XS-1.)/(XS*XS)
    C5=XNPP*(WH*TGO-SIN(WH*TGO))/(WH*WH*WH*TGO*TGO)
    XNC=C1*YH+C2*YDH+C3*YTDDH+C4*XNL+C5*YTDDDH
ENDIF
IF(XNC>XNCLIM)XNC=XNCLIM
IF(XNC<-XNCLIM)XNC=-XNCLIM
ERRYTDD=YTDD-YTDDH
ERRYTDDG=ERRYTDD/32.2
ERRYTDDD=YTDDD-YTDDDH
ERRYTDDDG=ERRYTDDD/32.2
SP55=SQRT(P(5,5))

```

(Contd.)

Listing 26.3 (Continued)

```

SP55P=-SP55
ERRW=W-WH
SP44=SQRT(P(4,4))
SP44P=-SP44
SP33=SQRT(P(3,3))
SP33P=-SP33
SP33G=SP33/32.2
SP33PG=SP33P/32.2
SP44G=SP44/32.2
SP44PG=SP44P/32.2
YTDDG=YTDD/32.2
YTDDHG=YTDDH/32.2
YTDDD=YTDDD/32.2
YTDDDHG=YTDDDH/32.2
XNCG=XNC/32.2
WRITE(9,*)T,YTDDG,YTDDHG,YTDDD,G,YTDDDH,G,W,WH
WRITE(1,*)T,YTDDG,YTDDHG,YTDDD,G,YTDDDH,G,W,WH
WRITE(2,*)T,ERRYTDD,SP33,-SP33,ERRYTDDD,SP44,-
1           SP44,ERRW,SP55,-SP55
GOTO 10
200 CONTINUE
YTDD=XNT*SIN(W*T)
TGO=TF-T+.00001
XNLD=(XNC-XNL)/TAU
YDD=YTDD-XNL
IF(STEP-1)66,66,55
999 CONTINUE
WRITE(9,*)Y
CLOSE(1)
CLOSE(2)
PAUSE
END

SUBROUTINE PROJECT(TP,TS,YPH,YDPH,YTDDPH,YTDDDPH,YB,YDB,
1           YTDBB,YTDDDB,HP,XNLP,WPH)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
T=0.
Y=YPH
YD=YDPH
YTDD=YTDDPH
YTDDD=YTDDDPH
W=WPH
XNL=XNLP
H=HP
10 IF(T>(TS-.0001))GOTO 999
YTDDDD=-W*W*YTDD
YTDDD=YTDDD+H*YTDDDD
YTDD=YTDD+H*YTDD
YDD=YTDD-XNL

```

(Contd.)

Listing 26.3 (Continued)

```

YD=YD+H*YDD
Y=Y+H*YD
T=T+H
GOTO 10
999 CONTINUE
YB=Y
YDB=YD
YTDDDB=YTDD
YTDDDDB=YTDDD
RETURN
END

SUBROUTINE GAUSS(X,SIG)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER*2 Random
INTEGER SUM
SUM=0
DO 14 J=1,6
IRAN=Random()
SUM=SUM+IRAN
14 CONTINUE
X=SUM/65536.
X=1.414*X*SIG
RETURN
END

SUBROUTINE MATTRN(A,IROW,ICOL,AT)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
REAL*8 A(IROW,ICOL),AT(ICOL,IROW)
DO 105 I=1,IROW
DO 105 J=1,ICOL
AT(J,I)=A(I,J)
105 CONTINUE
RETURN
END

SUBROUTINE MATMUL(A,IROW,ICOL,B,JROW,JCOL,C)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
REAL*8 A(IROW,ICOL),B(JROW,JCOL),C(IROW,JCOL)
DO 110 I=1,IROW
DO 110 J=1,JCOL
C(I,J)=0.
DO 110 K=1,ICOL
C(I,J)=C(I,J)+A(I,K)*B(K,J)
110 CONTINUE
RETURN
END

SUBROUTINE MATADD(A,IROW,ICOL,B,C)

```

Listing 26.3 (Continued)

```

IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
REAL*8 A(IROW,ICOL),B(IROW,ICOL),C(IROW,ICOL)
DO 120 I=1,IROW
DO 120 J=1,ICOL
      C(I,J)=A(I,J)+B(I,J)
120 CONTINUE
RETURN
END

SUBROUTINE MATSUB(A,IROW,ICOL,B,C)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
REAL*8 A(IROW,ICOL),B(IROW,ICOL),C(IROW,ICOL)
DO 120 I=1,IROW
DO 120 J=1,ICOL
      C(I,J)=A(I,J)-B(I,J)
120 CONTINUE
RETURN
END

```

to the frequency state, Fig. 26.25 shows that our estimate of the target weave frequency can actually diverge. Therefore, for safety reasons we will simply keep the second process noise source at zero (namely, $\Phi_{s2} = 0$) for future experiments.

To demonstrate the robustness of the five-state extended Kalman filter, another experiment was conducted. This time a case was considered in which the target maneuvered, but not sinusoidally. Instead, the target performed a constant $3 g$

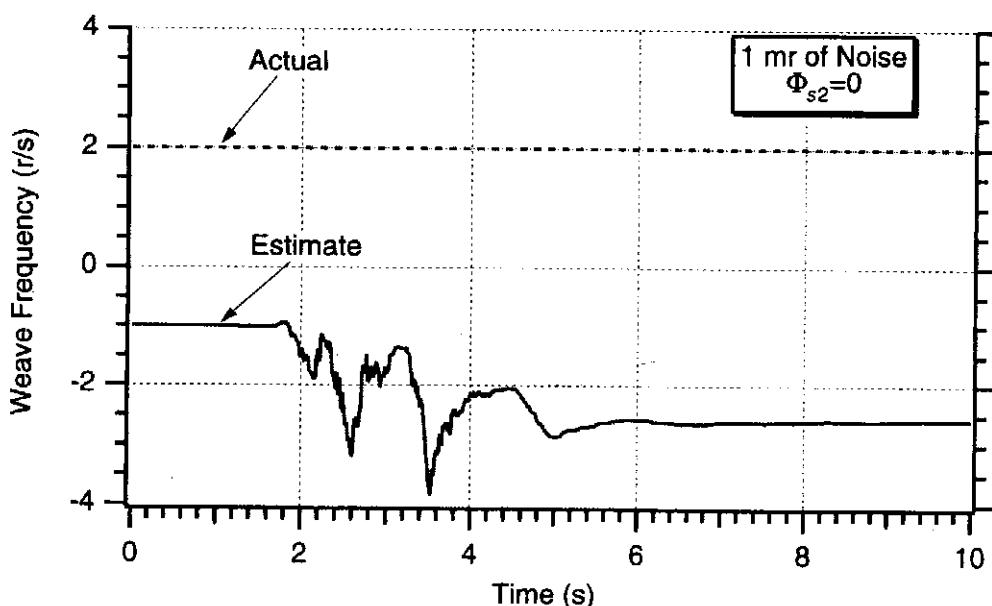


Fig. 26.18 Extended Kalman filter is able to estimate target weave frequency magnitude but not the sign.

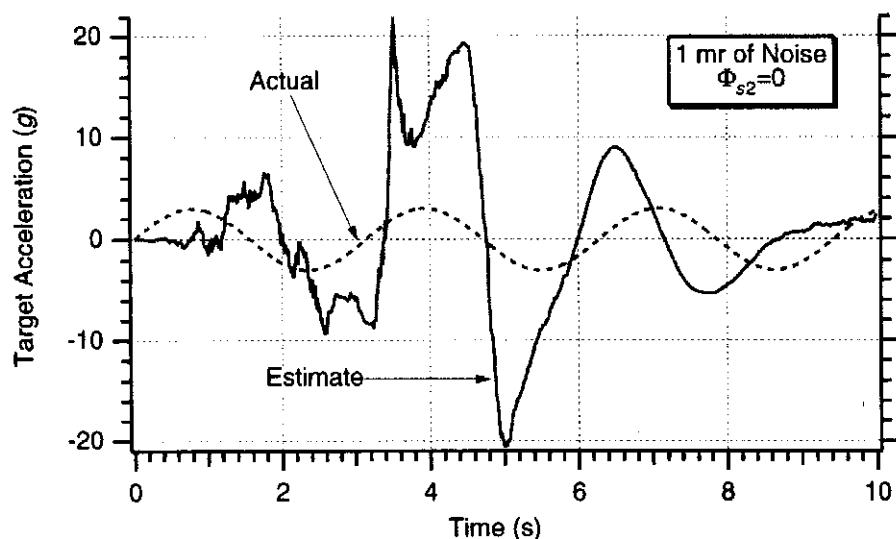


Fig. 26.19 Except for the end of flight, large amount of measurement noise causes bad estimates of target acceleration.

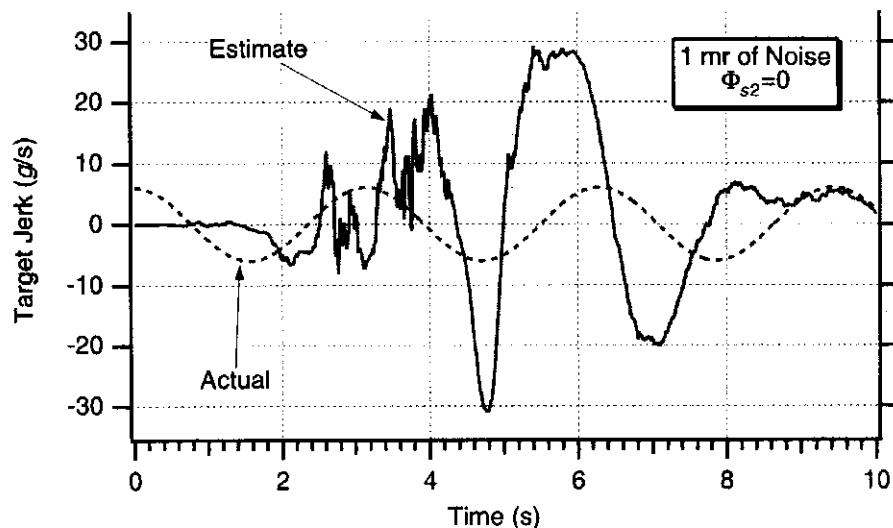


Fig. 26.20 Except for the end of flight, large amount of measurement noise causes bad estimate of target jerk.

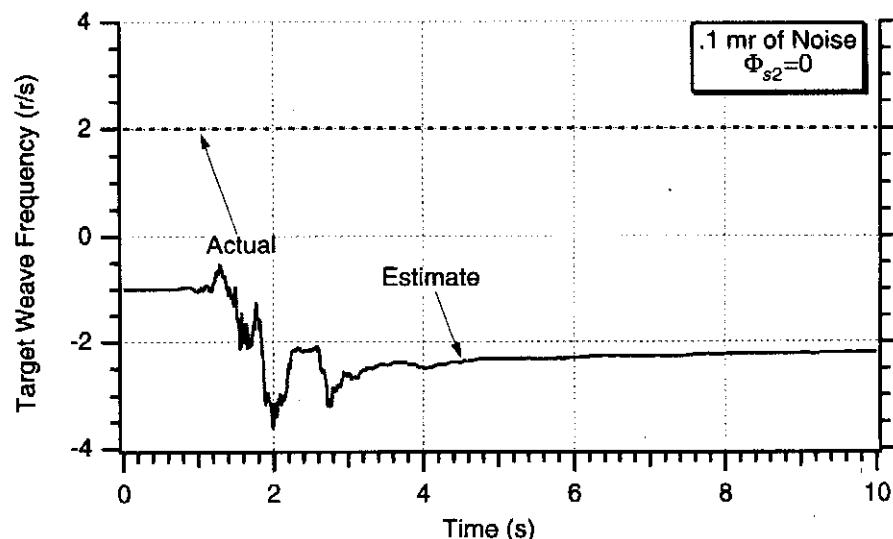


Fig. 26.21 Reducing the measurement noise improves the extended Kalman filter's estimate of the magnitude of the target weave frequency.

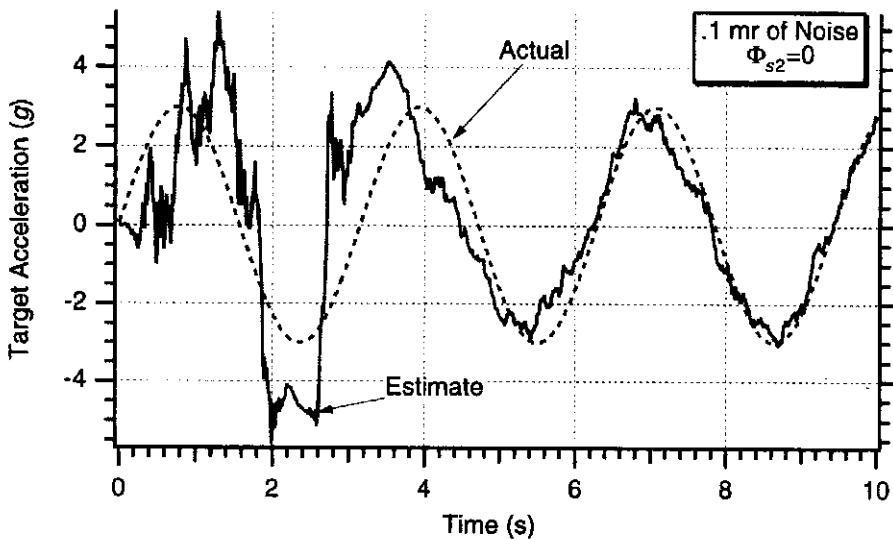


Fig. 26.22 Extended Kalman filter's estimate of target acceleration is fairly good after a transient period when measurement noise is reduced by an order of magnitude.

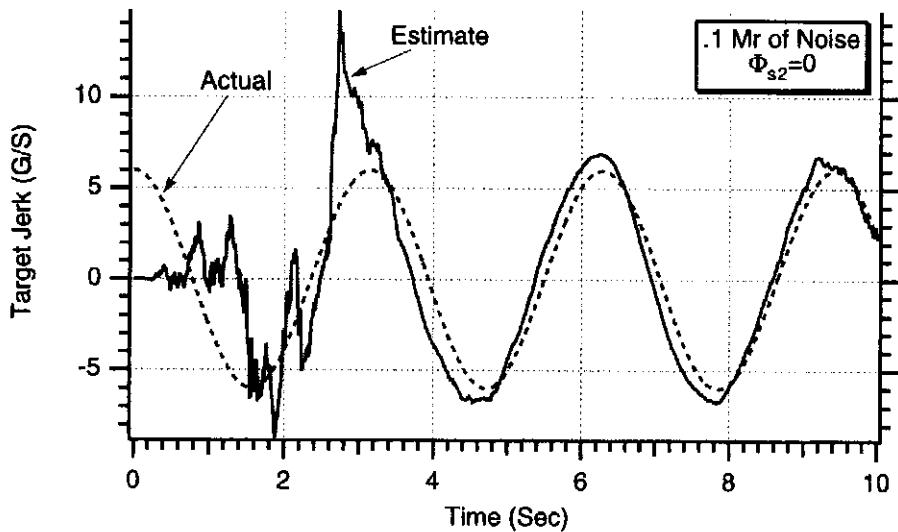


Fig. 26.23 Extended Kalman filter's estimate of target jerk is also fairly good after a transient period when measurement noise is reduced an order of magnitude.

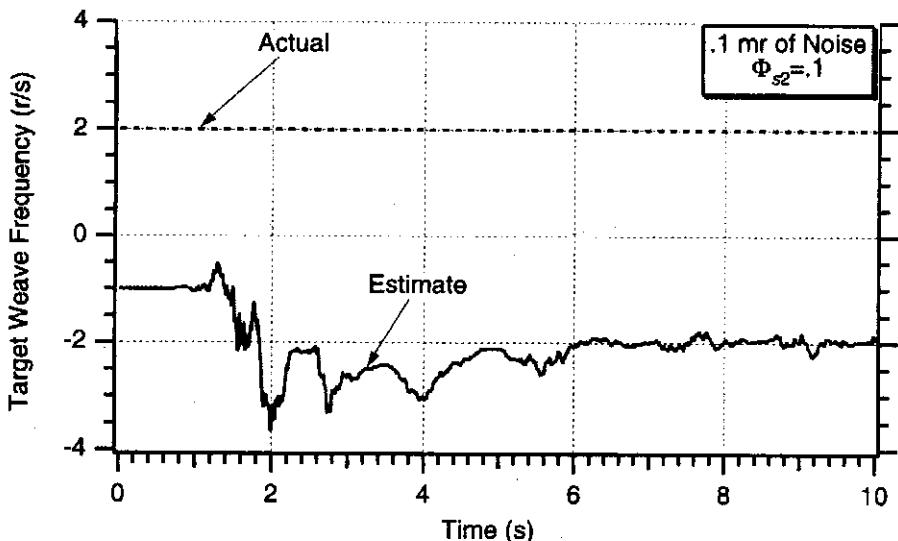


Fig. 26.24 Adding small amount of process noise to frequency state slightly improves frequency estimate.

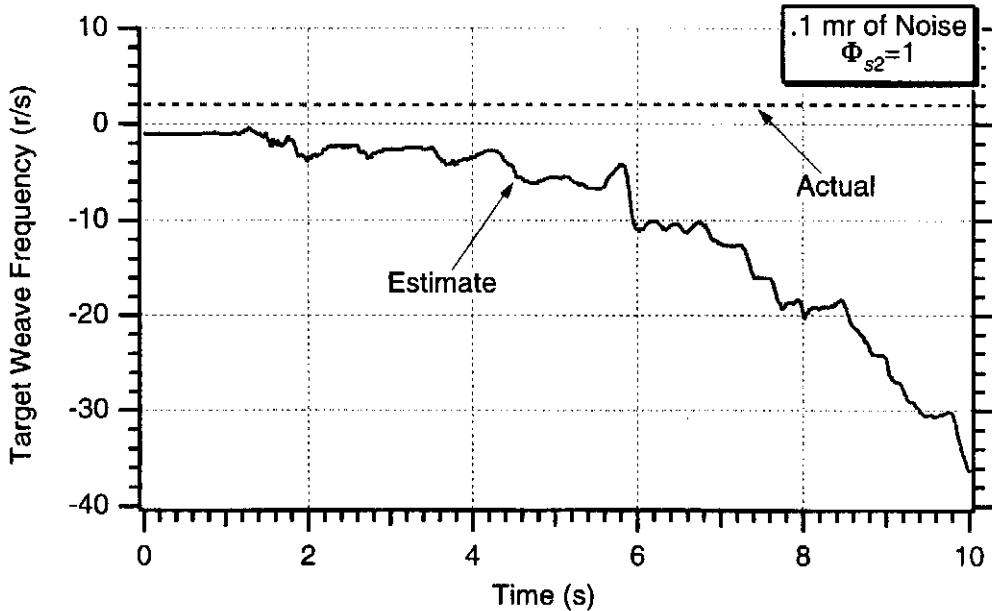


Fig. 26.25 Adding too much process noise to frequency state causes frequency estimate to diverge.

maneuver. Figure 26.26 shows that in this case the extended Kalman filter correctly estimated, after an initial transient period, that the target weave frequency was zero. Figure 26.27 shows that after 4 s the filter is able to estimate that the level of the constant target maneuver is 3 g. Finally, Fig. 26.28 shows that the estimate of the target jerk is zero. This should be the case when the target maneuver is constant. Thus, we can conclude that the five-state extended Kalman filter is indeed robust.

Miss distance experiments were conducted with the three-state linear Kalman filter and the five-state extended Kalman filter when both were part of the homing loop. The guidance law options for the linear Kalman filter were proportional navigation and optimal guidance. The extended Kalman filter was always used

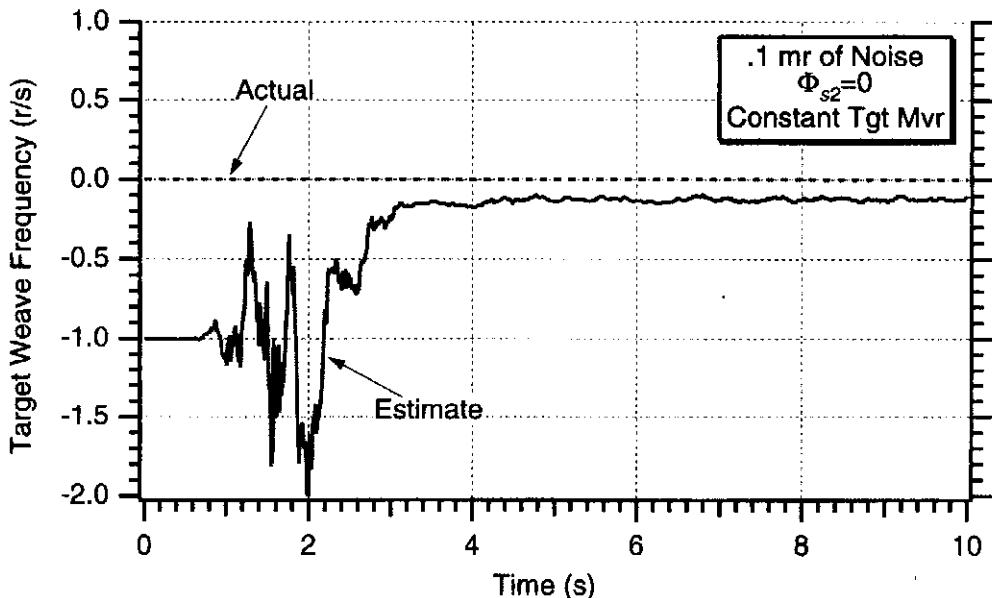


Fig. 26.26 Filter correctly estimates weave frequency of zero in presence of constant target maneuver.

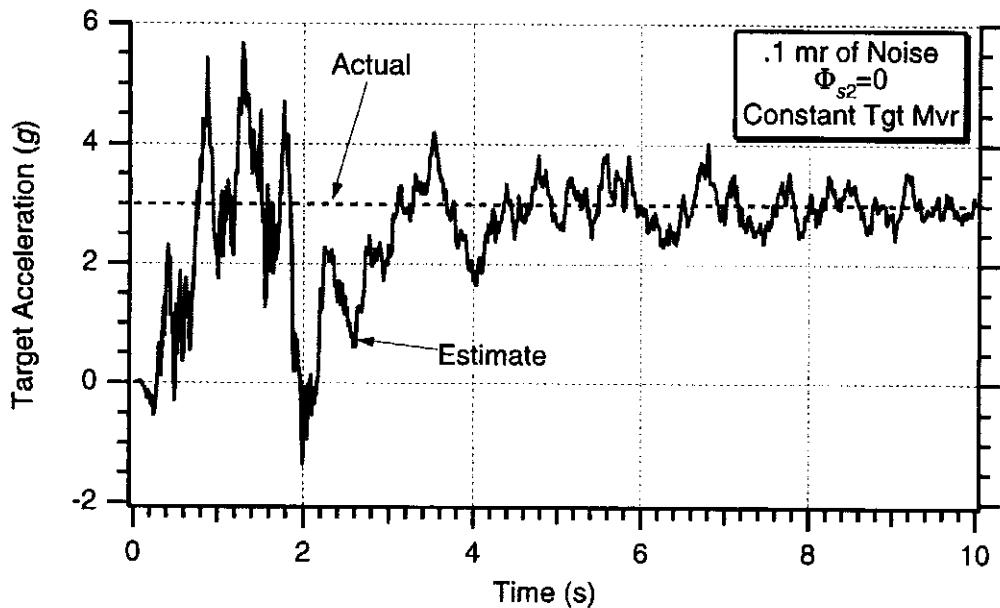


Fig. 26.27 After initial transient period extended Kalman filter is able to estimate constant target maneuver.

in conjunction with the compensated weave guidance law. A close examination of this guidance law reveals that the sign of the target weave frequency does not influence the guidance command. Therefore, the inability of the five-state extended Kalman filter to correctly determine the sign of the target weave frequency should not be important for guidance purposes.

Figure 26.29 shows that when there are no constraints on the missile acceleration and there is 1 mr of measurement noise, both the three-state linear Kalman filter with optimal guidance and five-state extended Kalman filter with compensated weave guidance yield superior miss distance performance to that of a proportional navigation guidance system. Both optimal guidance and compensated weave

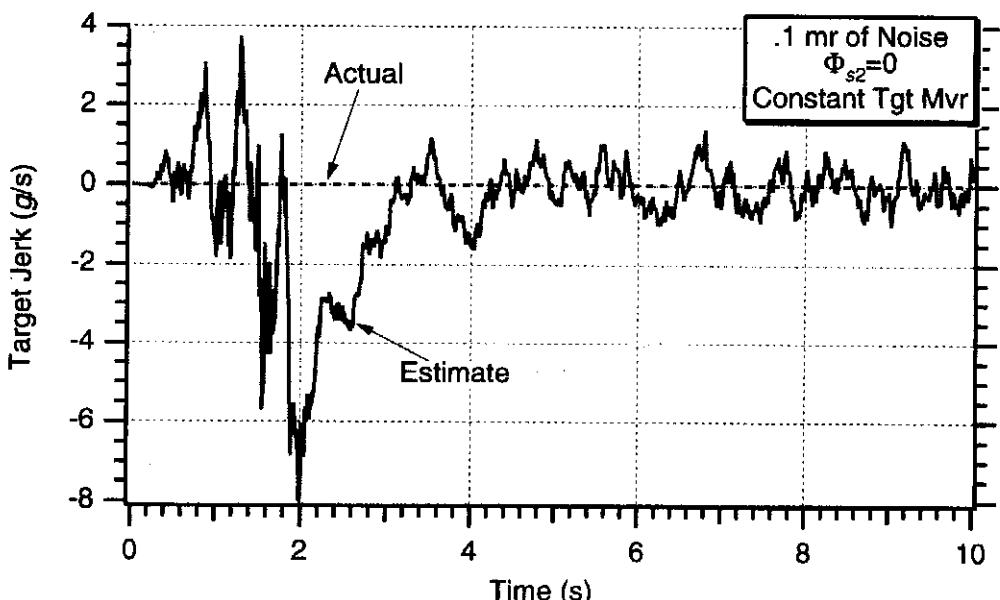


Fig. 26.28 Extended Kalman filter is able to correctly estimate that constant target maneuver has no jerk term.

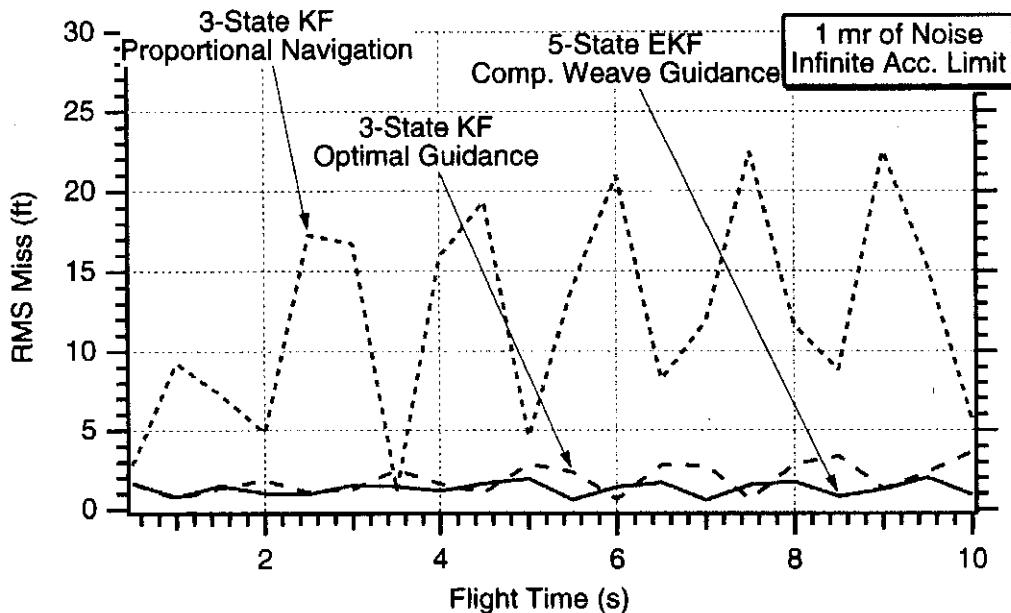


Fig. 26.29 Both compensate weave guidance and optimal guidance yield similar performance when there are no missile acceleration constraints.

guidance yield similar performance as measured by the RMS miss distance. However, we can see from Fig. 26.30 that if there is a 10 g missile acceleration limit, optimal guidance degrades severely and the best performance is obtained by the five-state extended Kalman filter with compensated weave guidance. In this case, the performance of optimal guidance and proportional navigation are similar and both yield much larger RMS miss distances when compared with the compensated weave guidance law.

Reducing the measurement noise by an order of magnitude to .1 mr improves the performance of all the guidance systems. When there is no constraint on the

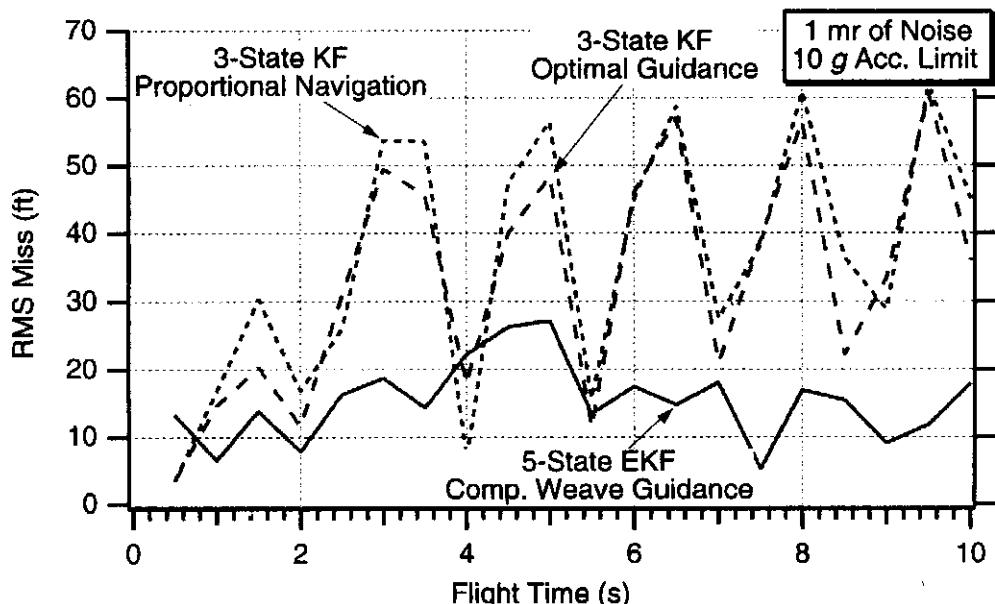


Fig. 26.30 With missile acceleration constraints, five-state extended Kalman filter with compensated weave guidance yields the best results.

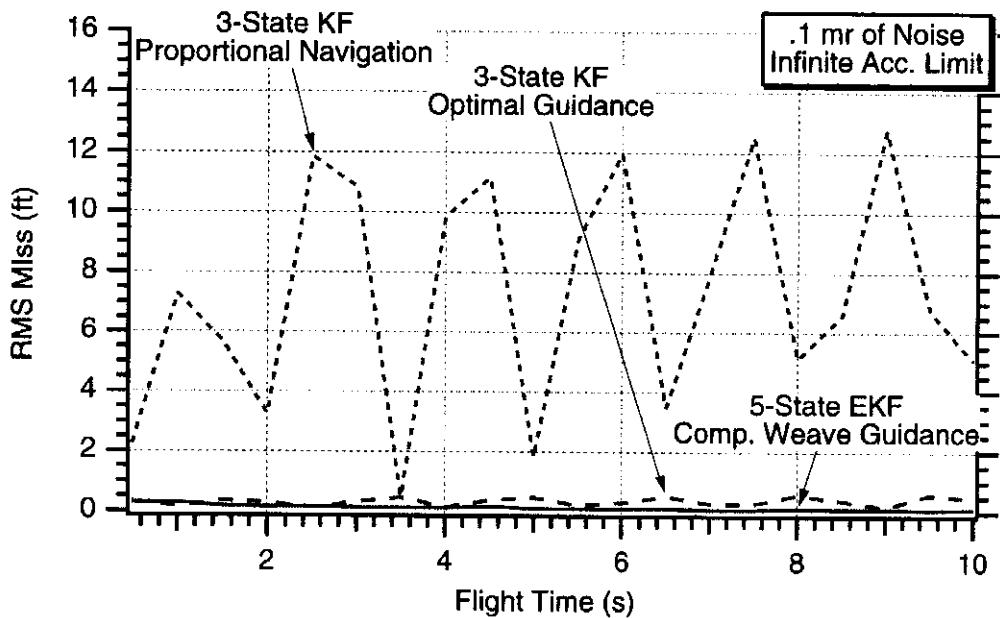


Fig. 26.31 With reduced measurement noise, both compensated weave guidance and optimal guidance still yield similar performance when there are no missile acceleration constraints.

available missile acceleration Fig. 26.31 again shows that optimal guidance and compensated weave guidance yield much better performance than a proportional navigation guidance system. We can also see from Fig. 26.31 that the performances of both optimal guidance and compensated weave guidance are virtually identical. However, we can see from Fig. 26.32 that the performance of the optimal guidance system degrades when there is a 10 g missile acceleration limit, and the performance of the extended Kalman filter with the compensated weave guidance law is much better. The trends are identical to the case in which the measurement noise was an order of magnitude larger.

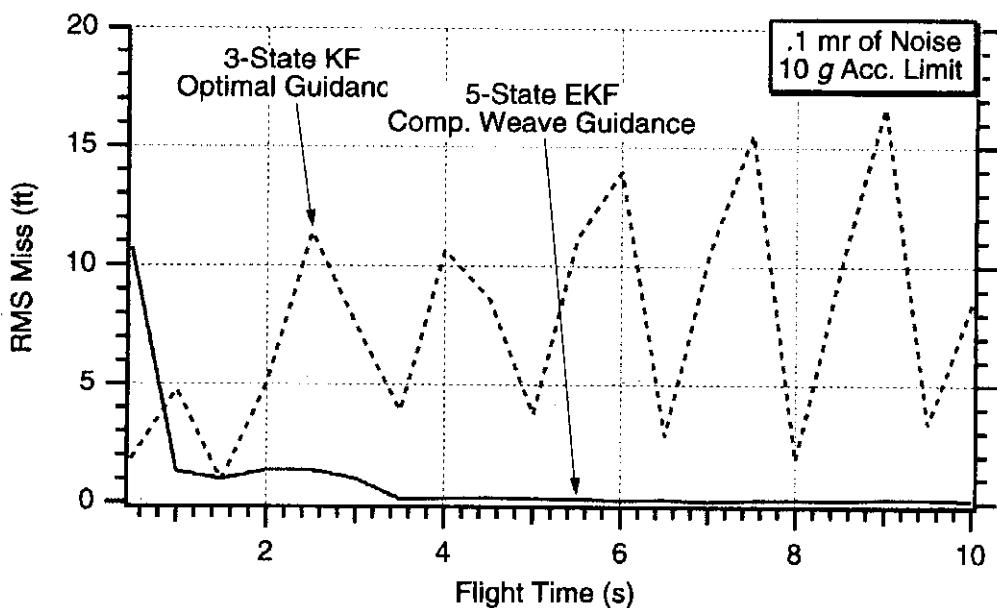


Fig. 26.32 With missile acceleration constraints, five-state extended Kalman filter with compensated weave guidance yields the best results when measurement noise is reduced.

Summary

This chapter shows various filtering options that can be used in conjunction with advance guidance laws to improve the performance of missile guidance system against weaving targets. A conventional linear three-state Kalman filter can be used in conjunction with the optimal guidance law to yield significant performance improvements compared with proportional navigation. Similar performance improvements can also be obtained with a four-state weave Kalman filter in conjunction with the compensated weave guidance law if the target weave frequency is known. This filtering and guidance approach will perform better than optimal guidance when there is a low missile-to-target acceleration advantage. If the target weave frequency is not known it can be estimated with a five-state extended Kalman filter. Similar performance improvements can be obtained with this nonlinear filter and guidance law approach.

References

- ¹Zarchan, P., "Tracking and Intercepting Spiraling Ballistic Missiles," *Proceedings of IEEE PLANS Conference*, San Diego, CA, March 2000.
- ²Zarchan, P., and Musoff, H., *Fundamentals of Kalman Filtering: A Practical Approach*, Progress in Astronautics and Aeronautics, AIAA, Reston, VA, 2000.

Predictor–Corrector Guidance

Introduction

So far, most of the guidance methods we have studied in this text are variants of proportional navigation in which the guidance commands are proportional to the zero effort miss and inversely proportional to the square of time to go. We have seen that guidance laws of this type are very effective and fairly robust. The space community often uses predictor–corrector guidance to achieve intercepts or near-intercepts for their applications. In this chapter we will first see how the predictor–corrector method can be applied to a problem in which other, more conventional guidance laws are also applicable. We will compare all of the guidance laws in terms of performance and robustness. Next, we will select a problem in which the direct application of a proportional navigation type guidance law is not possible. In this application we will see how predictor–corrector guidance can be applied and how it performs. We will also see how robust the predictor–corrector guidance law is to errors in knowledge of acceleration.

Surface-to-Surface Missile Problem

Let us return to the problem of an impulsively launched surface-to-surface missile that is acted on by only drag and gravity as was originally modeled in Listing 11.1. In that simulation, the missile was not guided and we simply studied where the missile would impact the Earth for different launch or flight path angles. In this application we will try to intercept a stationary target located on the surface of the Earth. The only accelerations acting on the missile are drag and gravity; these are displayed in Fig. 27.1.

In the model of Fig. 27.1, the “1” axis is downrange, while the “2” axis is altitude. The downrange and altitude components of the accelerations acting on the missile can be obtained by inspection of Fig. 27.1 and are given by

$$a_{M1} = -\text{Drag} \cos \gamma$$

$$a_{M2} = -\text{Drag} \sin \gamma - g$$

where the drag acceleration can be expressed in terms of gravity g , the dynamic pressure, Q , and the ballistic coefficient, β , as

$$\text{Drag} = \frac{g Q}{\beta}$$

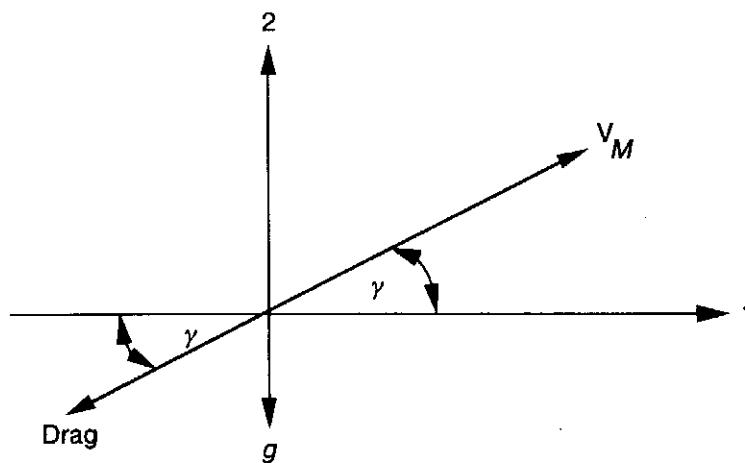


Fig. 27.1 Accelerations acting on impulsively launched missile.

The dynamic pressure can be expressed in terms of the air density, ρ , and missile velocity, V_M , as

$$Q = .5\rho V_M^2$$

Recall that in Chapter 11 we used an exponential approximation for the air density. Below 30,000 ft we used

$$\rho = .002378 e^{-R_{M2}/30000} \quad (R_{M2} < 30,000 \text{ ft})$$

and above 30,000 ft we used

$$\rho = .0034 e^{-R_{M2}/22000} \quad (R_{M2} \geq 30,000 \text{ ft})$$

where the altitude R_{M2} was measured in units of feet and the air density had units of slug/ft³. Because the altitude and downrange missile acceleration components are in an inertial coordinate system, they can be integrated twice to yield the location of the missile. For a proportional navigation guided missile, the acceleration command obtained from the guidance law is assumed to be perpendicular to the line of sight, as shown in Fig. 27.2.

With proportional navigation guidance, the guidance command is simply

$$n_c = N' V_c \dot{\lambda}$$

The component of missile drag perpendicular to the line of sight will appear to the missile as a target maneuver. For a missile with small acceleration capability, excessive acceleration requirements may result from the apparent target maneuver because of the missile's own drag. We can compensate for missile drag in the

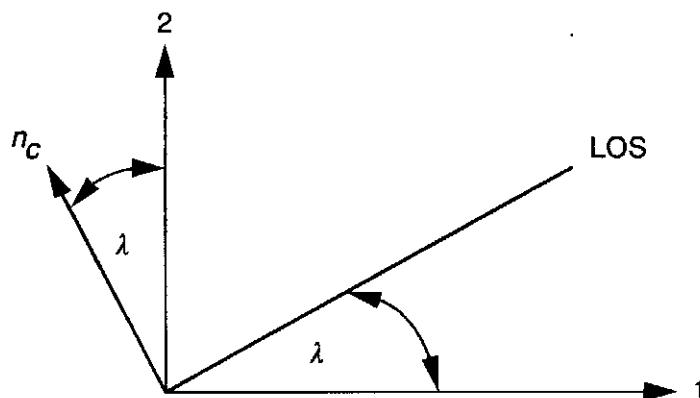


Fig. 27.2 Guidance command is perpendicular to line of sight.

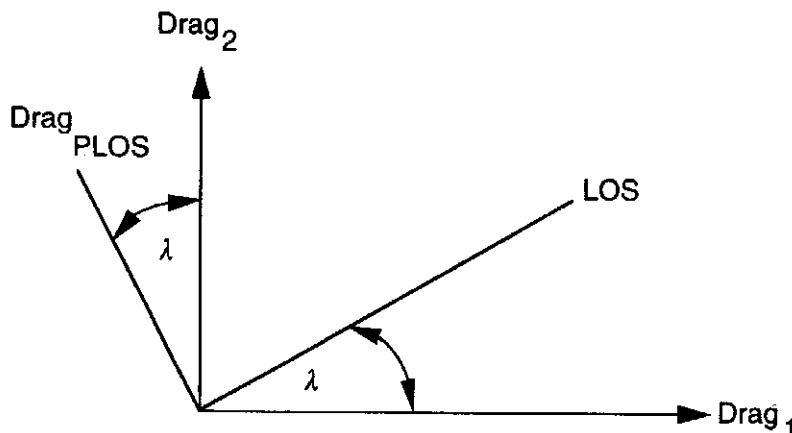


Fig. 27.3 Calculating the component of drag perpendicular to the line of sight.

guidance law by using a form of augmented proportional navigation, or

$$n_c = N' V_c \dot{\lambda} + .5N'(a_{TPLOS} - \text{Drag}_{PLOS})$$

where a_{TPLOS} is the target acceleration perpendicular to the line of sight and Drag_{PLOS} is the component of the missile drag acceleration that is perpendicular to the line of sight. Because in this application the target is stationary, a_{TPLOS} is zero. To calculate the component of drag and gravity perpendicular to the line of sight we first observe from Fig. 27.1 that the components of drag and gravity in the downrange and altitude directions are given by

$$\text{Drag}_1 = -\text{Drag} \cos \gamma$$

$$\text{Drag}_2 = -\text{Drag} \sin \gamma - g$$

From Fig. 27.3 we can see how the acceleration of drag perpendicular to the line of sight is related to the downrange–altitude coordinate system.

Therefore, from Fig. 27.3 we can see that the component of drag perpendicular to the line of sight is given by

$$\text{Drag}_{PLOS} = -\text{Drag}_1 \sin \lambda + \text{Drag}_2 \cos \lambda$$

An engagement simulation of the impulsively launched missile and stationary target appears in Listing 27.1. Initially there is no guidance, and the missile is launched at 3000 ft/s with an initial flight path angle of 45 deg. For the first 30 s of flight, only drag and gravity act on the missile. However, after 30 s of flight guidance is turned on and the missile can either use proportional navigation (APN=0) or augmented proportional navigation (APN=1) to guide toward the stationary target located at downrange and altitude coordinates of RT1 and RT2. Because the target is located on the surface of the Earth, RT2 is zero. The simulation has a 5 g acceleration limit on the acceleration commands (XLIMG=5). Notice that the simulation stops when the missile hits the ground rather than when the closing velocity goes negative (missile and target are at point of closest approach). This minor adjustment to our normal stopping condition (that is, normally we stop engagement simulations when the closing velocity goes negative because this is the point of closest approach) will prevent the missile from traveling through the Earth to reach the target.

Cases were run with the engagement simulation of Listing 27.1 in which the target downrange location RT1 was made a parameter. We can see from Fig. 27.4 that when proportional navigation guidance is used the missile can hit the stationary

Listing 27.1 Missile-target engagement simulation with conventional guidance

```

IMPLICIT REAL*8(A-H,O-Z)
INTEGER STEP,APN
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
TS=.1
APN=0
XLIMG=5.
VM=3000.
BETA=1000.
T=0.
S=0.
GAMDEG=45.
VM1=VM*COS(GAMDEG/57.3)
VM2=VM*SIN(GAMDEG/57.3)
RM1=0.
RM2=0.
XNC=0.
RT1=60000.
RT2=0.
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=0.-VM1
VTM2=0.-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
XLIM=XLIMG*32.2
XNC=0.
10 IF(T>0..AND.RM2<=0.)GOTO 999
IF(RTM<1000.)THEN
    H=.0001
ELSE
    H=.01
ENDIF
RM1OLD=RM1
RM2OLD=RM2
VM1OLD=VM1
VM2OLD=VM2
STEP=1
GOTO 200
66 STEP=2
RM1=RM1+H*VM1
RM2=RM2+H*VM2
VM1=VM1+H*AM1
VM2=VM2+H*AM2
T=T+H
GOTO 200
55 CONTINUE
RM1=.5*(RM1OLD+RM1+H*VM1)
RM2=.5*(RM2OLD+RM2+H*VM2)
VM1=.5*(VM1OLD+VM1+H*AM1)
VM2=.5*(VM2OLD+VM2+H*AM2)

```

(Contd.)

Listing 27.1 (Continued)

```

S=S+H
IF(S<(TS-.00001))GOTO 10
S=0.
RM1K=RM1/1000.
RM2K=RM2/1000.
RT1K=RT1/1000.
RT2K=RT2/1000.
DRAG1=-DRAG*COS(GAM)
DRAG2=-DRAG*SIN(GAM)-32.2
DRAGPLOS=-DRAG1*SIN(XLAM)+DRAG2*COS(XLAM)
ATPLOS=0.
IF(T>30.)THEN
    IF(APN.EQ.0)THEN
        XNC=3.*VC*XLAMD
    ELSE
        XNC=3.*VC*XLAMD+1.5*(ATPLOS-DRAGPLOS)
    ENDIF
ELSE
    XNC=0.
ENDIF
IF(XNC>XLIM)XNC=XLIM
IF(XNC<-XLIM)XNC=-XLIM
XNCG=XNC/32.2
DRAGPLOSG=DRAGPLOS/32.2
WRITE(9,*)T,RM1K,RM2K,RT1K,RT2K,XNCG
WRITE(1,*)T,RM1K,RM2K,RT1K,RT2K,XNCG
GOTO 10
200 CONTINUE
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=0.-VM1
VTM2=0.-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
XLAM=ATAN2(RTM2,RTM1)
XLAMD=(RTM1*VTM2-RTM2*VTM1)/(RTM*RTM)
IF(RM2<30000.)THEN
    RHO=.002378*EXP(-RM2/30000)
ELSE
    RHO=.0034*EXP(-RM2/22000)
ENDIF
VM=SQRT(VM1**2+VM2**2)
Q=.5*RHO*VM*VM
GAM=ATAN2(VM2,VM1)
DRAG=Q*32.2/BETA
XNE1=-XNC*SIN(XLAM)
XNE2=XNC*COS(XLAM)
AM1=-DRAG*COS(GAM)+XNE1
AM2=-32.2-DRAG*SIN(GAM)+XNE2
IF(STEP-1)66,66,55

```

(Contd.)

Listing 27.1 (Continued)

```

999  CONTINUE
      RM1K=RM1/1000.
      RM2K=RM2/1000.
      RT1K=RT1/1000.
      RT2K=RT2/1000.
      XNCG=XNC/32.2
      DRAGPLOSSG=DRAGPLOS/32.2
      WRITE(9,*)T,RM1K,RM2K,RT1K,RT2K,XNCG
      WRITE(1,*)T,RM1K,RM2K,RT1K,RT2K,XNCG
      WRITE(9,*)RTM
      PAUSE
      CLOSE(1)
      END

```

target when it is located from 40 to 60 kft downrange of the missile launch point. However, if the target is 70 kft downrange of the missile launch point the missile will hit the ground too soon and will miss the target when proportional navigation guidance is used. Figure 27.5 shows that the lack of success in the 70 kft case was not because of acceleration saturation because the acceleration level never reached 5 g. In this case, proportional navigation does not realize that the missile cannot travel through the ground to reach the target. Cases were run in which the drag was reduced and then the missile was able to hit the target at the 70-kft range and beyond. Clearly the drag is causing problems for proportional navigation.

As was mentioned previously, the drag component that is perpendicular to the line of sight appears as a target maneuver to the missile and causes potential miss distance problems. Because we should know our own drag it seems reasonable that we can use augmented proportional navigation (APN=1) as was previously described in this section. When augmented proportional navigation was used, we can see from Fig. 27.6 that now we can hit targets at 70 kft and beyond. Thus, we can see that in this application more advanced guidance laws can expand the zone of effectiveness of a missile against stationary targets. Figure 27.7 shows that

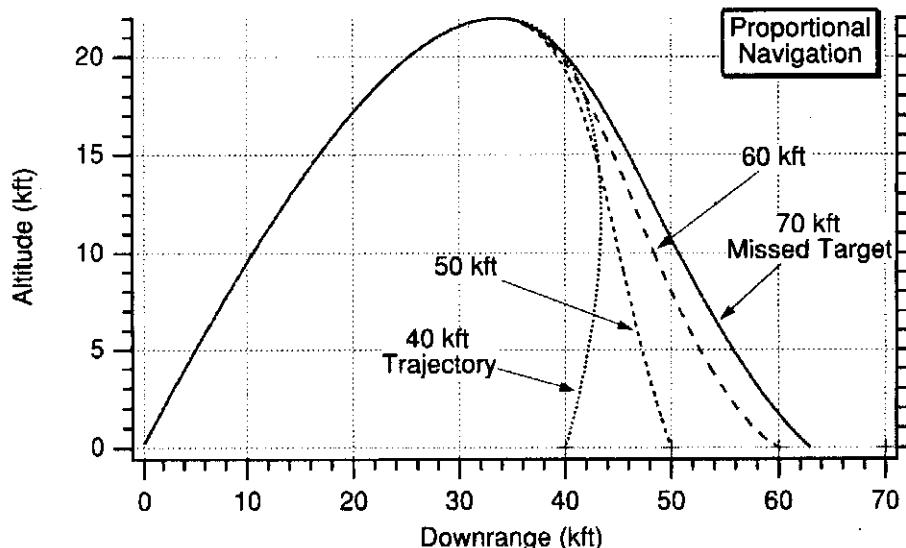


Fig. 27.4 With proportional navigation we can hit stationary targets out to 60 kft downrange.

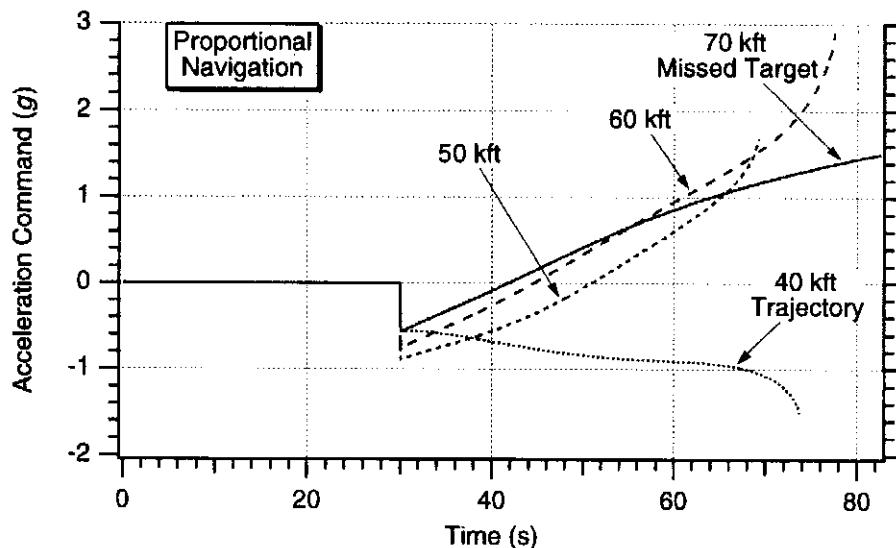


Fig. 27.5 We missed at 70 kft because we hit the ground first and not because we acceleration saturated.

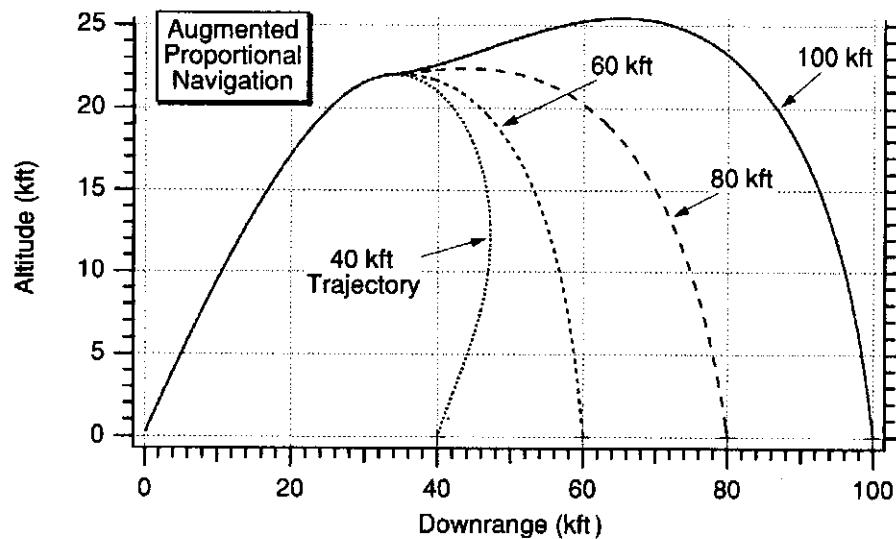


Fig. 27.6 Augmented proportional navigation extends zone of effectiveness of missile.

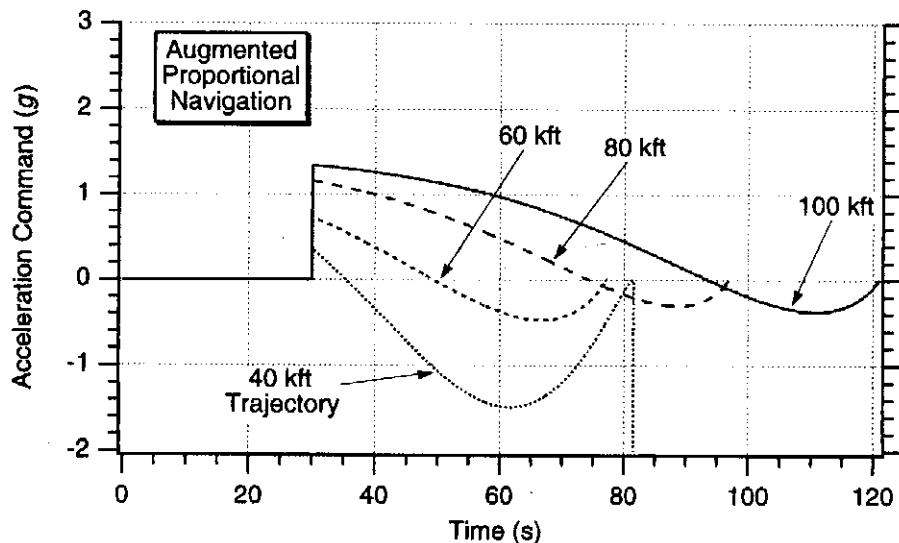


Fig. 27.7 All acceleration profiles are well behaved and do not saturate with augmented proportional navigation.

the missile acceleration profiles for all trajectories considered are very reasonable. The acceleration requirements are always well below the simulation limit of 5 g.

Predictor-Corrector Guidance With Surface-to-Surface Intercept Problem

Predictor-corrector guidance is based on having a model of the interceptor and target and trying to determine at each guidance update how certain changes in important parameters influence the missile trajectory. With predictor-corrector guidance we must first assume a form of the acceleration command in this application. For simplicity, the form of the acceleration command we will assume is a constant. Using the example of the previous section in conjunction with predictor-corrector guidance we first integrate the missile equations of motion forward assuming no acceleration commands (that is, our nominal constant is zero) are issued to see where the missile impacts the ground. We then measure the difference between where the target really is, R_{T1} , and where the missile actually hits the ground, x_1 , and call the resultant quantity Δx or

$$\Delta x = R_{T1} - x_1$$

We now add a little bit to the previous constant acceleration command n_c (which is zero the first time we initiate this procedure) and call the new constant acceleration command n'_c . We again integrate the missile equations of motion forward, assuming the new constant acceleration command, and calling the place where the missile impacts the ground x_2 . Thus, we can manually construct a partial derivative for the change in impact point with respect to the change in acceleration command by differencing the two impact points and dividing by the difference in the two constant acceleration commands or

$$\frac{\partial x}{\partial n_c} = \frac{x_2 - x_1}{n'_c - n_c}$$

Using the notation of the preceding two equations we can compute, using calculus, the total differential as

$$\Delta x = \frac{\partial x}{\partial n_c} \Delta n_c$$

Because we have already calculated Δx and the partial derivative, we can invert the preceding equation and solve for the change in missile acceleration as

$$\Delta n_c = \frac{\Delta x}{\frac{\partial x}{\partial n_c}}$$

The new acceleration command, assuming the acceleration command is a constant, in the predictor-corrector scheme is simply the previous command plus the suggested change from the previous equation or

$$n_{c_k} = n_{c_{k-1}} + \Delta n_c$$

We now repeat the cycle at each guidance update (solving for Δx and partial derivative by integrating equations of motion forward) until the missile hits the ground. The calculations performed at each guidance update are also depicted in Fig. 27.8.

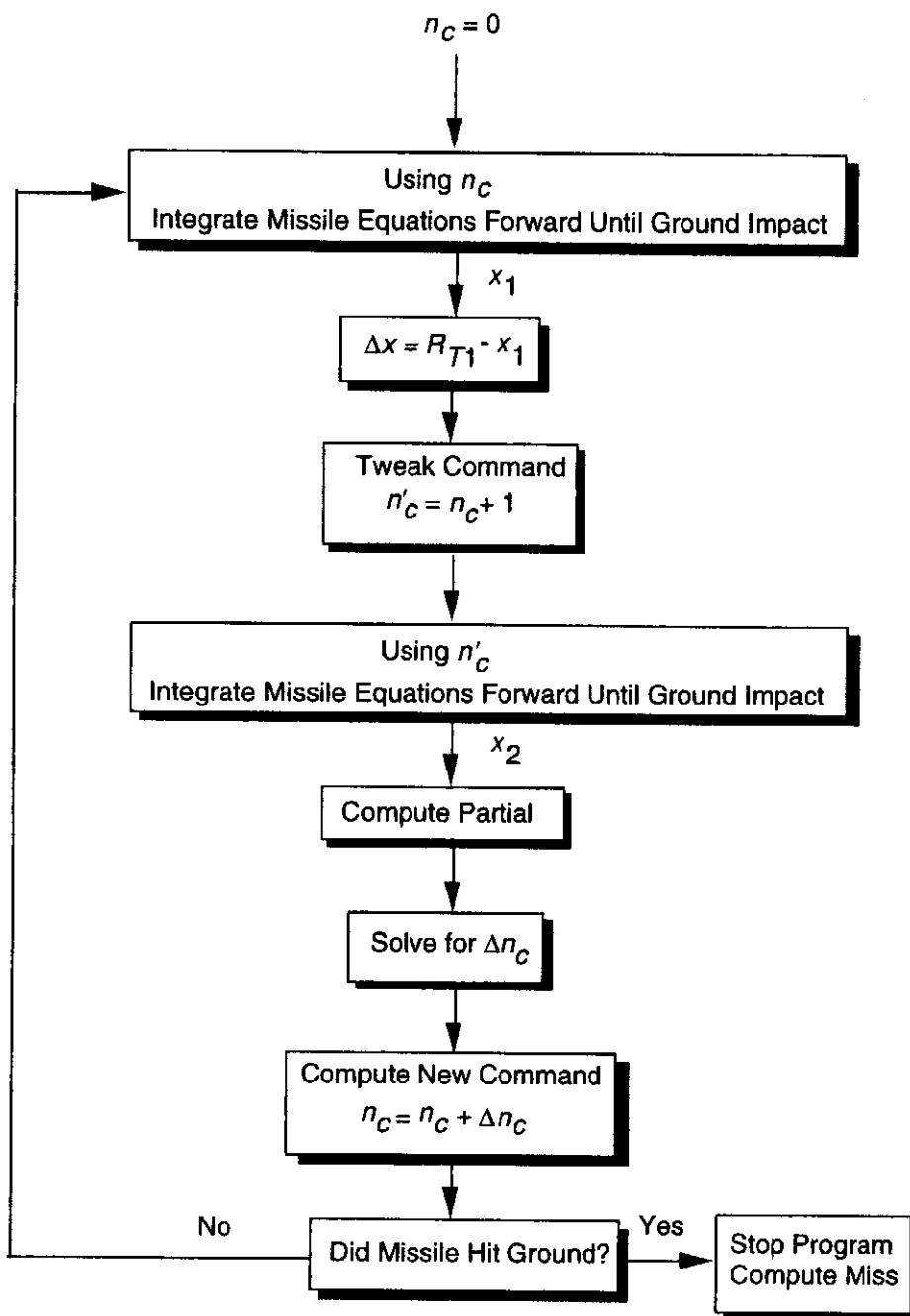


Fig. 27.8 Predictor-corrector guidance logic for first problem.

The predictor-corrector guidance scheme is similar to predictive guidance (see Chapter 16), except that in this application we integrate the equations of motion forward twice rather than once to obtain the desired acceleration command at each guidance update. An engagement simulation using the predictor-corrector method appears in Listing 27.2. As can be seen from the listing, we integrate the equations of motion forward until ground impact twice during each guidance update by calling subroutine PREDICT. Each time we call subroutine PREDICT we must supply the current missile and target location, missile speed, ballistic coefficient, and acceleration command. The output of the subroutine is the missile location at ground impact. In Listing 27.2 we have the ability to supply the estimated missile ballistic coefficient BETAH, which may be different than the actual missile ballistic coefficient BETA.

Listing 27.2 Engagement simulation with predictor-corrector guidance

```

IMPLICIT REAL*8(A-H,O-Z)
INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
H=.01
VM=3000.
BETA=1000.
BETAH=1000.
XLIMG=5.
T=0.
S=0.
GAMDEG=45.
VM1=VM*COS(GAMDEG/57.3)
VM2=VM*SIN(GAMDEG/57.3)
RM1=0.
RM2=0.
XNC=0.
RT1=60000.
RT2=0.
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
XLIM=XLIMG*32.2
10 IF(T>0..AND.RM2<=0.)GOTO 999
IF(RTM<1000.)THEN
    H=.0001
ELSE
    H=.01
ENDIF
RM1OLD=RM1
RM2OLD=RM2
VM1OLD=VM1
VM2OLD=VM2
STEP=1
GOTO 200
66 STEP=2
RM1=RM1+H*VM1
RM2=RM2+H*VM2
VM1=VM1+H*AM1
VM2=VM2+H*AM2
T=T+H
GOTO 200
55 CONTINUE
RM1=.5*(RM1OLD+RM1+H*VM1)
RM2=.5*(RM2OLD+RM2+H*VM2)
VM1=.5*(VM1OLD+VM1+H*AM1)
VM2=.5*(VM2OLD+VM2+H*AM2)
S=S+H
IF(S<.99999)GOTO 10
S=0.

```

(Contd.)

Listing 27.2 (Continued)

```

IF(T>30.)THEN
    CALL PREDICT(T,TF,RM1,RM2,VM1,VM2,X1,XNC,
                RT1,RT2,BETAH)
    DELX=RT1-X1
    XNCP=XNC+1.
    CALL PREDICT(T,TF,RM1,RM2,VM1,VM2,X2,XNCP,
                RT1,RT2,BETAH)
    DXDNC=(X2-X1)/(XNCP-XNC)
    DELXNC=DELX/DXDNC
    XNC=XNC+DELXNC
    IF(XNC>XLIM)XNC=XLIM
    IF(XNC<-XLIM)XNC=-XLIM
ENDIF
RM1K=RM1/1000.
RM2K=RM2/1000.
RT1K=RT1/1000.
RT2K=RT2/1000.
XNCG=XNC/32.2
WRITE(9,*)T, RM1K, RM2K, RT1K, RT2K, XNCG
WRITE(1,*)T, RM1K, RM2K, RT1K, RT2K, XNCG
GOTO 10
200  CONTINUE
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
XLAM=ATAN2(RTM2,RTM1)
XNE1=-XNC*SIN(XLAM)
XNE2=XNC*COS(XLAM)
IF(RM2<30000.)THEN
    RHO=.002378*EXP(-RM2/30000)
ELSE
    RHO=.0034*EXP(-RM2/22000)
ENDIF
VM=SQRT(VM1**2+VM2**2)
Q=.5*RHO*VM*VM
GAM=ATAN2(VM2,VM1)
DRAG=Q*32.2/BETA
AM1=-DRAG*COS(GAM)+XNE1
AM2=-32.2-DRAG*SIN(GAM)+XNE2
IF(STEP-1)66,66,55
999  CONTINUE
VM=SQRT(VM1**2+VM2**2)
RM1K=RM1/1000.
RM2K=RM2/1000.
RT1K=RT1/1000.
RT2K=RT2/1000.
XNCG=XNC/32.2
WRITE(9,*)T, RM1K, RM2K, RT1K, RT2K, XNCG
WRITE(1,*)T, RM1K, RM2K, RT1K, RT2K, XNCG

```

(Contd.)

Listing 27.2 (Continued)

```

        WRITE(9,*)RTM
        PAUSE
        CLOSE(1)
        END

        SUBROUTINE PREDICT(TP,TF,RM1P,RM2P,VM1P,VM2P,RM1F,XNC1F,
1          RT1P,RT2P,BETAH)
        IMPLICIT REAL*8(A-H,O-Z)
        INTEGER STEP
        H=.01
        RM1=RM1P
        RM2=RM2P
        VM1=VM1P
        VM2=VM2P
        XNC=XNC1F
        RT1=RT1P
        RT2=RT2P
        BETA=BETAH
        T=TP
        RTM1=RT1-RM1
        RTM2=RT2-RM2
        RTM=SQRT(RTM1**2+RTM2**2)
10      IF(RM2<0.)GOTO 999
        IF(RTM<1000.)THEN
          H=.0001
        ELSE
          H=.01
        ENDIF
        RM1OLD=RM1
        RM2OLD=RM2
        VM1OLD=VM1
        VM2OLD=VM2
        STEP=1
        GOTO 200
66      STEP=2
        RM1=RM1+H*VM1
        RM2=RM2+H*VM2
        VM1=VM1+H*AM1
        VM2=VM2+H*AM2
        T=T+H
        GOTO 200
55      CONTINUE
        RM1=.5*(RM1OLD+RM1+H*VM1)
        RM2=.5*(RM2OLD+RM2+H*VM2)
        VM1=.5*(VM1OLD+VM1+H*AM1)
        VM2=.5*(VM2OLD+VM2+H*AM2)
        GOTO 10
200     CONTINUE
        RTM1=RT1-RM1
        RTM2=RT2-RM2

```

(Contd.)

Listing 27.2 (Continued)

```

RTM=SQRT(RTM1**2+RTM2**2)
XLAM=ATAN2(RTM2,RTM1)
XNE1=-XNC*SIN(XLAM)
XNE2=XNC*COS(XLAM)
IF(RM2<30000.)THEN
    RHO=.002378*EXP(-RM2/30000)
ELSE
    RHO=.0034*EXP(-RM2/22000)
ENDIF
VM=SQRT(VM1**2+VM2**2)
Q=.5*RHO*VM*VM
GAM=ATAN2(VM2,VM1)
DRAG=Q*32.2/BETA
AM1=-DRAG*COS(GAM)+XNE1
AM2=-32.2-DRAG*SIN(GAM)+XNE2
IF(STEP-1)66,66,55
999  CONTINUE
RM1F=RM1
RETURN
END

```

The predictor-corrector engagement simulation of Listing 27.2 was run for cases in which the target location RT1 was made a parameter. We can see from Fig. 27.9 that, as with augmented proportional navigation, we can hit targets from 40 to 100 kft downrange of the missile launch point. The acceleration profiles for the various trajectories appear in Fig. 27.10. We can see that when guidance is turned on, after a brief transient period the acceleration commands for the predictor-corrector guidance approach are approximately a constant, thus matching our assumed nominal acceleration profile. The most demanding part of the acceleration

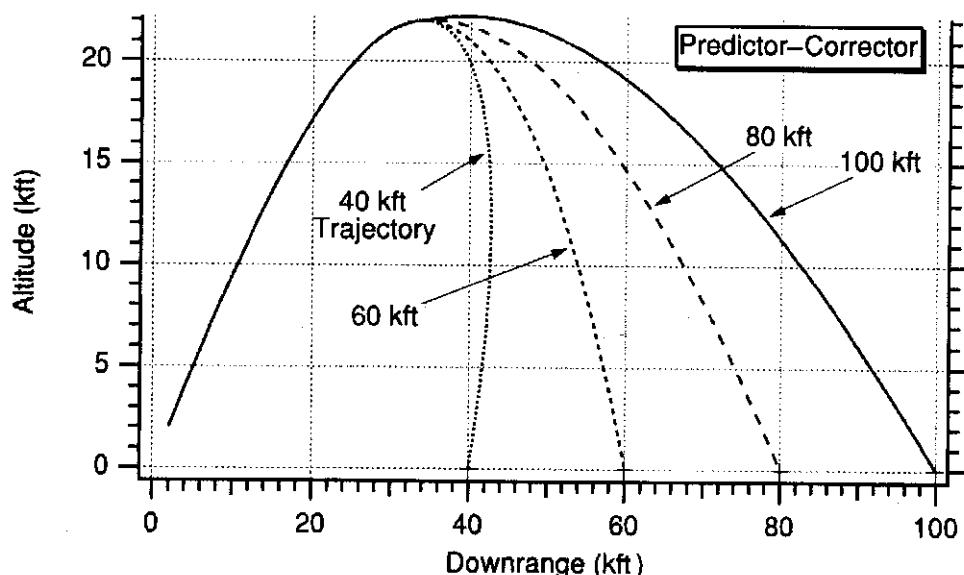


Fig. 27.9 Predictor-corrector guidance is as effective as augmented proportional navigation in this application.

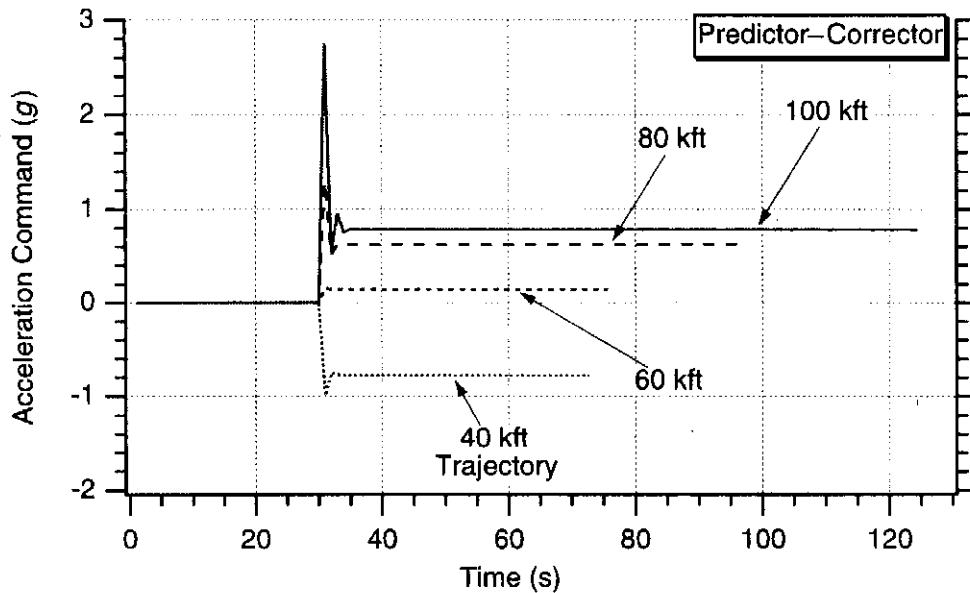


Fig. 27.10 Predictor-corrector guidance commands are near constant in this application.

command is initially during the transient portion of flight. The acceleration commands in the steady state with predictor-corrector guidance are actually slightly smaller than those obtained with augmented proportional navigation.

Recall from Chapter 14 that predictive guidance was demonstrated to be very sensitive to modeling errors. To test the robustness of predictor-corrector guidance in this application, the estimated ballistic coefficient was varied from the correct value of 1000 lb/ft^2 to as low as 500 lb/ft^2 to as high as 1500 lb/ft^2 . We can see from Fig. 27.11 that even with these large ballistic coefficient errors we are still able to hit the target. Figure 27.12 shows that the acceleration profiles hardly change when there are significant errors in the estimated ballistic coefficient. Therefore, we can conclude that in this application predictor-corrector guidance does not appear to be sensitive to modeling errors.

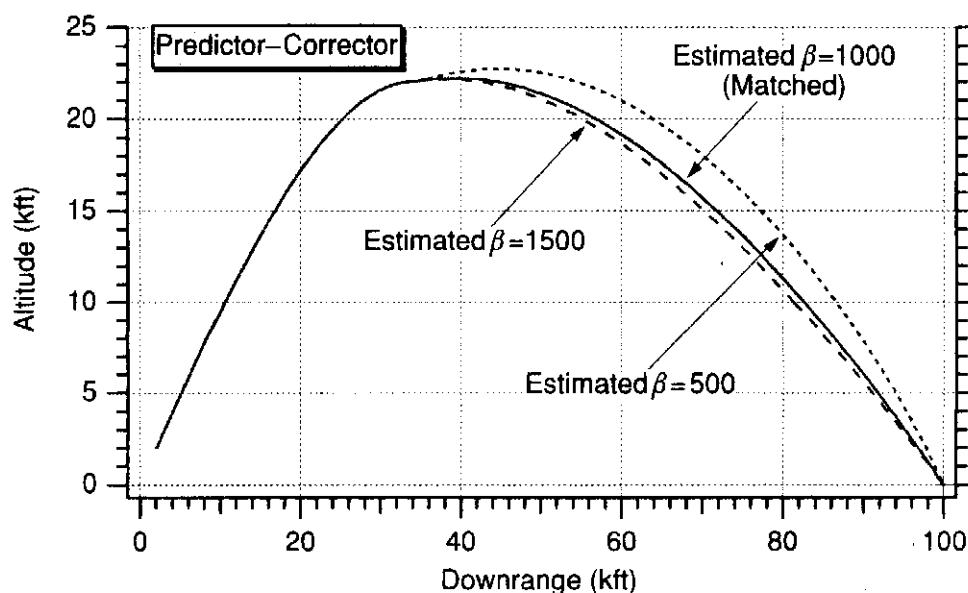


Fig. 27.11 Predictor-corrector guidance still hits target if there is 50% error in knowledge of ballistic coefficient.

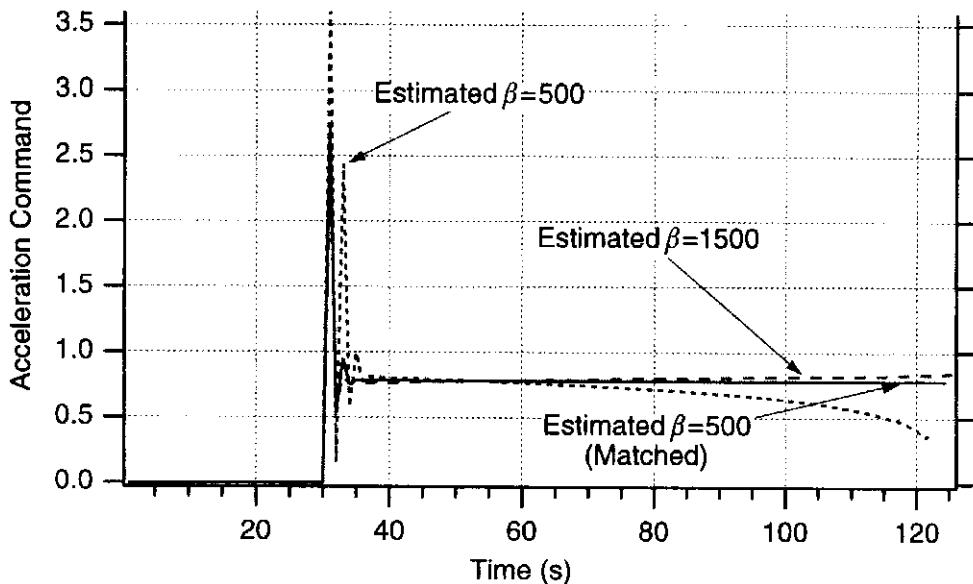


Fig. 27.12 Acceleration commands only change slightly if there is a 50% error in estimate of ballistic coefficient.

Roll Angle or Rate Guidance Problem

As a further test of predictor-corrector guidance we will consider an unusual problem in which the missile is severely constrained in its ability to issue guidance commands. Consider the case of Fig. 27.13, in which we have a missile that is attempting to hit a stationary target. In this case, the acceleration command of the missile, n_c , is fixed in amplitude and we only have control over the orientation of the acceleration magnitude by controlling the either the roll-angle, ϕ , or the roll-rate. This type of guidance might be representative of a rolling missile that is at a fixed trim angle. From Fig. 27.13 we can see that the components of the missile acceleration command in the x and y directions are given by

$$\ddot{x}_M = n_c \cos \phi$$

$$\ddot{y}_M = n_c \sin \phi$$

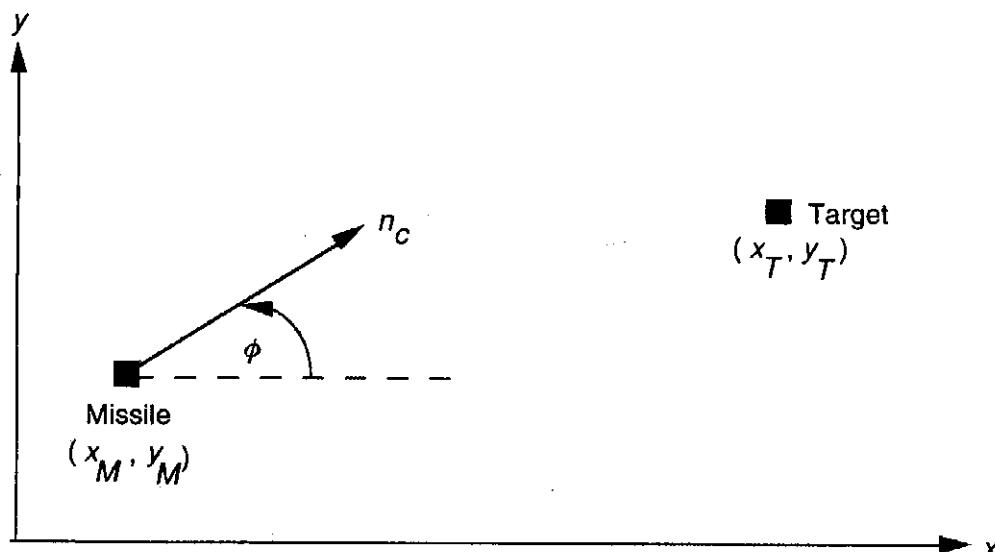


Fig. 27.13 Unusual guidance problem in which we can only control roll-angle.

In this guidance problem we would like to hit a stationary target at time t_F . Mathematically, this means that

$$x_M(t_F) = x_T$$

$$y_M(t_F) = y_T$$

Therefore, the guidance problem is to figure out what to do with the roll-angle or roll-rate so that the missile hits the target at the final time.

Before we attempt to use predictor-corrector guidance let us first see if it is possible to use some form of proportional navigation on this problem. Recall that the relative ranges between the missile and target are given by

$$R_{TM1} = x_T - x_M$$

$$R_{TM2} = y_T - y_M$$

In this problem the target is stationary and therefore its velocity must be zero. The components of the relative velocity are given by

$$V_{TM1} = \dot{x}_T - \dot{x}_M = -\dot{x}_M$$

$$V_{TM2} = \dot{y}_T - \dot{y}_M = -\dot{y}_M$$

We can think of proportional navigation as a guidance law whose acceleration commands are proportional to the line of sight rate or proportional to the zero-effort miss and inversely proportional to the square of time to go until intercept (see Chapter 2). Therefore, if we want to use proportional navigation in this application the commanded acceleration components in the x and y directions are given by

$$n_{cx} = \frac{N'}{t_{go}^2} (R_{TM1} + V_{TM1} t_{go})$$

$$n_{cy} = \frac{N'}{t_{go}^2} (R_{TM2} + V_{TM2} t_{go})$$

Therefore, from Fig. 27.13 the appropriate roll-angle command given a fixed acceleration level for the implementation of proportional navigation should be

$$\phi = \tan^{-1} \frac{n_{cy}}{n_{cx}} = \tan^{-1} \left[\frac{\frac{N'}{t_{go}^2} (R_{TM2} + V_{TM2} t_{go})}{\frac{N'}{t_{go}^2} (R_{TM1} + V_{TM1} t_{go})} \right] = \tan^{-1} \left[\frac{R_{TM2} + V_{TM2} t_{go}}{R_{TM1} + V_{TM1} t_{go}} \right]$$

Listing 27.3 is an engagement simulation of the missile attempting to hit the stationary target after 100 s using the roll-angle command of the preceding equation as the guidance law. In this application we are not attempting to control the roll-rate. In a practical application we would have to ensure that the resultant roll-rates are not too high. The missile is initially at the origin (namely, XM=0, YM=0) while the target has coordinates XT=-4000 ft and YT=5000 ft. In the simulation the magnitude of the acceleration command is 12 ft/s² and the roll-angle command is computed every .1 s. Notice that the simulation does not stop at the point of closest approach but instead stops after 100 s of flight (that is, the time that we would like to hit the target).

Listing 27.3 An engagement simulation using an unusual form of proportional navigation

```

IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
XT=-4000.
YT=5000.
TF=100.
H=.01
TS=.1
XNC=12.
XM=0.
YM=0.
XMD=0.
YMD=0.
S=0.
T=0.
PHI=0.
10 IF(T>TF)GOTO 999
XMOLD=XM
YMOLD=YM
XMDOLD=XMD
YMDOLD=YMD
STEP=1
GOTO 200
66 STEP=2
XM=XM+H*XMD
YM=YM+H*YMD
XMD=XMD+H*XMDD
YMD=YMD+H*YMDD
T=T+H
GOTO 200
55 CONTINUE
XM=(XMOLD+XM)/2+.5*H*XMD
YM=(YMOLD+YM)/2+.5*H*YMD
XMD=(XMDOLD+XMD)/2+.5*H*XMDD
YMD=(YMDOLD+YMD)/2+.5*H*YMDD
S=S+H
IF(S.LT.(TS-.0001))GOTO 10
S=0.
TGO=TF-T+.0001
RTM1=XT-XM
RTM2=YT-YM
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=-XMD
VTM2=-YMD
PHI=ATAN2(RTM2+VTM2*TGO,RTM1+VTM1*TGO)
WRITE(9,*)T,XM,YM,XT,YT,PHI*57.3
WRITE(1,*)T,XM,YM,XT,YT,PHI*57.3
GOTO 10

```

(Contd.)

Listing 27.3 (Continued)

```

200  CONTINUE
XMDD=XNC*COS(PHI)
YMDD=XNC*SIN(PHI)
IF(STEP-1)66,66,55
999  CONTINUE
RTM1=XT-XM
RTM2=YT-YM
RTM=SQRT(RTM1**2+RTM2**2)
WRITE(9,*)T,XM,YM,XT,YT,PHI*57.3
WRITE(1,*)T,XM,YM,XT,YT,PHI*57.3
WRITE(9,*)RTM
PAUSE
CLOSE(1)
END

```

The nominal case of Listing 27.3 was run, and we can see from Fig. 27.14 that the missile flies directly to the stationary target and scores a direct hit. The roll-angle command required to achieve this successful intercept is displayed in Fig. 27.15. We can see that the roll-angle continuously oscillates between -50 and 130 deg. This means that the accelerations are 180 deg apart. In other words, the missile first accelerates in one direction and then after a very brief period of time accelerates in the opposite direction. In practice it might be very difficult to achieve the high roll-rates needed to switch nearly instantaneously between the two roll-angles. From Fig. 27.14 we can see that the angle required for the missile to fly directly to the target is simply

$$\phi = \tan^{-1} \frac{5000}{-4000} = 129 \text{ deg}$$

which is approximately the same roll-angle chosen by the guidance law. If the guidance law senses that there is too much acceleration and that the missile will arrive at the target too soon, a command is issued in the opposite direction.

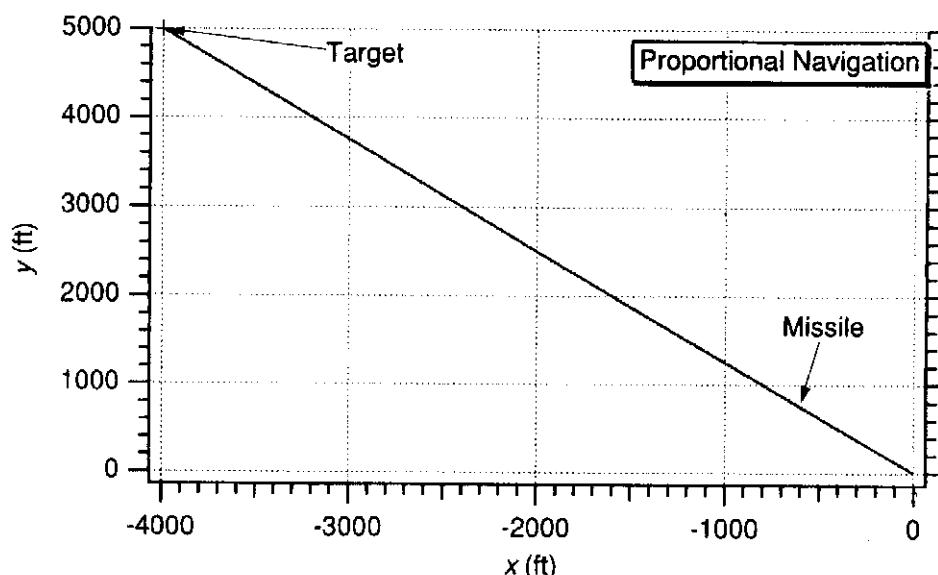


Fig. 27.14 Missile flies directly to target with proportional navigation.

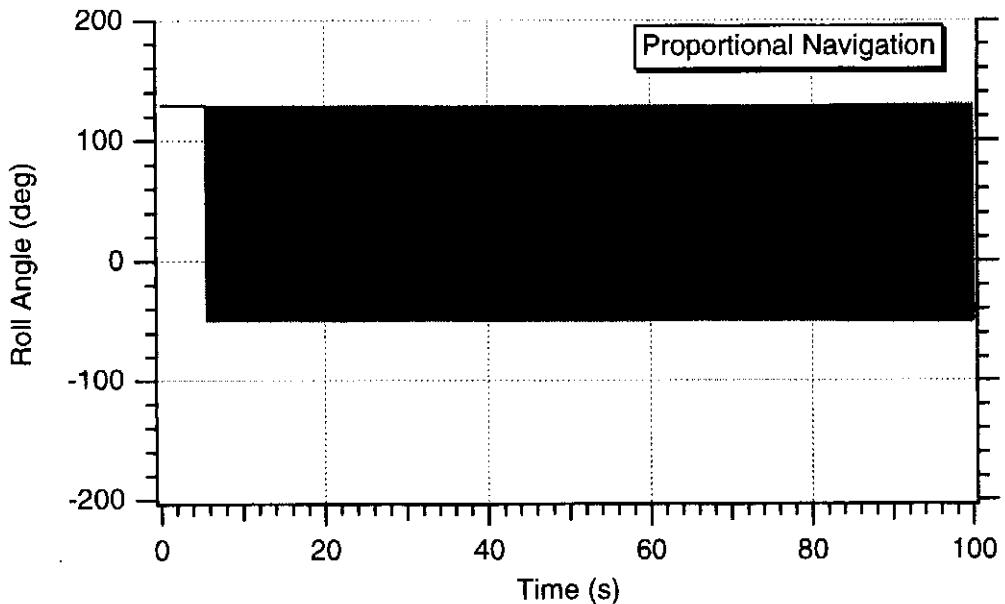


Fig. 27.15 With proportional navigation missile roll-angle oscillates between -50 and 130 deg.

Similarly, when the command is in the opposite direction and the guidance law believes that the missile will be late in arriving at the target, a command is issued in the opposite direction.

The target was then moved to all four quadrants [that is, $(-4000, -5000)$, $(-4000, 5000)$, $(4000, -5000)$ and $(4000, 5000)$], and successful intercepts resulted in all cases. Thus, we can conclude that if we ignore the high roll-rate question, at least mathematically proportional navigation type guidance is effective in this application.

Predictor-Corrector Guidance With Roll-Rate Problem^{1,2,3}

As was mentioned previously in this chapter, predictor-corrector guidance is based on having a model of the interceptor and target and then trying to determine how certain changes in important parameters influence the missile trajectory. In the previous application of predictor-corrector guidance we assumed a nominal acceleration command that was a constant. In this problem we will assume a linear form for the nominal roll-rate command as shown in Fig. 27.16. In this model we have specified the nominal maximum and minimum roll-rates along with the time

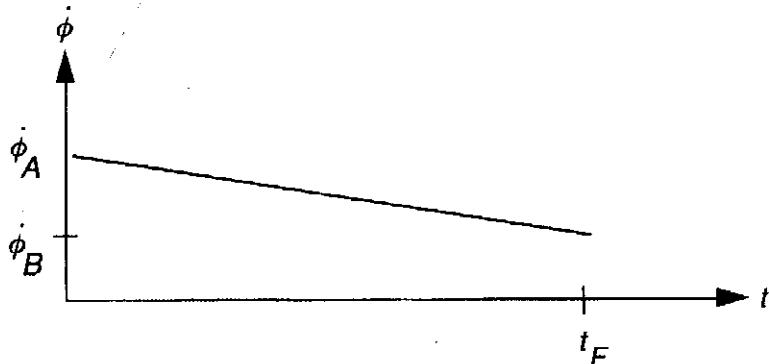


Fig. 27.16 Nominal model for roll-rate profile to be used with predictor-corrector guidance.

of flight. Two parameters are required in our nominal model because we will be attempting to control two parameters.

Using the example of the previous section in conjunction with predictor-corrector guidance, we first integrate the missile equations of motion forward assuming the nominal roll-rate profile of Fig. 27.16 (with initial specified values for the two roll-rate parameters and the flight time). The fixed acceleration command at the appropriate roll-angle profile (that is, obtained by integrating the roll-rate) is issued to see where the missile is at the final time t_F . We then measure the difference between where the target really is (x_T, y_T) and where the missile actually ends up (x_1, y_1) and call the resultant quantities Δx and Δy or

$$\Delta x = x_T - x_1$$

$$\Delta y = y_T - y_1$$

We now add a little bit or tweak one of the roll-rate parameters. For example, let us assume that the new minimum roll-rate parameter is given by

$$\dot{\phi}_{B2} = \dot{\phi}_B + .001$$

We again integrate the missile equations of motion forward using the new minimum roll-rate parameter to see where the missile is at the final time t_F . We then note where the missile actually ends up (x_2, y_2) and manually compute partial derivatives. In this example, we can say that

$$\begin{aligned}\frac{\partial x}{\partial \dot{\phi}_B} &= \frac{x_2 - x_1}{\dot{\phi}_{B2} - \dot{\phi}_B} = \frac{x_2 - x_1}{.001} \\ \frac{\partial y}{\partial \dot{\phi}_B} &= \frac{y_2 - y_1}{\dot{\phi}_{B2} - \dot{\phi}_B} = \frac{y_2 - y_1}{.001}\end{aligned}$$

We now add a little bit or tweak the maximum roll-rate parameter. The new maximum roll-rate parameter is given by

$$\dot{\phi}_{A2} = \dot{\phi}_A + .001$$

We again integrate the missile equations of motion forward using the new maximum roll-rate parameter to see where the missile is at the final time t_F . We then note where the missile actually ends up (x_3, y_3) and manually compute another set of partial derivatives. In this example, we can say that

$$\begin{aligned}\frac{\partial x}{\partial \dot{\phi}_A} &= \frac{x_3 - x_1}{\dot{\phi}_{A2} - \dot{\phi}_A} = \frac{x_3 - x_1}{.001} \\ \frac{\partial y}{\partial \dot{\phi}_A} &= \frac{y_3 - y_1}{\dot{\phi}_{A2} - \dot{\phi}_A} = \frac{y_3 - y_1}{.001}\end{aligned}$$

From calculus we know that the total differential can be written as

$$\Delta x = \frac{\partial x}{\partial \dot{\phi}_A} \Delta \dot{\phi}_A + \frac{\partial x}{\partial \dot{\phi}_B} \Delta \dot{\phi}_B$$

$$\Delta y = \frac{\partial y}{\partial \dot{\phi}_A} \Delta \dot{\phi}_A + \frac{\partial y}{\partial \dot{\phi}_B} \Delta \dot{\phi}_B$$

We now have two equations with two unknowns (because we already know the partial derivatives and Δx and Δy). After solving for the new change in the roll-rate

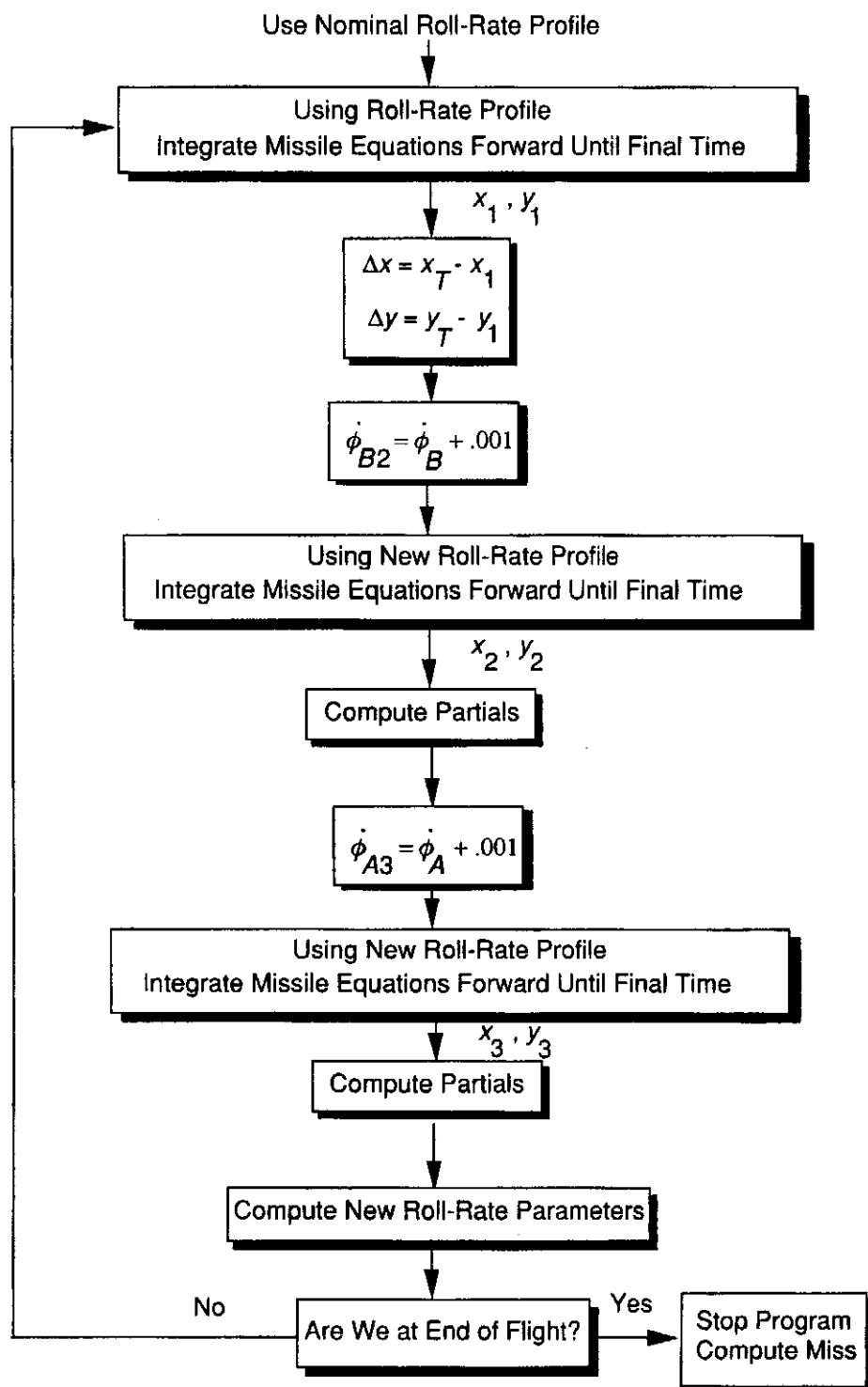


Fig. 27.17 In predictor-corrector guidance logic for second problem.

parameters, we can say that the new maximum and minimum roll-rate parameters are given by the previous values plus the changes or

$$\dot{\phi}_{A_k} = \dot{\phi}_{A_{k-1}} + \Delta\dot{\phi}_A$$

$$\dot{\phi}_{B_k} = \dot{\phi}_{B_{k-1}} + \Delta\dot{\phi}_B$$

We now repeat the cycle (that is, solving for Δx , Δy , and partial derivative by integrating equations of motion forward until the final time using new nominal roll-rate profile). The cycle is also depicted in Fig. 27.17.

Listing 27.4 is an engagement simulation of the missile attempting to hit the stationary target after 100 s using the previously described predictor–corrector guidance scheme. The missile is initially at the origin (namely, XM=0, YM=0) while the target has coordinates XT=-4000 ft and YT=5000 ft. In the simulation, the magnitude of the acceleration command is 12 ft/s² and the roll-angle command is computed every .1 s. The nominal maximum and minimum roll-rates are 10 deg/s and 5 deg/s, respectively. Again, notice that the simulation does not stop at the point of closest approach but stops after 100 s of flight. In addition, the simulation slightly modifies the formulas for the updated roll-rates to be

$$\dot{\phi}_{A_k} = \dot{\phi}_{A_{k-1}} + \text{GAIN } \Delta \dot{\phi}_A$$

$$\dot{\phi}_{B_k} = \dot{\phi}_{B_{k-1}} + \text{GAIN } \Delta \dot{\phi}_B$$

The addition of the new parameter GAIN will give us an extra degree of flexibility in modifying the iterated roll-rates if we run into problems. Nominally, GAIN=1, so we have simulated the scheme that was derived in this section. Unlike the previous application of predictor–corrector guidance in which we called subroutine PREDICT twice (we only needed to calculate one partial derivative), we now have to call subroutine PREDICT three times.

The nominal case of Listing 27.4 was run, and we can see from Fig. 27.18 that a direct hit was scored. The trajectory is totally different from the straight-line approach of proportional navigation. The missile loops around before hitting the target. The required roll-rate profile for the successful intercept is displayed in Fig. 27.19. Here we can see that the roll-rate profile is approximately linear with a maximum roll-rate of only 12 deg/s. The required roll-rate is much smaller here than it would have been using proportional navigation.

Another case was run in which the target was moved to another location. Figure 27.20 shows that in this case predictor–corrector guidance failed and the missile did not even get close to the target. Normally predictor–corrector guidance acts as a feedback scheme in which the Δs should be driven to zero. However, we can see from Fig. 27.21 that Δx and Δy do not approach zero, which is another way of letting us know something is wrong.

Numerous experiments were conducted in which various parameters were varied in the simulation to see if predictor–corrector guidance could be made to work in this application. It was discovered that if GAIN could be reduced from 1 to .1 the predictor–corrector guidance scheme could be made very effective. Cases were then run in which the stationary target was moved to all four quadrants. We can see from Fig. 27.22 that now the missile can hit the target in any quadrant when GAIN is reduced to the new value of .1. The resultant roll-rate profiles are displayed in Fig. 27.23. We can see that no matter where the target is placed the roll-rate profiles are all well behaved.

To further test the robustness of predictor–corrector guidance, a modeling error was introduced. The actual missile acceleration is 12 ft/s² and a 10% error was introduced (namely, ACCERR=1.2) so that subroutine PREDICT would be in error. We can see from Fig. 27.24 that the small acceleration error causes the missile to miss the target when predictor–corrector guidance is used. The Δs no longer approach zero. Figure 27.25 shows that the roll-rate profiles start to oscillate when the acceleration error is introduced.

Thus, we can conclude that in this application predictor–corrector guidance is very sensitive to modeling errors. Its lack of robustness is similar to that of predictive guidance, which is also a brute-force, model-based approach to guidance.

Listing 27.4 An unusual problem using predictor-corrector guidance

```

IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
GAIN=1.
XT=-4000.
YT=5000.
PHIADDEG=10.
PHIBDDEG=5.
PHIDMAXDEG=15.
TF=100.
ACCERR=0.
H=.01
TS=.1
XNC=12.
PHIDMAX=PHIDMAXDEG/57.3
PHI=0.
XM=0.
YM=0.
XMD=0.
YMD=0.
PHIAD=PHIADDEG/57.3
PHIBD=PHIBDDEG/57.3
S=0.
T=0.
10 IF(T>(TF-.00001))GOTO 999
XMOLD=XM
YMOLD=YM
XMDOLD=XMD
YMDOLD=YMD
PHIOLD=PHI
STEP=1
GOTO 200
66 STEP=2
XM=XM+H*XMD
YM=YM+H*YMD
XMD=XMD+H*XMDD
YMD=YMD+H*YMDD
PHI=PHI+H*PHID
T=T+H
GOTO 200
55 CONTINUE
XM=(XMOLD+XM)/2+.5*H*XMD
YM=(YMOLD+YM)/2+.5*H*YMD
XMD=(XMDOLD+XMD)/2+.5*H*XMDD
YMD=(YMDOLD+YMD)/2+.5*H*YMDD
PHI=(PHIOLD+PHI)/2+.5*H*PHID
S=S+H
IF(S.LT.(TS-.0001))GOTO 10
S=0.

```

(Contd.)

Listing 27.4 (Continued)

```

CALL PREDICT(T,XM,YM,XMD,YMD,PHI,TF,PHIAD,PHIBD,X1,Y1,XNC,
1      H,PHIDMAX,ACCERR)
DELX=XT-X1
DELY=YT-Y1
PHIBD2=PHIBD+.001
CALL PREDICT(T,XM,YM,XMD,YMD,PHI,TF,PHIAD,PHIBD2,X2,Y2,XNC,
1      H,PHIDMAX,ACCERR)
DXDPB=(X2-X1)/(PHIBD2-PHIBD)
DYDPB=(Y2-Y1)/(PHIBD2-PHIBD)
PHIAD3=PHIAD+.001
CALL PREDICT(T,XM,YM,XMD,YMD,PHI,TF,PHIAD3,PHIBD,X3,Y3,XNC,
1      H,PHIDMAX,ACCERR)
DXDPA=(X3-X1)/(PHIAD3-PHIAD)
DYDPA=(Y3-Y1)/(PHIAD3-PHIAD)
IF((DXDPB*DYDPA-DXDPA*DYDPB).EQ.0.)THEN
    DELPHIAD=0.
    DELPHIBD=0.
ELSE
    DELPHIAD=(DXDPB*DELY-DELX*DYDPB)/(DXDPB*DYDPA-
1      DXPB*DYDPB)
    DELPHIBD=(DELX*DYDPA-DXDPA*DELY)/(DXDPB*DYDPA-
1      DXPB*DYDPB)
ENDIF
PHIAD=PHIAD+GAIN*DELPHIAD
PHIBD=PHIBD+GAIN*DELPHIBD
WRITE(9,*)T,XM,YM,XT,YT,PHID*57.3
WRITE(1,*)T,XM,YM,XT,YT,PHID*57.3
GOTO 10
200 CONTINUE
SLOPE=(PHIBD-PHIAD)/TF
BINT=PHIBD-SLOPE*TF
PHID=SLOPE*T+BINT
IF(PHID>PHIDMAX)PHID=PHIDMAX
IF(PHID<-PHIDMAX)PHID=-PHIDMAX
XMDD=XNC*COS(PHI)
YMDD=XNC*SIN(PHI)
IF(STEP-1)66,66,55
999 CONTINUE
RTM=SQRT((XT-XM)**2+(YT-YM)**2)
WRITE(9,*)T,XM,YM,XT,YT,PHID*57.3
WRITE(1,*)T,XM,YM,XT,YT,PHID*57.3
WRITE(9,*)RTM
PAUSE
CLOSE(1)
END

SUBROUTINE PREDICT(TP,XP,YP,XDP,YDP,THETP,TF,THADP,THBDP,XF,
1      YF,ACC,HP,THDMAX,ACCERR)
SAVE

```

(Contd.)

Listing 27.4 (Continued)

```

IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER STEP
H=HP*10.
T=TP
X=XP
Y=YP
XD=XDP
YD=YDP
THET=THETP
THAD=THADP
THBD=THBDP
S=0.
10 IF(T>(TF-.00001))GOTO 999
XOLD=X
YOLD=Y
XDOLD=XD
YDOLD=YD
THETOLD=THET
STEP=1
GOTO 200
66 STEP=2
X=X+H*XD
Y=Y+H*YD
XD=XD+H*XDD
YD=YD+H*YDD
THET=THET+H*THETD
T=T+H
GOTO 200
55 CONTINUE
X=(XOLD+X)/2+.5*H*XD
Y=(YOLD+Y)/2+.5*H*YD
XD=(XDOLD+XD)/2+.5*H*XDD
YD=(YDOLD+YD)/2+.5*H*YDD
THET=(THETOLD+THET)/2+.5*H*THETD
GOTO 10
200 CONTINUE
SLOPE=(THBD-THAD)/TF
BINT=THBD-SLOPE*TF
THETD=SLOPE*T+BINT
IF(THETD>THDMAX)THETD=THDMAX
IF(THETD<-THDMAX)THETD=-THDMAX
XDD=(ACC+ACCERR)*COS(THET)
YDD=(ACC+ACCERR)*SIN(THET)
IF(STEP-1)66,66,55
999 CONTINUE
XF=X
YF=Y
RETURN
END

```

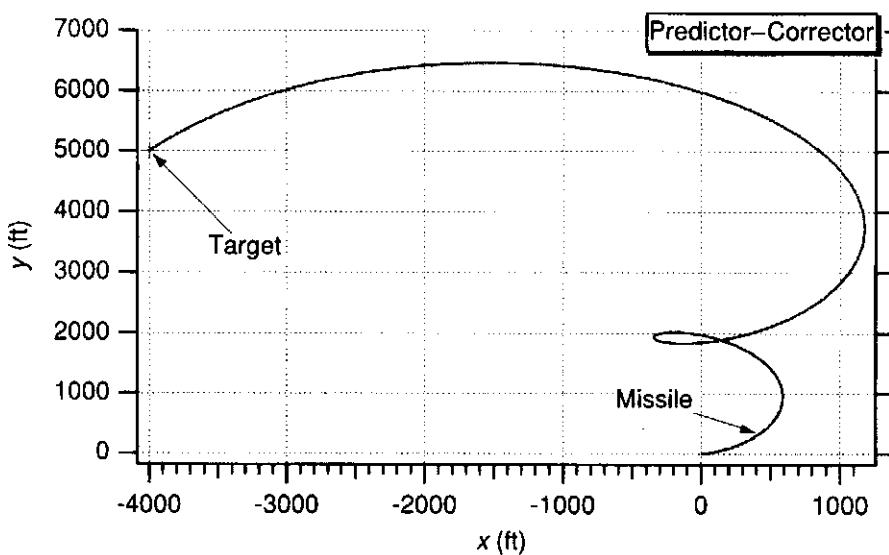


Fig. 27.18 With predictor-corrector guidance missile hits target by flying unusual trajectory.

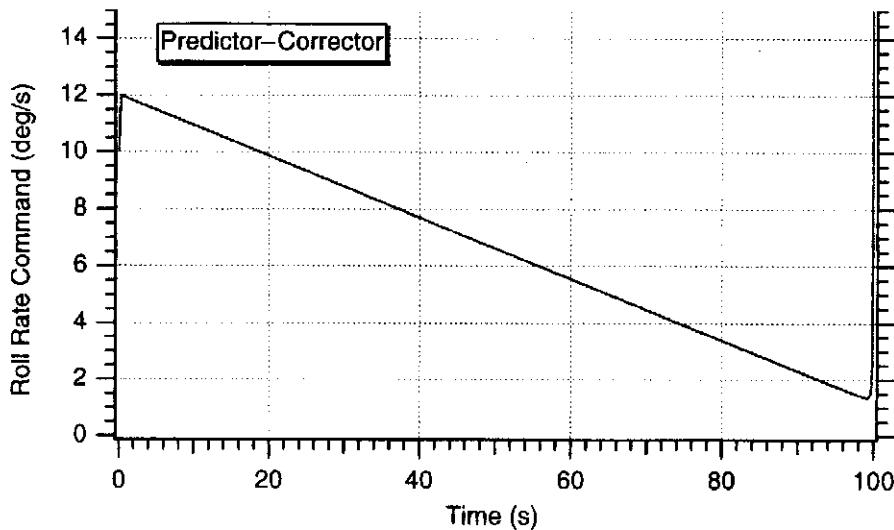


Fig. 27.19 Predictor-corrector guidance yields much smaller roll-rates than proportional navigation.

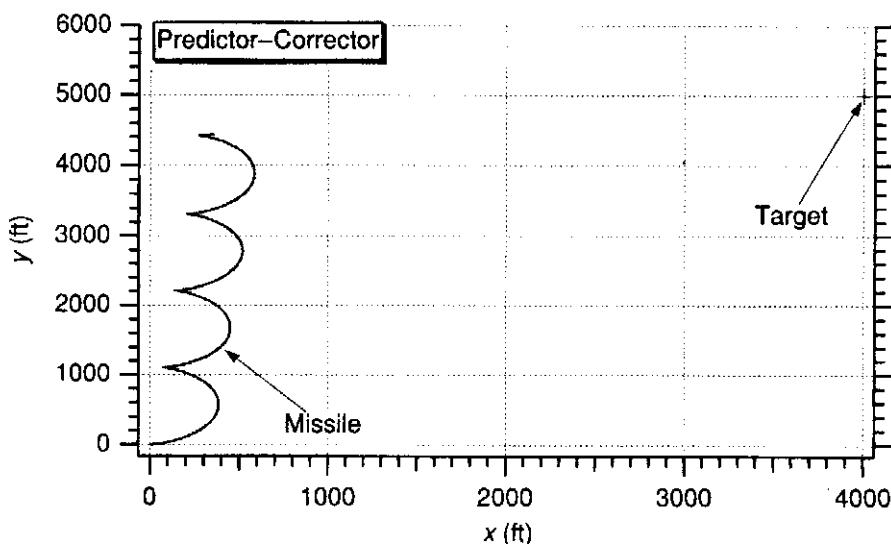


Fig. 27.20 Predictor-corrector guidance has a problem when we move the target.

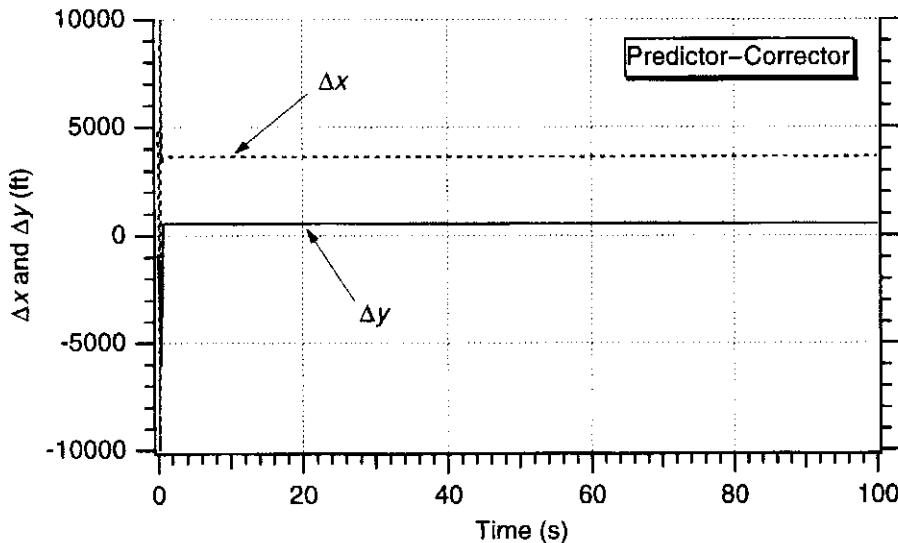


Fig. 27.21 Something is now wrong because errors do not go to zero when target is moved.

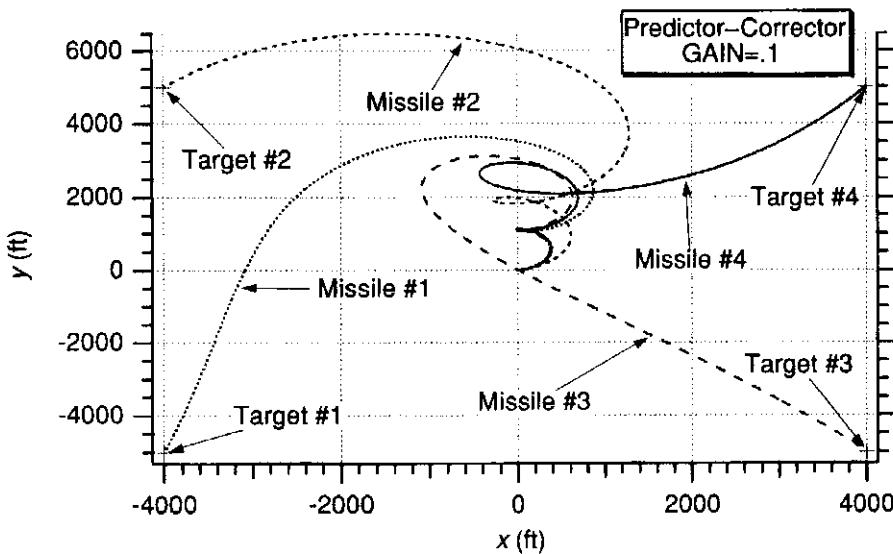


Fig. 27.22 Can hit target in any quadrant when gain reduced from 1 to .1.

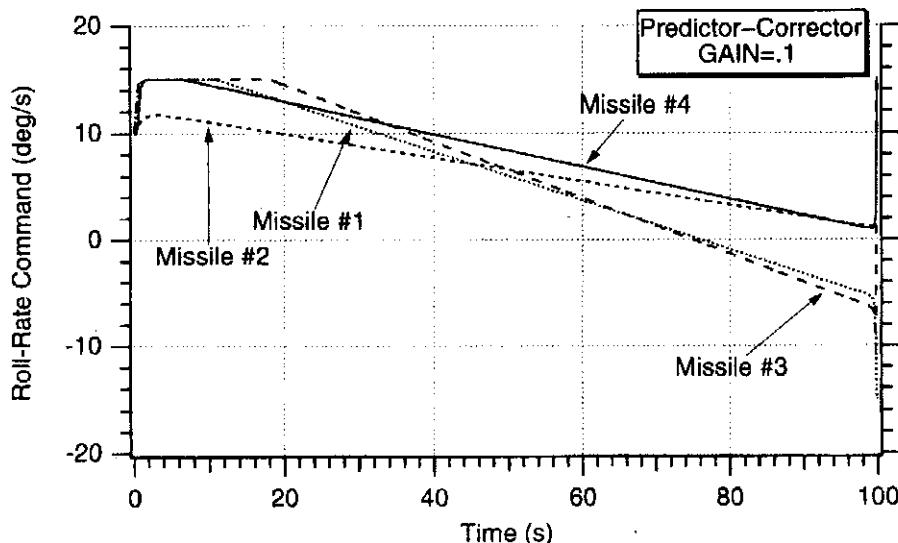


Fig. 27.23 Roll-rate profiles are well behaved and vary with location of target.

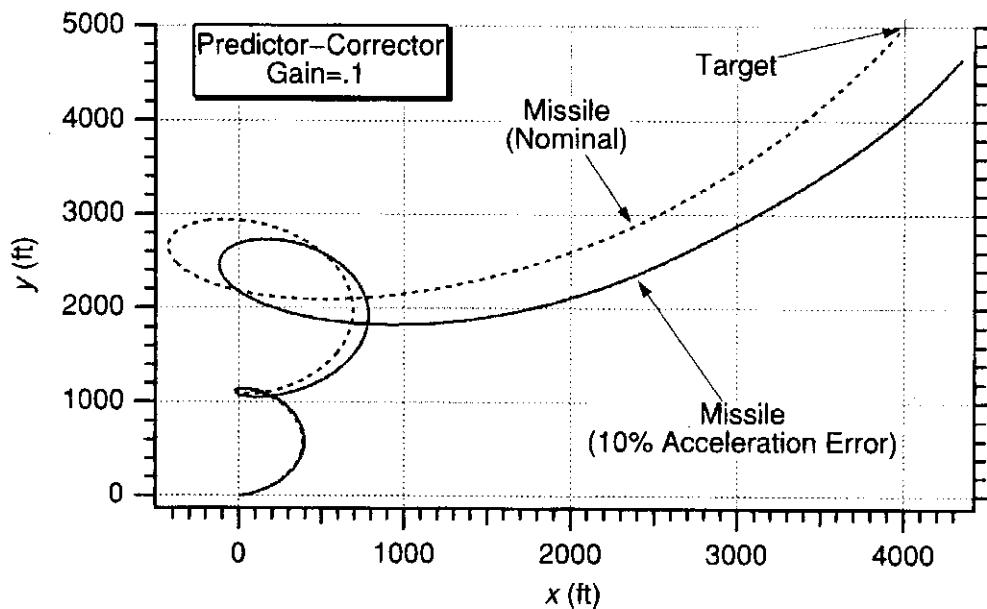


Fig. 27.24 Small error in knowledge of missile acceleration causes predictor guidance to miss target.

We also saw that the application of predictor-corrector guidance is very ad hoc because we must guess a nominal acceleration profile and perform lots of experimentation on various system parameters to achieve success. Thus, the application of predictor-corrector guidance is not as straightforward as the application of proportional navigation types of guidance laws, but under certain circumstances it might be the only guidance choice available.

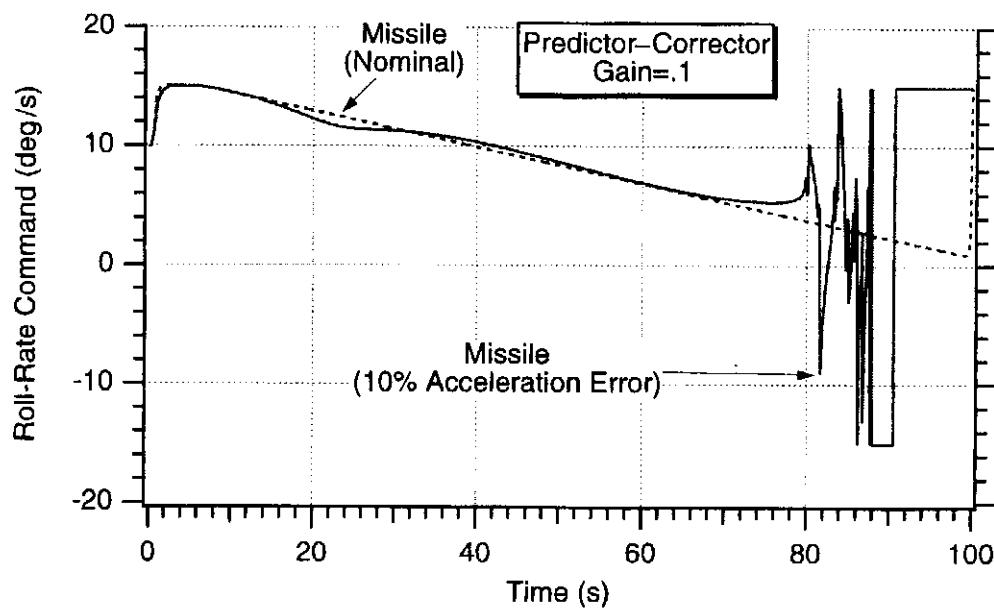


Fig. 27.25 Small error in knowledge of missile acceleration causes roll-rate command to oscillate.

Summary

Two problems were selected in which a predictor-corrector guidance method was introduced and compared with proportional navigation. We showed that predictor-corrector guidance was similar to predictive guidance in the sense that equations representing our best guess at a model of the real world had to be integrated forward several times at each guidance update. The successful application of this guidance approach often requires significant experimentation to pick appropriate nominal models and to choose appropriate gains.

References

- ¹Haroz, C., "A Predictor-Corrector Guidance Algorithm Design For a Low L/D Autonomous Re-Entry Vehicle," Master of Science Thesis, MIT, December 1998.
- ²Braun, R., and Powell, R., "Predictor-Corrector Guidance Algorithm for Use in High-Energy Aerobraking System Studies," *Journal of Guidance, Control, and Dynamics*, Vol.15, No.3, May-June 1992, pp. 672-678.
- ³Miele, A., and Wang, T., "Robust Predictor-Corrector Guidance for Aeroassisted Orbital Transfer," *Journal of Guidance, Control, and Dynamics*, Vol.19, No. 5, September-October 1996, pp. 1134-1141.



Tactical and Strategic Missile Guidance Software

Introduction

IN this appendix additional examples will be presented showing the interested reader how to use the source code listings presented in "Tactical and Strategic Missile Guidance." The listings are often slightly modified so that the reader will understand examples of specific changes that can be made in order to explore issues beyond the scope of the text.

Software Details

To facilitate learning, a CD-ROM that is formatted for both IBM and Macintosh computers containing all of the text's FORTRAN source code listings are included. As a special feature for those who work in the C, MATLAB[®], or True BASIC languages, duplicate files can be found on the CD-ROM that are the C, MATLAB[®], and True BASIC equivalents of the FORTRAN listings.

The FORTRAN and C source code should run, as is, with either Version 6.2 of the Absoft Macintosh Pro Fortran compiler or the Absoft Windows Pro Fortran 7.0 compiler for IBM-compatible machines. Use of different FORTRAN or C compilers, in either the Macintosh or IBM compatible world, may require some slight modification of the source code. The MATLAB[®] source code should run as is using MATLAB[®] 5.2 on the Macintosh or version 5.0 on IBM-compatible computers. The True BASIC source code should run as is with the Bronze edition on the Macintosh or with the Silver edition on IBM-compatible computers.

The naming conventions for the source code files on both languages are slightly different. The FORTRAN naming convention is CxLy.F where x corresponds to Chapter number and y corresponds to Listing number. In other words C4L2.F corresponds to FORTRAN Listing 4.2 of the text (i.e., Chapter 4, Listing 2). The C naming convention is similar for the Macintosh and IBM-compatible machines (C4L2.C). The MATLAB[®] naming convention is the same for both Macintosh and IBM-compatible machines (C4L2.M).

Each of the source code files on the enclosed CD-ROM has a few lines of extra code to make data files so that the user can plot or store the results after a run is made. The name of the generated data file is DATFIL for the Macintosh code and DATFIL.TXT for IBM compatible code. The execution time for most of the FORTRAN and C source code files should only be a few seconds on today's powerful microcomputers. The execution time for the MATLAB[®] and True BASIC source code is considerably longer for the brute force and Monte Carlo listings. In

those cases the number of runs has been reduced and/or the integration step size was increased in order to make the running time more reasonable.

The data statements or definition of constants in each of the source code files corresponds to those used in the numerical examples presented in the text. The user should first run the program of interest as is in order to verify that the data file generated corresponds to the appropriate figure in the text. Other cases of interest can be run by either changing constants and recompiling or by modifying the source code to read input from the keyboard.

In the source code listings that make use of random numbers (i.e., uniform or Gaussian noise generators), use has been made of the FORTRAN random number generators supplied by Absoft. If other FORTRAN compilers are used, the user will have to invoke the appropriate random number language extension for the particular compiler or may have to write a random number generator (such as the one appearing in this appendix) if it is not supplied by the compiler publisher. C users, in either the Macintosh or IBM compatible world, do not have to worry about random number generators since the C language includes the `rand()` statement to simulate a uniformly distributed random number generator. Similarly, MATLAB® users have special statements for uniform and Gaussian distributions, whereas True BASIC offers a special statement for uniformly distributed random numbers.

Integration Example

To show how issues beyond the scope of the text can be explored, let us consider changing the method of integration used in Listing 1.1. We will use the simpler Euler numerical integration method¹ rather than the second-order Runge-Kutta integration technique. Given a first-order differential equation of the form

$$\dot{x} = f(x, t)$$

Euler integration says that the value of the state x at the next integration interval h is related to the previous value by the recursive relationship

$$x_{K+1} = x_K + hf(x, t)$$

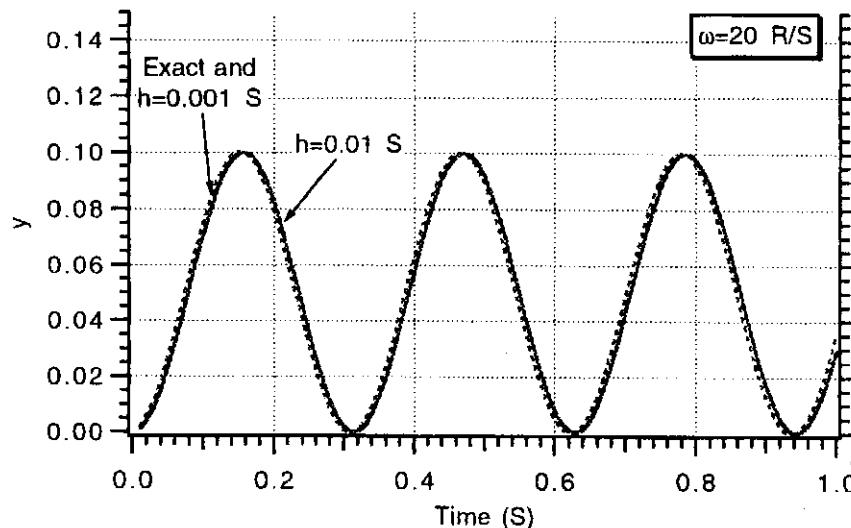


Fig. A.1 Euler integration is less accurate than second-order Runge-Kutta numerical integration.

**Listing A.1 Simulation of second-order shaping network
using Euler integration**

```

OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
W=20.
T=0.
S=0.
Y=0.
YD=0.
X=1.
H=.001
10 IF(T>1.) GOTO 999
S=S+H
YDD=W*X-W*W*Y
YD=YD+H*YDD
Y=Y+H*YD
T=T+H
IF (S <.009999) GOTO 10
S=0.
YTHEORY=(1.-COS(W*T))/W
WRITE(*,97)T,Y,YTHEORY
WRITE(1,97)T,Y,YTHEORY
GOTO 10
97 FORMAT(3F10.5)
999 CONTINUE
PAUSE
CLOSE(1)
END

```

The modified second-order shaping network using Euler integration appears in Listing A.1. The modified section of code is highlighted in boldface so that the interested reader can quickly identify changes to the original listing. Note that this program is physically smaller than the one of Listing 1.1 because the implementation of Euler integration requires fewer lines of code than does the second-order Runge-Kutta technique.

The price paid for the compact code of Euler integration is less accuracy, compared to other methods, for a given integration interval.¹ By comparing Fig. A.1 with Fig. 5.2, we can see that the Euler method of integration is less accurate than second-order Runge-Kutta for an integration step size of 0.01 s. Being prudent and using small integration step sizes is one way of getting accurate answers with Euler integration. However, sometimes Euler integration does not work at all—no matter how small the integration step size! For this reason second-order Runge-Kutta numerical integration is used in most of the listings in the text.

Pursuit Guidance

In Chapter 2 we convinced ourselves that proportional navigation could hit targets in the absence of guidance lags and acceleration saturation effects. A guidance law that is not as effective as proportional navigation, but offers simpler mechanization advantages, is known as pursuit guidance. In this guidance law an attempt is made to keep the turning rate of the missile equal to the line-of-sight rate or

$$\dot{\gamma} = \lambda$$

where γ is the missile flight path angle and λ is the line-of-sight angle. The turning rate of the missile is related to the missile acceleration n_c and velocity V_M according to

$$\dot{\gamma} = \frac{n_c}{V_M}$$

Therefore we can say that the pursuit guidance law can be expressed mathematically as

$$n_c = V_M \dot{\lambda}$$

When expressed in terms of the line-of-sight rate, pursuit guidance appears to be very similar to proportional navigation except the acceleration depends on missile velocity rather than the closing velocity and the gain is unity rather than an effective navigation ratio.

Listing 2.1, which represents a two-dimensional proportional navigation engagement simulation, was modified to have a pursuit guidance option. The slightly modified version of Listing 2.1 appears in Listing A.2. Setting the new logical PURSUIT=.FALSE. yields proportional navigation while setting PURSUIT=.TRUE. yields pursuit guidance. Note that the simulation is set up with a -20 deg heading error and an effective navigation ratio of 4 when proportional navigation is used. The WRITE statements have also been modified so that trajectory information is printed in units of kft. Important differences with the original FORTRAN version of Listing 2.1 have been highlighted in boldface in Listing A.2.

The guidance law in Listing A.2 was made a parameter, and the nominal case with a 20-deg heading error was run. Trajectory information for both proportional navigation and pursuit guidance laws are displayed in Fig. A.2. We can see that although both guidance laws appear to be mathematically similar, the resultant trajectories are vastly different. Both proportional navigation and pursuit enable the missile to hit the nonmaneuvering target, but the pursuit guidance trajectory has tremendous curvature and ends up in a tail-chase.

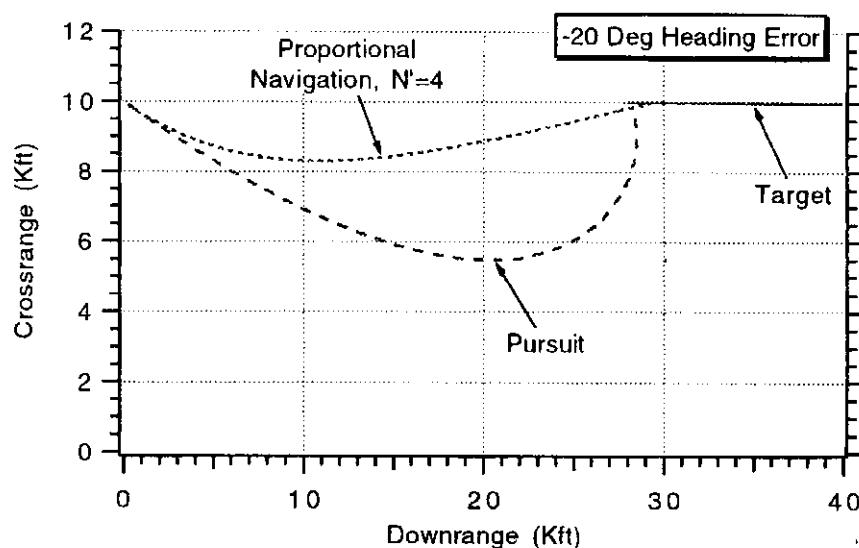


Fig. A.2 Pursuit guidance hits target but ends up in tail-chase.

**Listing A.2 Engagement simulation with pursuit guidance option
(modified Listing 2.1)**

```

INTEGER STEP
LOGICAL PURSUIT
DATA VM,VT,XNT,HEDEG,XNP/3000.,1000.,0.,-20.,4./
DATA RM1,RM2,RT1,RT2/0.,10000.,40000.,10000./
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
PURSUIT=.TRUE.
BETA=0.
VT1=-VT*COS(BETA)
VT2=VT*SIN(BETA)
HE=HEDEG/57.3
T=0.
S=0.
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
XLAM=ATAN2(RTM2,RTM1)
XLEAD=ASIN(VT*SIN(BETA+XLAM)/VM)
THET=XLAM+XLEAD
VM1=VM*COS(THET+HE)
VM2=VM*SIN(THET+HE)
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
10 IF(VC<0.)GOTO 999
IF(RTM<1000.)THEN
    H=.0002
ELSE
    H=.01
ENDIF
BETAOLD=BETA
RT1OLD=RT1
RT2OLD=RT2
RM1OLD=RM1
RM2OLD=RM2
VM1OLD=VM1
VM2OLD=VM2
STEP=1
GOTO 200
66 STEP=2
BETA=BETA+H*BETAD
RT1=RT1+H*VT1
RT2=RT2+H*VT2
RM1=RM1+H*VM1
RM2=RM2+H*VM2
VM1=VM1+H*AM1
VM2=VM2+H*AM2
T=T+H
GOTO 200
55 CONTINUE

```

(Contd.)

Listing A.2 (Continued)

```

BETA=.5*(BETAOLD+BETA+H*BETAD)
RT1=.5*(RT1OLD+RT1+H*VT1)
RT2=.5*(RT2OLD+RT2+H*VT2)
RM1=.5*(RM1OLD+RM1+H*VM1)
RM2=.5*(RM2OLD+RM2+H*VM2)
VM1=.5*(VM1OLD+VM1+H*AM1)
VM2=.5*(VM2OLD+VM2+H*AM2)
S=S+H
IF(S<.09999)GOTO 10
S=0.
WRITE(*,97)T,RT1/1000.,RT2/1000.,RM1/1000.,RM2/1000.,XNC/32.2
WRITE(1,97)T,RT1/1000.,RT2/1000.,RM1/1000.,RM2/1000.,XNC/32.2
GOTO 10
97 FORMAT(6F10.3)
200 CONTINUE
RTM1=RT1-RM1
RTM2=RT2-RM2
RTM=SQRT(RTM1**2+RTM2**2)
VTM1=VT1-VM1
VTM2=VT2-VM2
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
XLAM=ATAN2(RTM2,RTM1)
XLAMD=(RTM1*VTM2-RTM2*VTM1)/(RTM*RTM)
IF(.NOT.PURSUIT)THEN
    XNC=XNP*VC*XLAMD
ELSE
    XNC=VM*XLAMD
ENDIF
AM1=-XNC*SIN(XLAM)
AM2=XNC*COS(XLAM)
VT1=-VT*COS(BETA)
VT2=VT*SIN(BETA)
BETAD=XNT/VT
IF(STEP-1)66,66,55
999 CONTINUE
WRITE(*,*)T,RTM
PAUSE
CLOSE(1)
END

```

Figure A.3 shows that the price paid for the tremendous curvature in the pursuit guidance trajectory is very large acceleration requirements when compared to proportional navigation, which has a monotonically decreasing acceleration profile against heading error, the acceleration profile for pursuit guidance is monotonically increasing. In fact, the pursuit guidance command approaches infinity and will cause any practical guidance system to acceleration saturate.

The interested reader should conduct more experiments with both proportional navigation and pursuit in which the missile velocity is made a parameter and the

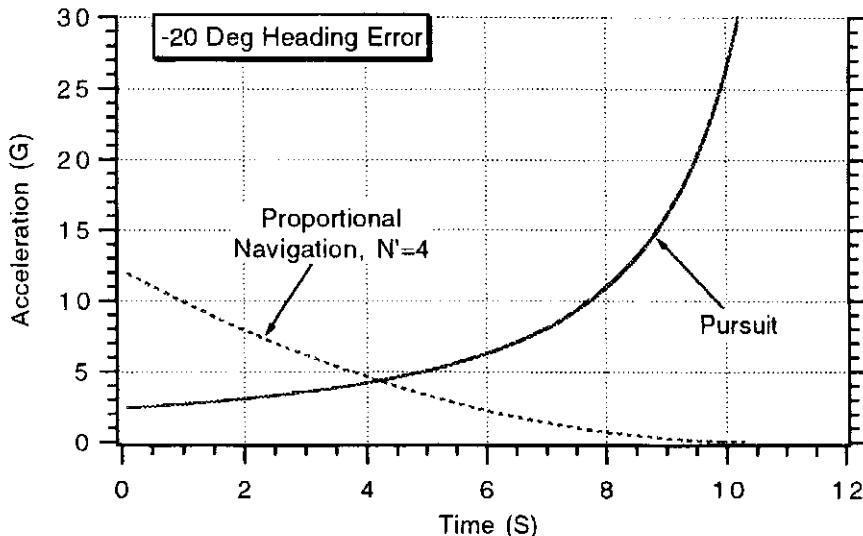


Fig. A.3 Pursuit guidance acceleration requirements are much larger than that of proportional navigation.

target is maneuvering. It will soon become apparent that pursuit guidance is not an effective guidance law against moving targets when the missile and target speeds are comparable. However, Ref. 2 shows that against stationary targets pursuit guidance can be made to work in a very cost-effective fashion.

Sensitivity of Optimal Guidance to Time to Go Errors

In evaluating the performance of the optimal guidance law, we have seen tremendous performance benefits over proportional navigation in the presence of guidance system dynamics. It has been assumed that the time to go information, required by optimal guidance, was perfect. Adjoint Listing 8.2 was modified to include scale factor SF and bias BIAS errors on the estimated time to go TGOH when the optimal guidance option was used (APN=2). An error-free time to go case would require SF=1 and BIAS=0. Time to go errors are not introduced into the proportional navigation option (APN=0) because in practice this guidance law does not require time to go but works directly on the line-of-sight rate. Statements that have been modified from the original Listing 8.2 are highlighted in boldface in Listing A.3.

Cases were rerun for proportional navigation (APN=0), optimal guidance with a time to go scale factor error of 0.9 (APN=2, SF=0.9, BIAS=0), and optimal guidance with a time to go bias error of 0.1 s (APN=2, SF=1, BIAS=0.1). The miss distance sensitivity to a 3-g target maneuver when the guidance system time constant is 1 s is displayed in Fig. A.4. We can see that both scale factor and bias errors degrade the optimal guidance performance. Figure A.4 shows that the performance of optimal guidance with a 0.9 scale factor error is worse than that of proportional navigation for flight times greater than 8 s and with a 0.1 s bias error is worse than that of proportional navigation for flight times greater than 5.5 s. (Ref. 3). Therefore time to go must be known accurately in order for optimal guidance to perform better than proportional navigation.

**Listing A.3 Adjoint simulation of optimal guidance system
(modified Listing 8.2)**

```

INTEGER STEP,APN
DATA XNT,XNP,TAU,TF,VM,HEDEG/ 96.6,4,1,10,3000.,-20./
DATA APN,BIAS,SF/2,0.,9/
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
T=0.
S=0.
TP=T+.00001
X1=0
X2=0
X3=1
X4=0
XNPP=0.
H=.01
HE=HEDEG/57.3
10 IF(TP >(TF-.00001))GOTO 999
S=S+H
X1OLD=X1
X2OLD=X2
X3OLD=X3
X4OLD=X4
STEP=1
GOTO 200
66 STEP=2
X1=X1+H*X1D
X2=X2+H*X2D
X3=X3+H*X3D
X4=X4+H*X4D
TP=TP+H
GOTO 200
55 CONTINUE
X1=(X1OLD+X1)/2+.5*H*X1D
X2=(X2OLD+X2)/2+.5*H*X2D
X3=(X3OLD+X3)/2+.5*H*X3D
X4=(X4OLD+X4)/2+.5*H*X4D
IF(S<.09999)GOTO 10
S=0.
XMNT=XNT*X1
XMHE=-VM*HE*X2
WRITE(*,97)TP,XMNT,XMHE
WRITE(1,97)TP,XMNT,XMHE
GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
TGO=TP+.00001
IF(APN.EQ.0)THEN
    C1=XNP/(TGO*TGO)
    C2=XNP/TGO
    C3=0.

```

(Contd.)

Listing A.3 (Continued)

```

C4=0.
ELSEIF(APN.EQ.1)THEN
  C1=XNP/(TGO*TGO)
  C2=XNP/TGO
  C3=.5*XNP
  C4=0.
ELSE
  TGOH=SF*TGO+BIAS
  IF(TGOH<0.)TGO=.0001
  X=TGOH/TAU
  TOP=6.*X*X*(EXP(-X)-1.+X)
  BOT1=2.*X*X*X+3.+6.*X-6.*X*X
  BOT2=-12.*X*EXP(-X)-3.*EXP(-2.*X)
  XNPP=TOP/(.0001+BOT1+BOT2)
  C1=XNPP/(TGOH*TGOH)
  C2=XNPP/TGOH
  C3=.5*XNPP
  C4=-XNPP*(EXP(-X)+X-1.)/(X*X)
ENDIF
X1D=X2+C3*X4/TAU
X2D=X3+C2*X4/TAU
X3D=C1*X4/TAU
X4D=-X4/TAU-X2+C4*X4/TAU
IF(STEP>1)66,66,55
999  CONTINUE
PAUSE
CLOSE(1)
END

```

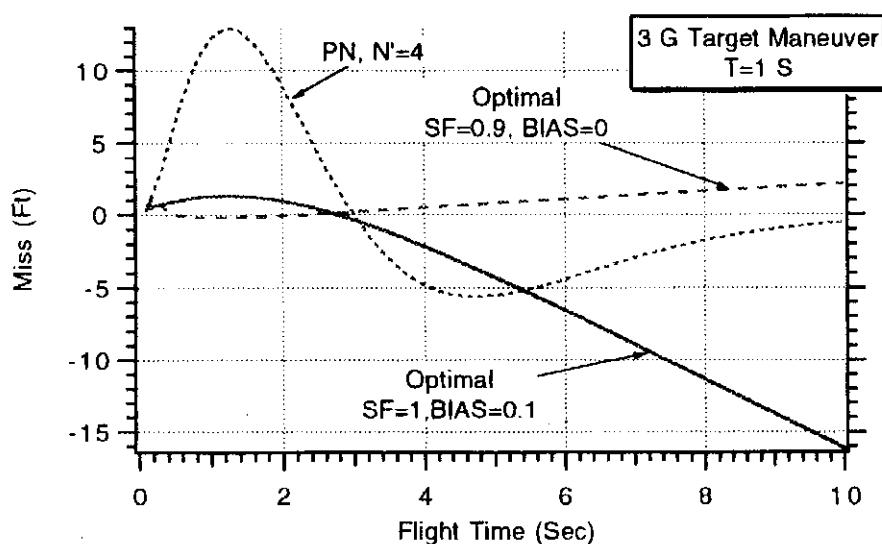


Fig. A.4 Time to go must be known accurately for optimal guidance to yield performance benefits.

Alternative Formulation of Radome Effects

In Chapters 6 and 19 the radome refraction angle r was assumed to be proportional to the missile gimbal angle θ_H or

$$r = R\theta_H$$

where R was assumed to be a constant known as the radome slope. Based on that definition, stability constraints on the minimum permissible guidance system time constant were derived in Chapter 19. Actually the radome refraction angle is proportional to the missile look angle $\lambda - \theta$ as shown in Fig. A.5. Therefore from Fig. A.5 we can express the measured line of sight angle λ^* in terms of the radome slope as

$$\lambda^* = \lambda + r = \lambda + R(\lambda - \theta) = \lambda(1 + R) - R\theta$$

Using the preceding relationship, which does not depend on the gimbal angle, the fifth-order binomial model of the guidance system with radome effects simplifies considerably. The alternative guidance system model appears in Fig. A.6.

Listing 19.2, based on the gimbal angle definition of radome slope, was modified to reflect the alternative guidance system formulation of Fig. A.6. The FORTRAN source code, representing a simulation of the alternative formulation of the fifth-order binomial model of the guidance system with radome effects, appears in Listing A.4. The modifications to the original simulation are highlighted in bold.

The nominal inputs of Listing A.4, representing a ballistic target case, were used and the guidance system time constant TAU was made a parameter. The simulation results were identical to those presented in Fig. 19.7, thus experimentally confirming that the gimbal angle and look angle definitions of the radome slope slope are approximately the same. This conclusion is also reasonable because the look angle $\lambda - \theta$ and gimbal angle θ_H only differ by an amount smaller than the boresight error (i.e., see Fig. 6.20 which shows $\lambda - \theta - \theta_H = \varepsilon - r$). Generally, the boresight error is much smaller than either the gimbal angle or look angle.

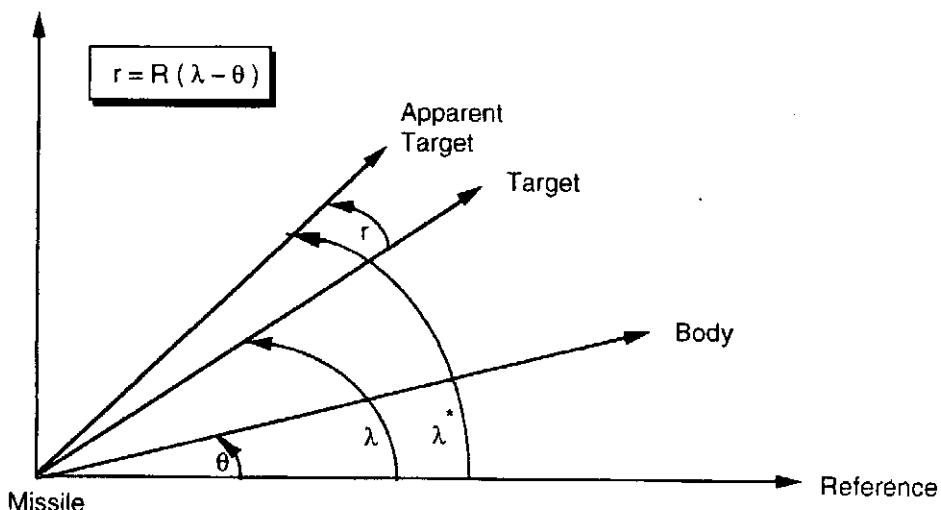


Fig. A.5 Alternative formulation of radome slope.

**Listing A.4 Alternative model of homing loop with radome effects
(modified Listing 19.2)**

```

OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
VC=9000.
XNT=32.2
XNCLIMG=7.
YIC=0.
VM=3000.
HEDEG=0.
TAU=.5
XNP=3.
TA=5.
R=-.01
TF=10.
Y=YIC
YD=-VM*HEDEG/57.3
YDIC=YD
XNL=0.
ELAMDH=0.
X4=0.
X5=0.
TH=0.
D=0.
T=0.
H=.01
S=0.
XNCLIM=XNCLIMG*32.2
10 IF(T >(TF-.0001))GOTO 999
YOLD=Y
YDOLD=YD
XNLOLD=XNL
ELAMDHOLD=ELAMDH
X4OLD=X4
X5OLD=X5
THOLD=TH
DOLD=D
STEP=1
GOTO 200
66 STEP=2
Y=Y+H*YD
YD=YD+H*YDD
XNL=XNL+H*XNLD
ELAMDH=ELAMDH+H*ELAMDHD
X4=X4+H*X4D
X5=X5+H*X5D
TH=TH+H*THD
D=D+H*DD
T=T+H
GOTO 200
55 CONTINUE
Y=.5*(YOLD+Y+H*YD)

```

(Contd.)

Listing A.4 (Continued)

```

    YD=.5*(YDOLD+YD+H*YDD)
    XNL=.5*(XNLOLD+XNL+H*XNLD)
    ELAMDH=.5*(ELAMDHOLD+ELAMDH+H*ELAMDHD)
    X4=.5*(X4OLD+X4+H*X4D)
    X5=.5*(X5OLD+X5+H*X5D)
    TH=.5*(THOLD+TH+H*THD)
    D=.5*(DOLD+D+H*DD)
    S=S+H
    IF(S<.09999)THEN
        S=0.
        WRITE(*,97)T,Y,XNC/32.2
        WRITE(1,97)T,Y,XNC/32.2
    ENDIF
    GOTO 10
97   FORMAT(3F10.3)
200   CONTINUE
        TGO=TF-T+.00001
        XLAM=Y/(VC*TGO)
XLAMS=XLAM*(1.+R)-R*TH
DD=5.*(XLAMS-D)/TAU
        ELAMDHD=.5*(DD-ELAMDH)/TAU
        XNC=XNP*VC*ELAMDH
        IF(XNC>XNCLIM)XNC=XNCLIM
        IF(XNC<-XNCLIM)XNC=-XNCLIM
        X4D=.5*(XNC-X4)/TAU
        X5D=.5*(X4-X5)/TAU
        XNLD=.5*(X5-XNL)/TAU
        THD=XNL/VM+TA*XNLD/VM
        THHD=DD-THD
        YDD=XNT-XNL
        IF(STEP-1)66,66,55
999   CONTINUE
        PAUSE
        CLOSE(1)
        END

```

Another Way of Generating Random Numbers

Readers working in a language that does not have a random number generator will have difficulty in duplicating the many Monte Carlo experiments outlined in the text. In this section code will be provided in pure FORTRAN of a Gaussian random number generator based on the multiplicative congruential method, which is fully described in Refs. 4 and 5. Listing A.5 is a modified form of Listing 4.2 in which 1000 Gaussian distributed random numbers are generated with the multiplicative congruential method and compared to the theoretical probability density function.

The nominal case was run and the resulting probability density function is compared to the Gaussian probability density function in Fig. A.7. We can see that the comparison is excellent and readers should have no problem in using this Gaussian random number generator for Monte Carlo experiments.

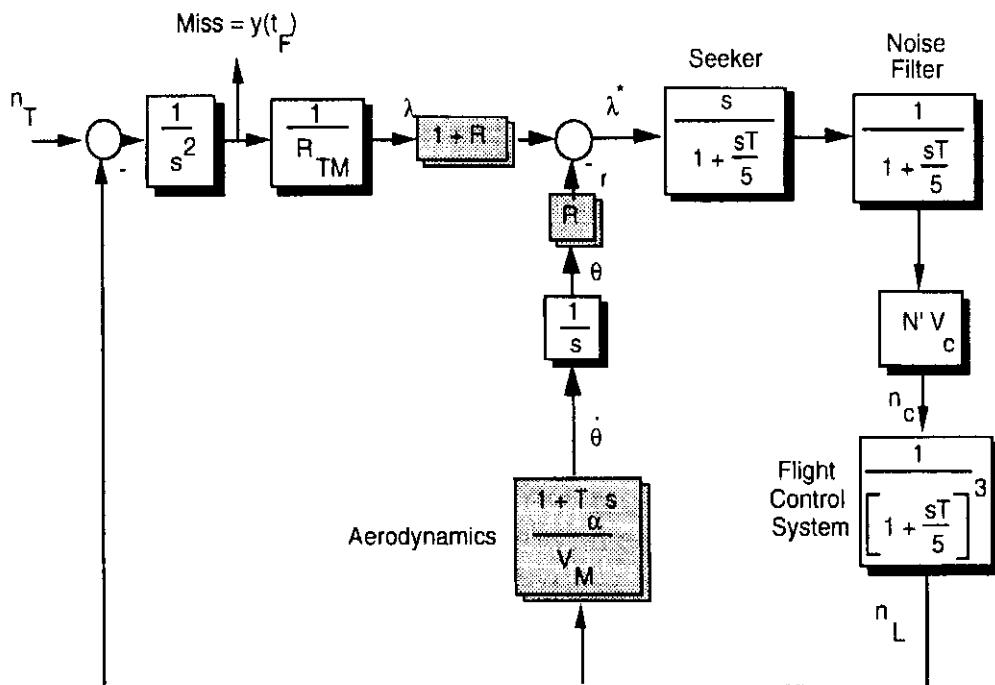


Fig. A.6 Alternative formulation of fifth-order binomial model of the guidance system with radome effects.

Simulating an Impulse

So far, when the adjoint simulation technique has been used the impulse required for the implementation of the adjoint method has been simulated by finding the appropriate initial conditions on the various integrators in the system. In some applications, in which the adjoint method is being automated, it may be advantageous to actually simulate the impulse rather than trying to develop the logic in finding the appropriate initial conditions.⁶ Listing 3.1 has been modified to show how the impulse can be simulated. Changes to the original source code have been highlighted in boldface. We can see after statement label 10 that the height of the

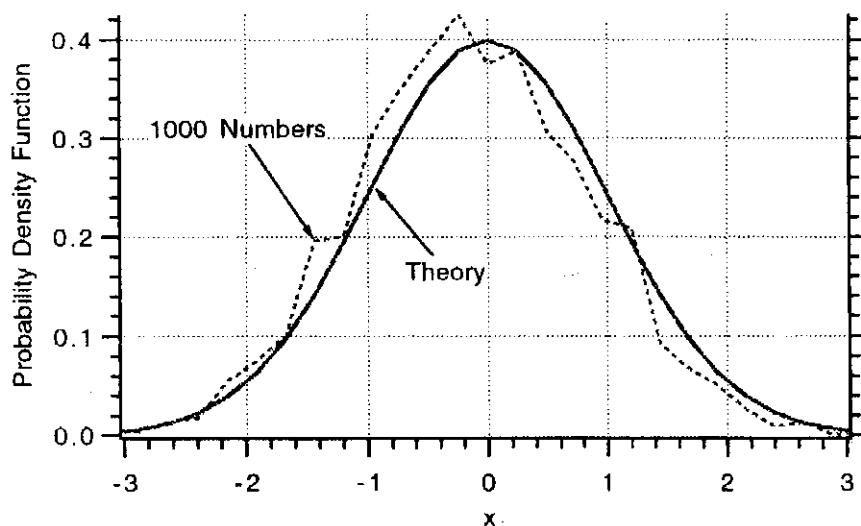


Fig. A.7 Random numbers based on multiplicative congruential method fit Gaussian distribution.

**Listing A.5 FORTRAN program to generate Gaussian probability density function using the multiplicative congruential method
(equivalent to Listing 4.2)**

```

INTEGER BIN
DIMENSION H(10000),X(10000)
OPEN(1,STATUS= 'UNKNOWN ',FILE= 'DATFIL')
XMAX=6.
XMIN=-6.
RANGE=XMAX-XMIN
TMP=1./SQRT(6.28)
N=1000
BIN=50
DO 10 I=1,N
CALL GAUSS(PZ,1.)
X(I)=PZ
10 CONTINUE
DO 20 I=1,BIN
H(I)=0
20 CONTINUE
DO 30 I=1,N
K=INT(((X(I)-XMIN)/RANGE)*BIN)+.99
IF(K<1) K=1
IF(K>BIN)K=BIN
H(K)=H(K)+1
30 CONTINUE
DO 40 K=1,BIN
PDF=(H(K)/N)*BIN/RANGE
AB=XMIN+K*RANGE/BIN
TH=TMP*EXP(-AB*AB/2.)
WRITE(*,97)AB,PDF,TH
WRITE(1,97)AB,PDF,TH
40 CONTINUE
97 FORMAT(3F10.3)
PAUSE
CLOSE(1)
END

SUBROUTINE GAUSS(X,SIG)
SAVE
SUM=0.
DO 14 J=1,12
Z=RAND(0)
SUM=SUM+Z
14 CONTINUE
X=SUM-6.
RETURN
END

FUNCTION RAND(X)
SAVE
REAL*4 REAL4,NORM,SEND

```

(Contd.)

Listing A.5 (Continued)

```

REAL*8 SEED,MULT,PROD
INTEGER*4 X,HOLDER
DATA MULT/16807.D0/
DATA NORM/4.65661288E-10/
DATA SEED/524287.D0/
IF(X.NE.0)SEED = X
PROD=SEED*MULT
HOLDER=PROD/2147483647.D0
SEED=PROD-HOLDER*2147483647.D0
REAL4=SEED
SEND=REAL4*NORM
RAND=SEND
RETURN
END

```

impulse is chosen so that the simulated impulse has unit area. The width of the impulse is half an integration interval since the differential equations are called twice with the second-order Runge-Kutta integration technique. After statement label 200 we can see that the impulse is added only once to the derivative of x_3 . When we come back to this section of code at other times the value of the simulated impulse will be zero.

We can see from Figs. A.8 and A.9 that the miss due to target maneuver and heading error is identical whether we are using Listing 3.1 where initial conditions are used for the impulse or Listing A.6 where the impulse is actually simulated. However it is important to note that when the impulse is simulated the answers are more sensitive to the integration step size than when the initial condition method is used. This means that the answers in Listing A.6 will start to diverge from the true answers sooner than the answers from Listing 3.1 if the integration interval is made larger.

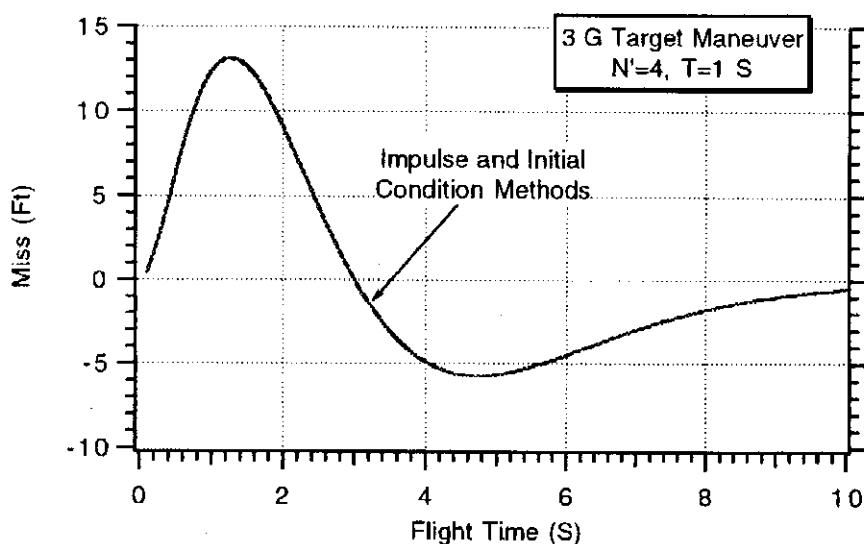


Fig. A.8 Both impulse and initial condition methods are equivalent for target maneuver disturbance.

Listing A.6 Simulating an impulse rather than using initial conditions for use in the adjoint method (equivalent to Listing 3.1)

```

INTEGER STEP
REAL IMPULSE
DATA XNT,XNP,TAU,TF,VM,HEDEG/ 96.6,4.,1.,10.,3000.,-20./
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
T=0.
S=0.
TP=TP+.00001
X1=0
X2=0
X3=0.
X4=0
H=.01
HE=HEDEG/57.3
10 IF(TP>(TF-.00001))GOTO 999
S=S+H
X1OLD=X1
X2OLD=X2
X3OLD=X3
X4OLD=X4
IF(TP<H/2.)THEN
    IMPULSE=2./H
ELSE
    IMPULSE=0.
ENDIF
STEP=1
GOTO 200
66 STEP=2
X1=X1+H*X1D
X2=X2+H*X2D
X3=X3+H*X3D
X4=X4+H*X4D
TP=TP+H
GOTO 200
55 CONTINUE
X1=(X1OLD+X1)/2+.5*H*X1D
X2=(X2OLD+X2)/2+.5*H*X2D
X3=(X3OLD+X3)/2+.5*H*X3D
X4=(X4OLD+X4)/2+.5*H*X4D
IF(S <.09999)GOTO 10
S=0.
XMNT=XNT*X1
XMHE=-VM*HE*X2
WRITE(*,97)TP,XMNT,XMHE
WRITE(1,97)TP,XMNT,XMHE
GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
X1D=X2
X2D=X3

```

(Contd.)

Listing A.6 (Continued)

```

Y1=(X4-X2)/TAU
TGO=TP+.00001
IF(STEP.EQ.2)IMPULSE=0.
X3D=XNP*Y1/TGO+IMPULSE
X4D=-Y1
IF(STEP-1)66,66,55
999  CONTINUE
PAUSE
CLOSE(1)
END

```

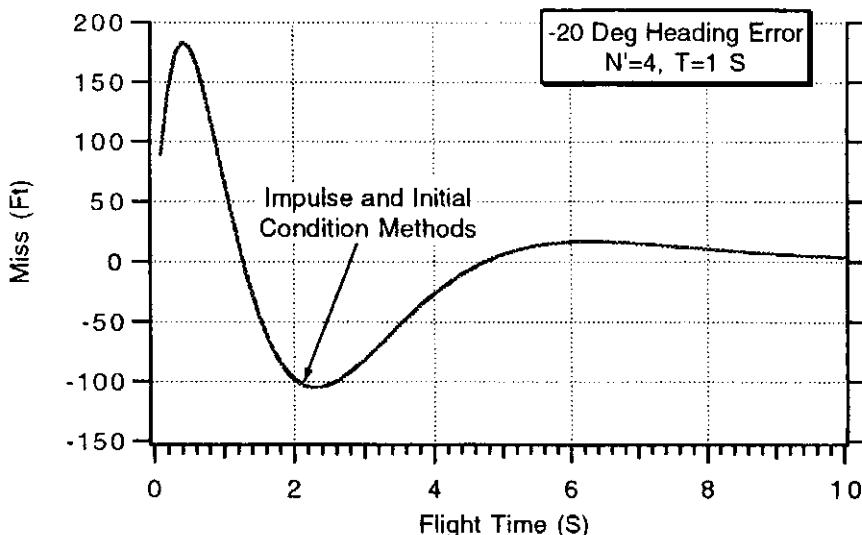


Fig. A.9 Both impulse and initial condition methods are equivalent for heading error disturbance.

Different Guidance System Distributions

So far in the text we have chosen the binomial distribution as our canonic guidance system form (i.e., all equal time constants). The canonic guidance system form was chosen for simplicity since only the total guidance system time constant had to be specified. It was never demonstrated how the binomial guidance system compared to other canonic guidance system forms in terms of the resultant performance projections.

In this section we will consider two additional guidance system forms. The only feature that the various guidance systems will have in common with the binomial guidance system is that the total guidance system time constant will be the same. The fifth-order binomial guidance system adjoint block diagram of Fig. 6.5 was modified so that other guidance system configurations could be studied by input changes and the resultant adjoint block diagram appears in Fig. A.10. We can see that if $Q_D = 0$ and all the time constants are the same (i.e., $T_1 = T_2 = T_3 = T_4 = T_5$) we have the fifth-order binomial guidance system that we have already studied. However, now we have the flexibility to make the individual time constants different.

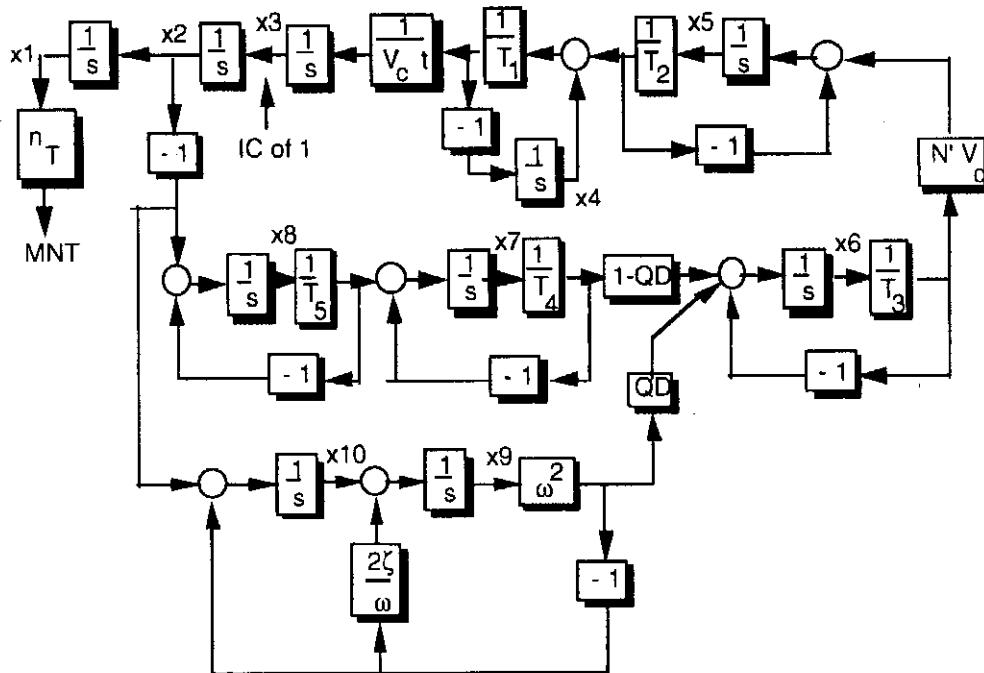


Fig. A.10 Canonic guidance system adjoint block diagram.

from one another. If $QD = 1$ in Fig. A.10, we have three real poles and a quadratic distribution for the guidance system. Again the real poles can all have different values. Note that the only disturbance in the guidance system of Fig. A.10 is a step target maneuver.

The adjoint simulation of Listing 6.1 was modified to correspond to Fig. A.10, and the resultant simulation appears in Listing A.7. The statements that have changed from Listing 6.1 have been highlighted in boldface. We can see from the listing that the only disturbance to the guidance system is a 1-g target maneuver.

For reference the first guidance system studied is the fifth-order binomial guidance system in which all the individual time constants are 0.2 s or

$$\frac{n_L}{\lambda} = \frac{N'V_c}{(1 + 0.2 \text{ s})(1 + 0.2 \text{ s})(1 + 0.2 \text{ s})(1 + 0.2 \text{ s})(1 + 0.2 \text{ s})}$$

We can see that the total time constant of the preceding expression is 1 s since

$$T = 0.2 + 0.2 + 0.2 + 0.2 + 0.2 = 1 \text{ s}$$

The next guidance system under consideration has five unequal time constants where the second time constant is twice as big as the first, the third time constant is three times as big as the first, the fourth time constant is four times bigger than the first, and the fifth time constant is five times bigger than the first or

$$\frac{n_L}{\lambda} = \frac{N'V_c}{(1 + 0.0667 \text{ s})(1 + 0.133 \text{ s})(1 + 0.2 \text{ s})(1 + 0.267 \text{ s})(1 + 0.333 \text{ s})}$$

Again we can see that the total time constant of the preceding expression is also 1 s since

$$T = 0.0667 + 0.133 + 0.2 + 0.267 + 0.333 = 1 \text{ s}$$

Listing A.7 Canonic guidance system adjoint simulation

```

INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
QD=0.
XNT=32.2
XNP=4.
T1=.0667
T2=.133
T3=.2
T4=.267
T5=.333
W=10.
Z=.7
TF=10.
VC=4000.
T=0.
S=0.
TP=T+.00001
X1=0
X2=0
X3=1
X4=0
X5=0.
X6=0.
X7=0.
X8=0.
X9=0.
X10=0.
H=.01
10 IF(TP>(TF-.00001))GOTO 999
S=S+H
X1OLD=X1
X2OLD=X2
X3OLD=X3
X4OLD=X4
X5OLD=X5
X6OLD=X6
X7OLD=X7
X8OLD=X8
X9OLD=X9
X10OLD=X10
STEP=1
GOTO 200
66 STEP=2
X1=X1+H*X1D
X2=X2+H*X2D
X3=X3+H*X3D
X4=X4+H*X4D
X5=X5+H*X5D
X6=X6+H*X6D

```

(Contd.)

Listing A.7 (Continued)

```

X7=X7+H*X7D
X8=X8+H*X8D
X9=X9+H*X9D
X10=X10+H*X10D
TP=TP+H
GOTO 200
55  CONTINUE
X1=(X1OLD+X1)/2+.5*H*X1D
X2=(X2OLD+X2)/2+.5*H*X2D
X3=(X3OLD+X3)/2+.5*H*X3D
X4=(X4OLD+X4)/2+.5*H*X4D
X5=(X5OLD+X5)/2+.5*H*X5D
X6=(X6OLD+X6)/2+.5*H*X6D
X7=(X7OLD+X7)/2+.5*H*X7D
X8=(X8OLD+X8)/2+.5*H*X8D
X9=(X9OLD+X9)/2+.5*H*X9D
X10=(X10OLD+X10)/2+.5*H*X10D
IF(S < .09999)GOTO 10
S=0.
XMNT=XNT*X1
WRITE(*,97)TP,XMNT
WRITE(1,97)TP,XMNT
GOTO 10
97  FORMAT(2F10.3)
200  CONTINUE
X1D=X2
X2D=X3
TGO=TP+.00001
X3D=(X4+X5/T2)/(VC*TGO*T1)
X4D=-(X4+X5/T2)/T1
X5D=-X5/T2+XNP*VC*X6/T3
X6D=-X6/T3+QD*W*W*X9+(1.-QD)*X7/T4
X7D=-X7/T4+X8/T5
X8D=-X8/T5-X2
X9D=X10-2.*Z*W*X9
X10D=-W*W*X9-X2
IF(STEP-1)66,66,55
999 CONTINUE
PAUSE
CLOSE(1)
END

```

The last fifth-order guidance system configuration studied is the one with three real poles and a quadratic distribution or

$$\frac{n_L}{\lambda} = \frac{N'V_c}{(1 + 0.1 s)(1 + 0.2 s)(1 + 0.56 s)[1 + (2 * 0.7/10)s + (s^2/10^2)]}$$

Again we can see that the total time constant of the preceding expression is 1 s since

$$T = 0.1 + 0.2 + 0.56 + 2 * 0.7/10 = 1 \text{ s}$$

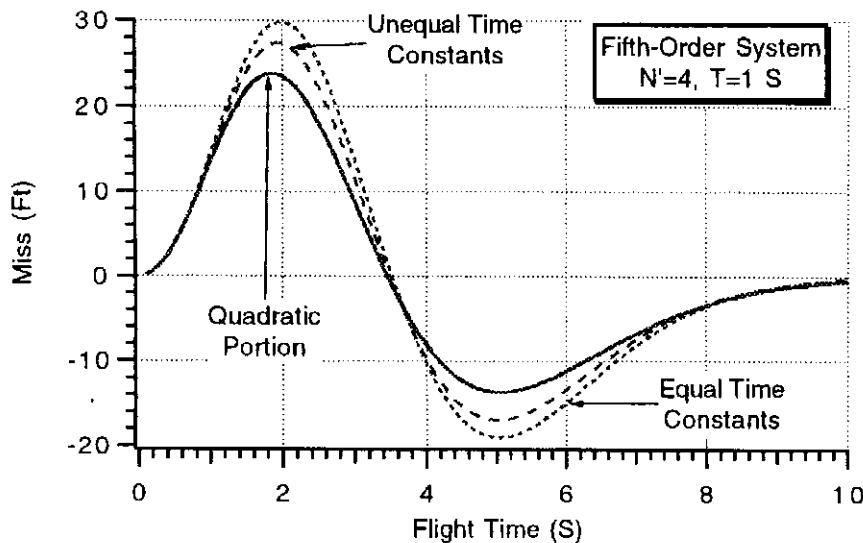


Fig. A.11 All fifth-order guidance system configurations yield approximately same answers if total guidance system time constant is same.

The adjoint simulation results appear in Fig. A.11 where the miss distance due to a 1-g step target maneuver disturbance is presented as a function of flight time. We can see that all of the miss distance curves are close to one another, indicating that the performance of all of the guidance system configurations considered is approximately the same since the total guidance system time constant of each guidance systems is 1 s. This means that the value of the total guidance system time constant is far more important than the guidance system pole distribution.

Fading Memory Filters for Booster Estimation

In Chapter 16 it was shown how two decoupled polynomial Kalman filters could be used to estimate booster acceleration from line-of-sight angle measurements if the range from the missile to target was known. In Chapter 7 we studied the properties of some constant gain fading memory filters and showed how a three-state fading memory filter could be used to estimate target acceleration. In this section we shall see how well the fading memory filters perform for estimating booster acceleration. Recall that the three-state fading memory filter was of the form

$$\begin{aligned}\hat{x}_n &= \hat{x}_{n-1} + \hat{\dot{x}}_{n-1}T_s + .5\hat{\ddot{x}}_{n-1}T_s^2 + G[x_n^* - (\hat{x}_{n-1} + \hat{\dot{x}}_{n-1}T_s + .5\hat{\ddot{x}}_{n-1}T_s^2)] \\ \hat{\dot{x}}_n &= \hat{x}_{n-1} + \hat{\dot{x}}_{n-1}T_s + \frac{H}{T_s}[x_n^* - (\hat{x}_{n-1} + \hat{\dot{x}}_{n-1}T_s + .5\hat{\ddot{x}}_{n-1}T_s^2)] \\ \hat{\ddot{x}}_n &= \hat{\dot{x}}_{n-1} + \frac{2K}{T_s^2}[x_n^* - (\hat{x}_{n-1} + \hat{\dot{x}}_{n-1}T_s + .5\hat{\ddot{x}}_{n-1}T_s^2)]\end{aligned}$$

where the gains G , H , and K are constants and are given by

$$\begin{aligned}G &= 1 - \beta^3 \\ H &= 1.5(1 - \beta)^2(1 + \beta) \\ K &= 0.5(1 - \beta)^3\end{aligned}$$

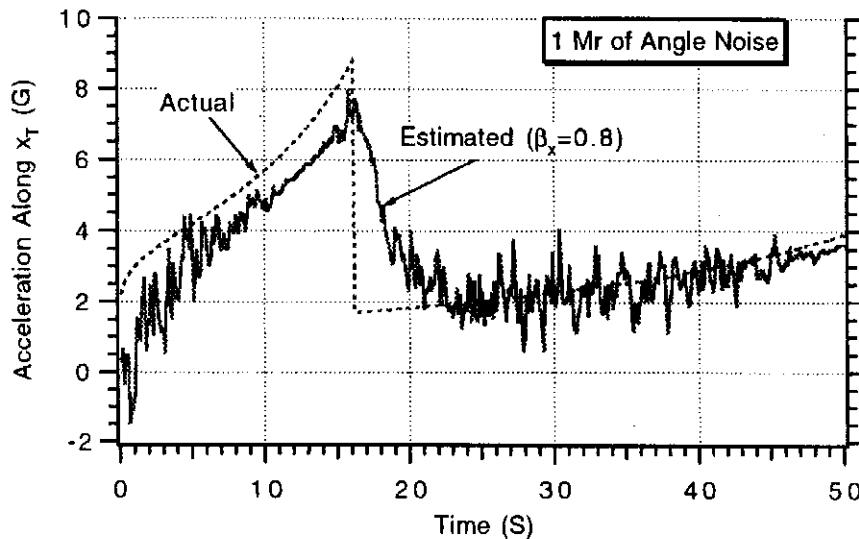


Fig. A.12 Constant gain fading memory filter tracks acceleration along x quite well.

and β is a parameter between zero and unity used to adjust the filter bandwidth. Small values of β will cause a large filter bandwidth, and large values of β will result in small filter bandwidth.

Listing 16.2 was modified to replace the two decoupled Kalman filters with two three-state fading memory filters. The modified listing appears in Listing A.8. Highlighted in boldface are all the changes required in replacing the Kalman filters with fading memory filters.

The nominal case was run in which there was 1 mr of angle noise. Because we already know from Chapter 16 that there is more noise in the y channel than in the x channel, different values of β were used to adjust the bandwidth of each filter. We can see from Figs. A.12 and A.13 that the fading memory filter acceleration estimates are comparable to the Kalman filter estimates of Figs. 16.10 and 16.11 showing that there are potentially a variety of filters that could be used for this application.

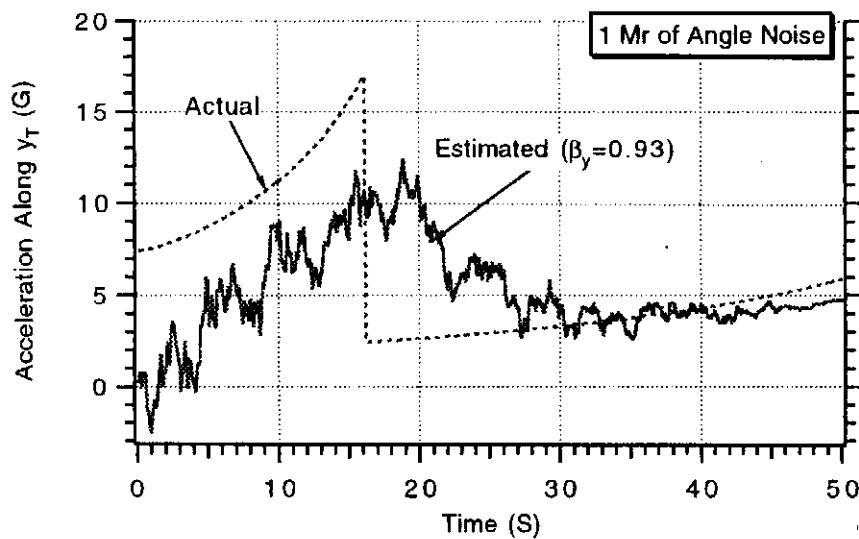


Fig. A.13 Constant gain fading memory filter does poorer job of tracking acceleration along y axis.

Listing A.8 Using fading memory filters for booster estimation

```

GLOBAL DEFINE
  INCLUDE 'quickdraw.inc'
END

IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
REAL*8 KX,KY
INTEGER STEP
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
BETAX=.8
BETAY=.93
A=2.0926E7
GM=1.4077E16
XNP=3.
TS=.1
XLONGTDEG=90.
XLONGMDEG=85.
DEGRAD=57.3
SIGLAM=.001
PREDERR=0.
XISP1=250.
XISP2=250.
XMF1=.85
XMF2=.85
WPAY=100.
DELV=20000.
DELV1=.3333*DELV
DELV2=.6667*DELV
AMAX1=20.
AMAX2=20.
XKICKDEG=80.
TOP2=WPAY*(EXP(DELV2/(XISP2*32.2))-1.)
BOT2=1/XMF2-((1.-XMF2)/XMF2)*EXP(DELV2/(XISP2*32.2))
WP2=TOP2/BOT2
WS2=WP2*(1-XMF2)/XMF2
WTOT2=WP2+WS2+WPAY
TRST2=AMAX2*(WPAY+WS2)
TB2=XISP2*WP2/TRST2
TOP1=WTOT2*(EXP(DELV1/(XISP1*32.2))-1.)
BOT1=1/XMF1-((1.-XMF1)/XMF1)*EXP(DELV1/(XISP1*32.2))
WP1=TOP1/BOT1
WS1=WP1*(1-XMF1)/XMF1
WTOT=WP1+WS1+WTOT2
TRST1=AMAX1*(WTOT2+WS1)
TB1=XISP1*WP1/TRST1
ALTNMT=0.
ALTNMM=0.
ALTT=ALTNMT*6076.
ALTM=ALTNMM*6076.
S=0.
SCOUNT=0.

```

(Contd.)

Listing A.8 (Continued)

```

XLONGM=XLONGMDEG/DEGRAD
XLONGT=XLONGTDEG/DEGRAD
XM=A*COS(XLONGM)
YM=A*SIN(XLONGM)
XT=(A+ALTT)*COS(XLONGT)
YT=(A+ALTT)*SIN(XLONGT)
XFIRSTT=XT
YFIRSTT=YT
X1T=COS(XKICKDEG/57.3)
Y1T=SIN(XKICKDEG/57.3)
XTH=XT
XTDH=X1T
XTDDH=0.
YTH=YT
YTDH=Y1T
YTDDH=0.
T=0.
TF=50.
CALLPREDICT (TF,XT,YT,X1T,Y1T,XTF,YTF
1      ,WP1,WTOT,TB1,TRST1,TB2,WP2,WTOT2,TRST2,WPAY)
YTF=YTF+PREDEERR
CALL LAMBERT(XM,YM,TF,XTF,YTF,VRXM,VRYM,XLONGM,XLONGT
1      ,ICOUNT)
X1M=VRXM
Y1M=VRYM
RTM1=XT-XM
RTM2=YT-YM
RTM=SQRT(RTM1**2+RTM2**2)
XLAM=ATAN2(RTM2,RTM1)
SIGX2=(SIGLAM*RTM*SIN(XLAM))**2
SIGX=SQRT(SIGX2)
SIGY2=(SIGLAM*RTM*COS(XLAM))**2
SIGY=SQRT(SIGY2)
VTM1=X1T-X1M
VTM2=Y1T-Y1M
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
DELV=0.
GX=1.-BETAX**3
HX=1.5*(1.-BETAX)*(1.-BETAX)*(1.+BETAX)
KX=.5*((1.-BETAX)**3)
GY=1.-BETAY**3
HY=1.5*(1.-BETAY)*(1.-BETAY)*(1.+BETAY)
KY=.5*((1.-BETAY)**3)
10   IF(VC<0.)GOTO 999
      TGO=RTM/VC
      IF(TGO>.1)THEN
          H=.01
      ELSE
          H=.0001
      ENDIF

```

(Contd.)

Listing A.8 (Continued)

```

XOLDT=XT
YOLDT=YT
X1OLDT=X1T
Y1OLDT=Y1T
XOLDM=XM
YOLDM=YM
X1OLDM=X1M
Y1OLDM=Y1M
DELVOLD=DELV
STEP=1
GOTO 200
66 STEP=2
XT=XT+H*XDT
YT=YT+H*YDT
X1T=X1T+H*X1DT
Y1T=Y1T+H*Y1DT
XM=XM+H*XDM
YM=YM+H*YDM
X1M=X1M+H*X1DM
Y1M=Y1M+H*Y1DM
DELV=DELV+H*DELVD
T=T+H
GOTO 200
55 XT=(XOLDT+XT)/2+.5*H*XDT
YT=(YOLDT+YT)/2+.5*H*YDT
X1T=(X1OLDT+X1T)/2+.5*H*X1DT
Y1T=(Y1OLDT+Y1T)/2+.5*H*Y1DT
XM=(XOLDM+XM)/2+.5*H*XDM
YM=(YOLDM+YM)/2+.5*H*YDM
X1M=(X1OLDM+X1M)/2+.5*H*X1DM
Y1M=(Y1OLDM+Y1M)/2+.5*H*Y1DM
DELV=(DELVOLD+DELV)/2+.5*H*DELVD
ALTT=SQRT(XT**2+YT**2)-A
ALTM=SQRT(XM**2+YM**2)-A
S=S+H
SCOUNT=SCOUNT+H
IF(SCOUNT.LT.(TS-.00001))GOTO 10
SCOUNT=0.
TS2=TS*TS
TS3=TS2*TS
TS4=TS3*TS
TS5=TS4*TS
SIGX2=(SIGLAM*RTM*SIN(XLAM))**2
SIGY2=(SIGLAM*RTM*COS(XLAM))**2
SIGX=SQRT(SIGX2)
SIGY=SQRT(SIGY2)
CALL GAUSS(XLAMNOISE,SIGLAM)
YTMEAS=YM+RTM*SIN(XLAM+XLAMNOISE)
XTMEAS=XM+RTM*COS(XLAM+XLAMNOISE)
XNOISE=XT-XTMEAS

```

(Contd.)

Listing A.8 (Continued)

```

YNOISE=YT-YTMEAS
RESX=XTMEAS-XTH-TS*XTDH-.5*TS2*XTDDH
XTH=GX*RESX+XTH+TS*XTDH+.5*TS2*XTDDH
XTDH=HX*RESX/TS+XTDH+TS*XTDDH
XTDDH=KX*RESX/TS2+XTDDH
RESY=YTMEAS-YTH-TS*YTDH-.5*TS2*YTDDH
YTH=GY*RESY+YTH+TS*YTDH+.5*TS2*YTDDH
YTDH=HY*RESY/TS+YTDH+TS*YTDDH
YTDDH=KY*RESY/TS2+YTDDH
XNMT=XT/6076.
YNMT=YT/6076.
XNMM=XM/6076.
YNMM=YM/6076.
ALTNMT=ALTT/6076.
CALL DISTANCE(XT,YT,XFIRSTT,YFIRSTT,DISTNMT)
ALTNMM=ALTM/6076.
CALL DISTANCE(XM,YM,XFIRSTT,YFIRSTT,DISTNMM)
X1DTG=X1DT/32.2
XTDDHG=XTDDH/32.2
Y1DTG=Y1DT/32.2
YTDDHG=YTDDH/32.2
WRITE(*,97)T,X1DTG,XTDDHG,Y1DTG,YTDDHG
WRITE(1,97)T,X1DTG,XTDDHG,Y1DTG,YTDDHG
GOTO 10
97  FORMAT(5F10.3)
200  CONTINUE
IF(T<TB1)THEN
    WGT=-WP1*T/TB1+WTOT
    TRST=TRST1
ELSEIF(T<(TB1+TB2))THEN
    WGT=-WP2*T/TB2+WTOT2+WP2*TB1/TB2
    TRST=TRST2
ELSE
    WGT=WPAY
    TRST=0.
ENDIF
AT=32.2*TRST/WGT
VEL=SQRT(X1T**2+Y1T**2)
AXT=AT*X1T/VEL
AYT=AT*Y1T/VEL
TEMBOTT=(XT**2+YT**2)**1.5
X1DT=-GM*XT/TEMBOTT+AXT
Y1DT=-GM*YT/TEMBOTT+AYT
ATPLOS=Y1DT*COS(XLAM)-X1DT*SIN(XLAM)
XDT=X1T
YDT=Y1T
RTM1=XT-XM
RTM2=YT-YM
RTM=SQRT(RTM1**2+RTM2**2)

```

(Contd.)

Listing A.8 (Continued)

```

VTM1=X1T-X1M
VTM2=Y1T-Y1M
VC=-(RTM1*VTM1+RTM2*VTM2)/RTM
TGO=RTM/VC
XLAM=ATAN2(RTM2,RTM1)
XLAMD=(RTM1*VTM2-RTM2*VTM1)/(RTM*RTM)
XNC=XNP*VC*XLAMD+.5*XNP*ATPLOS
DELVD=ABS(XNC)
AM1=-XNC*SIN(XLAM)
AM2=XNC*COS(XLAM)
TEMBOTM=(XM**2+YM**2)**1.5
X1DM=-GM*XM/TEMBOTM+AM1
Y1DM=-GM*YM/TEMBOTM+AM2
XDM=X1M
YDM=Y1M
IF(STEP-1)66,66,55
999  CONTINUE
XNMT=XT/6076.
YNMT=YT/6076.
XNMM=XM/6076.
YNMM=YM/6076.
ALTNMT=ALTT/6076.
CALL DISTANCE(XT,YT,XFIRSTT,YFIRSTT,DISTNMT)
ALTNMM=ALTM/6076.
CALL DISTANCE(XM,YM,XFIRSTT,YFIRSTT,DISTNMM)
WRITE(*,*)T,RTM,DELV
PAUSE
CLOSE(1)
END

SUBROUTINE DISTANCE(XT,YT,XF,YF,DISTNM)
SAVE
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
R=SQRT(XT**2+YT**2)
RF=SQRT(XF**2+YF**2)
A=2.0926E7
CBETA=(XT*XF+YT*YF)/(R*RF)
IF(CBETA<1.)THEN
    BETA=ACOS(CBETA)
    DISTNM=A*BETA/6076.
ELSE
    DISTNM=(XT-XF)/6076.
ENDIF
RETURN
END

SUBROUTINE PREDICT (TF,XDUM,YDUM,X1DUM,Y1DUM,XTF,YTF
1      ,WP1,WTOT,TB1,TRST1,TB2,WP2,WTOT2,TRST2,WPAY)
IMPLICIT REAL*8 (A-H)

```

(Contd.)

Listing A.8 (Continued)

```

IMPLICIT REAL*8 (O-Z)
INTEGER STEP
SAVE
H=.01
A=2.0926E7
GM=1.4077E16
T=0.
X=XDUM
Y=YDUM
X1=X1DUM
Y1=Y1DUM
10 IF(T>(TF-.00001))GOTO 999
XOLD=X
YOLD=Y
X1OLD=X1
Y1OLD=Y1
STEP=1
GOTO 200
66 STEP=2
X=X+H*XD
Y=Y+H*YD
X1=X1+H*X1D
Y1=Y1+H*Y1D
T=T+H
GOTO 200
55 X=(XOLD+X)/2+.5*H*XD
Y=(YOLD+Y)/2+.5*H*YD
X1=(X1OLD+X1)/2+.5*H*X1D
Y1=(Y1OLD+Y1)/2+.5*H*Y1D
GOTO 10
200 CONTINUE
IF(T<TB1)THEN
    WGT=-WP1*T/TB1+WTOT
    TRST=TRST1
ELSEIF(T<(TB1+TB2))THEN
    WGT=-WP2*T/TB2+WTOT2+WP2*TB1/TB2
    TRST=TRST2
ELSE
    WGT=WPAY
    TRST=0.
ENDIF
AT=32.2*TRST/WGT
VEL=SQRT(X1**2+Y1**2)
AXT=AT*X1/VEL
AYT=AT*Y1/VEL
TEMBOOTT=(X**2+Y**2)**1.5
X1D=-GM*X/TEMBOOTT+AXT
Y1D=-GM*Y/TEMBOOTT+AYT
XD=X1
YD=Y1

```

(Contd.)

Listing A.8 (Continued)

```

IF(STEP-1)66,66,55
999 CONTINUE
XTF=X
YTF=Y
RETURN
END

SUBROUTINE LAMBERT(XIC,YIC,TFDES,XF,YF,VRX,VRY,XLONGM
1 ,XLONGT,ICOUNT)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
A=2.0926E7
GM=1.4077E16
RIC=SQRT(XIC**2+YIC**2)
RF=SQRT(XF**2+YF**2)
CPHI=(XIC*XF+YIC*YF)/(RIC*RF)
PHI=ACOS(CPHI)
SPHI=SIN(PHI)
R0=RIC
PI=3.14159
DEGRAD=360./(2.*PI)
ICOUNT=1
GMIN=ATAN2((SPHI-SQRT(2.*R0*(1.-CPHI)/RF)),(1-CPHI))
GMAX=ATAN2((SPHI+SQRT(2.*R0*(1.-CPHI)/RF)),(1-CPHI))
GAM=(GMIN+GMAX)/2.
DO
  TOP=GM*(1.-COS(PHI))
  TEMP=R0*COS(GAM)/RF-COS(PHI)+GAM
  BOT=R0*COS(GAM)*TEMP
  V=SQRT(TOP/BOT)
  IF (XLONGT>XLONGM) THEN
    VRX=V*COS(PI/2.-GAM+XLONGM)
    VRY=V*SIN(PI/2.-GAM+XLONGM)
  ELSE
    VRX=V*COS(-PI/2.+GAM+XLONGM)
    VRY=V*SIN(-PI/2.+GAM+XLONGM)
  ENDIF
  XLAM=R0*V*V/GM
  TOP1=TAN(GAM)*(1-COS(PHI))+(1-XLAM)*SIN(PHI)
  BOT1P=(1-COS(PHI))/(XLAM*COS(GAM)*COS(GAM))
  BOT1=(2-XLAM)*(BOT1P+COS(GAM+PHI)/COS(GAM))
  TOP2=2*COS(GAM)
  BOT2=XLAM*((2/XLAM-1)**1.5)
  TOP3=SQRT(2/XLAM-1)
  BOT3=COS(GAM)/TAN(PHI/2)-SIN(GAM)
  TEMP=(TOP2/BOT2)*ATAN2(TOP3,BOT3)
  TF=R0*(TOP1/BOT1+TEMP)/(V*COS(GAM))
  IF((ABS(TFDES-TF)<=.00000001*TFDES).OR.ICOUNT>100)THEN
    EXIT
  ENDIF

```

(Contd.)

Listing A.8 (Continued)

```

IF(TF>TFDES)THEN
  GMAX=GAM
ELSE
  GMIN=GAM
ENDIF
IF(ICOUNT.EQ.1)THEN
  XNEXT=(GMAX+GMIN)/2.
ELSE
  XNEXT=GAM+(GAM-GOLD)*(TFDES-TF)/(TF-TOLD)
  IF(XNEXT>GMAX.OR.XNEXT<GMIN)THEN
    XNEXT=(GMAX+GMIN)/2.
  ENDIF
ENDIF
GOLD=GAM
TOLD=TF
GAM=XNEXT
ICOUNT=ICOUNT+1
REPEAT
RETURN
END

SUBROUTINE GAUSS(X,SIG)
IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER SUM
SUM=0
DO 14 J=1,6
IRAN=Random()
SUM=SUM+IRAN
14 CONTINUE
X=SUM/65536.
X=1.414*X*SIG
RETURN
END

```

Adjoint of Discrete Inputs

In Chapter 8 mixed continuous discrete adjoints were introduced, and it was shown that for discrete noise inputs Monte Carlo and adjoint miss distance performance projections were in total agreement. In this section we will show how adjoints can also be used in mixed continuous discrete systems for discrete deterministic inputs.

Consider the proportional navigation guidance system in which the second-order fading memory filter is part of the homing loop as was originally shown in Fig. 7.1. For convenience the guidance system is redrawn and shown in Fig. A.14. Note that the original input noise disturbance on the line-of-sight angle has been removed and a discrete sinusoidal bias has been added to the acceleration command.

Before we can take the adjoint of Fig. A.14, we must find the appropriate difference equation to describe the sinusoidal bias. From Table 1.2 of Chapter 1

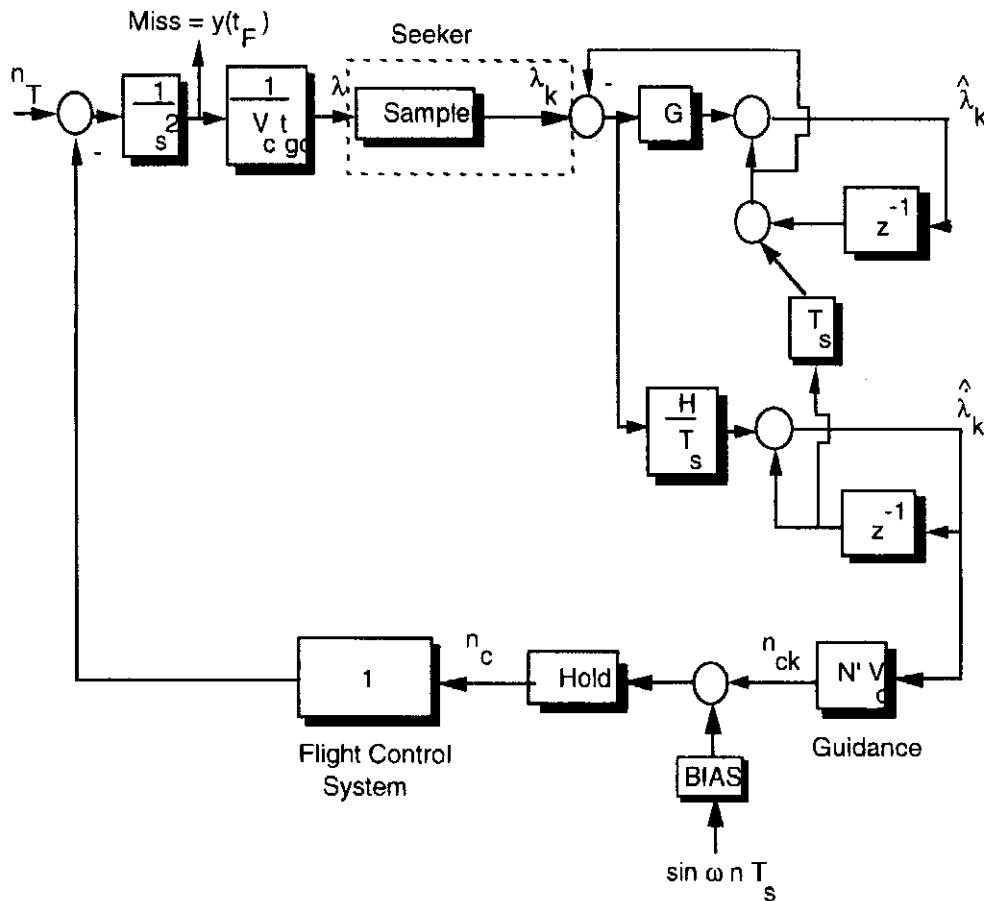


Fig. A.14 Discrete sinusoidal bias on acceleration command in homing loop with second-order fading memory filter.

we know that the Z transform of a sinusoid is given by

$$Z[\sin \omega n T_s] = \frac{z \sin \omega T_s}{z^2 - 2z \cos \omega T_s + 1}$$

Dividing the numerator and denominator by z^2 yields

$$Z[\sin \omega n T_s] = \frac{y}{x} = \frac{z^{-1} \sin \omega T_s}{1 - 2z^{-1} \cos \omega T_s + z^{-2}}$$

We know that, according to the chain rule, the preceding discrete transfer function can also be expressed as

$$\frac{y}{x} = \frac{e}{x} * \frac{y}{e}$$

Therefore, we can break the relationship into the following two equivalent transfer functions:

$$\frac{e}{x} = \frac{1}{1 - 2z^{-1} \cos \omega T_s + z^{-2}} \quad \text{and} \quad \frac{y}{e} = z^{-1} \sin \omega T_s$$

Cross multiplying both sides of the preceding two equations, rearranging terms, and converting Z transform notation to discrete notation yields the two difference

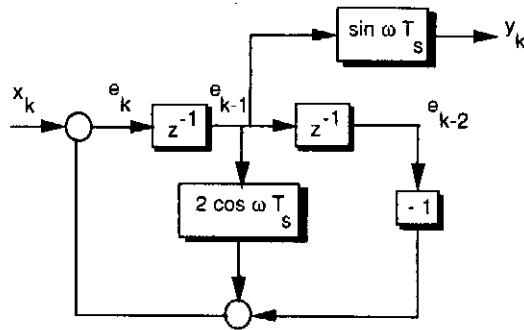


Fig. A.15 Block diagram representation of a discrete sinusoid.

equations

$$e_k = x_k + 2e_{k-1} \cos \omega T_s - e_{k-2}$$

$$y_k = \sin \omega T_s e_{k-1}$$

The preceding difference equations representing a discrete sinusoid can be converted to block diagram form and appear in Fig. A.15

The mixed continuous discrete adjoint of the Fig. A.15 can be found by reversing the signal flow and following the normal adjoint rules. The resultant adjoint block diagram for the guidance system of Fig. A.14 appears in Fig. A.16. An adjoint simulation was written based on Fig. A.16 and appears in Listing A.9.

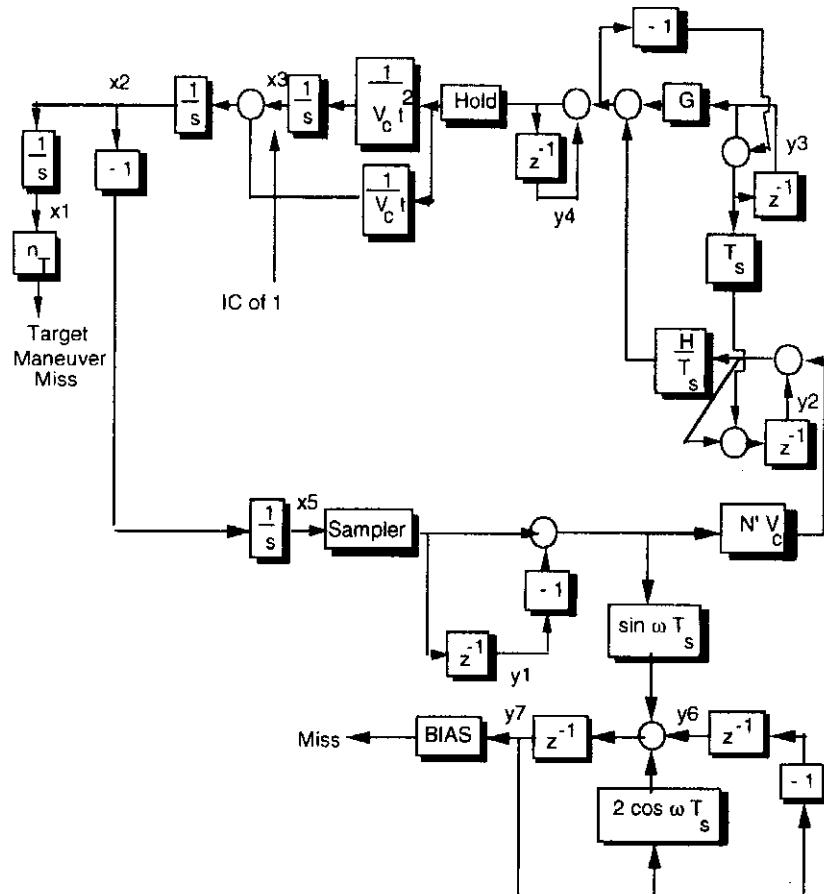


Fig. A.16 Adjoint of second-order fading memory filter in homing loop with discrete sinusoidal acceleration bias.

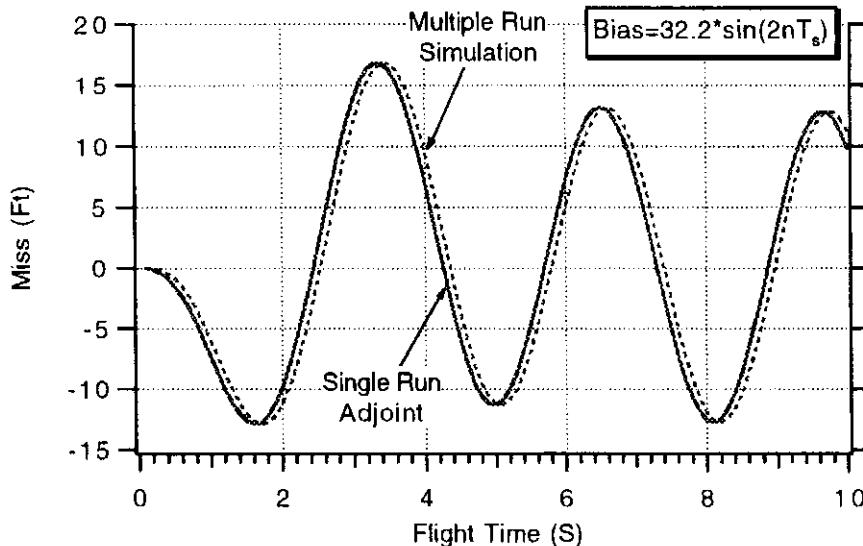


Fig. A.17 Discrete sinusoidal acceleration bias is modeled correctly in adjoint simulation.

This simulation was obtained by simply adding a few lines of code to the adjoint simulation of Listing 7.3 based on Fig. A.15 and removing the code related to the discrete noise source. The adjoint simulation output is the miss due to a sinusoidal acceleration bias where the bias amplitude is 1 g (BIAS=32.2) and bias frequency is 2 rad/s (W=2). All changes to the original adjoint simulation of Listing 7.3 are highlighted in boldface in Listing A.9.

The original mixed continuous discrete simulation Listing 7.2 was run in the brute force mode without noise to find how the miss due to a discrete sinusoidal acceleration bias depended on flight time. The simulation flight times were varied from 0.1 s to 10 s in steps of 0.1 s (i.e., 100 runs), and the miss distance performance projections appear as the dashed curve in Fig. A.17. A single adjoint run was made using Listing A.9, and the miss distance results appear as the solid line in Fig. A.17. We can see that both curves are virtually identical indicating that the discrete sinusoidal disturbance has been modeled correctly in the mixed continuous discrete adjoint.

Sampling Experiments

In Chapters 7 and 9 we conducted simplified data rate studies with both the digital fading memory and Kalman noise filters. We concluded that the miss distance due to noise and target maneuver tended to decrease as the data rate increased (i.e., sampling time decreased). For simplicity, in the data rate studies, the measurement noise standard deviation was held constant as the data rate changed. In many systems, when one gets into the details of the signal processing, it becomes readily apparent that the data rate and measurement noise standard deviation are *not* independent. In these systems the measurement noise spectral density Φ remains constant, which means that the standard deviation of the simulated digital measurement noise is proportional to the square root of the data rate (i.e., inversely proportional to the square root of the sampling time T_s) or

$$\sigma = \sqrt{\frac{\Phi}{T_s}}$$

Listing A.9 Mixed continuous discrete adjoint with discrete sinusoidal acceleration bias

```

INTEGER STEP
DATA XNT,XNP,TF,TS 96.6,3.,10.,.1
OPEN(1,STATUS=' UNKNOWN',FILE='DATFIL')
BETA=.8
BIAS=32.2
W=2.
VC=4000.
T=0.
S=0.
TP=T+.00001
X1=0
X2=0
X3=1
X5=0.
Y1OLD=0.
Y2OLD=0.
Y3OLD=0.
Y4OLD=0.
Y6OLD=0.
Y7OLD=0.
H=.01
GFILTER=1.-BETA**2
HFILTER=(1.-BETA)**2
10 IF(TP>(TF-.00001))GOTO 999
S=S+H
X1OLD=X1
X2OLD=X2
X3OLD=X3
X5OLD=X5
STEP=1
GOTO 200
66 STEP=2
X1=X1+H*X1D
X2=X2+H*X2D
X3=X3+H*X3D
X5=X5+H*X5D
TP=TP+H
GOTO 200
55 CONTINUE
X1=(X1OLD+X1)/2+.5*H*X1D
X2=(X2OLD+X2)/2+.5*H*X2D
X3=(X3OLD+X3)/2+.5*H*X3D
X5=(X5OLD+X5)/2+.5*H*X5D
IF(S<(TS-.0001))GOTO 10
S=0.
TEMP1=(X5-Y1OLD)*XNP*VC
TEMP2=HFILTER*(Y2OLD+TEMP1)/TS+GFILTER*Y3OLD
Y1NEW=X5
Y2NEW=TEMP1+Y2OLD+TS*(Y3OLD-TEMP2)

```

(Contd.)

Listing A.9 (Continued)

```

Y3NEW=Y3OLD-TEMP2
Y4NEW=Y4OLD+TEMP2
Y6NEW=-Y7OLD
Y7NEW=Y6OLD+SIN(W*TS)*(X5-Y1OLD)+2.*COS(W*TS)*Y7OLD
Y1OLD=Y1NEW
Y2OLD=Y2NEW
Y3OLD=Y3NEW
Y4OLD=Y4NEW
Y6OLD=Y6NEW
Y7OLD=Y7NEW
XMBIAS=BIAS*Y7OLD
WRITE(*,97)TP,XMBIAS
WRITE(1,97)TP,XMBIAS
GOTO 10
97  FORMAT(3F10.5)
200  CONTINUE
      TGO=TP+.00001
      X1D=X2
      X2D=X3+Y4OLD/(VC*TGO)
      X3D=(Y4OLD)/(VC*TGO*TGO)
      X5D=-X2
      IF(STEP-1)66,66,55
999  CONTINUE
      PAUSE
      CLOSE(1)
      END

```

Therefore if we double the data rate (i.e., sampling time halved), we must also increase the standard deviation of the simulated digital measurement noise by 41.4% (i.e., $2^5 = 1.414$). In this section we shall repeat the experiments of Chapters 7 and 9 to see if the miss due to digital measurement noise still decreases with increasing data rate when the measurement noise spectral density is held constant.

The miss distance adjoint program of Listing 7.3 represents the adjoint of a digital two-state fading memory filter in the homing loop. The program was modified so that a change in the data rate would cause a change in the standard deviation of the measurement noise according to the preceding relationship under the constant spectral density assumption. It was assumed that a sampling time of 0.1 s (i.e., 10 Hz data rate) corresponded to 1 mr of measurement noise. Adjoint runs were made in which the sampling time was considered a parameter. Figure A.18 shows that the standard deviation of the miss distance due to digital measurement noise still decreases with increasing data rate, although not as dramatically as was the case in Fig. 7.22.

The Monte Carlo simulation used to generate Fig. 9.12 was also modified so that the equivalent spectral density of the measurement noise would remain constant and the standard deviation of the digital measurement noise would vary with data rate. Fifty run Monte Carlo sets were run for 20 different values of flight time at data rates of 2 Hz ($T_s = 0.5$ s), 10 Hz ($T_s = 0.1$ s), and 20 Hz ($T_s = 0.05$ s). We can see from Fig. A.19 that, although the noise miss distance dependence on data rate

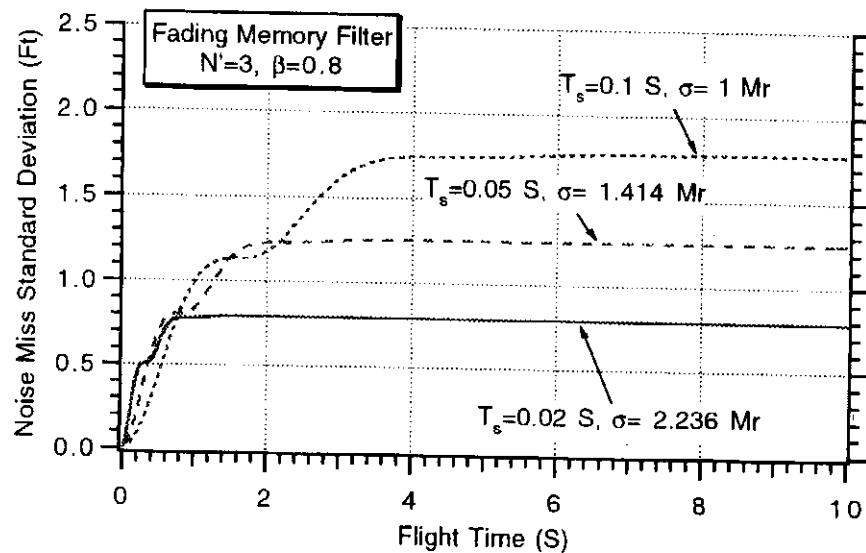


Fig. A.18 Increasing data rate still reduces miss due to digital measurement noise.

is not as dramatic as in Fig. 9.12, the miss distance still decreases with increasing data rate (decreasing sampling time).

In summary, we can say that even in cases in which it is appropriate to hold the measurement noise spectral density constant when data rate studies are conducted, the noise induced miss still decreases with increasing data rate when either a fading memory or Kalman filter is used. Both of these digital noise filters take into account information concerning the data rate so that they can both achieve good performance when the data rate changes.

Brute Force Frequency Response⁷

We have seen in Chapter 23 how we can find the frequency response of a linear system by first analytically deriving the open-loop transfer function of the system under consideration and then finding its magnitude and phase as a function of frequency. Because a great deal of algebraic manipulation of the system under

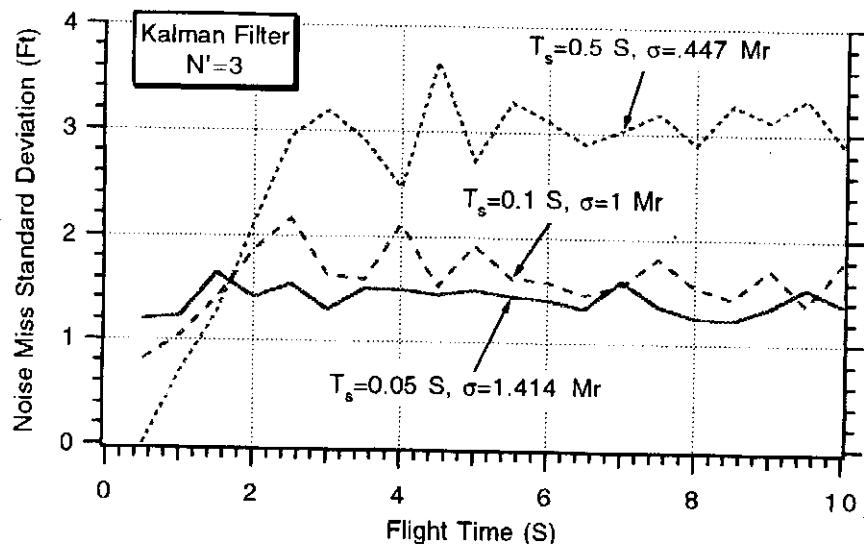


Fig. A.19 Measurement noise miss still increases with decreasing sampling rate.



Fig. A.20 Model for brute force frequency response method.

consideration is involved in this procedure, it is easy to make an error. This section will show that an independent check of the frequency response can be made by using a brute force simulation approach on the system under consideration.

Recall that when we are finding the frequency response of a system we are essentially finding the amplitude and phase of the steady-state sinusoidal output of a linear system driven by a sinusoidal input, as can be seen in Fig. A.20. It is important to note that for a frequency response the sinusoidal output magnitude and phase is found for different sinusoidal input frequencies.

Here, the sinusoidal input to the linear system of Fig. A.20 is given by

$$x = \sin \omega t$$

where ω is the sinusoidal input frequency. Because the system under consideration is linear, the output in the steady state (that is, after transients have died out) must also be a sine wave and can be expressed as

$$y = A \sin(\omega t + \phi)$$

where A is the amplitude of the sinusoidal output and ϕ is the phase angle. Because the system is linear, the frequency of the system output is the same as the frequency of the input. If we multiply the steady-state system output by a sine wave of the same frequency as the input and integrate the result over a period, we obtain P or

$$P = \int_0^{2\pi/\omega} y \sin \omega t dt = \int_0^{2\pi/\omega} A \sin(\omega t + \phi) \sin \omega t dt = \frac{A\pi}{\omega} \cos \phi$$

Similarly, if we multiply the steady-state system output by a cosine wave of the same frequency as the input and integrate the result over a period, we obtain Q or

$$Q = \int_0^{2\pi/\omega} y \cos \omega t dt = \int_0^{2\pi/\omega} A \sin(\omega t + \phi) \cos \omega t dt = \frac{A\pi}{\omega} \sin \phi$$

From the two preceding equations we can see that P and Q are related to the magnitude and phase of the system output according to

$$\sqrt{P^2 + Q^2} = \frac{A\pi}{\omega}$$

$$\phi = \tan^{-1} \frac{Q}{P}$$

In other words, information concerning the magnitude and phase of the system's steady-state sinusoidal output because of a sinusoidal input can be obtained from P and Q . To make the two preceding relationships useful we must first figure out a way to determine when we are in steady state (that is, transients have died out). If

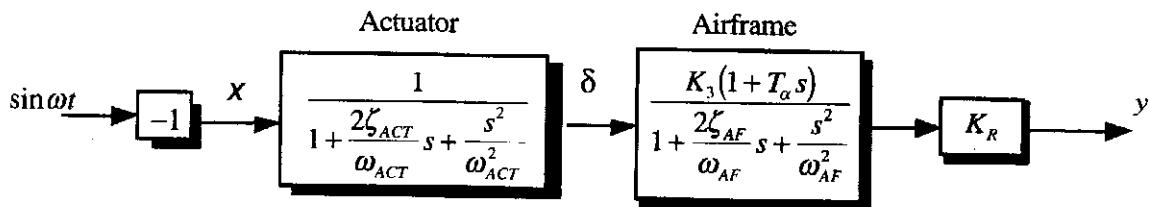


Fig. A.21 Computing frequency response by brute force.

we integrate over a period and evaluate P_0 , and then integrate again over another period and evaluate P_1 , the difference is

$$\Delta P_1 = P_1 - P_0$$

If the difference is very small or zero we know we are in steady state. If not, we evaluate other differences until we are in steady state or

$$\Delta P_2 = P_2 - P_1$$

⋮

$$\Delta P_n = P_n - P_{n-1}$$

To see if the brute-force frequency response technique works, let us again consider the rate gyro flight control system that was originally presented in Fig. 23.11 of Chapter 23. If we break the loop at the actuator (as was done in Fig. 23.15) the open-loop transfer function can be obtained by brute force from Fig. A.21. In this diagram, we will first choose a sinusoidal input frequency and then evaluate P and Q when steady state is reached. Another input frequency will be chosen and the process will be repeated. Enough input frequencies will be chosen to compute a proper frequency response.

To simulate Fig. A.21, we must first convert the transfer functions of the block diagram to differential equations. If

$$x = -\sin \omega t$$

we have already shown that by using the chain rule from calculus that the transfer function for the actuator can be converted to the differential equation

$$\ddot{\delta} = \omega_{ACT}^2 \left(x - \delta - \frac{2\zeta_{ACT}}{\omega_{ACT}} \dot{\delta} \right)$$

while the airframe transfer function becomes

$$\ddot{e} = \omega_{AF}^2 \left(\delta - e - \frac{2\zeta_{AF}}{\omega_{AF}} \dot{e} \right)$$

and the system output becomes

$$y = K_R K_3 (e + T_\alpha \dot{e})$$

The computerized method for finding the brute-force frequency response appears in Listing A.10. Notice that the integration interval is small as in other

Listing A.10 Brute-force frequency response program

```

IMPLICIT REAL*8(A-H,O-Z)
INTEGER STEP
REAL K3,KR
DATA ZACT,WAUT,K3,TA,ZAF,WAF,KR/.7,150.,-1.89,.457,.058,25.3,.1/
PI=3.1416
H=.0001
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
DO 22 I=2,160
W=10**(.025*I-1)
PERIOD=2.*PI/W
T=0.
S=0.
E=0.
ED=0.
DEL=0.
DELD=0.
P=0.
Q=0.
PPREV=0.
QPREV=0.
DELP=0.
DELQ=0.
DELPOLD=0.
DELQOLD=0.
DELDELP=100.
DELDELQ=100.
10 IF(T>20..AND.ABS(DELDELP)<.0001)GOTO 999
EOLD=E
EDOLD=ED
DELOLD=DEL
DELDOLD=DELD
POLD=P
QOLD=Q
STEP=1
GOTO 200
66 STEP=2
E=E+H*ED
ED=ED+H*EDD
DEL=DEL+H*DELD
DELD=DELD+H*DELDD
P=P+H*PD
Q=Q+H*QD
T=T+H
GOTO 200
55 CONTINUE
E=.5*(EOLD+E+H*ED)
ED=.5*(EDOLD+ED+H*EDD)
DEL=.5*(DELOLD+DEL+H*DELD)
DELD=.5*(DELDOLD+DELD+H*DELDD)
P=.5*(POLD+P+H*PD)

```

(Contd.)

Listing A.10 (Continued)

```

Q=.5*(QOLD+Q+H*QD)
S=S+H
IF(S<(PERIOD-.0001))GOTO 10
S=0.
DELP=P-PPREV
DELQ=Q-QPREV
PPREV=P
QPREV=Q
DELDELP=DELPOLD-DELP
DELDELQ=DELQOLD-DELQ
DELPOLD=DELP
DELQOLD=DELQ
GOTO 10
200 CONTINUE
X=-SIN(W*T)
DELDD=WACT*WACT*(X-DEL-2.*ZACT*DELD/WACT)
EDD=WAF*WAF*(DEL-E-2.*ZAF*ED/WAF)
Y=KR*K3*(E+TA*ED)
PD=Y*SIN(W*T)
QD=Y*COS(W*T)
IF(STEP>1)66,66,55
999 CONTINUE
PHASE=57.3*ATAN2(DELQ,DELP)
IF(PHASE>90.)PHASE=PHASE-360.
GAIN=10.*DLOG10((DELP**2+DELQ**2)*W*W/(PI*PI))
WRITE(9,*)W,GAIN,PHASE
WRITE(1,*)W,GAIN,PHASE
22 CONTINUE
PAUSE
CLOSE(1)
END

```

simulations of the rate gyro flight control system. At statement label 10 error criteria are set to ensure that P is in steady-state. For example, we assume that it takes at least 20 s for transients to die out. At the lower frequencies it will take longer for the transients to die out, and therefore we have an additional error criteria (that is, test when differences in P over a period are sufficiently small). We can see from Listing A.10 that the “do loop” increments the sinusoidal input frequency in an intelligent manner. The resultant magnitude and phase of the system output for each input frequency is printed out and also written to a file.

Listing A.10 was run for the nominal case, and Fig. A.22 displays the open-loop frequency response obtained by the method of brute force. By comparing this open-loop response to that of Fig. 23.16, we can see that both responses are identical in magnitude and phase. However, the analytic frequency response method of Chapter 23 (namely, Listing 23.2) yields the answers faster because numerical integration techniques are not involved. It is important to note that when using the analytical frequency response method, it is easier to make mistakes in setting up the program. On the other hand, the brute-force method has a longer computer

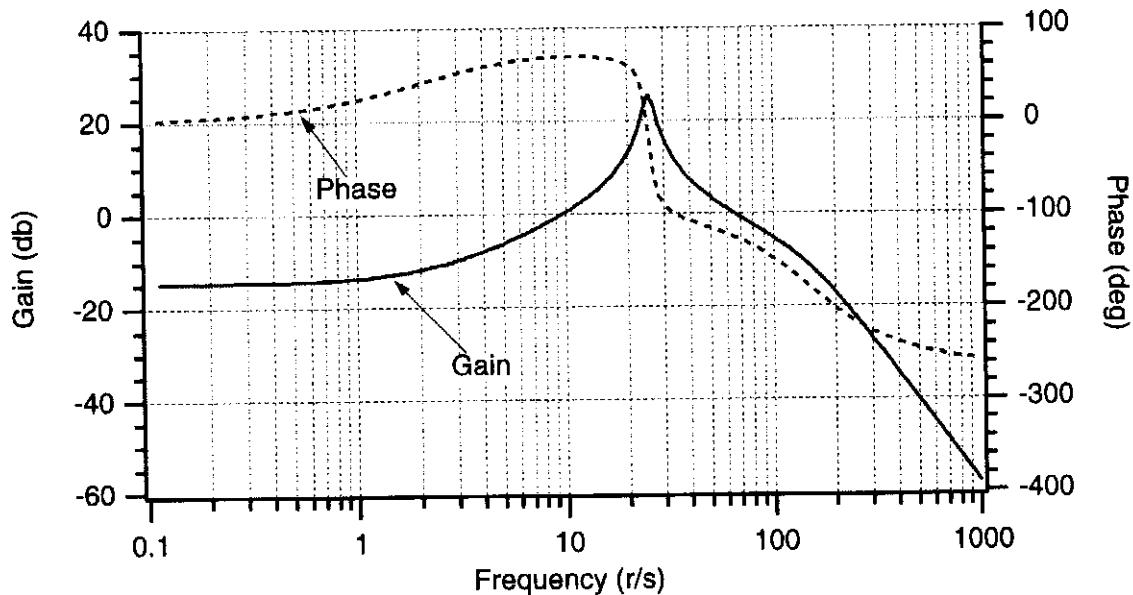


Fig. A.22 Brute-force method results are identical to those of Fig. 23.16.

running time because numerical integration is involved for each input frequency, but there is less chance of making a mistake in setting up the program because it is similar to simulation.

Table A.1 shows, for selected frequencies, how the frequency and time domain approaches compare. It can be assumed that the frequency domain answers are exact and the time domain answers are approximate. We can see that the gain is identical for both methods, but at the higher frequencies the phase angle obtained by the time domain method is in slight error.

In summary, we can say that the brute-force method for finding the open-loop frequency response of a system can be used as a useful check on analytically derived answers.

Why Stability Margins Are Important

In the three chapters on autopilot design, special emphasis was placed on having adequate phase and gain margins. To some, it might appear that system performance can be improved by reducing the stability margins. That might be true if

Table A.1 Both methods are numerically identical

$\omega, \text{r/s}$	Frequency domain		Time domain	
	Gain, db	Phase, deg	Gain, db	Phase, deg
0.112	-14.46	2.85	-14.46	2.85
0.501	-14.25	12.5	-14.25	12.5
3.981	-7.91	58.0	-7.91	58.0
33.4966	11.53	-100.43	11.53	-100.45
149.624	-11.30	-179.5	-11.30	-179.4
1000	-58.1	-257.85	-58.1	-258.53

we had perfect knowledge of the real world. In this section, we will demonstrate that if the real world is not known perfectly, having large stability margins is prudent.

In the autopilot work done so far, the force and moment equations were linearized so that we could apply linear methods of analysis in deriving autopilot gains. After autopilot gains are tested in the linearized world it is a good idea to test the autopilot design in the presence of the nonlinear force and moment equations to ensure that the autopilot still works.

To provide a practical test of stability margins, the three-loop autopilot step response simulation of Listing 24.1 was slightly modified to include the nonlinear force and moment equations. In addition, the autopilot designer's estimate of $C_{M\delta}$ is allowed to be in error according to

$$\hat{C}_{M\delta} = \text{SCALE}^* C_{M\delta}$$

If $\text{SCALE}=1$ in the preceding expression then the autopilot designer has perfect knowledge of $C_{M\delta}$. The resultant simulation is shown below in Listing A.11 with the changes from the original simulation of Listing 24.1 being highlighted in bold. Note that the simulation is set up for a missile traveling at 3000 ft/s at sea level with an autopilot time constant of 0.1 s.

The first case examined with Listing A.11 was one in which the design goals of the three-loop autopilot were a time constant of 0.1 s, a crossover frequency of 50 r/s, and a damping of 0.7 (namely, $\omega_{CR} = 50$, $\tau = 0.1$, $\zeta = 0.7$). If we run the open-loop frequency response program of Listing 24.2 with these parameters, we find that the system gain margin is 11 db while the phase margin is 46 deg. Usually, if the system has a gain margin in excess of 6 db and a phase margin in excess of 30 deg, the system is considered to have adequate stability margins. If we increase the design crossover frequency to 70 r/s, running Listing 24.2 tells us that the stability margins decrease. With the higher crossover frequency the resultant gain and phase margins are now 6.5 db and 27 deg, respectively. Thus, the system stability margins have been considerably reduced by increasing the crossover frequency. However the time domain results of Fig. A.23 give the appearance that the autopilot step response gets faster when the stability margins are reduced.

The linearized aerodynamic parameters are not usually known precisely. Let us conduct an experiment with an error in one of the aerodynamic parameters when the system stability margins are good (that is, crossover frequency is 50 r/s). For example, if we assumed that the estimated $C_{M\delta}$ was half its actual value (namely, $\text{SCALE}=0.5$) then we can see from Fig. A.24 that the system step response degrades but is still stable compared with when $C_{M\delta}$ was known perfectly (namely, $\text{SCALE}=1$).

Let us now repeat the same experiment when the system stability margins are degraded (namely, crossover frequency is 70 r/s). If we now assume that the estimated $C_{M\delta}$ was half its actual value (namely, $\text{SCALE}=0.5$) then we can see from Fig. A.25 that the system step response is on the verge of instability because it oscillates. Thus, we can see that it may be prudent to have large stability margins if there is considerable uncertainty in the aerodynamic parameters required for the derivation of the autopilot gains.

Listing A.11 Three-loop autopilot with nonlinear force and moment equations

```

FR=3.
DIAM=1.
XL=20.
CTW=0.
CRW=6.
HW=2.
CTT=0.
CRT=2.
HT=2.
XN=4.
XCG=10.
XHL=19.5
WACT=150.
ZACT=.7
TF=1.
VM=3000.
XNCG=10.
WCR=50.
ZETA=.7
TAU=.1
ALT=0.
SCALE=1.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
A=1000.
IF(ALT.LE.30000.)THEN
    RHO=.002378*EXP(-ALT/30000.)
ELSE
    RHO=.0034*EXP(-ALT/22000.)
ENDIF
WGT=1000.
XNLLIN=0.
XACC=XCG
SWING=.5*HW*(CTW+CRW)
STAIL=.5*HT*(CTT+CRT)
SREF=3.1416*DIAM*DIAM/4.
XLP=FR*DIAM
SPLAN=(XL-XLP)*DIAM+1.33*XLP*DIAM/2.
XCPN=2*XLP/3
AN=.67*XLP*DIAM
AB=(XL-XLP)*DIAM
XCPB=(.67*AN*XLP+AB*(XLP+.5*(XL-XLP)))/(AN+AB)
XCPW=XLP+XN+.7*CRW-.2*CTW
XMACH=VM/A
XIYY=WGT*(3*((DIAM/2)**2)+XL*XL)/(12*32.2)
TMP1=(XCG-XCPW)/DIAM
TMP2=(XCG-XHL)/DIAM
TMP3=(XCG-XCPB)/DIAM
TMP4=(XCG-XCPN)/DIAM
B=SQRT(XMACH**2-1)

```

(Contd.)

Listing A.11 (Continued)

```

Q=.5*RHO*VM*VM
P1=WGT*XNCG/(Q*SREF)
Y1=2+8*SWING/(B*SREF)+8*STAIL/(B*SREF)
Y2=1.5*SPLAN/SREF
Y3=8*STAIL/(B*SREF)
Y4=2*TMP4+8*SWING*TMP1/(B*SREF)+8*STAIL*TMP2/(B*SREF)
Y5=1.5*SPLAN*TMP3/SREF
Y6=8*STAIL*TMP2/(B*SREF)
P2=Y2-Y3*Y5/Y6
P3=Y1-Y3*Y4/Y6
ALFTR=(-P3+SQRT(P3*P3+4.*P2*P1))/(2.*P2)
DELTR=-Y4*ALFTR/Y6-Y5*ALFTR*ALFTR/Y6
CNA=2+1.5*SPLAN*ALFTR/SREF+8*SWING/(B*SREF)+8*STAIL/(B*SREF)
CND=8*STAIL/(B*SREF)
CMAP=2*TMP4+1.5*SPLAN*ALFTR*TMP3/SREF+
     8*SWING*TMP1/(B*SREF)
CMA=CMAP+8*STAIL*TMP2/(B*SREF)
CMD=8*STAIL*TMP2/(B*SREF)
CMD=SCALE*CMD
XMA=Q*SREF*DIAM*CMA/XIYY
XMD=Q*SREF*DIAM*CMD/XIYY
ZA=-32.2*Q*SREF*CNA/(WGT*VM)
ZD=-32.2*Q*SREF*CND/(WGT*VM)
WZ=SQRT((XMA*ZD-ZA*XMD)/ZD)
WAF=SQRT(-XMA)
ZAF=.5*WAF*ZA/XMA
XK1=-VM*(XMA*ZD-XMD*ZA)/(1845*XMA)
XK2=XK1
TA=XMD/(XMA*ZD-XMD*ZA)
XK3=1845*XK1/VM
W=(TAU*WCR*(1+2.*ZAF*WAF/WCR)-1)/(2*ZETA*TAU)
W0=W/SQRT(TAU*WCR)
Z0=.5*W0*(2*ZETA/W+TAU-WAF**2/(W0*W0*WCR))
XKC=(-W0**2/WZ**2-1.+2.*Z0*W0*TA)/(1.-2.*Z0*W0*TA+W0*W0*TA*TA)
XKA=XK3/(XK1*XKC)
XK0=-W*W/(TAU*WAF*WAF)
XK=XK0/(XK1*(1+XKC))
WI=XKC*TA*W0*W0/(1+XKC+W0**2/WZ**2)
XKR=XK/(XKA*WI)
XKDC=1.+1845./(XKA*VM)
E=0.
ED=0.
DELD=0.
DEL=0.
X=0.
T=0
H=.0001
S=0
THD=0
ALF=0

```

(Contd.)

Listing A.11 (Continued)

```

10 IF(T>(TF-.00001))GOTO 999
S=S+H
THDOLD=THD
ALFOLD=ALF
DELOLD=DEL
DELDOLD=DELD
XOLD=X
STEP=1
GOTO 200
66 STEP=2
THD=THD+H*THDD
ALF=ALF+H*ALFD
DEL=DEL+H*DELD
DELD=DELD+H*DELDD
X=X+H*XD
T=T+H
GOTO 200
55 CONTINUE
THD=.5*(THDOLD+THD+H*THDD)
ALF=.5*(ALFOLD+ALF+H*ALFD)
DEL=.5*(DELOLD+DEL+H*DELD)
DELD=.5*(DELDOLD+DELD+H*DELDD)
X=.5*(XOLD+X+H*XD)
IF(S.LE..0099999)GOTO 10
S=0.
WRITE(1,97)T,XNLG,XNCG
WRITE(*,97)T,XNLG,XNCG
GOTO 10
97 FORMAT(3F10.3)
200 CONTINUE
DELR=DEL/57.3
CN=2*ALF+1.5*SPLAN*ALF*ALF/SREF+8*SWING*ALF/(B*SREF)
1      +8*STAIL*(ALF+DELR)/(B*SREF)
CM=2*ALF*TMP4+1.5*SPLAN*ALF*ALF*TMP3/SREF
1      +8*SWING*ALF*TMP1/(B*SREF)+8*STAIL*(ALF+DELR)
2      *TMP2/(B*SREF)
THDD=Q*SREF*DIAM*CM/XIYY
XNL=32.2*Q*SREF*CN/WGT
ALFD=THD-XNL/VM
THDDEG=THD*57.3
XNLG=XNL/32.2
DELC=XKR*(X+THDDEG)
DELDD=WACT*WACT*(DELC-DEL-2.*ZACT*DELD/WACT)
XD=WI*(THDDEG+XKA*(XNLG-XNCG*XKDC))
IF(STEP-1)66,66,55
999 CONTINUE
CLOSE(1)
PAUSE
END

```

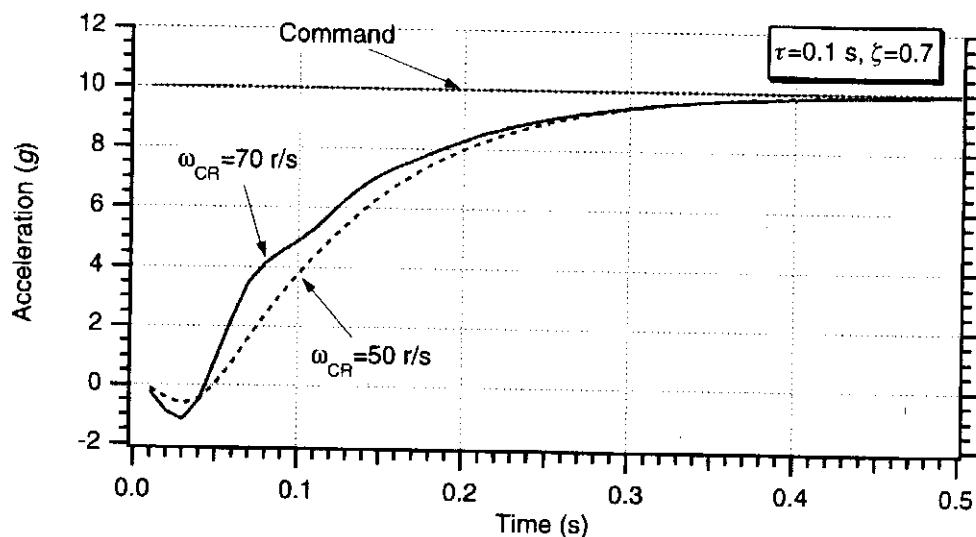


Fig. A.23 Step response appears to get faster when crossover frequency is increased.

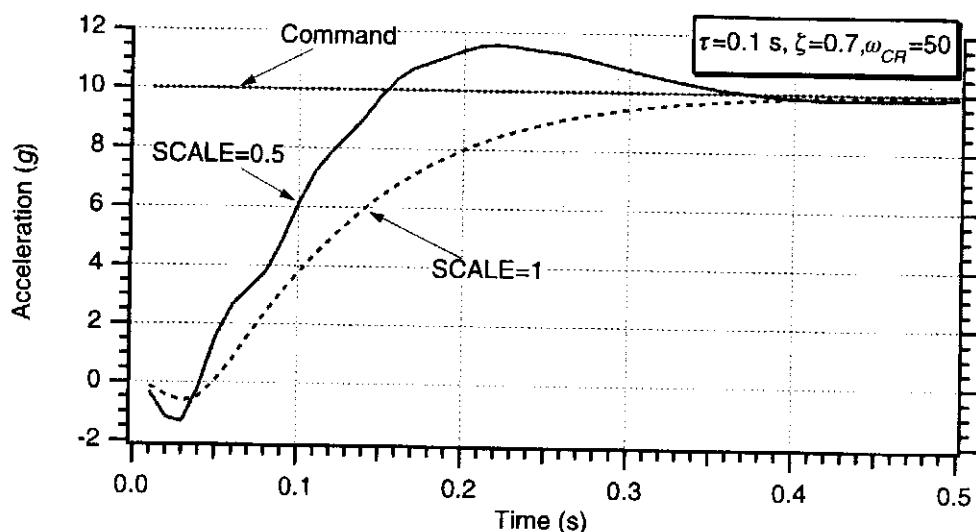


Fig. A.24 Response degrades when estimate of $C_{M\delta}$ is half of actual value if original stability margins are good.

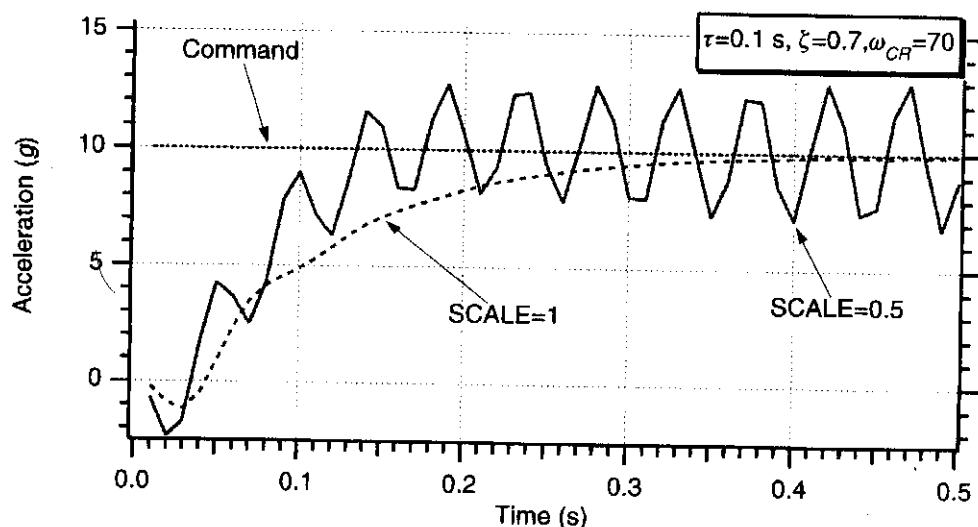


Fig. A.25 Autopilot response oscillates when estimate of $C_{M\delta}$ is half of actual value if original stability margins were reduced.

Minimum Energy Trajectories

We derived the hit equation in Chapter 12. The formula for the required velocity for an impulsive missile to travel a certain distance was given by

$$V = \sqrt{\frac{gm(1 - \cos \phi)}{a \cos \gamma [\cos \gamma - \cos(\phi + \gamma)]}}$$

where ϕ was related to the distance to be traveled, gm was a gravitational constant, a was the radius of the Earth, and γ was the flight path angle.

A minimum energy trajectory is one in which the velocity to travel a certain distance is minimized. To minimize the velocity in the preceding expression, we set the derivative of the velocity with respect to the flight path angle to zero or

$$\frac{dV}{d\gamma} = 0$$

After much algebra one can show that the flight path angle that yields minimum energy trajectories γ_{ME} is given by⁸

$$\gamma_{ME} = \frac{\pi}{4} - \frac{\phi}{4}$$

Recall that when we use Lambert guidance we specify where we are, where we want to go, and the amount of time it takes to get there. Therefore, in future uses of Lambert guidance for minimum energy trajectories we would like to specify the flight time that yields the appropriate flight path angle γ_{ME} . Recall in Chapter 12 we showed that the formula for the flight time was given by

$$t_F = \frac{a}{V \cos \gamma} \left\{ \frac{\tan \gamma (1 - \cos \phi) + (1 - \lambda) \sin \phi}{(2 - \lambda) \left[\frac{1 - \cos \phi}{\lambda \cos^2 \gamma} + \frac{\cos(\gamma + \phi)}{\cos \gamma} \right]} \right. \\ \left. + \frac{2 \cos \gamma}{\lambda \left(\frac{2}{\lambda} - 1 \right)^{1.5}} \tan^{-1} \left[\frac{\sqrt{\frac{2}{\lambda} - 1}}{\frac{\cos \gamma}{\tan \frac{\phi}{2}} - \sin \gamma} \right] \right\}$$

Listing A.12 programs the preceding formulas for the minimum energy flight path angle, velocity, and flight time. We can see that a loop appears in which the downrange distance to be traveled is varied from 2000 to 10,000 km in steps of 100 km. Listing A.12 then calculates the flight path angle that corresponds to each minimum energy trajectory.

The simulation of Listing A.12 was run, and the minimum energy flight time corresponding to each downrange to be traveled is displayed in Fig. A.26. We can see that it appears that the flight time for a minimum energy trajectory is a linear function of downrange (that is, distance to be traveled in kilometers). Superimposed on the graph is the best linear least-squares curve fit to the simulation results. We can see that if the downrange to be traveled is expressed in kilometers then the minimum energy flight time in units of seconds can be expressed as

$$t_{F_{ME}} = 452 + 0.1528 DR_{km}$$

Listing A.12 Calculating flight time for a minimum energy trajectory

```

GM=1.4077E16
A=2.0926E7
CONST=SQRT(GM/A)
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
DO 10 DISTKM=2000.,10000.,100.
PHI=DISTKM*3280./A
GAM=3.14159/4.-PHI/4.
GAMDEG=57.3*GAM
TOP=GM*(1.-COS(PHI))
TEMP=A*COS(GAM)/A-COS(PHI+GAM)
BOT=A*COS(GAM)*TEMP
V=SQRT(TOP/BOT)
XLAM=A*V*V/GM
TOP1=TAN(GAM)*(1-COS(PHI))+(1-XLAM)*SIN(PHI)
BOT1P=(1-COS(PHI))/(XLAM*COS(GAM)*COS(GAM))
BOT1=(2-XLAM)*(BOT1P+COS(GAM+PHI)/COS(GAM))
TOP2=2*COS(GAM)
BOT2=XLAM*((2/XLAM-1)**1.5)
TOP3=SQRT(2/XLAM-1)
BOT3=COS(GAM)/TAN(PHI/2)-SIN(GAM)
TEMP=(TOP2/BOT2)*ATAN2(TOP3,BOT3)
TF=A*(TOP1/BOT1+TEMP)/(V*COS(GAM))
WRITE(9,*)DISTKM,TF
WRITE(1,*)DISTKM,TF
10 CONTINUE
PAUSE
CLOSE(1)
END

```

To test the minimum energy formula for flight time, Listing 14.3 was modified to include the new formula. We can see that the distance to be traveled is 40 deg (namely, 70 deg – 30 deg) or approximately 2400 km. In this simulation, a two-stage booster flies to its intended target according to Lambert guidance. The extra code that was required for a minimum energy trajectory is highlighted in bold in Listing A.13.

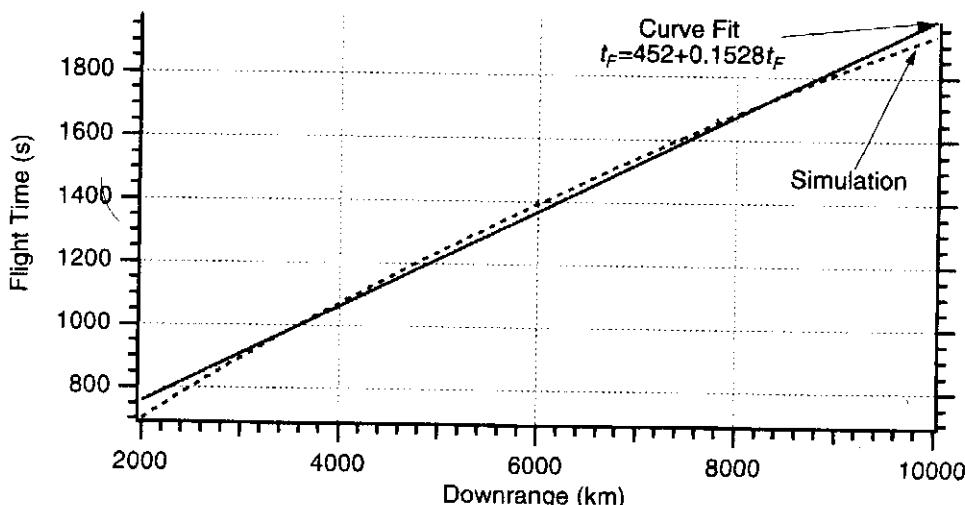


Fig. A.26 Deriving flight time formula for minimum energy trajectories.

Listing A.13 Modified form of Listing 14.3

```

IMPLICIT REAL*8 (A-H)
IMPLICIT REAL*8 (O-Z)
INTEGER STEP
LOGICAL LEFT,QBOOST
LEFT=.TRUE.
QBOOST=.TRUE.
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
OPEN(2,STATUS='UNKNOWN',FILE='LAMFIL')
XISP1=300.
XISP2=300.
XMF1=.90
XMF2=.90
WPAY=100.
DELV=20000.
DELV1=.3333*DELV
DELV2=.6667*DELV
AMAX1=20.
AMAX2=20.
TOP2=WPAY*(EXP(DELV2/(XISP2*32.2))-1.)
BOT2=1/XMF2-((1.-XMF2)/XMF2)*EXP(DELV2/(XISP2*32.2))
WP2=TOP2/BOT2
WS2=WP2*(1-XMF2)/XMF2
WTOT2=WP2+WS2+WPAY
TRST2=AMAX2*(WPAY+WS2)
TB2=XISP2*WP2/TRST2
TOP1=WTOT2*(EXP(DELV1/(XISP1*32.2))-1.)
BOT1=1/XMF1-((1.-XMF1)/XMF1)*EXP(DELV1/(XISP1*32.2))
WP1=TOP1/BOT1
WS1=WP1*(1-XMF1)/XMF1
WTOT=WP1+WS1+WTOT2
TRST1=AMAX1*(WTOT2+WS1)
TB1=XISP1*WP1/TRST1
DELVK=DELV/1000.
H=.01
T=0.
S=0.
A=2.0926E7
GM=1.4077E16
ALTNM=0.
ALT=ALTNM*6076.
ANGDEG=30.
ANG=ANGDEG/57.3
XLONGM=ANG
X=(A+ALT)*COS(ANG)
Y=(A+ALT)*SIN(ANG)
ALT=SQRT(X**2+Y**2)-A
XFIRST=X
YFIRST=Y
X1=0.

```

(Contd.)

Listing A.13 (Continued)

```

Y1=0.
AXT=0.
AYT=0.
XLONGTDEG=70.
XLONGT=XLONGTDEG/57.3
XF=A*COS(XLONGT)
YF=A*SIN(XLONGT)
CALL DISTANCE(XF,YF,XFIRST,YFIRST,DISTNM)
DISTKM=DISTNM*6076./3280.
TF=452.+.1528*DISTKM
WRITE(9,*)DISTNM,DISTKM,TF
PAUSE
10 IF(ALT<0..AND.T>10.)GOTO 999
XOLD=X
YOLD=Y
X1OLD=X1
Y1OLD=Y1
STEP=1
GOTO 200
66 STEP=2
X=X+H*XD
Y=Y+H*YD
X1=X1+H*X1D
Y1=Y1+H*Y1D
T=T+H
GOTO 200
55 CONTINUE
X=(XOLD+X)/2+.5*H*XD
Y=(YOLD+Y)/2+.5*H*YD
X1=(X1OLD+X1)/2+.5*H*X1D
Y1=(Y1OLD+Y1)/2+.5*H*Y1D
ALT=SQRT(X**2+Y**2)-A
S=S+H
IF(QBOOST)THEN
    TGOLAM=TF-T
    XLONGM=ATAN2(Y,X)
    CALL LAMBERT(X,Y,TGOLAM,XF,YF,VRX,VRY,XLONGM,
1          XLONGT,ICOUNT)
    DELX=VRX-X1
    DELY=VRY-Y1
    DEL=SQRT(DELX**2+DELY**2)
    WRITE(2,*)T,VRX,X1,VRY,Y1,DISTNM,ALTNM,ICOUNT
    IF(TRST>0..AND.DEL>500.)THEN
        AXT=AT*DELX/DEL
        AYT=AT*DELY/DEL
    ELSEIF(DEL<500.)THEN
        TRST=0.
        QBOOST=.FALSE.
        AXT=0.
        AYT=0.

```

(Contd.)

Listing A.13 (Continued)

```

        X1=VRX
        Y1=VRY
        X1OLD=X1
        Y1OLD=Y1
    ELSE
        QBOOST=.FALSE.
        AXT=0.
        AYT=0.
    ENDIF
ENDIF
99 FORMAT(7F10.3)
IF(S.LT.9.99999)GOTO 10
S=0.
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTNM)
ALTNM=(SQRT(X**2+Y**2)-A)/6076.
VELK=SQRT(X1**2+Y1**2)/1000.
GAMDEG=57.3*ATAN2(Y1,X1)
XNM=X/6076.
YNM=Y/6076.
WRITE(9,*)T,DISTNM,ALTNM,VELK
WRITE(1,*)T,DISTNM,ALTNM,VELK
GOTO 10
200 CONTINUE
IF(T<TB1)THEN
    WGT=-WP1*T/TB1+WTOT
    TRST=TRST1
ELSEIF(T<(TB1+TB2))THEN
    WGT=-WP2*T/TB2+WTOT2+WP2*TB1/TB2
    TRST=TRST2
ELSE
    WGT=WPAY
    TRST=0.
ENDIF
AT=32.2*TRST/WGT
XD=X1
YD=Y1
TEMBOT=(X**2+Y**2)**1.5
X1D=-GM*X/TEMBOT+AXT
Y1D=-GM*Y/TEMBOT+AYT
IF(STEP-1)66,66,55
999 CONTINUE
CALL DISTANCE(X,Y,XFIRST,YFIRST,DISTNM)
ALTNM=(SQRT(X**2+Y**2)-A)/6076.
VELK=SQRT(X1**2+Y1**2)/1000.
WRITE(9,*)T,DISTNM,ALTNM,VELK
WRITE(1,*)T,DISTNM,ALTNM,VELK
PAUSE
CLOSE(1)
CLOSE(2)
END

```

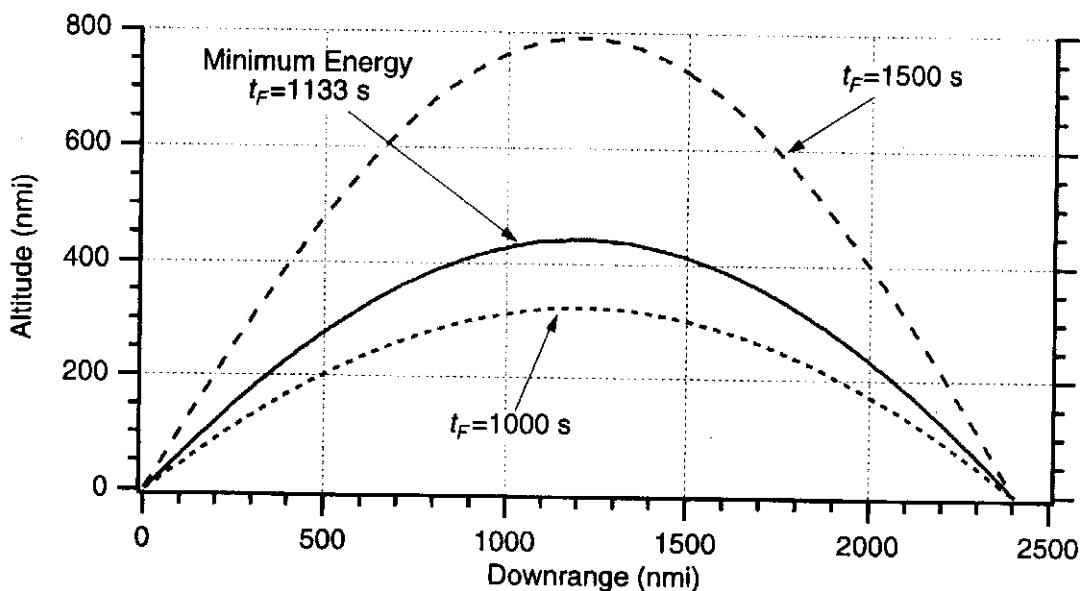


Fig. A.27 Minimum energy trajectory compared with lofted and depressed trajectories.

The first case run with Listing A.13 was for a minimum energy trajectory. In this case, the flight time turned out to be 1133 s. Trajectories corresponding to flight times of 1000 s and 1500 s were also run. We can see from Fig. A.27 that decreasing the flight time from the minimum energy value depresses the trajectory, whereas increasing the flight time from its minimum energy value lofts the trajectory.

To ensure that a flight time of 1133 s corresponds to a minimum energy trajectory, a comparison was made of the velocity profiles for each of the three trajectories. Fig. A.28 shows that the final velocity for the 1000 s trajectory is 18.86 kft/s, the final velocity for the 1500 s trajectory is 19.1 kft/s, and the final velocity for the minimum energy trajectory is 18.52 kft/s. As expected, the minimum energy trajectory yields the smallest final velocity. However, it is important to note that the velocity differential between all three trajectories is not very large.

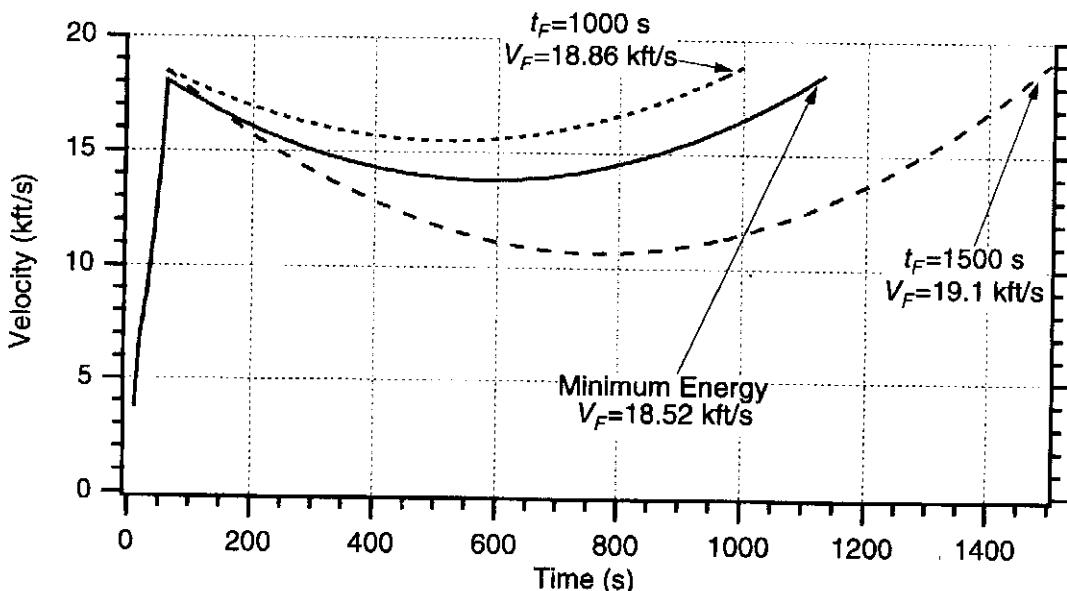


Fig. A.28 Minimum energy trajectory has least velocity at the end of the flight.

References

- ¹Rosko, J. S., *Digital Simulation of Physical Systems*, Addison-Wesley, Reading, MA, 1972.
- ²Chadwick, W. R., and Rose, C. M., "A Guidance Law for General Surface Targets," *Journal of Guidance, Control and Dynamics*, Vol. 6, Nov.-Dec. 1983, pp. 526-529.
- ³Nesline, F. W., and Zarchan, P., "A New Look at Classical Versus Modern Homing Guidance," *Journal of Guidance and Control*, Vol. 4, Jan.-Feb. 1981, pp. 78-85.
- ⁴Knuth, D. E., *The Art of Computer Programming*, Vol. 2, Addison-Wesley, Reading, MA, 1981.
- ⁵Bell, J., "Fast Random Numbers," *MacTutor*, Vol. 8, July 1992, pp. 16-22.
- ⁶Bucco, D., "Adjoints Revisited: A Software Tool To Facilitate Their Application," *Proceedings of 1997 AIAA Guidance and Control Conference*, Paper No. AIAA 97-3566, Washington, DC.
- ⁷Mitchell and Gauthier Associates (MGA) Inc., *Advanced Continuous Simulation Language (ACSL) Reference Manual*, Concord, MA, 1991, pp. A62-A69.
- ⁸Regan, F. J., and Anandakrishnan, J. M., *Dynamics of Atmospheric Re-Entry*, AIAA, Washington, DC, 1993, pp. 154-156.



Converting Programs to C

OFTEN engineers must convert their code from FORTRAN to C simply because the target computer on which the algorithms will run only has a C compiler. Although there are many books written describing the C language, most of the examples chosen to illustrate the implementation of C usually are not relevant to engineers performing scientific computations. In this appendix some simple ways of converting FORTRAN to C are presented and two examples taken from the text are used to illustrate the conversion algorithm. All of the text's source code has also been converted to C using the algorithm described in this appendix and is included on the enclosed floppy disks.

The goal of the conversion algorithm is simply to get the code to run with any C compiler and to look as similar to the text's FORTRAN code as possible. Although the resultant C code might be offensive to software professionals, it is important to note that the C compiler is more understanding and simply does not care. First the FORTRAN code is converted to lowercase and a semicolon is placed after each line. Next, all variables of the original FORTRAN listing are declared at the beginning of the new C program. Table B.1 presents some examples of how typical FORTRAN fragments of code are converted to C.

Finally, Table B.2 demonstrates how some of the FORTRAN logical expressions are converted to C. Some of the C equivalents are identical to FORTRAN extensions with many compilers.

Rather than talk more about conversion techniques, two examples are presented to show the interested reader how the conversion is accomplished and provide examples of complete C programs that not only work but give identical answers to their FORTRAN cousins.

To show how random numbers can be generated in C, the FORTRAN program to generate probability density functions (Listing 4.2) has been converted. We can see from Listing B.1 that the C `rand()` statement is equivalent to a uniform random number generator. In this example we are summing up six uniform distributions to yield a Gaussian distribution. Note that the C equivalent of a FORTRAN subroutine takes some getting use to.

A case was run with Listing B.1 in which there were 1000 random numbers. Figure B.1 shows that the fit to the theoretical bell-shaped Gaussian distribution is equivalent in accuracy to the results obtained with the FORTRAN code (i.e., see Fig. 4.6). Therefore we can conclude that the random number generator supplied by the C compiler vendor is as good as the one supplied by the FORTRAN compiler vendor.

Table B.1 FORTRAN fragments of code converted to C

FORTRAN	C
INTEGER STEP REAL*4 X REAL*8 Y	int step; float x; double y;
OPEN(1,STATUS='UNKNOWN', FILE='DATFIL')	FILE *fptr; fptr=fopen("DATFIL","w");
WRITE(*,97)T,X,Y	printf("%10.3f %10.3%10.3f\n", t,x,y);
WRITE(1,97)T,X,Y	fprintf(fptr,"%10.3f %10.3f %10.3f\\n", t,x,y);
97 FORMAT(3F10.3)	
IF(S<.0999) GOTO 10 X=0. Y=3.	if(s<.0999)goto L10; x=0.; L10: Y=3.;
10 DO 100 I=1,10 X=A Y=X/10. CONTINUE	for(i=1;i<=10;i=i+1){ x=a; y=x/10.; }
100 IF(I.EQ.1)THEN X=A Y=B ELSEIF(I.EQ.2.OR.I.EQ.3.)THEN X=C Y=D ELSE X=E Y=F ENDIF	if(i==1){ x=a; y=b; } else if(i==2 i==3){ x=c; y=d; } else{ x=e; y=f; }

**Listing B.1 C program to generate probability density function
(equivalent to Listing 4.2)**

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
void gauss(float sig, float *xnoise);
main( )
{
    int i,x,n,j,bin,k;
```

(Contd.)

Listing B.1 (Continued)

```

float y,sum,xmax,xmin,range,tmp,pdf,ab,th,sig,xnoise;
float h[2000],z[2000];
FILE *fptr;
sig=1.;
n=1000;
xmax=6.;
xmin=-6.;
bin=50;
range=xmax-xmin;
tmp=1./sqrt(6.28);
fptr=fopen("DATFIL","w");
for (i=1; i<=n; i=i+1){
    gauss(sig, &xnoise);
    z[i]=xnoise;
}
for (i=1; i<=bin; i=i+1){
    h[i]=0.;
}
for (i=1; i<=n; i=i+1){
    k=(((z[i]-xmin)/range)*bin)+.99;
    if(k<1)
        k=1;
    if(k>bin)
        k=bin;
    h[k]=h[k]+1;
}
for (k=1; k<=bin; k=k+1) {
    pdf=(h[k]/n)*bin/range;
    ab=xmin+k*range/bin;
    th=tmp*exp(-ab*ab/2.);
    printf("%10.3f %10.3f %10.3f\n", ab,pdf,th);
    fprintf(fptr,"%10.3f %10.3f %10.3f\n", ab,pdf,th);
}
fclose (fptr);
}

void gauss(float sig,float *xnoise)
{
    float sum,x,y,temp;
    int j;
    sum=0.;
    for (j=1; j<=6; j=j+1){
        x=rand();
        y=x/32767.;
        sum=sum+y;
    }
    temp=sum-3.;
    *xnoise=1.414*temp*sig;
}

```

Listing B.2 C version of efficient Lambert subroutine (Listing 14.2)

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
void lambert(double xm,double ym,double tfdes,double xt,double yt,
             double*vrx,double*vry,double xlongm,double xlongt);
main( )
{
    double xlongmdeg,xlongtdeg,altnmt,altnmm,tfdes,pi,degrad,a,gm,altt;
    double altm,xlongm,xlongt,xm,ym,xt,yt,vrx,vry;
    xlongmdeg=45.;
    xlongtdeg=90.;
    altnmt=0.;
    altnmm=0.;
    tfdes=1000.;
    pi=3.14159;
    degrad=360./(2.*pi);
    a=2.0926E7;
    gm=1.4077E16;
    altt=altnmt*6076.;
    altm=altnmm*6076.;
    xlongm=xlongmdeg/degrad;
    xlongt=xlongtdeg/degrad;
    xm=(a+alm)*cos(xlongm);
    ym=(a+alm)*sin(xlongm);
    xt=(a+altt)*cos(xlongt);
    yt=(a+altt)*sin(xlongt);
    lambert(xm,ym,tfdes,xt,yt, &vrx, &vry,xlongm,xlongt);
}
void lambert(double xm,double ym,double tfdes,double xt,double yt,
             double*vrx,double*vry,double xlongm,double xlongt)
{
    FILE *fptr;
    double a,gm,ric,rf,cphi,phi,sphi,r0,pi,degrad,gmin,gmax,gam;
    int icount;
    double top,temp,bot,v,xlam,top1,bot1p,bot1,top2,bot2;
    double top3,bot3,tf,xnext,told,gold;
    fptr=fopen("DATFIL","w");
    a=2.0926E7;
    gm=1.4077E16;
    ric=sqrt(xm*xm+ym*ym);
    rf=sqrt(xt*xt+yt*yt);
    cphi=(xm*xt+ym*yt)/(ric*rf);
    phi=atan2(sqrt(1.-cphi*cphi),cphi);
    sphi=sin(phi);
    r0=ric;
    pi=3.14159;
    degrad=360./(2.*pi);
    gmin=atan2((sphi-sqrt(2.*r0*(1.-cphi)/rf)),(1.-cphi));
}

```

(Contd.)

Listing B.2 (Continued)

```

gmax=atan2((sphi+sqrt(2.*r0*(1.-cphi)/rf)),(1.-cphi));
gam=(gmin+gmax)/2.;

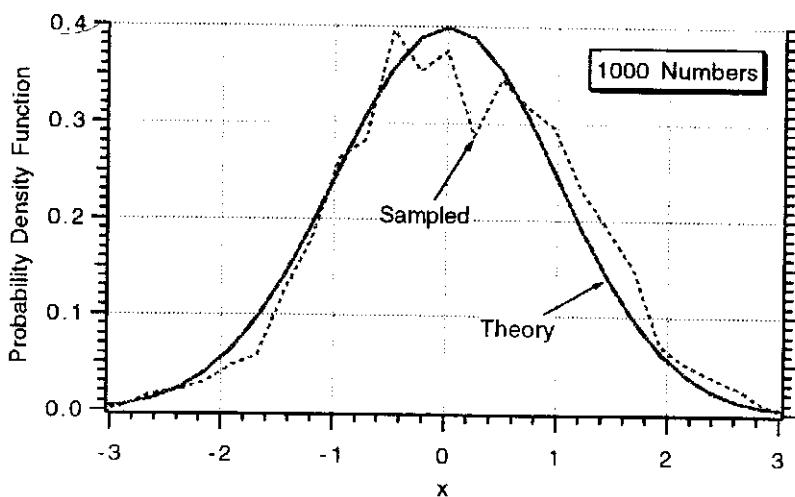
for (icount=1; icount<=100; icount=icount+1){
    top=gm*(1.-cos(phi));
    temp=r0*cos(gam)/rf-cos(phi+gam);
    bot=r0*cos(gam)*temp;
    v=sqrt(top/bot);
    if(xlongt>xlongm){
        *vrx=v*cos(pi/2.-gam+xlongm);
        *vry=v*sin(pi/2.-gam+xlongm);
    }
    if(xlongt<=xlongm){
        *vrx=v*cos(-pi/2.+gam+xlongm);
        *vry=v*sin(-pi/2.+gam+xlongm);
    }
    xlam=r0*v*v/gm;
    top1=tan(gam)*(1.-cos(phi))+(1.-xlam)*sin(phi);
    bot1p=(1.-cos(phi))/(xlam*cos(gam)*cos(gam));
    bot1=(2.-xlam)*(bot1p+cos(gam+phi)/cos(gam));
    top2=2.*cos(gam);
    bot2=xlam*pow((2./xlam-1.),1.5);
    top3=sqrt(2./xlam-1.);
    bot3=cos(gam)/tan(phi/2.)-sin(gam);
    temp=(top2/bot2)*atan2(top3,bot3);
    tf=r0*(top1/bot1+temp)/(v*cos(gam));
    printf("%5d %10.4f %8.2f %8.2f %9.4f\n",icount,57.3*gam,
           *vrx,*vry,tf);
    fprintf(fp,"%5d %10.4f %8.2f %8.2f %9.4f\n",icount,57.3*gam,
           *vrx,*vry,tf);
    if(fabs(tfdes-tf)<=.00000001*tfdes))
        goto L100;
    if(tf>tfdes)
        gmax=gam;
    else
        gmin=gam;
    if(icount==1)
        xnext=(gmax+gmin)/2.;
    else{
        xnext=gam+(gam-gold)*(tfdes-tf)/(tf-told);
        if(xnext>gmax || xnext<gmin)
            xnext=(gmax+gmin)/2.;
    }
    gold=gam;
    told=tf;
    gam=xnext;
}
L100:
fclose (fp);
}

```

Table B.2 Some logical expressions in FORTRAN and C

FORTRAN	.EQ.	.GE.	.LE.	.AND.	.OR.	.NE.
C	==	>=	<=	&&		!=

To give the interested reader more practice, Listing 14.2 was simply converted to C. The example was chosen because all the changes are not obvious plus we now have to work with double-precision arithmetic. We can see from Listing B.2 that the C statement *double* is used to convert variables to double-precision. The DO...REPEAT loop with an EXIT statement in the FORTRAN version of Listing 14.2 has also been converted to the inelegant but effective *goto* statement in Listing B.2. The reader will also note that raising a number to a power in C requires the *pow* statement. This program yields identical answers to its FORTRAN counterpart.

**Fig. B.1 C compiler also yields good random numbers.**

Converting Programs to MATLAB®

MANY engineers prefer to work in MATLAB® because of its friendly environment and powerful plotting statements. In this appendix some simple ways of converting the FORTRAN code to MATLAB® line by line are presented, and several examples taken from the text are used to illustrate the conversion algorithm. All of the text's source code has also been converted to MATLAB® M-files in this way and is included on the enclosed floppy disks. In addition, the MATLAB® code allows the reader to obtain the figures in the book directly without the need for porting the data to other plotting packages. The structure of the translation was provided by Dr. Michael Dutton of the Weapons Systems Division, Defence Science and Technology Organisation (DSTO) in Salisbury, South Australia.

The method of translation adopted was chosen so that the resultant M-files would closely resemble the original FORTRAN listings. This was done to facilitate a line-by-line conversion and to allow the code to run under earlier MATLAB® versions (including the student editions available on both PC and Macintosh platforms) without the need for invoking special library functions and various toolboxes.

MATLAB® is an interpretative language, and therefore its code in general will run much slower than compiled source code written in either C or FORTRAN. In particular the use of FOR or WHILE loops are normally avoided in MATLAB® because their use dramatically reduces program run-time performance. In this sense no serious attempt was made to optimize the running time of the M-files. However, basic measures such as pre-allocation of arrays, the use of "vectoring" or matrix operations have been taken, where appropriate.

Unfortunately, if the MATLAB® code is to closely mirror the FORTRAN listings, the use of FOR and/or WHILE statements becomes necessary in order that the GOTO statements (not supported under MATLAB®) can be removed from the original FORTRAN listings. This means that some of the programs will run more than an order of magnitude slower than their FORTRAN or C counterparts. In general those programs that run in the brute force mode (i.e., Monte Carlo) or those simulations having a small integration interval or long flight times will have excessive MATLAB® running times. In those cases the number of runs made has been reduced and/or the integration step size has been increased to get acceptable MATLAB® running times. Comment statements have been added to the M-files in these cases to alert the reader that the code is different than the FORTRAN listing shown in the text.

As has already been mentioned, the FORTRAN code in the text involving the integration of differential equations makes use of GOTO statements. Because MATLAB® has eliminated the GOTO statement, an alternative way of structuring

the FORTRAN programs is first required before a MATLAB® line-by-line translation can begin. For example, Listing C.1 eliminates the FORTRAN GOTO statements of Listing 3.1 by making use of the WHILE statement (which is an Abssoft extension to FORTRAN 77).

The MATLAB® equivalent of Listing C.1 appears in Listing C.2a. We can see that the MATLAB® listing is identical to the FORTRAN listing except that semicolons have been added to the end of each line and that the output has been saved as individual arrays. At the end of the program the resultant data is plotted by using MATLAB® statements and the data is sent to an ASCII file called DATFIL. It is important to note that use has not been made of the special MATLAB® statements for the integration of differential equations. In general, these specialized statements will enable the program to run more rapidly because a variable integration step size is used by the ODE solvers algorithm.

An example of the use of the optimized code using a MATLAB® ordinary differential equation solver "ode23" is given for comparison. Listing C.2b shows how the WHILE loop can be avoided and the library function ode23 can be used to optimize run time. The resultant code is written as two separate M-files, one that contains the differential equation to be solved and the other as the calling routine. This code runs much faster than Listing C.2a and produces the same results. There is also no speed penalty when using multiple M-files. Notice, however, that the code no longer resembles the original listing and in this case uses a variable integration step size algorithm.

The resultant plots from C3L1.M (or from S3L1.M and DE3L1.M) appear in Figs. C.1 and C.2. A careful examination of the figures reveals that these plots are

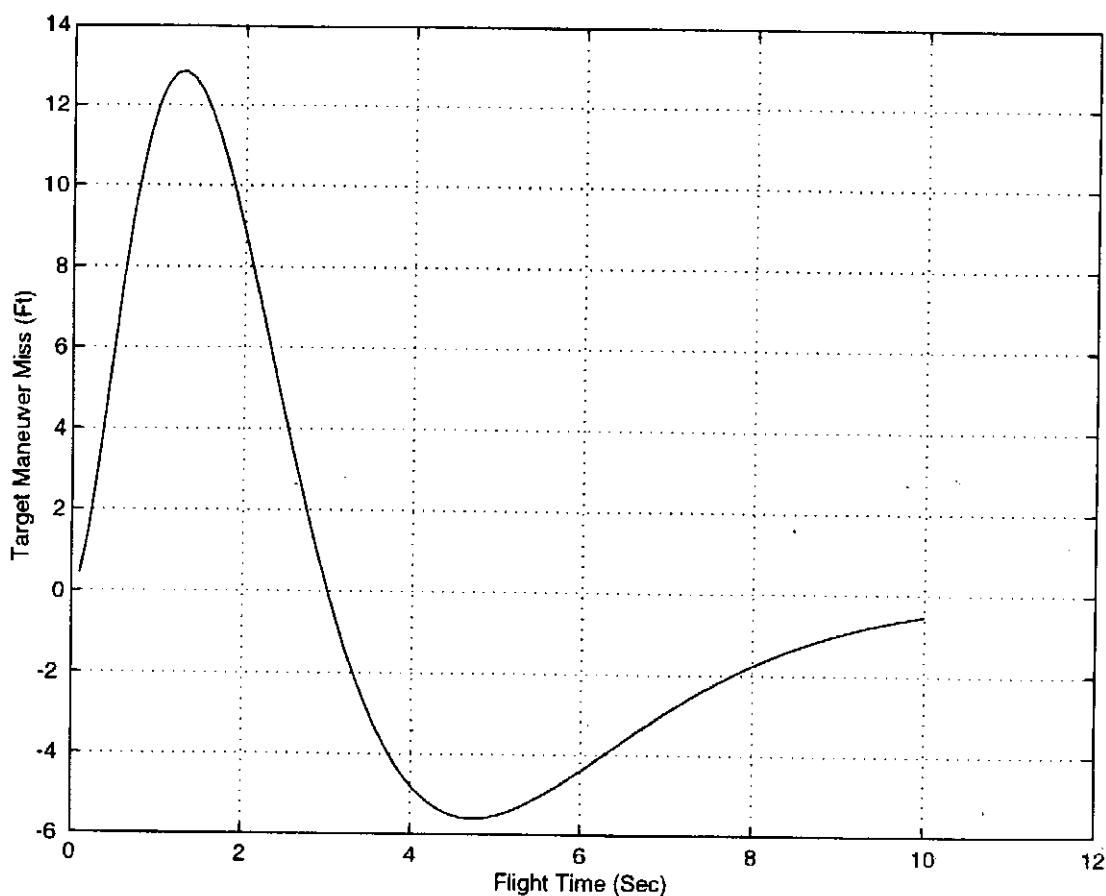


Fig. C.1 MATLAB® output for miss due to target maneuver.

Listing C.1 GOTO statements of Listing 3.1 can be eliminated

```

INTEGER STEP,FLAG
DATA XNT,XNPTAU,TF,VM,HEDEG/ 96.6,4.,1.,10.,3000.,-20./
OPEN(1,STATUS='UNKNOWN',FILE='DATFIL')
T=0.
S=0.
TP=T+.00001
X1=0
X2=0
X3=1
X4=0
H=.01
HE=HEDEG/57.3
WHILE (TP<=(TF-.00001))
    X1OLD=X1
    X2OLD=X2
    X3OLD=X3
    X4OLD=X4
    STEP=1
    FLAG=0
    WHILE (STEP<=1)
        IF(FLAG.EQ.1)THEN
            STEP=2
            X1=X1+H*X1D
            X2=X2+H*X2D
            X3=X3+H*X3D
            X4=X4+H*X4D
            TP=TP+H
        ENDIF
        X1D=X2
        X2D=X3
        Y1=(X4-X2)/TAU
        TGO=TP+.00001
        X3D=XNP*Y1/TGO
        X4D=-Y1
        FLAG=1
    END DO
    FLAG=0
    X1=(X1OLD+X1)/2+.5*H*X1D
    X2=(X2OLD+X2)/2+.5*H*X2D
    X3=(X3OLD+X3)/2+.5*H*X3D
    X4=(X4OLD+X4)/2+.5*H*X4D
    S=S+H
    IF(S>=.09999)THEN
        S=0.
        XMNT=XNT*X1
        XMHE=-VM*HE*X2
        WRITE(*,97)TP,XMNT,XMHE
        WRITE(1,97)TP,XMNT,XMHE
    ENDIF

```

(Contd.)

Listing C.1 (Continued)

```

END DO
97 FORMAT(3F10.3)
PAUSE
CLOSE(1)
END

```

identical to Figs. 3.13 and 3.14 of the text, indicating that MATLAB® is giving answers equivalent to that of the FORTRAN code.

A more complex conversion example is that of Listing 7.1. This example was chosen because it represented a mixed continuous discrete system (i.e., differential and difference equations) driven by noise. Listing C.3 provides the line-by-line translation of Listing 7.1 and is known as C7L1.M on the enclosed floppy disks. We can see that the integration section of the program is identical to that of Listing C.2a whereas the difference equation section is identical to that of the original FORTRAN listing. In this example it is necessary to have a fixed integration step size and also to maintain a large integer ratio for the sampling time divided by the integration step size in order to avoid aliasing effects. Therefore the use of specialized variable step size ODE MATLAB® statements for the integration of differential equations is not recommended for mixed continuous discrete systems. Although a special MATLAB® function could have been used for the generation of Gaussian noise, a separate M-file was used instead, in order for MATLAB® to

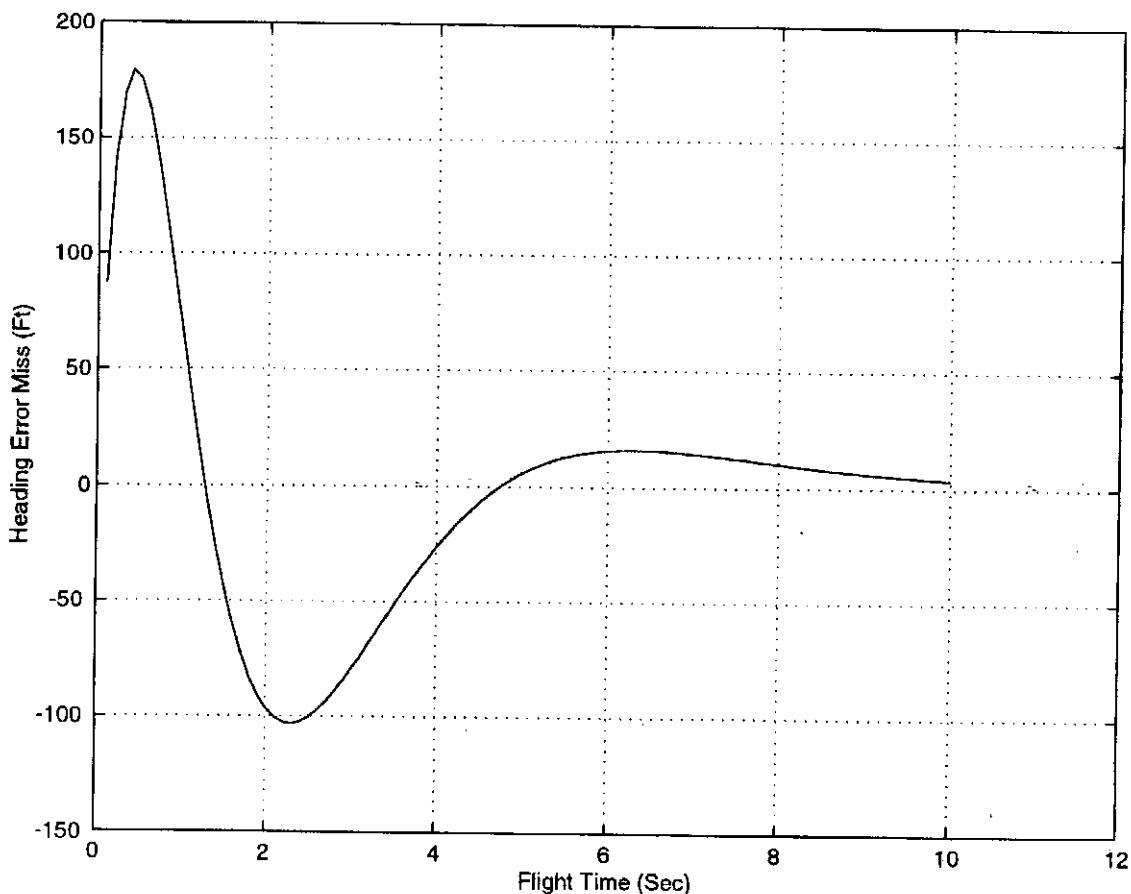


Fig. C.2 MATLAB® output for miss due to heading error.

**Listing C.2a MATLAB® line-by-line
translation of Listing 3.1 is called C3L1.M**

```

n=0;
XNT=96.6;
XNP=4.:
TAU=1.:
TF=10;
VM=3000.:
HEDEG=-20.:
T=0.:
S=0.:
TP=T+.00001;
X1=0;
X2=0;
X3=1;
X4=0;
H=.01;
HE=HEDEG/57.3;
while TP<=(TF-1e-5)
    X1OLD=X1;
    X2OLD=X2;
    X3OLD=X3;
    X4OLD=X4;
    STEP=1;
    FLAG=0;
    while STEP<=1
        if FLAG==1
            STEP=2;
            X1=X1+H*X1D;
            X2=X2+H*X2D;
            X3=X3+H*X3D;
            X4=X4+H*X4D;
            TP=TP+H;
        end
        X1D=X2;
        X2D=X3;
        Y1=(X4-X2)/TAU;
        TGO=TP+.00001;
        X3D=XNP*Y1/TGO;
        X4D=-Y1;
        FLAG=1;
    end
    FLAG=0;
    X1=(X1OLD+X1)/2+.5*H*X1D;
    X2=(X2OLD+X2)/2+.5*H*X2D;
    X3=(X3OLD+X3)/2+.5*H*X3D;
    X4=(X4OLD+X4)/2+.5*H*X4D;
    S=S+H;
    if S>=.0999
        S=0.;
        n=n+1;
        ArrayTP(n)=TP;

```

(Contd.)

Listing C.2a (Continued)

```

        ArrayXMNT(n)=XNT*X1;
        ArrayXMHE(n)=-VM*HE*X2;
    end
end
figure
plot(ArrayTP,ArrayXMNT),grid
xlabel('Flight Time (Sec)')
ylabel('Target Maneuver Miss (Ft)')
figure
plot(ArrayTP,ArrayXMHE),grid
xlabel('Flight Time (Sec)')
ylabel('Heading Error Miss (Ft)')
clc
output=[ArrayTP',ArrayXMNT',ArrayXMHE'];
save datfil output -ascii
disp 'simulation finished'

```

mimic the FORTRAN subroutine. A scaling statement was also used to make the plots more interesting because the last value of the line-of-sight rate is near infinite. The equivalent of the Gaussian noise subroutine is a separate M file known as GAUSSC7.M and appears in Listing C.4. The program was run, and the line-of-sight rate and estimate appear in Fig. C.3. We can see that these results are virtually identical to that of Fig. 7.3.

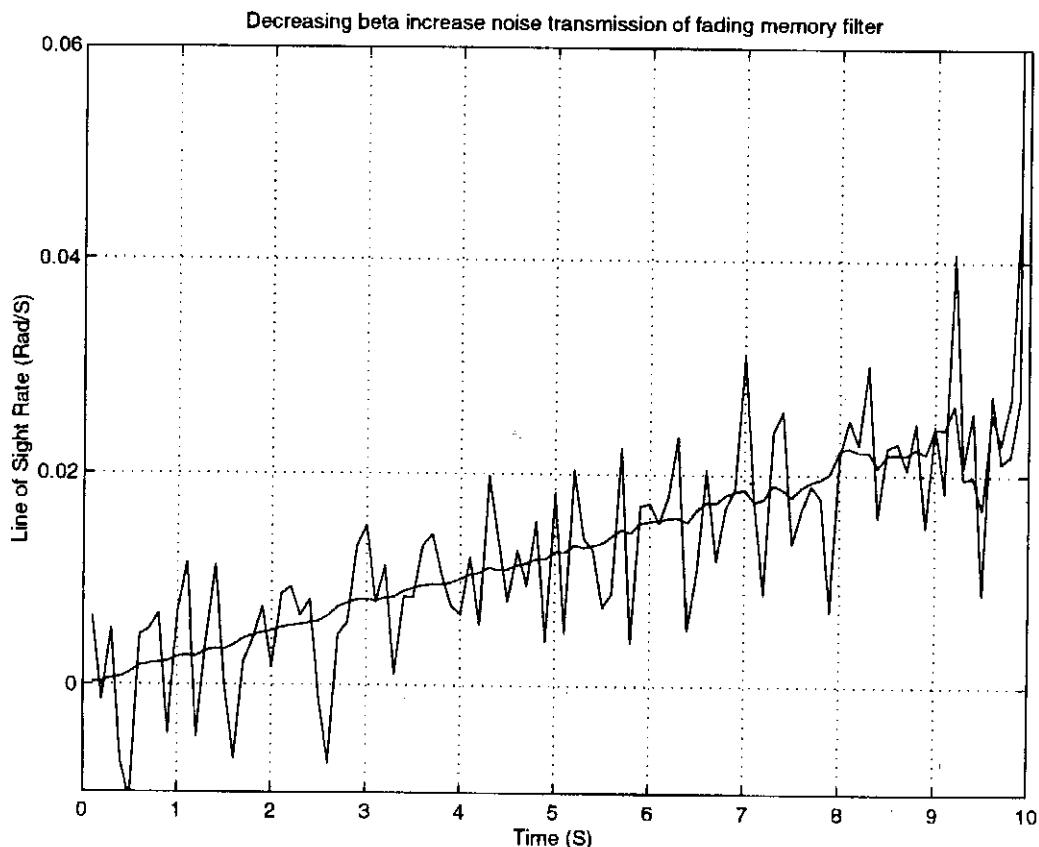


Fig. C.3 MATLAB® yields correct line-of-sight rate estimate.

**Listing C.2b MATLAB® optimized code for
Listing 3.1 called S3L1.M**

```

XNT=96.6;
VM=3000.;
HEDEG=-20.;
HE=HEDEG/57.3;
tol=1e-6;
t0=0; %
tf=10; %
x0=[0,0,1,0]';
[t,x]=ode23('de3l1',t0,tf,x0,tol);
XMNT=XNT*x(:,1);
XMHE=-VM*HE*x(:,2);
figure
plot(t,XMHE),grid
title('single-lag guidance system')
xlabel('Flight Time (sec)')
ylabel('Miss due to Heading Error (Ft)')
figure
plot(t,XMNT),grid
title('single-lag guidance system')
xlabel('Flight Time (sec)')
ylabel('Miss due to Target Maneuver (Ft)')
clc
output=[t,XMHE,XMNT];
save datfil output -ascii
disp 'simulation finished'

```

The final example presented is that of a Listing 14.2 in the text in which a small program is calling the efficient Lambert subroutine. The calling program appears in Listing C.5. The last statement is missing a semicolon so that the two component velocities generated by the Lambert subroutine will be printed to the screen. The equivalent of the Lambert subroutine is a separate M file known as LAMBERT.M and appears in Listing C.6. Notice the GOTO statements of the original FORTRAN listing have been eliminated by the use of the MATLAB® WHILE statement. This combination of C14L2.M and LAMBERT.M yields answers that are identical to its FORTRAN counterpart.

**Listing C.2c Differential equations
for Listing 3.1 called DE3L1.M**

```

function xdot=de3l1(t,x)
TAU=1;
XNP=4;
xdot=zeros(4,1);
xdot(1)=x(2);
xdot(2)=x(3);
y1=(x(4)-x(2))/TAU;
TGO=t+.00001;
xdot(3)=XNP.*y1/TGO;
xdot(4)=-y1;

```

**Listing C.3 MATLAB® equivalent of Listing 7.1
without the Gaussian noise subroutine**

```

count=0;
VC=4000;
XNT=96.6;
YIC=0;
VM=3000;
HEDEG=0;
BETA=0.3;
XNP=3;
SIGNOISE=.001;
TF=10;
TS=.1;
NOISE=1;
Y=YIC;
YD=-VM*HEDEG/57.3;
YDIC=YD;
T=0;
H=.01;
S=0;
GFILTER=1.-BETA^2;
HFILTER=(1.-BETA)^2;
XLAMH=0;
XLAMDH=0;
XNC=0;
while T <= (TF - 1e-5)
    YOLD=Y;
    YDOLD=YD;
    STEP=1;
    FLAG=0;
    while STEP <=1
        if FLAG==1
            Y=Y+H*YD;
            YD=YD+H*YDD;
            T=T+H;
            STEP=2;
        end;
        TGO=TF-T+1e-5;
        RTM=VC*TGO;
        XLAM=Y/(VC*TGO);
        XLAMD=(RTM*YD+Y*VC)/(RTM^2);
        YDD=XNT-XNC;
        FLAG=1;
    end;
    FLAG=0;
    Y=.5*(YOLD+Y+H*YD);
    YD=.5*(YDOLD+YD+H*YDD);
    S=S+H;

```

(Contd.)

Listing C.3 (Continued)

```

if S>=(TS - 1e-5)
    S=0.;
    if NOISE==1,
        XLAMNOISE=gaussc7(SIGNOISE);
    else
        XLAMNOISE=0.;

end;
RES=XLAM-(XLAMH+TS*XLAMDH)+XLAMNOISE;
XLAMH=GFILTER*RES+XLAMH+TS*XLAMDH;
XLAMDH=HFILTER*RES/TS+XLAMDH;
XNC=XNP*VC*XLAMDH;
count=count+1;
ArrayT(count)=T;
ArrayY(count)=Y;
ArrayXNC(count)=XNC;
ArrayXLAMD(count)=XLAMD;
ArrayXLAMDH(count)=XLAMDH;
end;

end;
figure
plot(ArrayT,ArrayXLAMD,ArrayT,ArrayXLAMDH),grid
title('Decreasing beta increase noise transmission of fading memory filter')
xlabel('Time (S)')
ylabel('Line of Sight Rate (Rad/S)')
axis([0 10 -.01 .06])
output=[ArrayT',ArrayY',ArrayXNC',ArrayXLAMD',ArrayXLAMDH'];
save datfil output /ascii
disp('Simulation Complete')

```

**Listing C.4 Gaussian noise subroutine
known as GAUSSC7.M**

```

function y=gaussc7(sig)
sum=0;
for j=1:6;
    RAND=rand(1);
    sum=sum+RAND;
end;
x=sum-3;
y=1.414*x*sig;

```

Listing C.5 MATLAB® equivalent of Listing 14.2 without the Lambert subroutine

```
XLONGMDEG=45.;
XLONGTDEG=90.;
ALTNMT=0.;
ALTNMM=0.;
TF=1000.;
PI=3.14159;
DEGRAD=360./(2.*PI);
A=2.0926e7;
GM=1.4077e16;
ALTT=ALTNMT*6076.;
ALTM=ALTNMM*6076.;
XLONGM=XLONGMDEG/DEGRAD;
XLONGT=XLONGTDEG/DEGRAD;
XM=(A+ALTM)*cos(XLONGM);
YM=(A+ALTM)*sin(XLONGM);
XT=(A+ALTT)*cos(XLONGT);
YT=(A+ALTT)*sin(XLONGT);
[VRXM,VRYM]=lambert(XM,YM,TF,XT,YT,XLONGM,XLONGT)
```

Listing C.6 Lambert subroutine known as LAMBERT.M

```
function [vrx,vry]=lambert(xic,yic,tfdes,xf,yf,xlongm,xlongt)
a=2.0926e7;
gm=1.4077e16;
ric=sqrt(xic^2+yic^2);
rf=sqrt(xf^2+yf^2);
cphi=(xic*xf+yic*yf)/(ric*rf);
phi=acos(cphi);
sphi=sin(phi);
r0=ric;
degrad=360./(2.*pi);
icount=0;
gmin=atan2((sphi-sqrt(2.*r0*(1.-cphi)/rf)),(1-cphi));
gmax=atan2((sphi+sqrt(2.*r0*(1.-cphi)/rf)),(1-cphi));
gam=(gmin+gmax)/2.;
tf=0;
while ~ (abs(tfdes-tf)<=(.00000001* tfdes))
    top=gm*(1.-cos(phi));
    temp=r0*cos(gam)/rf-cos(phi+gam);
    bot=r0*cos(gam)*temp;
    v=sqrt(top/bot);
    if xlongt>xlongm
        vrz=v*cos(pi/2.-gam+xlongm);
        vry=v*sin(pi/2.-gam+xlongm);
```

(Contd.)

Listing C.6 (Continued)

```

else
    vrx=v*cos(-pi/2.+gam+xlongm);
    vry=v*sin(-pi/2.+gam+xlongm);
end
xlam=r0*v/v/gm;
top1=tan(gam)*(1-cos(phi))+(1-xlam)*sin(phi);
bot1p=(1-cos(phi))/(xlam*cos(gam)*cos(gam));
bot1=(2-xlam)*(bot1p+cos(gam+phi)/cos(gam));
top2=2*cos(gam);
bot2=xlam*((2/xlam-1)^1.5);
top3=sqrt(2/xlam-1);
bot3=cos(gam)/tan(phi/2)-sin(gam);
temp=(top2/bot2)*atan2(top3,bot3);
tf=r0*(top1/bot1+temp)/(v*cos(gam));
icount=icount+1;
if tf>tfdes
    gmax=gam;
else
    gmin=gam;
end
if icount==1
    xnext=(gmax+gmin)/2.;
else
    xnext=gam+(gam-gold)*(tfdes-tf)/(tf-told);
    if (xnext>gmax|xnext<gmin)
        xnext=(gmax+gmin)/2.;
    end
end
gold=gam;
told=tf;
gam=xnext;
end

```



True BASIC¹

To many older engineers, BASIC is simply known as "FORTRAN without grief." BASIC was originally developed in 1963 at Dartmouth College by John Kemeny and Thomas Kurtz. Although many dialects of BASIC were originally implemented on microcomputers during the 1980s and early 1990s, the major nonobject-oriented dialect to survive was the original, which is known commercially as True BASIC. True BASIC, like MATLAB[®], is an interpretive language and is therefore much slower than a compiled language such as FORTRAN. However, as with MATLAB[®], most of the programs in the text take only from a few seconds to a minute of running time with True BASIC on today's powerful microcomputers. Its low cost (approximately \$35 for the regular version and the student version is free) and ease of use also make it a natural choice for implementing the code in this text on personal computers or workstations. As with MATLAB[®], True BASIC also has special functions for matrix manipulation.

As was the case with MATLAB[®], True BASIC also does not allow the use of the GOTO statement. Therefore, code involving numerical integration was modified by using the WHILE statement. Therefore, the True BASIC code will closely resemble the MATLAB[®] code of this text.

As an example of how the FORTRAN source code was converted to True BASIC, let us again consider the adjoint program of Listing 3.1 that involved numerical integration. The True BASIC equivalent of Listing 3.1 appears in Listing D.1. We can see that the True BASIC listing is virtually identical to MATLAB[®] Listing C.2a. At the end of the program the resultant data is sent to an ASCII file called DATFIL. True BASIC has a problem creating a new file if an old file with the same name already exists. Therefore, the user must be cautious when running True BASIC several times in a row when data files are automatically created. Either the user must manually delete previously created data files or have statements in the program automatically deleting the files. Listing D.1 has two statements at the beginning of the program to delete previously created file DATFIL. This statement is initially commented out by use of the REM statement. After the program is run once, the word REM can be deleted from the listing so that the file of the previous run will automatically be removed.

The True BASIC simulation of Listing D.1 was run, and data concerning the miss because of target maneuver and heading error are written to a file DATFIL. In addition, the miss due to target maneuver and heading error are displayed in Figs. D.1 and D.2, respectively. Notice that Fig. D.1 is equivalent to the FORTRAN-generated results of Fig. 3.13, while Fig. D.2 is equivalent to the FORTRAN-generated results of Fig. 3.14. These results also agree with the MATLAB[®]-generated results of the previous section (see Figs. C.1 and C.2).

Listing D.1 True BASIC equivalent of Listing 3.1

```

OPTION NOLET
REM UNSAVE "DATFIL"
OPEN #1:NAME "DATFIL",ACCESS OUTPUT,CREATE NEW,
ORGANIZATION TEXT
SET #1: MARGIN 1000
XNT=96.6
XNP=4.
TAU=1.
TF=10
VM=3000.
HEDEG=-20.
T=0.
S=0.
TP=T+.00001
X1=0
X2=0
X3=1
X4=0
H=.01
HE=HEDEG/57.3
DO WHILE TP<=(TF-1e-5)
    X1OLD=X1
    X2OLD=X2
    X3OLD=X3
    X4OLD=X4
    STEP=1
    FLAG=0
    DO WHILE STEP<=1
        IF FLAG=1 THEN
            STEP=2
            X1=X1+H*X1D
            X2=X2+H*X2D
            X3=X3+H*X3D
            X4=X4+H*X4D
            TP=TP+H
        END IF
        X1D=X2
        X2D=X3
        Y1=(X4-X2)/TAU
        TGO=TP+.00001
        X3D=XNP*Y1/TGO
        X4D=-Y1
        FLAG=1
    LOOP
    FLAG=0
    X1=(X1OLD+X1)/2+.5*H*X1D
    X2=(X2OLD+X2)/2+.5*H*X2D
    X3=(X3OLD+X3)/2+.5*H*X3D
    X4=(X4OLD+X4)/2+.5*H*X4D
    S=S+H

```

(Contd.)

Listing D.1 (Continued)

```

IF S>=.0999 THEN
  S=0.
  XMNT=XNT*X1
  XMHE=-VM*HE*X2
  PRINT TP,XMNT,XMHE
  PRINT #1:TP,XMNT,XMHE
END IF
LOOP
CLOSE #1
END

```

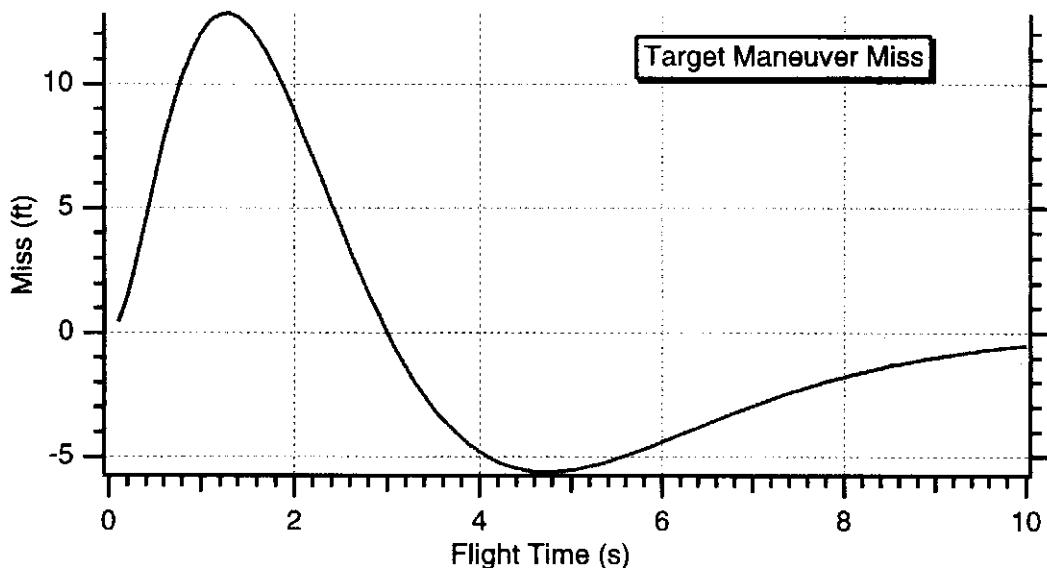


Fig. D.1 True BASIC output is equivalent to Fig. 3.13.

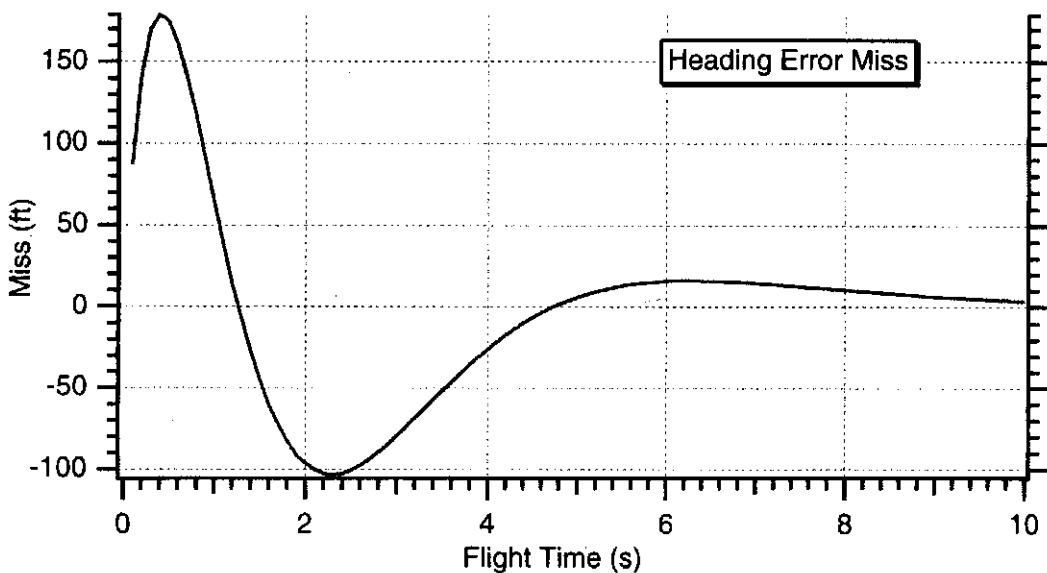


Fig. D.2 True BASIC output is equivalent to Fig. 3.14.

Listing D.2 True BASIC equivalent of Listing 7.1

```

OPTION NOLET
REM UNSAVE "DATFIL"
OPEN #1:NAME "DATFIL",ACCESS OUTPUT,CREATE NEW,
ORGANIZATION TEXT
SET #1: MARGIN 1000
VC=4000
XNT=96.6
YIC=0
VM=3000
HEDEG=0
BETA=0.3
XNP=3
SIGNOISE=.001
TF=10
TS=.1
NOISE=1
Y=YIC
YD=VM*HEDEG/57.3
YDIC=YD
T=0
H=.01
S=0
GFILTER=1.-BETA^2
HFILTER=(1.-BETA)^2
XLAMH=0
XLAMDH=0
XNC=0
DO WHILE T <= (TF - 1e-5)
    YOLD=Y
    YDOLD=YD
    STEP=1
    FLAG=0
    DO WHILE STEP <=1
        IF FLAG=1 THEN
            Y=Y+H*YD
            YD=YD+H*YDD
            T=T+H
            STEP=2
        END IF
        TGO=TF-T+1e-5
        RTM=VC*TGO
        XLAM=Y/(VC*TGO)
        XLAMD=(RTM*YD+Y*VC)/(RTM^2)
        YDD=XNT-XNC
        FLAG=1
    LOOP
    FLAG=0
    Y=.5*(YOLD+Y+H*YD)
    YD=.5*(YDOLD+YD+H*YDD)
    S=S+H

```

(Contd.)

Listing D.2 (Continued)

```

IF S>=(TS - 1e-5) THEN
    S=0.
    IF NOISE=1 THEN
        CALL GAUSS(XLAMNOISE,SIGNOISE)
    ELSE
        XLAMNOISE=0.
    END IF
    RES=XLAM-(XLAMH+TS*XLAMDH)+XLAMNOISE
    XLAMH=GFILTER*RES+XLAMH+TS*XLAMDH
    XLAMDH=HFILTER*RES/TS+XLAMDH
    XNC=XNP*VC*XLAMDH
    PRINT T,Y,XNC,XLAMD,XLAMDH
    PRINT #1:T,Y,XNC,XLAMD,XLAMDH
END IF
LOOP
CLOSE #1
END
SUB GAUSS(X,SIG)
LET X=RND+RND+RND+RND+RND+RND-3
LET X=1.414*X*SIG
END SUB

```

Unlike MATLAB[®], Gaussian random numbers are not automatically available with True BASIC. However, True BASIC provides a random number generator yielding uniformly distributed numbers between 0 and 1 with the statement RND. A subroutine, similar to the FORTRAN subroutine GAUSS, makes use of that random number generator to provide zero-mean Gaussian distributed random numbers.

A more complex conversion example is that of Listing 7.1. This example was chosen because it represented a mixed continuous discrete system (that is, differential and difference equations) driven by noise. Listing D.2 provides the line-by-line

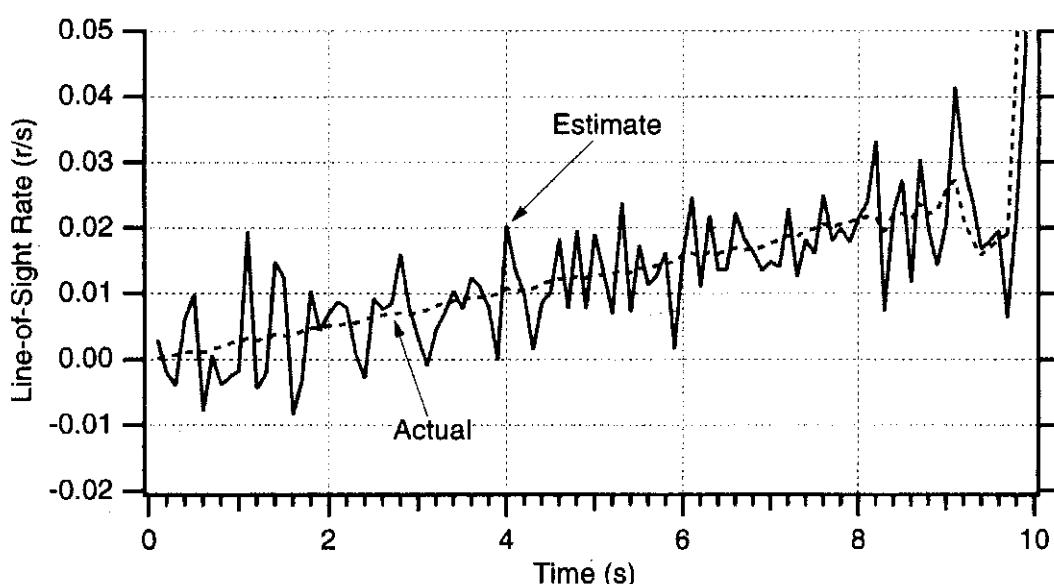


Fig. D.3 True BASIC output is equivalent to Fig. 7.3.

translation of Listing 7.1. We can see that the integration section of the program is identical to that of Listing D.1, while the difference equation section is identical to that of the original FORTRAN listing. In this example, it is necessary to have a fixed integration step size and to also maintain a large integer ratio for the sampling time divided by the integration step size to avoid aliasing effects.

The simulation of Listing D.2 was run and the true line-of-sight rate, along with its estimate, are displayed in Fig. D.3. We can see that this figure is equivalent to the FORTRAN-generated results of Fig. 7.3.

Reference

¹Zarchan, P., and Musoff, H., *Fundamentals of Kalman Filtering: A Practical Approach*, Progress in Astronautics and Aeronautics, AIAA, Reston, VA, 2000, pp. 654–662.

Units

TO most of the civilized world the units used in this text appear to be unconventional and in violation of everything taught in the universities. However, for many senior U.S. engineers it is not unusual to express altitude in ft, velocity in km/s, and downrange in yards—all in the same breath! As a special favor to those who do not appreciate the importance of the slug, Table E.1 shows how to convert the text's English units (which are no longer used in England) to either the metric system or to more popular units. For those who must pretend that they are familiar with the metric system, simply take the reciprocal of the conversion factor to get back to the text's units.

Table E.1 Unit conversion table

Quantity	Text's units	Multiply by	To get
Angle	rad	57.3	deg
Length	ft	0.305	m
	n.mi.	1.85	km
Velocity	ft/s	0.305	m/s
Acceleration	ft/s ²	0.031	g
		0.305	m/s ²
Area	ft ²	0.093	m ²
Weight	lb	0.45	kg
Air density	slug/ft ³	515	kg/m ³



Listing Index

- Listing 1.1**
FORTRAN simulation of second-order system, 6
- Listing 2.1**
two-dimensional tactical missile-target engagement simulation, 16–17
- Listing 2.2**
linearized engagement simulation, 22
- Listing 3.1**
single-lag adjoint with second-order Runge–Kutta integration, 42–43
- Listing 4.1**
Gaussian random number generator, 55
- Listing 4.2**
FORTRAN program used to generate probability density function, 56
- Listing 4.3**
program for computing sampled standard deviation, 59
- Listing 4.4**
simulation of low-pass-filter driven by white noise, 64
- Listing 4.5**
simulation of homing loop with random target maneuver, 71–72
- Listing 4.6**
adjoint model using shaping filter approach, 74
- Listing 5.1**
simulation of second-order system with fourth-order Runge–Kutta integration technique, 83–84
- Listing 5.2**
homing loop covariance analysis program, 86–89
- Listing 5.3**
acceleration and miss distance adjoint program, 93–94
- Listing 6.1**
adjoint of fifth-order binomial guidance system, 100–101
- Listing 6.2**
engagement simulation with radome effects, 113–114
- Listing 7.1**
engagement simulation with second-order fading memory filter, 122–123
- Listing 7.2**
Monte Carlo version of fading memory filter in homing loop, 125–126
- Listing 7.3**
adjoint engagement simulation with two-state fading memory filter, 132–133
- Listing 7.4**
engagement simulation with three-state fading memory filter, 138–139
- Listing 8.1**
adjoint simulation for inverse, 154
- Listing 8.2**
adjoint simulation of optimal guidance system, 159–160
- Listing 9.1**
Listing of FORTRAN program to solve Riccati equations, 169–170
- Listing 9.2**
listing of Kalman filter in homing loop, 171–173
- Listing 9.3**
Monte Carlo engagement simulation to test optimal guidance, 179–182
- Listing 10.1**
proportional navigation command guidance simulation, 188–190
- Listing 10.2**
FORTRAN beam rider engagement simulation, 195–196
- Listing 11.1**
trajectory simulation, 217–218
- Listing 12.1**
gravity field simulation, 226–227
- Listing 12.2**
gravity field simulation with different coordinate systems, 231–232
- Listing 12.3**
FORTRAN orbit generator, 237–238
- Listing 12.4**
simulation to demonstrate validity of velocity formula, 240–241
- Listing 13.1**
FORTRAN thrust-weight computations, 253–254
- Listing 13.2**
gravity turn simulation, 257–259
- Listing 14.1**
Lambert routine using brute force approach, 268–270
- Listing 14.2**
more efficient Lambert routine, 272–273
- Listing 14.3**
booster simulation with Lambert guidance, 275–279

- Listing 14.4**
general energy management simulation, 284–288
- Listing 15.1**
engagement simulation with ballistic target, 296–301
- Listing 15.2**
engagement simulation for boosting target, 307–313
- Listing 16.1**
predictive guidance engagement simulation of ballistic missile and booster target, 322–329
- Listing 16.2**
booster engagement simulation with Kalman filter based on range and angle measurement, 392–340
- Listing 16.3**
pulsed guidance simulation, 345–351
- Listing 17.1**
ballistic target simulation, 355–356
- Listing 17.2**
ballistic target engagement simulation, 366–369
- Listing 18.1**
one-dimensional extended Kalman filter for ballistic coefficient estimation, 380–383
- Listing 19.1**
adjoint of fifth-order binomial guidance system for noise miss distance calculations, 395–396
- Listing 19.2**
simulation of homing loop with radome effects, 400–401
- Listing 20.1**
nonlinear engagement simulation with step in target displacement, 409–410
- Listing 20.2**
adjoint simulation of single time constant guidance system with step in target displacement, 418–419
- Listing 20.3**
multiple run nonlinear single time constant guidance system engagement simulation, 420–421
- Listing 20.4**
adjoint of fifth-order binomial guidance system used to find normalized miss distance, 424–425
- Listing 20.5**
brute force simulation for finding normalized miss due to step in target displacement, 428–429
- Listing 21.1**
nonlinear engagement simulation for a single time constant guidance system with weaving target, 434–435
- Listing 21.2**
adjoint simulation of single time constant guidance system and weaving target, 438–439
- Listing 21.3**
brute force simulation for generating normalized design curves against weaving target, 446–447
- Listing 21.4**
brute force simulation for guidance law evaluation against weaving target, 458–459
- Listing 22.1**
airframe simulation, 466–467
- Listing 22.2**
linear airframe simulation, 475–476
- Listing 23.1**
open-loop flight-control system, 487–488
- Listing 23.2**
rate gyro flight-control system in presence of linear airframe, 493–494
- Listing 23.3**
open-loop Bode response for rate gyro flight-control system, 499
- Listing 23.4**
rate gyro flight-control system with pure delay, 502–504
- Listing 24.1**
three-loop autopilot step response simulation, 515–517
- Listing 24.2**
open-loop response of three-loop autopilot, 519–520
- Listing 24.3**
adjoint simulation of homing loop with detailed flight-control system, 531–535
- Listing 25.1**
linear engagement simulation for guidance law comparison, 549–550
- Listing 25.2**
nonlinear engagement simulation to test trajectory shaping guidance law, 561–563
- Listing 26.1**
original three-state linear Kalman filter and weaving target, 574–576
- Listing 26.2**
weave Kalman filter and weaving target, 584–589
- Listing 26.3**
extended Kalman filter, 599–605
- Listing 27.1**
missile-target engagement simulation with conventional guidance, 616–618
- Listing 27.2**
engagement simulation with predictor-corrector guidance, 622–625
- Listing 27.3**
an engagement simulation using an unusual form of proportional navigation, 629–630

- Listing 27.4
an unusual problem using predictor-corrector guidance, 635-637
- Listing A.1
simulation of second-order shaping network using Euler integration, 645
- Listing A.2
engagement simulation with pursuit guidance option, 647-648
- Listing A.3
adjoint simulation of optimal guidance system, 650-651
- Listing A.4
alternative model of homing loop with radome effects, 653-654
- Listing A.5
FORTRAN program to generate Gaussian probability density function, multiplicative congruential method, 656-657
- Listing A.6
simulating an impulse rather than using initial conditions for use in the adjoint method, 658-659
- Listing A.7
canonic guidance system adjoint simulation, 661-662
- Listing A.8
using fading memory filters for booster estimation, 665-672
- Listing A.9
mixed continuous discrete adjoint with discrete sinusoidal acceleration bias, 676-677
- Listing A.10
brute-force frequency response program, 681-682
- Listing A.11
three-loop autopilot with nonlinear force and moment equations, 685-687
- Listing A.12
calculating flight time for a minimum energy trajectory, 690
- Listing A.13
modified form of Listing 14.3, 691-693
- Listing B.1
FORTRAN fragments of code converted to C, 698-699
- Listing B.2
C version of efficient Lambert subroutine, 700-701
- Listing C.1
GOTO statements of Listing 3.1 can be eliminated, 705-706
- Listing C.3
MATLAB® equivalent of Listing 7.1 without the Gaussian noise subroutine, 710-711
- Listing C.4
Gaussian noise subroutine known as GAUSSC7.M, 711
- Listing C.5
MATLAB® equivalent of Listing 14.2 without the Lambert guidance, 712
- Listing C.6
Lambert subroutine known as LAMBERT.M, 712-713
- Listing C.2a
MATLAB® line-by-line translation of Listing 3.1 is called C3L1.M, 707-708
- Listing C.2b
MATLAB® optimized code for Listing 3.1 called S3L1.M, 709
- Listing C.2c
differential equations for Listing 3.1 called DE3L1.M, 709
- Listing D.1
True BASIC equivalent of Listing 3.1, 716-717
- Listing D.2
True Basic equivalent of Listing 7.1, 718-719



Index

- Acceleration saturation
for multiple targets, 426–431
for weaving targets, 448–451
- Adjoints
closed-form solutions, 43–50
example of deterministic adjoint, 40–43
for deterministic systems, 38–40
for noise-driven systems, 65–66
homing loop diagrams, 31–32
in advanced guidance laws, 143–161
in covariance analysis, 91–94
in digital fading memory noise filters,
 homing loop, 127–134
in proportional navigation, 98–104
in software package, 658–675
in three-loop autopilot, 527–538
introduction and overview, 31
mathematics, 37–38
method for construction, 34–37
normalization, 48–50
simulation for weaving targets, 437–440
simulation of single time constant guidance
 system, 418–419
single time constant guidance systems,
 32–34
stochastic adjoint example, 70–75
target maneuver, 33–34
- Airframe, transfer functions
airframe simulation, 465–469
effects of flight conditions, 478–482
force and moment equations, 461–465
introduction, 461
linearization of airframe, 469–472
 numerical example, 472–478
- Apollo guidance law, 546
- Autopilot. *see* three-loop autopilot
- Ballistic targets
closed-form solutions, 359–362
fifth-order binomial guidance system miss
 distances, 393–397
interception of ballistic target, 364–371
introduction and overview, 353
minimum guidance system time constant
 constraints, 399–403
miss distance due to noise, 389–393
miss due to noise, 403–405
missile aerodynamics, 362–364
model, 353–354
Runge–Kutta integration technique,
 356
target experiments, 354–359
turning rate time constant, 397–399
- Beam rider guidance
definitions and overview, 193
FORTRAN simulation, 194–196
linearized example, 197–201
- Booster estimation with range and angle,
 330–342
- Boosters
estimation with range and angle, 330–342
gravity turn, 255–261
in strategic interceptors, 305–316
numerical example, 251–255
review of applicable equations, 247–249
staging techniques, 249–251
steering, Lambert guidance, 274–281
- CD-ROM description and instructions,
 643–644
- Closed-form solutions
for random target maneuver in noise analysis,
 75–76
for weaving targets, 440–444
in ballistic target properties, 359–362
in Lambert guidance, 264–267
in proportional navigation, 24–27, 143–160
in proportional navigation and miss distance,
 97–104
in strategic intercepts, 291–314
in tactical zones, 211–214
in trajectory shaping guidance, 554–559
strategic considerations, 233–239
 with adjoints, 43–50
- Closing velocity, definition, 13
- Command to line-of-sight guidance
definitions and overview, 201–202
example case, 202–204
- Conversion of units, table, 721
- Converting programs: FORTRAN to C,
 697–702
- Covariance analysis
acceleration adjoint, 91–94
background and theory, 79–80
homing loop example, 82–91
low-pass filter example, 80–81
numerical considerations, 881–82
Runge–Kutta integration, 81–89
- Digital fading memory noise filters, homing
 loop
estimating target maneuver, 136–140
fading memory filters, 119–127
introduction, 119
mixed continuous discrete adjoint theory,
 127–134

- Digital fading memory (*cont.*)
 Runge-Kutta integration, 121–127
 using adjoint theory to evaluate filter, 130–134
- Flight control design
 constructed adjoint example, 489–490
 guidance system interactions, 489–490
 introduction, 483
 open-loop crossover frequency, 504–506
 open-loop flight-control system, 483–488
 open-loop transfer function, 496–500
 rate gyro flight-control system, 490–496
 simulation, rate gyro system, 492–495
 three-loop autopilot, 507–539
 time domain verification, open-loop results, 500–504
- Flight time, 243–244
- FORTRAN
 adjoint simulation, 40–43
 converting programs: FORTRAN to C, 697–702
 converting programs: FORTRAN to MATLAB®, 703–713
 in adjoint simulation, 153–160
 in digital fading memory noise filters, homing loop, 121–127, 131–140
 in Kalman filters, 168–183
 in proportional navigation, miss distance, 99–104
 in software package, 643–694
 simulation of second-order system, 6
 two-dimensional tactical missile-target engagement simulation, 15–18
 use in Lambert guidance, 267–289
 use in noise analysis, 56–75
 use with boosters, 252–261
 with orbit generator, 237–239
 with strategic intercepts, 295–316
- Fourier transform in noise analysis, 60–62
- General energy management (GEM) steering, 281–289
- Gravitational model, 221–228
- Gravity compensation, 317–320
- Hit equation, 239–243
- Hs. 298 missile, 11
- Hughes Aircraft Company, 11–12
- Kalman filters, ballistic coefficient estimation
 differential equations for target, 375–376
 extended Kalman filter for target, 376–379
 introduction, 373
 numerical example, 379–387
 theoretical equations, 373–375
- Kalman filters, filtering and weaving targets
 extended Kalman filter, 595–611
 four-state weave Kalman filter, 592–595
- review of three-state Kalman filter, 571–578
- Kalman filters, homing loop
 experiments with optimal guidance, 178–183
 filter gains, 167–168
 introduction, 163
 numerical examples, 168–178
 Riccati equations, 163–172
 theoretical equations, 163–164
 zero-lag homing loop model, 164–166
- Lambert guidance
 booster steering, 274–281
 definition of Lambert's problem, 263–264
 GEM steering, 281–289
 numerical example using FORTRAN, 267–270
 solution to Lambert's problem, 264–267
 solving more quickly, 270–274
- Laplace transforms
 common inverse Laplace transforms, 4
 definition and overview of methods, 1–3
- Lark missile, 11
- Line of sight, definition, 13
- Listings, *see* Listing Index
- MATLAB®
 in software package, 643–694
 MATLAB® conversion, 703–713
 Monte Carlo method, definition, 58
- Noise analysis
 adjoints for noise-driven systems, 65–66
 closed-form solution for random target maneuver, 75–76
 computational issues, 58–60
 Fourier transform, 60–61
 Gaussian distribution, 53–54
 Gaussian noise example, 54–58
 introduction and basic definitions, 51–54
 low-pass-filter example, 61–65
 random variables, 51–54
 second-order solutions, 60–61
 shaping filters and random processes, 66–70
 stochastic adjoint example, 70–75
 white noise, 60–64
- Polar coordinate system, 228–233
- Predictive guidance, 320–330
- Predictor-corrector guidance
 introduction, 613
 roll angle or rate guidance problem, 627–631
 roll-rate problem, 631–640
 surface-to-surface intercept problem, 620–626
 surface-to-surface missile problem, 613–620
- Project Hermes, 221

- Proportional navigation
 augmented proportional navigation law, 145–161
 closed-form solutions, 24–27
 definition, 12
 influence of time constraints, 152–155
 linearization, 20–22
 linearized engagement simulation, 22–24
 optimal guidance, 155–160
 tactical endoatmospheric missiles, 12
 tactical radar homing missiles, 12
 two-dimensional engagement simulation, 15–20
 two-dimensional proportional simulation, 12–15
 zero effort miss, 27–28
- Proportional navigation command guidance
 definitions and overview, 185–192
 development of line-of-sight rate, 186–188
 similarity to homing loop, 191–193
 simulation example, 188–190
- Proportional navigation, miss distance
 barrel roll, 106–107
 design relationships, 96–104
 guidance time constant, 100–105
 introduction and overview, 95
 optimal target evasive maneuvers, 104–106
 parasitic or unwanted feedback paths, 110–116
 practical evasive maneuvers, 106–110
 radome slope, 114–116
 saturation, 108–110
 system order, 95–96
 thrust vector control, 116–118
 turning rate time constant, 111–116
 Vertical-S maneuver, 106–109
- Pulsed guidance, 342–352
- Raytheon, early work in proportional navigation, 11–12
- RCA Laboratories, early work in proportional navigation, 11–12
- Rheintochter (R-1) missile, 11
- Runge–Kutta integration
 definition and overview, 3–6
 in ballistic target experiments, 356
 in covariance analysis, 81–89
 in digital fading memory noise filters, homing loop, 121–127
 tactical missile-target engagement simulation, 15–20
 with adjoints, 40–43
- Software for tactical and strategic missile guidance
 adjoint of discrete inputs, 672–675
 alternative formulation, Radome, 652–654
 brute force frequency response, 678–683
- canonic guidance system adjoint, 659–660
 CD-ROM description and instructions, 643–644
 fading memory filters, booster est., 663–672
 generating random numbers, 654–655
 integration example, 644–645
 introduction, 643
 minimum energy trajectories, 689–694
 optimal guidance to time to go errors, 649–651
 pursuit guidance, 645–649
 sampling experiments, 675–678
 simulating impulse, 655–659
 stability margins, 683–688
- Strategic ballistic missiles, history of, 221
- Strategic intercepts
 ballistic engagement simulation, 293–305
 boosters, 305–316
 closed-form solution, 291–314
 FORTRAN use, 295–316
 guidance review, 291–293
 introduction, 291
 line-of-sight changes, 301–303
 prediction error, 303–305
- Tactical zones
 acceleration effects, 214–216
 definitions and overview, 207
 drag effects, 210–214
 gravity effects, 216–219
 trajectory simulation, 217–218
 velocity computation, 207–210
- Targets, filtering and weaving
 extended Kalman filter, 595–611
 four-state weave Kalman filter, 579–591
 introduction, 571
 miss distance analysis, 592–595
 review of three-state linear Kalman filter, 571–578
- Targets, multiple
 acceleration saturation, 426–431
 higher-order guidance system, 422–426
 introduction and background, 407
 linear model development, 407–415
 single time constant guidance system, 415–422
- Targets, weaving
 acceleration saturation, 448–451
 adjoint simulation, 437–440
 closed-form solutions, 440–444
 higher-order guidance system dynamics, 444–448
 improving performance with advanced guidance, 453–460
 introduction and background, 433
 maneuver in single time constant guidance system, 433–439
 reducing time constant, 451–453

- Three-loop autopilot
adjoint analysis, 527–538
closed-loop analysis, 510–522
configuration, 507
crossover frequency, 518–523
flight condition experiments, 523–527
guidance system analysis, 527–539
introduction, 507
open-loop analysis, 510
- Trajectory shaping guidance
alternate law, 547–548
closed-form solutions, 554–559
introduction, 541
mathematics, 541–543
nonlinear results, 560–568
problem explanation, 541
Schwartz inequality, 543–546
testing in linear world, 548–554
- Transfer functions. *see* airframe, transfer functions
True BASIC, 715–720
- Units, conversion table, 721
- V-2 missile, 221
- White noise
in ballistic target challenges, 389–392
in covariance analysis, 80–81
in noise analysis, 60–61
- Z transforms, definition and overview, 6–10
Zero-effort miss
in advanced guidance laws, 143–160
in proportional navigation, 27–28

PROGRESS IN ASTRONAUTICS AND AERONAUTICS SERIES VOLUMES

- *1. Solid Propellant Rocket Research (1960)**
Martin Summerfield
Princeton University
- *2. Liquid Rockets and Propellants (1960)**
Loren E. Bollinger
Ohio State University
Martin Goldsmith
The Rand Corp.
Alexis W. Lemmon Jr.
Battell Memorial Institute
- *3. Energy Conversion for Space Power (1961)**
Nathan W. Snyder
Institute for Defense Analyses
- *4. Space Power Systems (1961)**
Nathan W. Snyder
Institute for Defense Analyses
- *5. Electrostatic Propulsion (1961)**
David B. Langmuir
Space Technology Laboratories, Inc.
Ernst Stuhlinger
NASA George C. Marshall Space Flight Center
J. M. Sellen Jr.
Space Technology Laboratories, Inc.
- *6. Detonation and Two-Phase Flow (1962)**
S. S. Penner
California Institute of Technology
F.A. Williams
Harvard University
- *7. Hypersonic Flow Research (1962)**
Frederick R. Riddell
AVCO Corp.
- *8. Guidance and Control (1962)**
Robert E. Roberson
Consultant
James S. Farrior
Lockheed Missiles and Space Co.
- *9. Electric Propulsion Development (1963)**
Ernst Stuhlinger
NASA George C. Marshall Space Flight Center
- *10. Technology of Lunar Exploration (1963)**
Clifford I. Cumming
Harold R. Lawrence
Jet Propulsion Laboratory
- *11. Power Systems for Space Flight (1963)**
Morris A. Zipkin
Russell N. Edwards
General Electric Co.
- *12. Ionization in High-Temperature Gases (1963)**
Kurt E. Shuler, Editor
National Bureau of Standards
John B. Fenn, Associate Editor
Princeton University
- *13. Guidance and Control-II (1964)**
Robert C. Langford
General Precision Inc.
Charles J. Mundo
Institute Naval Studies
- *14. Celestial Mechanics and Astrodynamics (1964)**
Victor G. Szebehely
Yale University Observatory
- *15. Heterogeneous Combustion (1964)**
Hans G. Wolfson
Institute for Defense Analyses
Irvin Glassman
Princeton University
Leon Green Jr.
Air Force Systems Command
- *16. Space Power Systems Engineering (1966)**
George C. Szego
Institute for Defense Analyses
J. Edward Taylor
TRW Inc.
- *17. Methods in Astrodynamics and Celestial Mechanics (1966)**
Raynor L. Duncombe
U.S. Naval Observatory
Victor G. Szebehely
Yale University Observatory
- *18. Thermophysics and Temperature Control of Spacecraft and Entry Vehicles (1966)**
Gerhard B. Heller
NASA George C. Marshall Space Flight Center
- *19. Communication Satellite Systems Technology (1966)**
Richard B. Marsten
Radio Corporation of America
- *20. Thermophysics of Spacecraft and Planetary Bodies: Radiation Properties of Solids and the Electromagnetic Radiation Environment in Space (1967)**
Gerhard B. Heller
NASA George C. Marshall Space Flight Center

*Out of print.

- *21. Thermal Design Principles of Spacecraft and Entry Bodies (1969)**
Jerry T. Bevans
TRW Systems
- *22. Stratospheric Circulation (1969)**
Willis L. Webb
Atmospheric Sciences Laboratory, White Sands, and University of Texas at El Paso
- *23. Thermophysics: Applications to Thermal Design of Spacecraft (1970)**
Jerry T. Bevans
TRW Systems
- *24. Heat Transfer and Spacecraft Thermal Control (1971)**
John W. Lucas
Jet Propulsion Laboratory
- 25. Communication Satellites for the 70's: Technology (1971)**
Nathaniel E. Feldman
The Rand Corp.
Charles M. Kelly
The Aerospace Corp.
- 26. Communication Satellites for the 70's: Systems (1971)**
Nathaniel E. Feldman
The Rand Corp.
Charles M. Kelly
The Aerospace Corp.
- 27. Thermospheric Circulation (1972)**
Willis L. Webb
Atmospheric Sciences Laboratory, White Sands, and University of Texas at El Paso
- 28. Thermal Characteristics of the Moon (1972)**
John W. Lucas
Jet Propulsion Laboratory
- *29. Fundamentals of Spacecraft Thermal Design (1972)**
John W. Lucas
Jet Propulsion Laboratory
- *30. Solar Activity Observations and Predictions (1972)**
Patrick S. McIntosh
Murray Dryer
Environmental Research Laboratories, National Oceanic and Atmospheric Administration
- *31. Thermal Control and Radiation (1973)**
Chang-Lin Tien
University of California at Berkeley
- *32. Communications Satellite Systems (1974)**
P. L. Bargellini
COMSAT Laboratories
- *33. Communications Satellite Technology (1974)**
P. L. Bargellini
COMSAT Laboratories
- *34. Instrumentation for Airbreathing Propulsion (1974)**
Allen E. Fuhs
Naval Postgraduate School
Marshall Kingery
Arnold Engineering Development Center
- *35. Thermophysics and Spacecraft Thermal Control (1974)**
Robert G. Hering
University of Iowa
- 36. Thermal Pollution Analysis (1975)**
Joseph A. Schetz
Virginia Polytechnic Institute
ISBN 0-915928-00-0
- *37. Aeroacoustics: Jet and Combustion Noise; Duct Acoustics (1975)**
Henry T. Nagamatsu, Editor
General Electric Research and Development Center
Jack V. O'Keefe,
Associate Editor
The Boeing Co.
Ira R. Schwartz, Associate Editor
NASA Ames Research Center
ISBN 0-915928-01-9
- *38. Aeroacoustics: Fan, STOL, and Boundary Layer Noise; Sonic Boom; Aeroacoustics Instrumentation (1975)**
Henry T. Nagamatsu, Editor
General Electric Research and Development Center
Jack V. O'Keefe,
Associate Editor
The Boeing Co.
Ira R. Schwartz, Associate Editor
NASA Ames Research Center
ISBN 0-915928-02-7
- *39. Heat Transfer with Thermal Control Applications (1975)**
M. Michael Yovanovich
University of Waterloo
ISBN 0-915928-03-5
- *40. Aerodynamics of Base Combustion (1976)**
S. N. B. Murthy, Editor
J. R. Osborn, Associate Editor
Purdue University
A. W. Barrows J. R. Ward,
Associate Editors
Ballistics Research Laboratories
ISBN 0-915928-04-3

*Out of print.

- *41. Communications Satellite Developments: Systems (1976)**
Gilbert E. LaVean
Defense Communications Agency
William G. Schmidt
CML Satellite Corp.
ISBN 0-915928-05-1
- *42. Communications Satellite Developments: Technology (1976)**
William G. Schmidt
CML Satellite Corp.
Gilbert E. LaVean
Defense Communications Agency
ISBN 0-915928-06-X
- *43. Aeroacoustics: Jet Noise, Combustion and Core Engine Noise (1976)**
Ira R. Schwartz, Editor
NASA Ames Research Center
Henry T. Nagamatsu,
Associate Editor
General Electric Research and Development Center
Warren C. Strahle,
Associate Editor
Georgia Institute of Technology
ISBN 0-915928-07-8
- *44. Aeroacoustics: Fan Noise and Control; Duct Acoustics; Rotor Noise (1976)**
Ira R. Schwartz, Editor
NASA Ames Research Center
Henry T. Nagamatsu,
Associate Editor
General Electric Research and Development Center
Warren C. Strahle,
Associate Editor
Georgia Institute of Technology
ISBN 0-915928-08-6
- *45. Aeroacoustics: STOL Noise; Airframe and Airfoil Noise (1976)**
Ira R. Schwartz, Editor
NASA Ames Research Center
Henry T. Nagamatsu,
Associate Editor
General Electric Research and Development Center
Warren C. Strahle,
Associate Editor
Georgia Institute of Technology
ISBN 0-915928-09-4
- *46. Aeroacoustics: Acoustics Wave Propagation; Aircraft Noise Prediction; Aeroacoustic Instrumentation (1976)**
Ira R. Schwartz, Editor
NASA Ames Research Center
Henry T. Nagamatsu,
Associate Editor
General Electric Research and Development Center
Warren C. Strahle,
Associate Editor
Georgia Institute of Technology
ISBN 0-915928-10-8
- *47. Spacecraft Charging by Magnetospheric Plasmas (1976)**
Alan Rosen
TRW Inc.
ISBN 0-915928-11-6
- *48. Scientific Investigations on the Skylab Satellite (1976)**
Marion I. Kent
Ernst Stuhlinger
NASA George C. Marshall Space Flight Center
Shi-Tsan Wu
University of Alabama
ISBN 0-915928-12-4
- *49. Radiative Transfer and Thermal Control (1976)**
Allie M. Smith
ARO Inc.
ISBN 0-915928-13-2
- *50. Exploration of the Outer Solar System (1976)**
Eugene W. Greenstadt
TRW Inc.
Murray Dryer
National Oceanic and Atmospheric Administration
Devrie S. Intriligator
University of Southern California
ISBN 0-915928-14-0
- *51. Rarefied Gas Dynamics, Parts I and II (two volumes) (1977)**
J. Leith Potter
ARO Inc.
ISBN 0-915928-15-9
- *52. Materials Sciences in Space with Application to Space Processing (1977)**
Leo Steg
General Electric Co.
ISBN 0-915928-16-7
- *53. Experimental Diagnostics in Gas Phase Combustion Systems (1977)**
Ben T. Zinn, Editor
Georgia Institute of Technology
Craig T. Bowman,
Associate Editor
Stanford University
Daniel L. Hartley,
Associate Editor
Sandia Laboratories
Edward W. Price,
Associate Editor
Georgia Institute of Technology
James G. Skifstad,
Associate Editor
Purdue University
ISBN 0-915928-18-3

- *54. Satellite Communications: Future Systems (1977)**
David Jarett
TRW Inc.
ISBN 0-915928-18-3
- *55. Satellite Communications: Advanced Technologies (1977)**
David Jarett
TRW Inc.
ISBN 0-915928-19-1
- *56. Thermophysics of Spacecraft and Outer Planet Entry Probes (1977)**
Allie M. Smith
ARO Inc.
ISBN 0-915928-20-5
- *57. Space-Based Manufacturing from Nonterrestrial Materials (1977)**
Gerald K. O'Neill, Editor
Brian O'Leary,
Assistant Editor
Princeton University
ISBN 0-915928-21-3
- *58. Turbulent Combustion (1978)**
Lawrence A. Kennedy
State University of New York at Buffalo
ISBN 0-915928-22-1
- *59. Aerodynamic Heating and Thermal Protection Systems (1978)**
Leroy S. Fletcher
University of Virginia
ISBN 0-915928-23-X
- *60. Heat Transfer and Thermal Control Systems (1978)**
Leroy S. Fletcher
University of Virginia
ISBN 0-915928-24-8
- *61. Radiation Energy Conversion in Space (1978)**
Kenneth W. Billman
NASA Ames Research Center
ISBN 0-915928-26-4
- *62. Alternative Hydrocarbon Fuels: Combustion and Chemical Kinetics (1978)**
Craig T. Bowman
Stanford University
Jorgen Birkeland
Department of Energy
ISBN 0-915928-25-6
- *63. Experimental Diagnostics in Combustion of Solids (1978)**
Thomas L. Boggs
Naval Weapons Center
Ben T. Zinn
Georgia Institute of Technology
ISBN 0-915928-28-0
- *64. Outer Planet Entry Heating and Thermal Protection (1979)**
Raymond Viskanta
Purdue University
ISBN 0-915928-29-9
- *65. Thermophysics and Thermal Control (1979)**
Raymond Viskanta
Purdue University
ISBN 0-915928-30-2
- *66. Interior Ballistics of Guns (1979)**
Herman Krier
University of Illinois at Urbana-Champaign
Martin Summerfield
New York University
ISBN 0-915928-32-9
- *67. Remote Sensing of Earth from Space: Role of "Smart Sensors" (1979)**
Roger A. Breckenridge
NASA Langley Research Center
ISBN 0-915928-33-7
- *68. Injection and Mixing in Turbulent Flow (1980)**
Joseph A. Schetz
Virginia Polytechnic Institute and State University
ISBN 0-915928-35-3
- *69. Entry Heating and Thermal Protection (1980)**
Walter B. Olstad
NASA Headquarters
ISBN 0-915928-38-8
- *70. Heat Transfer, Thermal Control, and Heat Pipes (1980)**
Walter B. Olstad
NASA Headquarters
ISBN 0-915928-39-6
- *71. Space Systems and Their Interactions with Earth's Space Environment (1980)**
Henry B. Garrett
Charles P. Pike
Hanscom Air Force Base
ISBN 0-915928-41-8
- *72. Viscous Flow Drag Reduction (1980)**
Gary R. Hough
Vought Advanced Technology Center
ISBN 0-915928-44-2
- *73. Combustion Experiments in a Zero-Gravity Laboratory (1981)**
Thomas H. Cochran
NASA Lewis Research Center
ISBN 0-915928-48-5
- *74. Rarefied Gas Dynamics, Parts I and II (two volumes) (1981)**
Sam S. Fisher
University of Virginia
ISBN 0-915928-51-5

*Out of print.

- *75. Gasdynamics of Detonations and Explosions (1981)**
 J. R. Bowen
University of Wisconsin at Madison
 N. Manson
Universite de Poitiers
 A. K. Oppenheim
University of California at Berkeley
 R. I. Soloukhin
Institute of Heat and Mass Transfer, BSSR Academy of Sciences
 ISBN 0-915928-46-9
- *76. Combustion in Reactive Systems (1981)**
 J. R. Bowen
University of Wisconsin at Madison
 N. Manson
Universite de Poitiers
 A. K. Oppenheim
University of California at Berkeley
 R. I. Soloukhin
Institute of Heat and Mass Transfer, BSSR Academy of Sciences
 ISBN 0-915928-47-7
- *77. Aerothermodynamics and Planetary Entry (1981)**
 A. L. Crosbie
University of Missouri-Rolla
 ISBN 0-915928-52-3
- *78. Heat Transfer and Thermal Control (1981)**
 A. L. Crosbie
University of Missouri-Rolla
 ISBN 0-915928-53-1
- *79. Electric Propulsion and Its Applications to Space Missions (1981)**
 Robert C. Finke
NASA Lewis Research Center
 ISBN 0-915928-55-8
- *80. Aero-Optical Phenomena (1982)**
 Keith G. Gilbert
 Leonard J. Otten
Air Force Weapons Laboratory
 ISBN 0-915928-60-4
- *81. Transonic Aerodynamics (1982)**
 David Nixon
Nielsen Engineering & Research, Inc.
 ISBN 0-915928-65-5
- *82. Thermophysics of Atmospheric Entry (1982)**
 T. E. Horton
University of Mississippi
 ISBN 0-915928-66-3
- *83. Spacecraft Radiative Transfer and Temperature Control (1982)**
 T. E. Horton
University of Mississippi
 ISBN 0-915928-67-1
- *84. Liquid-Metal Flows and Magneto-hydrodynamics (1983)**
 H. Branover Ben-Gurion
University of the Negev
 P. S. Lykoudis
Purdue University
 A. Yakhot
Ben-Gurion University of the Negev
 ISBN 0-915928-70-1
- *85. Entry Vehicle Heating and Thermal Protection Systems: Space Shuttle, Solar Starprobe, Jupiter Galileo Probe (1983)**
 Paul E. Bauer
McDonnell Douglas Astronautics Co.
 Howard E. Collicott
The Boeing Co.
 ISBN 0-915928-74-4
- *86. Spacecraft Thermal Control, Design, and Operation (1983)**
 Howard E. Collicott
The Boeing Co.
 Paul E. Bauer
McDonnell Douglas Astronautics Co.
 ISBN 0-915928-75-2
- *87. Shock Waves, Explosions, and Detonations (1983)**
 J. R. Bowen
University of Washington
 N. Manson
Universite de Poitiers
 A. K. Oppenheim
University of California at Berkeley
 R. I. Soloukhin
Institute of Heat and Mass Transfer, BSSR Academy of Sciences
 ISBN 0-915928-76-0
- *88. Flames, Lasers, and Reactive Systems (1983)**
 J. R. Bowen
University of Washington
 N. Manson
Universite de Poitiers
 A. K. Oppenheim
University of California at Berkeley
 R. I. Soloukhin
Institute of Heat and Mass Transfer, BSSR Academy of Sciences
 ISBN 0-915928-77-9
- *89. Orbit-Raising and Maneuvering Propulsion: Research Status and Needs (1984)**
 Leonard H. Caveny
Air Force Office of Scientific Research
 ISBN 0-915928-82-5

SERIES LISTING

- *90. Fundamentals of Solid-Propellant Combustion (1984)**
Kenneth K. Kuo
Pennsylvania State University
Martin Summerfield
Princeton Combustion Research Laboratories, Inc.
ISBN 0-915928-84-1
- 91. Spacecraft Contamination: Sources and Prevention (1984)**
J. A. Roux
University of Mississippi
T. D. McCay
NASA Marshall Space Flight Center
ISBN 0-915928-85-X
- 92. Combustion Diagnostics by Nonintrusive Methods (1984)**
T. D. McCay
NASA Marshall Space Flight Center
J. A. Roux
University of Mississippi
ISBN 0-915928-86-8
- 93. The INTELSAT Global Satellite System (1984)**
Joel Alper
COMSAT Corp.
Joseph Pelton
INTELSAT
ISBN 0-915928-90-6
- 94. Dynamics of Shock Waves, Explosions, and Detonations (1984)**
J. R. Bowen
University of Washington
N. Manson
Universite de Poitiers
A. K. Oppenheim
University of California at Berkeley
R. I. Soloukhin
Institute of Heat and Mass Transfer, BSSR Academy of Sciences
ISBN 0-915928-91-4
- 95. Dynamics of Flames and Reactive Systems (1984)**
J. R. Bowen
University of Washington
N. Manson
Universite de Poitiers
A. K. Oppenheim
University of California at Berkeley
R. I. Soloukhin
Institute of Heat and Mass Transfer, BSSR Academy of Sciences
ISBN 0-915928-92-2
- 96. Thermal Design of Aeroassisted Orbital Transfer Vehicles (1985)**
H. F. Nelson
University of Missouri-Rolla
ISBN 0-915928-94-9
- 97. Monitoring Earth's Ocean, Land, and Atmosphere from Space—Sensors, Systems, and Applications (1985)**
Abraham Schnapf
Aerospace Systems Engineering
ISBN 0-915928-98-1
- 98. Thrust and Drag: Its Prediction and Verification (1985)**
Eugene E. Covert
Massachusetts Institute of Technology
C. R. James
Vought Corp.
William F. Kimzey
Sverdrup Technology
AEDC Group
George K. Richey
U.S. Air Force
Eugene C. Rooney
U.S. Navy Department of Defence
ISBN 0-930403-00-2
- 99. Space Stations and Space Platforms—Concepts, Designs, Infrastructure, and Uses (1985)**
Ivan Bekey
Daniel Herman
NASA Headquarters
ISBN 0-930403-01-0
- 100. Single- and Multi-Phase Flows in an Electromagnetic Field: Energy, Metallurgical, and Solar Applications (1985)**
Herman Branover
Ben-Gurion University of the Negev
Paul S. Lykoudis
Purdue University
Michael Mond
Ben-Gurion University of the Negev
ISBN 0-930403-04-5
- 101. MHD Energy Conversions: Physiotechnical Problems (1986)**
V. A. Kirillin
A. E. Sheyndlin
Soviet Academy of Sciences
ISBN 0-930403-05-3
- 102. Numerical Methods for Engine-Airframe Integration (1986)**
S. N. B. Murthy
Purdue University
Gerald C. Paynter
Boeing Airplane Co.
ISBN 0-930403-09-6
- 103. Thermophysical Aspects of Re-Entry Flows (1986)**
James N. Moss
NASA Langley Research Center
Carl D. Scott
NASA Johnson Space Center
ISBN 0-930403-10-X

*Out of print.

- *104. Tactical Missile Aerodynamics (1986)**
 M. J. Hemsch
PRC Kentron, Inc.
 J. N. Nielson
NASA Ames Research Center
 ISBN 0-930403-13-4
- 105. Dynamics of Reactive Systems Part I: Flames and Configurations; Part II: Modeling and Heterogeneous Combustion (1986)**
 J. R. Bowen
University of Washington
 J.-C. Leyer
Universite de Poitiers
 R. I. Soloukhin
Institute of Heat and Mass Transfer, BSSR Academy of Sciences
 ISBN 0-930403-14-2
- 106. Dynamics of Explosions (1986)**
 J. R. Bowen
University of Washington
 J.-C. Leyer
Universite de Poitiers
 R. I. Soloukhin
Institute of Heat and Mass Transfer, BSSR Academy of Sciences
 ISBN 0-930403-15-0
- *107. Spacecraft Dielectric Material Properties and Spacecraft Charging (1986)**
 A. R. Frederickson
U.S. Air Force Rome Air Development Center
 D. B. Cotts
SRI International
 J. A. Wall
U.S. Air Force Rome Air Development Center
 F. L. Bouquet
Jet Propulsion Laboratory, California Institute of Technology
 ISBN 0-930403-17-7
- *108. Opportunities for Academic Research in a Low-Gravity Environment (1986)**
 George A. Hazelrigg
National Science Foundation
 Joseph M. Reynolds
Louisiana State University
 ISBN 0-930403-18-5
- 109. Gun Propulsion Technology (1988)**
 Ludwig Stiefel
U.S. Army Armament Research, Development and Engineering Center
 ISBN 0-930403-20-7
- 110. Commercial Opportunities in Space (1998)**
 F. Shahrokhi
 K. E. Harwell
University of Tennessee Space Institute
 C. C. Chao
National Cheng Kung University
 ISBN 0-930403-39-8
- 111. Liquid-Metal Flows: Magnetohydrodynamics and Application (1988)**
 Herman Branover
 Michael Mond
 Yeshajahu Unger
Ben-Gurion University of the Negev
 ISBN 0-930403-43-6
- 112. Current Trends in Turbulence Research (1988)**
 Herman Branover
 Michael Mond
 Yeshajahu Unger
Ben-Gurion University of the Negev
 ISBN 0-930403-44-4
- 113. Dynamics of Reactive Systems Part I: Flames; Part II: Heterogeneous Combustion and Applications (1988)**
 A. L. Kuhl
R&D Associates
 J. R. Bowen
University of Washington
 J.-C. Leyer
Universite de Poitiers
 A. Borisov
USSR Academy of Sciences
 ISBN 0-930403-46-0
- 114. Dynamics of Explosions (1988)**
 A. L. Kuhl
R&D Associates
 J. R. Bowen
University of Washington
 J.-C. Leyer
Universite de Poitiers
 A. Borisov
USSR Academy of Sciences
 ISBN 0-930403-47-9
- 115. Machine Intelligence and Autonomy for Aerospace (1988)**
 E. Heer
Heer Associates, Inc.
 H. Lum
NASA Ames Research Center
 ISBN 0-930403-48-7
- 116. Rarefied Gas Dynamics: Space Related Studies (1989)**
 E. P. Muntz
University of Southern California
 D. P. Weaver
U.S. Air Force Astronautics Laboratory (AFSC)
 D. H. Campbell
University of Dayton Research Institute
 ISBN 0-930403-53-3

*Out of print.

SERIES LISTING

- 117. Rarefied Gas Dynamics: Physical Phenomena (1989)**
 E. P. Muntz
University of Southern California
 D. P. Weaver
U.S. Air Force Astronautics Laboratory (AFSC)
 D. H. Campbell
University of Dayton Research Institute
 ISBN 0-930403-54-1
- 118. Rarefied Gas Dynamics: Theoretical and Computational Techniques (1989)**
 E. P. Muntz
University of Southern California
 D. P. Weaver
U.S. Air Force Astronautics Laboratory (AFSC)
 D. H. Campbell
University of Dayton Research Institute
 ISBN 0-930403-55-X
- 119. Test and Evaluation of the Tactical Missile (1989)**
 Emil J. Eichblatt Jr.
Pacific Missile Test Center
 ISBN 0-930403-56-8
- 120. Unsteady Transonic Aerodynamics (1989)**
 David Nixon
Nielsen Engineering & Research, Inc.
 ISBN 0-930403-52-5
- 121. Orbital Debris from Upper-Stage Breakup (1989)**
 Joseph P. Loftus Jr.
NASA Johnson Space Center
 ISBN 0-930403-58-4
- 122. Thermal-Hydraulics for Space Power, Propulsion and Thermal Management System Design (1990)**
 William J. Krotiuk
General Electric Co.
 ISBN 0-930403-64-9
- 123. Viscous Drag Reduction in Boundary Layers (1990)**
 Dennis M. Bushnell
 Jerry N. Hefner
NASA Langley Research Center
 ISBN 0-930403-66-5
- *124. Tactical and Strategic Missile Guidance (1990)**
 Paul Zarchan
Charles Stark Draper Laboratory, Inc.
 ISBN 0-930403-68-1
- 125. Applied Computational Aerodynamics (1990)**
 P. A. Henne
Douglas Aircraft Company
 ISBN 0-930403-69-X
- 126. Space Commercialization: Launch Vehicles and Programs (1990)**
 F. Shahrokhi
University of Tennessee Space Institute
 J. S. Greenberg
Princeton Synergetics Inc.
 T. Al-Saud
Ministry of Defense and Aviation Kingdom of Saudi Arabia
 ISBN 0-930403-75-4
- 127. Space Commercialization: Platforms and Processing (1990)**
 F. Shahrokhi
University of Tennessee Space Institute
 G. Hazelrigg
National Science Foundation
 R. Bayuzick
Vanderbilt University
 ISBN 0-930403-76-2
- 128. Space Commercialization: Satellite Technology (1990)**
 F. Shahrokhi
University of Tennessee Space Institute
 N. Jasentuliyana
United Nations
 N. Tarabzouni
King Abulaziz City for Science and Technology
 ISBN 0-930403-77-0
- *129. Mechanics and Control of Large Flexible Structure (1990)**
 John L. Junkins
Texas A&M University
 ISBN 0-930403-73-8
- 130. Low-Gravity Fluid Dynamics and Transport Phenomena (1990)**
 Jean N. Koster
 Robert L. Sani
University of Colorado at Boulder
 ISBN 0-930403-74-6
- 131. Dynamics of Deflagrations and Reactive Systems: Flames (1991)**
 A. L. Kuhl
Lawrence Livermore National Laboratory
 J.-C. Leyer
Universite de Poitiers
 A. A. Borisov
USSR Academy of Sciences
 W. A. Sirignano
University of California
 ISBN 0-930403-95-9

*Out of print.

132. Dynamics of Deflagrations and Reactive Systems: Heterogeneous Combustion (1991)

A. L. Kuhl
Lawrence Livermore National Laboratory
 J.-C. Leyer
Universite de Poitiers
 A. A. Borisov
USSR Academy of Sciences
 W. A. Sirignano
University of California
 ISBN 0-930403-96-7

133. Dynamics of Detonations and Explosions: Detonations (1991)

A. L. Kuhl
Lawrence Livermore National Laboratory
 J.-C. Leyer
Universite de Poitiers
 A. A. Borisov
USSR Academy of Sciences
 W. A. Sirignano
University of California
 ISBN 0-930403-97-5

134. Dynamics of Detonations and Explosions: Explosion Phenomena (1991)

A. L. Kuhl
Lawrence Livermore National Laboratory
 J.-C. Leyer
Universite de Poitiers
 A. A. Borisov
USSR Academy of Sciences
 W. A. Sirignano
University of California
 ISBN 0-930403-98-3

135. Numerical Approaches to Combustion Modeling (1991)

Elaine S. Oran
 Jay P. Boris
Naval Research Laboratory
 ISBN 1-56347-004-7

136. Aerospace Software Engineering (1991)

Christine Anderson
U.S. Air Force Wright Laboratory
 Merlin Dorfman
Lockheed Missiles & Space Company, Inc.
 ISBN 1-56347-005-0

137. High-Speed Flight Propulsion Systems (1991)

S. N. B. Murthy
Purdue University
 E. T. Curran
Wright Laboratory
 ISBN 1-56347-011-X

138. Propagation of Intensive Laser Radiation in Clouds (1992)

O. A. Volkovitsky
 Yu. S. Sedenov
 L. P. Semenov
Institute of Experimental Meteorology
 ISBN 1-56347-020-9

139. Gun Muzzle Blast and Flash (1992)

Günter Klingerberg
Fraunhofer-Institut für Kurzzeitdynamik, Ernst-Mach-Institut
 Joseph M. Heimerl
U.S. Army Ballistic Research Laboratory
 ISBN 1-56347-012-8

140. Thermal Structures and Materials for High-Speed Flight (1992)

Earl. A. Thornton
University of Virginia
 ISBN 1-56347-017-9

141. Tactical Missile Aerodynamics: General Topics (1992)

Michael J. Hemsch
Lockheed Engineering & Sciences Company
 ISBN 1-56347-015-2

142. Tactical Missile Aerodynamics: Prediction Methodology (1992)
 Michael R. Mendenhall
Nielsen Engineering & Research, Inc.
 ISBN 1-56347-016-0

143. Nonsteady Burning and Combustion Stability of Solid Propellants (1992)

Luigi De Luca
Politecnico di Milano
 Edward W. Price
Georgia Institute of Technology
 Martin Summerfield
Princeton Combustion Research Laboratories, Inc.
 ISBN 1-56347-014-4

144. Space Economics (1992)

Joel S. Greenberg
Princeton Synergetics, Inc.
 Henry R. Hertzfeld
HRH Associates
 ISBN 1-56347-042-X

145. Mars: Past, Present, and Future (1992)

E. Brian Pritchard
NASA Langley Research Center
 ISBN 1-56347-043-8

146. Computational Nonlinear Mechanics in Aerospace Engineering (1992)

Satya N. Atluri
Georgia Institute of Technology
 ISBN 1-56347-044-6

147. Modern Engineering for Design of Liquid-Propellant Rocket Engines (1992)

Dieter K. Huzel
 David H. Huang
Rocketdyne Division of Rockwell International
 ISBN 1-56347-013-6

SERIES LISTING

148. Metallurgical Technologies, Energy Conversion, and Magneto-hydrodynamic Flows (1993)

Herman Branover
Yeshajahu Unger
Ben-Gurion University of the Negev
ISBN 1-56347-019-5

149. Advance in Turbulence Studies (1993)

Herman Branover
Yeshajahu Unger
Ben-Gurion University of the Negev
ISBN 1-56347-018-7

150. Structural Optimization: Status and Promise (1993)

Manohar P. Kamat
Georgia Institute of Technology
ISBN 1-56347-056-X

151. Dynamics of Gaseous Combustion (1993)

A. L. Kuhl
Lawrence Livermore National Laboratory
J.-C. Leyer
Universite de Poitiers
A. A. Borisov
USSR Academy of Sciences
W. A. Sirignano
University of California
ISBN 1-56347-060-8

152. Dynamics of Heterogeneous Gaseous Combustion and Reacting Systems (1993)

A. L. Kuhl
Lawrence Livermore National Laboratory
J.-C. Leyer
Universite de Poitiers
A. A. Borisov
USSR Academy of Sciences
W. A. Sirignano
University of California
ISBN 1-56347-058-6

153. Dynamic Aspects of Detonations (1993)

A. L. Kuhl
Lawrence Livermore National Laboratory
J.-C. Leyer
Universite de Poitiers
A. A. Borisov
USSR Academy of Sciences
W. A. Sirignano
University of California
ISBN 1-56347-057-8

154. Dynamic Aspects of Explosion Phenomena (1993)

A. L. Kuhl
Lawrence Livermore National Laboratory
J.-C. Leyer
Universite de Poitiers
A. A. Borisov
USSR Academy of Sciences
W. A. Sirignano
University of California
ISBN 1-56347-059-4

155. Tactical Missile Warheads (1993)

Joseph Carleone
Aerojet General Corporation
ISBN 1-56347-067-5

156. Toward a Science of Command, Control, and Communications (1993)

Carl R. Jones
Naval Postgraduate School
ISBN 1-56347-068-3

***157. Tactical and Strategic Missile Guidance Second Edition (1994)**

Paul Zarchan
Charles Stark Draper Laboratory, Inc.
ISBN 1-56347-077-2

158. Rarefied Gas Dynamics: Experimental Techniques and Physical Systems (1994)

Bernie D. Shizgal
University of British Columbia
David P. Weaver
Phillips Laboratory
ISBN 1-56347-079-9

159. Rarefied Gas Dynamics: Theory and Simulations (1994)

Bernie D. Shizgal
University of British Columbia
David P. Weaver
Phillips Laboratory
ISBN 1-56347-080-2

160. Rarefied Gas Dynamics: Space Sciences and Engineering (1994)

Bernie D. Shizgal
University of British Columbia
David P. Weaver
Phillips Laboratory
ISBN 1-56347-081-0

161. Teleoperation and Robotics in Space (1994)

Steven B. Skaar
University of Notre Dame
Carl F. Ruoff
Jet Propulsion Laboratory, California Institute of Technology
ISBN 1-56347-095-0

162. Progress in Turbulence Research (1994)

Herman Branover
Yeshajahu Unger
Ben-Gurion University of the Negev
ISBN 1-56347-095-0

*Out of print.

- 163. Global Positioning System: Theory and Applications, Volume I (1996)**
 Bradford W. Parkinson
Stanford University
 James J. Spilker Jr.
Stanford Telecom
 Penina Axelrad, Associate Editor
University of Colorado
 Per Enge, Associate Editor
Stanford University
 ISBN 1-56347-107-8
- 164. Global Positioning System: Theory and Applications, Volume II (1996)**
 Bradford W. Parkinson
Stanford University
 James J. Spilker Jr.
Stanford Telecom
 Penina Axelrad, Associate Editor
University of Colorado
 Per Enge, Associate Editor
Stanford University
 ISBN 1-56347-106-X
- 165. Developments in High-Speed Vehicle Propulsion Systems (1996)**
 S. N. B. Murthy
Purdue University
 E. T. Curran
Wright Laboratory
 ISBN 1-56347-176-0
- 166. Recent Advances in Spray Combustion: Spray Atomization and Drop Burning Phenomena, Volume I (1996)**
 Kenneth K. Kuo
Pennsylvania State University
 ISBN 1-56347-175-2
- 167. Fusion Energy in Space Propulsion (1995)**
 Terry Kammash
University of Michigan
 ISBN 1-56347-184-1
- 168. Aerospace Thermal Structures and Materials for a New Era (1995)**
 Earl A. Thornton
University of Virginia
 ISBN 1-56347-182-5
- 169. Liquid Rocket Engine Combustion Instability (1995)**
 Vigor Yang
 William E. Anderson
Pennsylvania State University
 ISBN 1-56347-183-3
- 170. Tactical Missile Propulsion (1996)**
 G. E. Jensen
United Technologies Corporation
 David W. Netzer
Naval Postgraduate School
 ISBN 1-56347-118-3
- 171. Recent Advances in Spray Combustion: Spray Combustion Measurements and Model Simulation, Volume II (1996)**
 Kenneth K. Kuo
Pennsylvania State University
 ISBN 1-56347-181-7
- 172. Future Aeronautical and Space Systems (1997)**
 Ahmed K. Noor
NASA Langley Research Center
 Samuel L. Venneri
NASA Headquarters
 ISBN 1-56347-188-4
- 173. Advances in Combustion Science: In Honor of Ya. B. Zel'dovich (1997)**
 William A. Sirignano
University of California
 Alexander G. Merzhanov
Russian Academy of Sciences
 Luigi De Luca
Politecnico di Milano
 ISBN 1-56347-178-7
- 174. Fundamentals of High Accuracy Inertial Navigation (1997)**
 Averil B. Chatfield
 ISBN 1-56347-243-0
- 175. Liquid Propellant Gun Technology (1997)**
 Günter Klingenberg
Fraunhofer-Institut für Kurzzeitdynamik, Ernst-Mach-Institut
 John D. Knapton
 Walter F. Morrison
 Gloria P. Wren
U.S. Army Research Laboratory
 ISBN 1-56347-196-5
- 176. Tactical and Strategic Missile Guidance Third Edition (1998)**
 Paul Zarchan
Charles Stark Draper Laboratory, Inc.
 ISBN 1-56347-279
- 177. Orbital and Celestial Mechanics (1998)**
 John P. Vinti
 Gim J. Der, Editor
TRW
 Nino L. Bonavito, Editor
NASA Goddard Space Flight Center
 ISBN 1-56347-256-2

SERIES LISTING

178. Some Engineering Applications in Random Vibrations and Random Structures (1998)

Giora Maymon

RAFAEL

ISBN 1-56347-258-9

179. Conventional Warhead Systems Physics and Engineering Design (1998)

Richard M. Lloyd

Raytheon Systems Company

ISBN 1-56347-255-4

180. Advances in Missile Guidance Theory (1998)

Joseph Z. Ben-Asher

Isaac Yaesh

Israel Military Industries—Advanced Systems Division

ISBN 1-56347-275-9

181. Satellite Thermal Control for Systems Engineers (1998)

Robert D. Karam

ISBN 1-56347-276-7

182. Progress in Fluid Flow Research: Turbulence and Applied MHD (1998)

Yeshajahu Unger

Herman Branover

Ben-Gurion University of the Negev

ISBN 1-56347-284-8

183. Aviation Weather Surveillance Systems (1999)

Pravas R. Mahapatra

Indian Institute of Science

ISBN 1-56347-340-2

184. Flight Control Systems (2000)

Roger W. Pratt, Editor

Loughborough University

ISBN 1-56347-404-2

185. Solid Propellant Chemistry, Combustion, and Motor Interior Ballistics (2000)

Vigor Yang

Pennsylvania State University

Thomas B. Brill

University of Delaware

Wu-Zhen Ren

China Ordnance Society

ISBN 1-56347-442-5

186. Approximate Methods for Weapons Aerodynamics (2000)

Frank G. Moore

ISBN 1-56347-399-2

187. Micropropulsion for Small Spacecraft (2000)

Michael M. Micci

Pennsylvania State University

Andrew D. Ketsdever

Air Force Research Laboratory, Edwards Air Force Base

ISBN 1-56347-448-4

188. Structures

Technology for Future Aerospace Systems (2000)

Ahmed K. Noor

NASA Langley Research Center

ISBN 1-56347-384-4

189. Scramjet Propulsion (2000)

E. T. Curran

Department of the Air Force

S. N. B. Murthy

Purdue University

ISBN 1-56347-322-4

190. Fundamentals of Kalman Filtering: A Practical Approach (2000)

Paul Zarchan

Howard Musoff

Charles Stark Draper Laboratory, Inc.

ISBN 1-56347-455-7

191. Gossamer Spacecraft: Membrane and Inflatable Structures Technology for Space Applications (2001)

Christopher H. M. Jenkins

South Dakota School of Mines

ISBN 1-56347-403-4

192. Theater Ballistic Missile Defense (2001)

Ben-Zion Naveh

Azriel Lorber

Wales Ltd.

ISBN 1-56347-385-2

193. Air Transportation Systems Engineering (2001)

George L. Donohue

George Mason University

Andres Zellweger

Embry Riddle Aeronautical University

ISBN 1-56347-474-3

194. Physics of Direct Hit and Near Miss Warhead Technology (2001)

Richard M. Lloyd

Raytheon Electronics Systems

ISBN 1-56347-473-5

195. Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications (2001)

Thomas J. Mueller

University of Notre Dame

ISBN 1-56347-517-0

196. Physical and Chemical Processes in Gas Dynamics: Cross Sections and Rate Constants for Physical and Chemical Processes, Volume I (2002)

G. G. Chernyi

S. A. Losev

Moscow State University

S. O. Macharet

Princeton University

B. V. Potapkin

Kurchatov Institute

ISBN 1-56347-518-9

*Out of print.

**197. Physical and
Chemical Processes in Gas
Dynamics: Physical and
Chemical Kinetics and
Thermodynamics of Gases
and Plasmas, Volume II
(2002)**

G. G. Chernyi
S. A. Losev

Moscow State University

S. O. Macharet

Princeton University

B. V. Potapkin

Kurchatov Institute

ISBN 1-56347-519-7

**198. Advanced Hypersonic
Test Facilities (2002)**

Frank K. Lu

University of Texas at

Arlington

Dan E. Marren

Arnold Engineering

Development Center

ISBN 1-56347-541-3

**199. Tactical and Strategic
Missile Guidance, Fourth
Edition (2002)**

Paul Zarchan

MIT Lincoln Laboratory

ISBN 1-56347-497-2