

NAVIGATION AND CONTROL STUDIES ON CRUISE MISSILES

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

VEDAT EKÜTEKİN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN  
MECHANICAL ENGINEERING

JANUARY 2007

Approval of the Graduate School of Natural and Applied Sciences

---

Prof. Dr. Canan ÖZGEN  
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

---

Prof. Dr. S. Kemal İDER  
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

---

Prof. Dr. M. Kemal ÖZGÖREN  
Supervisor

**Examining Committee Members**

Prof. Dr. Bülent E. PLATİN	(METU, ME) _____
Prof. Dr. M. Kemal ÖZGÖREN	(METU, ME) _____
Prof. Dr. M. Kemal LEBLEBİCİOĞLU	(METU, EEE) _____
Prof. Dr. Reşit SOYLU	(METU, ME) _____
Asst. Prof. Dr. Metin U. SALAMCI	(Gazi Univ., ME) _____

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name : Vedat EKÜTEKİN  
Signature :

## **ABSTRACT**

# **NAVIGATION AND CONTROL STUDIES ON CRUISE MISSILES**

EKÜTEKİN, Vedat

Ph. D., Department of Mechanical Engineering

Supervisor: Prof. Dr. M. Kemal ÖZGÖREN

January 2007, 298 pages

A cruise missile is a guided missile that uses a lifting wing and a jet propulsion system to allow sustained flight. Cruise missiles are, in essence, unmanned aircraft and they are generally designed to carry a large conventional or nuclear warhead many hundreds of miles with excellent accuracy. In this study, navigation and control studies on cruise missiles are performed. Due to the variety and complexity of the subsystems of the cruise missiles, the main concern is limited with the navigation system. Navigation system determines the position, velocity, attitude and time solutions of the missile. Therefore, it can be concluded that an accurate self-contained navigation system directly influences the success of the missile. In the study, modern radar data association algorithms are implemented as new Terrain Aided Navigation (TAN) algorithms which can be used with low-cost Inertial Measurement Units (IMU's). In order to perform the study, first a thorough survey of the literature on mid-course navigation of cruise missiles is performed.

Then, study on modern radar data association algorithms and their implementations to TAN are done with simple simulations. At the case study part, a six degree of freedom (6 DOF) flight simulation tool is developed which includes the aerodynamic and dynamic model of the cruise missile model including error model of the navigation system. Finally, the performances of the designed navigation systems with the implemented TAN algorithms are examined in detail with the help of the simulations performed.

Keywords: Cruise Missile, Terrain Aided Navigation (TAN), Probabilistic Data Association Filter (PDAF), Track Splitting Filter (TSF), Multiple Hypothesis Tracking (MHT).

# ÖZ

## SEYİR FÜZELERİ ÜZERİNE SEYRÜSEFER VE DENETİM ÇALIŞMALARI

EKÜTEKİN, Vedat

Doktora, Makina Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. M. Kemal ÖZGÖREN

Ocak 2007, 298 sayfa

Seyir füzesi, kaldırma kanatları ve jet itki sistemi ile kararlı uçuş sağlayan güdümlü bir füzedir. Seyir füzeleri genellikle, büyük konvansiyonel ya da nükleer savaş başlıklarını uzak mesafelere çok hassas olarak taşıyan insansız hava taşıtlarıdır. Bu çalışmada, seyir füzeleri üzerine seyrüsefer ve denetim çalışmaları gerçekleştirilmiştir. Seyir füzelerinin alt sistemlerindeki çeşitlilik ve karmaşıklık nedeniyle, çalışmanın ana konusu seyrüsefer sistemiyle kısıtlanmıştır. Seyrüsefer sistemi füzenin konum, hız, yönelim ve zaman çözümlerini belirler. Bu nedenle, hassas, kendi kendine yeterli bir seyrüsefer sisteminin füzenin başarısını doğrudan etkileyeceği sonucuna varılabilir. Çalışmada, modern radar veri ilişkilendirme algoritmaları, düşük maliyetli ataletsel seyrüsefer sistemleri ile kullanılabilecek yeni Arazi Destekli Seyrüsefer (ADS) algoritma uygulamaları için kullanılmıştır. Çalışmayı gerçekleştirmek için, ilk aşamada seyir füzelerinin seyrüsefer yöntemlerine ait ayrıntılı kaynak araştırması yapılmıştır. Daha sonra, modern radar

veri ilişkilendirme algoritmaları üzerine çalışılmış ve bunların ADS uygulamaları basit benzetimlerle gerçekleştirilmiştir. Örnek olay incelemesi kısmında, seyir füzesinin aerodinamik ve dinamik modeli ile seyrüsefer sisteminin hata modellemesini de içeren altı serbestlik dereceli bir uçuş benzetim aracı geliştirilmiştir. Son olarak, yeni uygulanan ADS algoritmaları kullanılarak tasarlanan seyrüsefer sistemlerinin başarıları, gerçekleştirilen uçuş benzetimlerinin yardımıyla ayrıntılı olarak incelenmiştir.

**Anahtar Kelimeler:** Seyir Füzesi, Arazi Destekli Seyrüsefer (ADS), Olasılıklı Veri İlişkilendirme Filtresi (OVİF), İz Ayırma Filtresi (İAF), Çoklu Varsayımlı Takip (ÇVT)

To The Child Who Didn't Know Crying

## **ACKNOWLEDGMENTS**

The author wishes to express his sincere appreciation to his supervisor, Prof. Dr. M. Kemal ÖZGÖREN for his guidance, helpful suggestions, prompt feedbacks, encouragements and endless patience throughout the study.

The author would also like to thank to his Thesis Supervising Committee members, Prof. Dr. Bülent E. PLATİN and Prof. Dr. M. Kemal LEBLEBİCİOĞLU for their constructive comments and guidance throughout his study.

Also, the author would like to thank to his colleagues and friends Dr. Bülent ÖZKAN, Mr. Özgür ATEŞOĞLU and other friends from TÜBİTAK-SAGE (The Scientific & Technological Research Council of Türkiye – Defense Industries Research & Development Institute) for their friendship and help.

The author's greatest thanks go to his wife, Yasemin, and his daughter, Elif for their endless support, patience and understanding.

The permit and support of TÜBİTAK-SAGE for this study is also acknowledged.

## TABLE OF CONTENTS

ABSTRACT .....	iv
ÖZ .....	vi
ACKNOWLEDGMENTS .....	ix
TABLE OF CONTENTS .....	x
LIST OF TABLES .....	xv
LIST OF FIGURES .....	xvii
LIST OF SYMBOLS .....	xxiii
LIST OF ABBREVIATIONS .....	xxx
CHAPTER	
1. INTRODUCTION.....	1
1.1.Scope of the Study .....	1
1.2.Cruise Missiles.....	3
1.2.1.    Background .....	3
1.2.2.    Cruise Missile Technology .....	5
1.2.2.1.    Airframe .....	8
1.2.2.2.    Propulsion System.....	9
1.2.2.3.    Guidance Systems .....	9
1.2.2.3.1.    In-flight Guidance .....	10
1.2.2.3.2.    Terminal Guidance.....	11
1.2.2.4.    Warhead .....	12
1.2.3.    Low-Cost Cruise Missiles .....	13
1.3.Terrain Aided Navigation (TAN) .....	14

1.3.1.	TAN Techniques .....	15
1.3.2.	TAN System Considerations.....	18
1.3.3.	Digital Terrain Elevation Data (DTED).....	19
1.4.	Literature Survey on TAN .....	20
1.4.1.	Cruise Missile System Performance .....	22
1.4.2.	Terrain Models and Path Optimization.....	23
1.4.3.	TAN Applications .....	25
1.4.4.	TAN Algorithms .....	26
1.4.4.1.	Batch TAN Algorithms .....	27
1.4.4.2.	Recursive TAN Algorithms .....	29
1.4.5.	TAN Patents .....	31
1.5.	Target Tracking.....	32
1.5.1.	Background .....	33
1.5.2.	Data Association Algorithms .....	35
1.6.	Outline of the Thesis .....	38
2.	MAJOR TERRAIN AIDED NAVIGATION METHODS.....	40
2.1.	Cruise Missile INS Errors .....	40
2.1.1.	INS Only Errors .....	40
2.1.2.	TAN INS Errors .....	49
2.2.	TERCOM .....	51
2.2.1.	Background .....	51
2.2.2.	TERCOM Concept.....	54
2.2.3.	TERCOM Data Correlation Techniques.....	56
2.2.4.	Terrain Roughness Characteristics.....	60
2.2.5.	Simulations and Discussion .....	63
2.3.	SITAN .....	71

2.3.1.	SITAN Fundamentals .....	71
2.3.2.	Simulations and Discussion .....	83
2.4.	VATAN.....	92
2.4.1.	VATAN Fundamentals .....	92
2.4.2.	Simulations and Discussion .....	97
3.	IMPLEMENTATION OF TARGET TRACKING ALGORITHMS TO TERRAIN AIDED NAVIGATION.....	101
3.1.	Target Tracking Background .....	101
3.1.1.	Target State Estimation.....	103
3.1.2.	Data Association .....	108
3.2.	Probabilistic Data Association Filter (PDAF) .....	113
3.2.1.	Theory .....	113
3.2.2.	Implementation of PDAF to TAN .....	118
3.2.2.1.	Implementation Methods of PDAF for TAN .....	118
3.2.2.2.	PDAF Equations Implemented for TAN.....	125
3.2.2.2.1.	Past Measurement Information .....	125
3.2.2.2.2.	Measurement Validation .....	130
3.2.2.2.3.	State & Covariance Estimation, Update and Prediction ....	131
3.2.2.2.4.	The Probabilistic Data Association.....	138
3.2.2.2.5.	Summary of PDAF Equations for TAN.....	139
3.2.2.3.	Discussion of Real-time PDAF Implementation for TAN....	141
3.3.	Multiple Hypothesis Tracking (MHT) and Track Splitting Filter (TSF).....	143
3.3.1.	Theory .....	143
3.3.2.	Implementation of TSF to TAN .....	149
3.3.3.	TSF Equations Implemented for TAN.....	152
3.3.4.	Discussion of TSF Implementation for TAN.....	158
3.4.	Simulations.....	159

3.4.1.	Simulation Model Development .....	159
3.4.1.1.	Trajectory and INS Model .....	161
3.4.1.2.	DTED Database Model.....	168
3.4.1.3.	TERCOM Model.....	172
3.4.1.4.	SITAN Model .....	174
3.4.1.5.	PDAF Model .....	177
3.4.1.6.	TSF Model .....	178
3.4.2.	Case Studies .....	179
3.4.2.1.	Simulations With DTED Level 1.....	180
3.4.2.1.1.	Terrain Selection .....	180
3.4.2.1.2.	INS Model Verification.....	183
3.4.2.1.3.	Simulation Results .....	185
3.4.2.2.	Simulations With DTED Level 2.....	199
3.4.2.2.1.	Terrain Properties.....	199
3.4.2.2.2.	Simulation Results .....	200
3.4.2.3.	Simulations with Various DTED Grid Sizes .....	206
3.4.2.4.	Discussion .....	209
4.	CASE STUDY .....	211
4.1.	Simulation Tool Development .....	211
4.1.1.	Airframe .....	214
4.1.1.1.	Reference Frames.....	215
4.1.1.2.	6 DOF Equations of Motion.....	219
4.1.1.3.	Kinematic Equations .....	223
4.1.1.4.	Aerodynamics and Propulsion .....	223
4.1.1.5.	Environmental Models .....	228
4.1.2.	Autopilots and Controls .....	230

4.1.2.1.	Mach Hold Control .....	231
4.1.2.2.	Roll Position Control .....	232
4.1.2.3.	Heading Angle Control .....	234
4.1.2.4.	Yaw Stability Augmentation.....	235
4.1.2.5.	Altitude Hold Control .....	236
4.1.2.6.	Acceleration Autopilot.....	237
4.1.3.	Actuators .....	239
4.1.4.	Sensors .....	241
4.2.	Simulations.....	243
5.	DISCUSSION AND CONCLUSION.....	260
	REFERENCES.....	273
	APPENDIX	
	PROBABILISTIC DATA ASSOCIATION EQUATIONS [52].....	286
	CURRICULUM VITAE .....	296

## LIST OF TABLES

Table 1. BGM-109 Tomahawk Cruise Missile Specifications [5].....	6
Table 2. AGM-86 Air-Launched Cruise Missile (ALCM) Specifications [6].....	7
Table 3. Properties of DTED Prepared by HGK.....	21
Table 4. Navigation System Performances of Various INS Aiding Systems [17]... ..	22
Table 5. INS Sensor Error Sources [57].....	46
Table 6. Chronological Overview of TERCOM Development [37].....	53
Table 7. TERCOM Simulation Results for Rough Terrain.....	68
Table 8. TERCOM Simulation Results for Smooth Terrain.....	68
Table 9. TERCOM Simulation Results for Terrain with Uniqueness .....	69
Table 10. Kalman Filter Equations for SITAN Process.....	80
Table 11. Terrain Parameters for SITAN Simulations.....	84
Table 12. SITAN Simulation Parameters for Tracking Mode .....	84
Table 13. Parallel KF Structure for SITAN Acquisition Mode .....	88
Table 14. SITAN Simulation Parameters for Acquisition Mode .....	89
Table 15. PDAF Equations for TAN Process .....	140
Table 16. TSF Equations for TAN Process.....	157
Table 17. Batch of Height Differences Formation for 5x5 DTED Grid .....	170
Table 18. Terrain Parameters for TAN Simulations .....	182
Table 19. INS Model Parameters for 1.0 nm/hr Quality .....	183
Table 20. Simulation Parameters for Tracking Mode.....	186
Table 21. Simulation Parameters for Acquisition Mode.....	194

Table 22. Simulation Parameters for DTED Level 2 Tracking Mode .....	200
Table 23. Simulation Parameters for DTED Level 2 Acquisition Mode .....	204
Table 24. Simulation Parameters for Various DTED Grid Sizes .....	206
Table 25. Cruise Missile Model Specifications .....	227
Table 26. 6 DOF Cruise Missile Model Simulation Parameters.....	246
Table 27. Tracking Mode Position Errors.....	259

## LIST OF FIGURES

Figure 1. BGM-109 Tomahawk Cruise Missile [5] .....	4
Figure 2. AGM-86 Air-Launched Cruise Missile (ALCM) [6] .....	5
Figure 3. Main Components of Tomahawk Cruise Missile .....	8
Figure 4. Tomahawk Cruise Missile Configuration.....	8
Figure 5. TERCOM and DSMAC Guidance .....	12
Figure 6. Terrain Aided Navigation (TAN) Techniques [11].....	17
Figure 7. TAN Measurements.....	19
Figure 8. NNF Implementation [51] .....	36
Figure 9. Horizontal Position Errors of the INS Error Model.....	48
Figure 10. Horizontal Velocity Errors of the INS Error Model.....	48
Figure 11. Attitude Errors of the INS Error Model.....	49
Figure 12. TERCOM Maps in Use [37].....	54
Figure 13. TERCOM Concept [37].....	55
Figure 14. TERCOM Measurements [37].....	56
Figure 15. Terrain Correlation Processing [37] .....	60
Figure 16. Terrain Standard Deviation (Sigma-T) [37] .....	61
Figure 17. Definition of Sigma-Z [37] .....	62
Figure 18. Sample Area Selection from OziExplorer Software [61] .....	65
Figure 19. Rendered Rough Surface Area .....	66
Figure 20. Rendered Smooth Surface Area and Area with Uniqueness .....	67
Figure 21. SITAN Process [12].....	72

Figure 22. SITAN Measurement Process.....	76
Figure 23. Terrain Stochastic Linearization (TSL) [12] .....	77
Figure 24. Standard EKF Divergence Problem [12].....	81
Figure 25. Parallel KF Configuration [12].....	81
Figure 26. Rough Terrain Northward Position Error vs. Time .....	85
Figure 27. Rough Terrain Eastward Position Error vs. Time .....	85
Figure 28. Smooth Terrain Northward Position Error vs. Time .....	86
Figure 29. Smooth Terrain Eastward Position Error vs. Time.....	86
Figure 30. Mountainous Terrain Northward Position Error vs. Time .....	87
Figure 31. Mountainous Terrain Eastward Position Error vs. Time .....	87
Figure 32. Northward Position Error vs. Time for Acquisition Mode.....	90
Figure 33. Eastward Position Error vs. Time for Acquisition Mode .....	90
Figure 34. Minimum AWRS KF Index vs. Time for Acquisition Mode.....	91
Figure 35. Contour Plot for VATAN Simulation Terrain Types [42] .....	98
Figure 36. Basic Tracking Functions [64] .....	102
Figure 37. PDAF Implementation [65] .....	116
Figure 38. PDAF Procedure [66] .....	117
Figure 39. Batch Algorithm for TAN Solution (Acquisition Mode) .....	121
Figure 40. Recursive Algorithm for TAN Solution (Tracking Mode).....	122
Figure 41. PDAF Implementation for TAN Solution .....	125
Figure 42. Height Differences Batch Used in PDAF .....	128
Figure 43. Horizontal Position Error Definitions.....	136
Figure 44. MHT Logic Overview [69].....	145
Figure 45. High-level Flow Chart of MHT Algorithm [69] .....	145

Figure 46. MHTF Implementation [65] .....	147
Figure 47. MHTF Implementation for TAN for a Single Time Step.....	151
Figure 48. TSF Track Formation and Pruning.....	155
Figure 49. General Simulink Model for TAN Models.....	160
Figure 50. DTED Height Measurement Difference.....	168
Figure 51. Simulink DTED Database Model.....	171
Figure 52. Simulink TERCOM Model .....	173
Figure 53. Simulink Single EKF SITAN Model.....	175
Figure 54. Simulink Bank of EKF SITAN Model for 5x5 Grids.....	176
Figure 55. Simulink Bank of EKF SITAN Model for 3x3 Grids.....	177
Figure 56. Simulink PDAF Model.....	178
Figure 57. Simulink TSF Model .....	179
Figure 58. Terrain Contours for TAN Simulations .....	181
Figure 59. Terrain Height vs. Time for Rough Terrain.....	181
Figure 60. Terrain Height vs. Time for Smooth Terrain.....	182
Figure 61. Terrain Height vs. Time for Mountainous Terrain .....	182
Figure 62. Horizontal Position Errors of the INS Model Used.....	184
Figure 63. Horizontal Velocity Errors of the INS Model Used.....	184
Figure 64. Northward Position Error vs. Time .....	186
Figure 65. Eastward Position Error vs. Time.....	187
Figure 66. Northward Position RMS Error vs. Time .....	187
Figure 67. Eastward Position RMS Error vs. Time .....	188
Figure 68. Total Position RMS Error vs. Time.....	188
Figure 69. Northward Position Error vs. Time .....	189

Figure 70. Eastward Position Error vs. Time .....	189
Figure 71. Total Position RMS Error vs. Time.....	190
Figure 72. Northward Position Error vs. Time .....	190
Figure 73. Eastward Position Error vs. Time.....	191
Figure 74. Northward Position RMS Error vs. Time.....	191
Figure 75. Eastward Position RMS Error vs. Time .....	192
Figure 76. Total Position RMS Error vs. Time.....	192
Figure 77. Northward Position Error vs. Time .....	194
Figure 78. Eastward Position Error vs. Time.....	195
Figure 79. TSF Indices vs. Time.....	195
Figure 80. Northward Position Error vs. Time .....	196
Figure 81. Eastward Position Error vs. Time.....	196
Figure 82. Northward Position Error vs. Time .....	197
Figure 83. Eastward Position Error vs. Time.....	197
Figure 84. TSF Indices vs. Time .....	198
Figure 85. DTED Level 2 Terrain for TAN Simulations.....	199
Figure 86. Terrain Height vs. Time for DTED Level 2 Terrain .....	200
Figure 87. Northward Position Error vs. Time .....	201
Figure 88. Eastward Position Error vs. Time.....	201
Figure 89. Northward Position RMS Error vs. Time .....	202
Figure 90. Eastward Position RMS Error vs. Time .....	202
Figure 91. Total Position RMS Error vs. Time.....	203
Figure 92. Northward Position Error vs. Time .....	204
Figure 93. Eastward Position Error vs. Time.....	204

Figure 94. TSF Indices vs. Time .....	205
Figure 95. PDAF Northward Position Error vs. Time .....	206
Figure 96. PDAF Eastward Position Error vs. Time.....	207
Figure 97. PDAF Total Position RMS Error vs. Time.....	207
Figure 98. TSF Northward Position Error vs. Time.....	208
Figure 99. TSF Eastward Position Error vs. Time .....	208
Figure 100. TSF Total Position RMS Error vs. Time .....	208
Figure 101. General 6 DOF Simulink Model with Implemented TAN Models ...	212
Figure 102. Inertial and ECEF Reference Frames .....	216
Figure 103. Geographic and Body Reference Frames .....	216
Figure 104. Cruise Missile Solid Model .....	227
Figure 105. Mach Hold Control Loop [79] .....	232
Figure 106. Roll Rate and Position Feedback Loops [79] .....	233
Figure 107. Heading Angle Tracker Loop [79] .....	234
Figure 108. Yaw Rate Feedback Loop [79] .....	235
Figure 109. Altitude Hold Autopilot [79] .....	237
Figure 110. Acceleration Autopilot Loop [79] .....	238
Figure 111. Second Order Actuator Model [79] .....	240
Figure 112. Simulation Path Used for 6 DOF Cruise Missile Simulation .....	244
Figure 113. Altitude vs. Time .....	247
Figure 114. Flight Profile over the Terrain .....	247
Figure 115. Latitude vs. Longitude over Terrain Contours .....	248
Figure 116. Roll Rate vs. Time .....	248
Figure 117. Pitch and Yaw Rates vs. Time .....	248

Figure 118. Attitudes vs. Time.....	249
Figure 119. Attitude Errors vs. Time .....	249
Figure 120. Total Velocity and Body Longitudinal Velocity vs. Time .....	250
Figure 121. Lateral and Vertical Body Velocities vs. Time .....	250
Figure 122. Angle of Attack and Side Slip Angle vs. Time .....	250
Figure 123. Wind Profile (Wind Velocity and Wind Heading) vs. Time .....	251
Figure 124. Mach vs. Time .....	251
Figure 125. Missile Mass vs. Time .....	251
Figure 126. Turbojet Thrust vs. Time .....	252
Figure 127. Missile Heading and Commanded Heading vs. Time .....	252
Figure 128. Body Accelerations vs. Time.....	252
Figure 129. Northward Position Errors vs. Time (Acquisition Mode) .....	255
Figure 130. Eastward Position Errors vs. Time (Acquisition Mode).....	255
Figure 131. Northward Position Errors vs. Time (Tracking Mode) .....	256
Figure 132. Eastward Position Errors vs. Time (Tracking Mode).....	256
Figure 133. Northward Velocity Errors vs. Time .....	257
Figure 134. Eastward Velocity Errors vs. Time.....	257

## LIST OF SYMBOLS

### **Symbols:**

$a$	:	Semi-major Axis
$a_N$	:	Normal Acceleration
$\alpha$	:	Angle of Attack
$\nabla$	:	Accelerometer Bias
$\beta_i$	:	Association Probability
$b$	:	Span
$b_F$	:	Specific Fuel Consumption (SFC)
$\beta$	:	Side-slip Angle
$C_x$	:	Axial Force Coefficient
$C_y$	:	Side Force Coefficient
$C_z$	:	Normal Force Coefficient
$C_d$	:	Drag Coefficient
$C_l$	:	Lift Coefficient
$C_r$	:	Rolling Moment Coefficient
$C_m$	:	Pitching Moment Coefficient
$C_n$	:	Yawing Moment Coefficient
$C_{est}$	:	Estimated Relative Height
$C_{meas}$	:	Measured Relative Height
$\hat{C}$	:	Transformation Matrix

$c$	:	Chord
$\delta$	:	Error Operator
$\delta a$	:	Aileron Command
$\Delta d$	:	Distance Between Profile Measurements (Cell Size)
$\delta e$	:	Elevator command
$\delta \bar{f}^{(b)}$	:	Accelerometer Errors
$\Delta g$	:	Gravity Error
$\delta h$	:	Height Position Error State
$D_i$	:	Differential Operator in Inertial Frame
$\delta i$	:	Actual Control Surface Deflection
$\delta i_c$	:	Fin Command
$\delta \Psi$	:	Attitude Errors
$\delta rN$	:	Northward Position Error
$\delta rE$	:	Eastward Position Error
$\delta r$	:	Rudder command
$\delta vN$	:	Northward Velocity Error
$\delta vE$	:	Eastward Velocity Error
$\delta \bar{x}$	:	Error State Vector
$\delta \hat{x}$	:	Estimated Error State Vector
$d_x$	:	DTED Spacing Along Longitude Direction
$d_y$	:	DTED Spacing Along Latitude Direction
$\varepsilon$	:	Gyro Drift
$E[*$	:	Expectation Operator
$e^2$	:	First Eccentricity Squared

$f$	: Specific Force
$\mathfrak{I}$	: Reference Frame
$\vec{F}$	: Force Vector
$F_r$	: Thrust Required
$g$	: Gravity
$\gamma$	: Gate Threshold
$G$	: Autopilot Gains
$h_{k,m}$	: The k'th Measured Terrain Elevation File
$H_{m,n}$	: Stored Reference Matrix Data
$\bar{H}$	: Mean Elevation
$h$	: Height Above Sea Level
$h_x$	: Terrain Slopes Along Eastward Direction
$h_y$	: Terrain Slopes Along Northward Direction
$H(k)$	: Measurement Matrix
$\vec{H}$	: Angular Momentum Vector
$\hat{I}$	: Moment of Inertia Matrix
$K$	: Number of Measured Terrain Elevation Files
$k$	: Discrete Time Step
$K(k)$	: Kalman Filter Gain
$K$	: Autopilot Gains
$\xi$	: Ambiguity Term for MAD and MSD Algorithms
$\lambda$	: Local Latitude
$\lambda(k)$	: Modified Log-likelihood Function
$\mu$	: Local Longitude

$M$	: Number of Reference Matrix Columns
$m_1$	: TSL Error at the Actual Point
$m_2$	: TSL Error at the Estimated Point
$m(k)$	: DTED Grid Size
$m$	: Mass
$\vec{M}$	: Moment Vector
$\dot{m}_F$	: Fuel Flow Rate
$\hat{M}$	: Misalignment Matrix
$N$	: Number of Samples in the Measured Terrain Elevation File
$n, m, k$	: Row, Column, and Terrain Elevation File Indices
$N^{[*]}$	: Normal Probability Density Function
$\Omega$	: Earth's Angular Velocity
$\omega$	: Angular Rate
$\omega_n$	: Natural Frequency
$\Phi$	: State Transition Matrix
$P(k)$	: Error State Covariance Matrix
$p$	: Probability Density Function
$\tilde{P}(k)$	: Spread of the Innovations
$p$	: Roll Rate
$\phi$	: Roll Angle
$\psi$	: Yaw Angle
$\psi_w$	: Wind Direction From North
$Q(k)$	: System Noise Covariance Matrix
$\bar{q}$	: Dynamic Pressure

$q$	:	Pitch Rate
$R$	:	Radius of Earth
$R_k$	:	The k'th Reference Profile
$R(k)$	:	Measurement Noise Covariance Matrix
$rN_{traj}$	:	Northward Position
$rE_{traj}$	:	Eastward Position
$R_N$	:	Earth's Polar Radius
$R_E$	:	Earth's Equatorial Radius
$\bar{r}$	:	Position Vector
$r$	:	Yaw Rate
$\rho$	:	Density of the Ambient Atmosphere
$\sigma_T$	:	Terrain Roughness
$\sigma_z$	:	Standard Deviation of the Point-to-point Changes in Terrain
$\sigma$	:	Standard Deviation
$\sigma^2$	:	Variance
$S(k)$	:	Innovation Covariance Matrix
$S$	:	Reference Area
$\hat{S}$	:	Scale Factor Error Matrix
$T$	:	Sampling period
$t$	:	Time
$\theta$	:	Pitch angle
$v_x$	:	Velocity Along $x$ Direction
$v_y$	:	Velocity Along $y$ Direction
$v(k)$	:	Innovation

$vN$	:	Northward velocity
$vE$	:	Eastward Velocity
$vD$	:	Down Velocity
$V_w$	:	Wind Magnitude
$V_s$	:	Sonic Speed
$v$	:	Measurement White Noise
$w$	:	Process White Noise
$X_T$	:	Terrain Correlation Length
$\bar{x}$	:	State Vector
$\tilde{\bar{x}}$	:	Measured State Vector
$\hat{\bar{x}}$	:	Estimated State Vector
$z_k$	:	Kalman Filter Measurement
$\zeta$	:	Damping

### Superscripts:

( . )	:	Time derivative
( ~ )	:	Skew symmetric matrix
( ^ )	:	Matrix
( → )	:	Vector
( - )	:	Column matrix
( T )	:	Transpose
( -I )	:	Inverse
( i )	:	Inertial frame

$(e)$	:	ECEF frame
$(g)$	:	Geographic frame
$(w)$	:	Wind frame
$(b)$	:	Body frame

### **Subscripts:**

$x$	:	East direction
$y$	:	North direction
$(0)$	:	Initial state
$(N)$	:	North
$(E)$	:	East
$(D)$	:	Down
$(i)$	:	Inertial frame
$(e)$	:	ECEF frame
$(g)$	:	Geographic frame
$(w)$	:	Wind frame
$(b)$	:	Body frame
$(a)$	:	Aerodynamic
$(p)$	:	Propulsion
$(c)$	:	Commanded

## **LIST OF ABBREVIATIONS**

6 DOF	: Six Degree of Freedom
ACM	: Advanced Cruise Missile
AGL	: Above Ground Level
AIAA	: The American Institute of Aeronautics and Astronautics
ALCM	: Air-Launched Cruise Missile
ASCM	: Anti Ship Cruise Missile
AWRS	: Average Weighted Residual Squared
BTT	: Bank-to-turn
BW	: Biological Warhead
CALCM	: Conventional Air Launched Cruise Missile
CCD	: Charged Coupled Device
CEP	: Circular Error Probable
CW	: Chemical Warhead
DCM	: Direction Cosine Matrix
DMA	: Defense Mapping Agency
DSMAC	: Digital Scene Matching Area Correlator
DTED	: Digital Terrain Elevation Data
ED50	: European 1950 Datum
EKF	: Extended Kalman Filter
ESA	: Electronically Steerable Antenna
GPB	: Generalized Pseudo Bayesian
GPS	: Global Positioning System

HGK	:	Turkish General Mapping Commandership
ICBM	:	Intercontinental Ballistic Missile
IEEE	:	The Institute of Electrical and Electronics Engineers
IMM	:	Interacting Multiple Model
IMU	:	Inertial Measurement Unit
INS	:	Inertial Navigation System
IR	:	Infrared
ISO	:	International Organization for Standardization
JPDA	:	Joint Probabilistic Data Association
JPDAF	:	Joint Probabilistic Data Association Filter
KF	:	Kalman Filter
LCCMD	:	Low Cost Cruise Missile Defense
LSE	:	Least Squares Estimation
MAD	:	Mean Absolute Difference
MAP	:	Maximum A Posteriori Estimator
MEMS	:	Micro Electromechanical Machine System
MHT	:	Multiple Hypothesis Tracking
MLE	:	Maximum Likelihood Estimator
MMAE	:	Multiple Model Adaptive Estimation
MSD	:	Mean Squared Difference
MSL	:	Mean Sea Level
NNF	:	Nearest Neighbor Filter
PDA	:	Probabilistic Data Association
PDAF	:	Probabilistic Data Association Filter
PTAN	:	Probability Based Terrain Aided Navigation

RCS	:	Radar Cross Section
RMS	:	Root Mean Squared
RPV	:	Remotely Piloted Vehicle
SAR	:	Synthetic Aperture Radar
SFC	:	Specific Fuel Consumption
SITAN	:	Sandia Inertial Terrain Aided Navigation
SNF	:	Strongest Neighbor Filter
TAN	:	Terrain Aided Navigation
TERCOM	:	Terrain Contour Matching
TF	:	Transfer Function
TS	:	Track Splitting
TSF	:	Track Splitting Filter
TSL	:	Terrain Stochastic Linearization
UAV	:	Unmanned Air Vehicle
UHF	:	Ultra High Frequency
USAF	:	United States Air Force
VA	:	Viterbi Algorithm
VATAN	:	Viterbi Algorithm Terrain Aided Navigation
VSIMM	:	Variable Structure Interacting Multiple Model
WGS84	:	World Geodetic System 1984
WP	:	Waypoint

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1. Scope of the Study**

A cruise missile is a guided missile that uses a lifting wing and a jet propulsion system to allow sustained flight. Cruise missiles are, in essence, unmanned aircraft and they are generally designed to carry a large conventional or nuclear warhead many hundreds of miles with excellent accuracy [1].

A cruise missile usually flies at subsonic speed and it would require several hours of continuously guided flight to cover its mission distance. Hence, guidance errors that accumulate with time would be almost 100 times larger for a cruise missile than for a ballistic missile which is guided for the first five of the twenty minutes. Therefore its accurate arrival on the target could be achieved only with continuous guidance that is updated and corrected from time to time by new location information. In order to obtain the necessary location information, a long-range cruise missile employs a device that can correlate information obtained by an onboard sensor about the terrain it is flying over with some kind of map stored in the memory of an onboard computer [2].

Navigation system of a cruise missile determines the position, velocity, attitude and time solutions of the missile. Therefore, it can be concluded that an accurate self-contained navigation system directly influences the success of the missile. In this study, navigation and control studies on cruise missiles will be performed.

Terrain Aided (Referenced) Navigation (TAN) is an important part of “Integrated Navigation Systems” in military and civil avionics. TAN provides position fixes, which can be used to aid a central navigation system. Especially, if other sources for position aids, like the Global Positioning System (GPS), are not available, TAN can provide reliable position information in low level flights over significant terrain [3].

The scope of the study is to implement some modern radar data association algorithms as new Terrain Aided Navigation (TAN) algorithms which can be used with low-cost Inertial Measurement Units (IMU’s).

In this chapter, theory about the study will be given. First, information about cruise missiles and cruise missile navigation performance will be given. Then, literature survey on TAN techniques will be discussed in detail. Finally, information about radar tracking techniques and possible implementations of radar data association algorithms to TAN will be given. At the last section of the chapter, outline of the thesis study will be summarized.

## **1.2. Cruise Missiles**

### **1.2.1. Background**

A cruise missile is a guided missile that uses a lifting wing and a jet propulsion system to allow sustained flight. Cruise missiles are, in essence, unmanned aircraft. They are generally designed to carry a large conventional or nuclear warhead many hundreds of miles with excellent accuracy. In 2001, modern cruise missiles normally travel at sub-sonic speeds, are self-navigating, and fly low in order to avoid radar detection [1]. The term cruise missile covers several vehicles and their capabilities, from the Chinese Silkworm (HY-2), which has a range of less than 105 km, to the U.S. Advanced Cruise Missile (ACM), which can fly to ranges of up to 3,000 km. These vehicles vary greatly in their speed and ability to penetrate defenses. All, however, meet the definition of a cruise missile: “an unmanned self-propelled guided vehicle that sustains flight through aerodynamic lift for most of its flight path and whose primary mission is to place an ordnance or special payload on a target”. This definition can include unmanned air vehicles (UAV’s) and unmanned control-guided helicopters or aircraft [4].

Cruise missiles were first developed by Nazi Germany during World War II. The V-1 (introduced in 1944) was the first weapon to use the classic cruise missile layout of a bomb-like fuselage with short wings and a dorsally mounted engine, along with a simple inertial guidance system. The V-1 was propelled by a crude pulse-jet engine, the sound of which gave the V-1 its nickname of “buzz bomb”. Japanese kamikaze aircraft could be viewed as manned cruise missiles. During the Cold War, both the United States and the Soviet Union experimented further with the concept, deploying early cruise missiles from submarines and aircraft. The Soviet Union was especially fond of large cruise missiles. The United States had a program to develop a nuclear-powered cruise missile, Project Pluto. Although the concept was proven sound, none were ever test-launched. While ballistic missiles were the weapons of choice for land targets, heavy nuclear and conventional tipped

cruise missiles were seen by the USSR as a primary weapon to destroy US carrier battle groups. Large submarines (e.g. Echo and Oscar class) were developed to carry these weapons and shadow US battle groups at sea, and large bombers (e.g. Backfire, Bear, and Blackjack models) were equipped with the weapons [1].

As of 2001, the Tomahawk missile (BGM-109) model has become a significant part of the US naval arsenal. It gives ships and submarines an extremely accurate, long-range, conventional land attack weapon. Each costs about \$1,000,000 USD. The United States Air Force deploys an air launched cruise missile, the AGM-86. It can be launched from bombers like the B-52 Stratofortress. Both the Tomahawk and the AGM-86 were used extensively during Operation Desert Storm [1].

In Figure 1, Figure 2, Table 1 and Table 2, well-known US cruise missiles with specifications were presented as examples to typical cruise missiles.



Figure 1. BGM-109 Tomahawk Cruise Missile [5]



Figure 2. AGM-86 Air-Launched Cruise Missile (ALCM) [6]

### 1.2.2. Cruise Missile Technology

Cruise missile technology has advanced substantially since the German V-1 of World War II. Modern cruise missiles fly at altitudes one-tenth those of the V-1, have Radar Cross-Sections (RCS) one hundred times smaller (which reduces detectability), and accuracies two hundred times better [7].

The technology of the cruise missile has four main component elements:

1. Airframe;
2. Propulsion system;

3. Guidance systems;

4. Warhead. [7]

As an example, the main components of Tomahawk cruise missile are shown in Figure 3 [5].

Table 1. BGM-109 Tomahawk Cruise Missile Specifications [5]

Primary Function:	Long-range subsonic cruise missile for attacking land targets.
Contractor:	Hughes Missile Systems Co., Tucson, Ariz.
Power Plant:	Williams International F107-WR-402 cruise turbo-fan engine; solid-fuel booster
Length:	18 feet 3 inches (5.56 meters); with booster: 20 feet 6 inches (6.25 meters)
Weight:	2,650 pounds (1192.5 kg); 3,200 pounds (1440 kg) with booster
Diameter:	20.4 inches (51.81 cm)
Wing Span:	8 feet 9 inches (2.67 meters)
Range:	Land attack, conventional warhead: 600 nautical miles (690 statute miles, 1104 km)
Speed:	Subsonic – about 550 mph (880 km/h)
Guidance System:	Inertial and TERCOM
Warheads:	Conventional: 1,000 pounds Bull pup, or conventional sub-munitions dispenser with combined effect bomblets, or WDU-36 warhead with PBXN-107 explosive & FMU-148 fuze, or <i>200 kt. W-80 nuclear device</i>
Date Deployed:	1983
Costs	\$500,000 - current production Unit Cost \$1,400,000 - average unit cost (TY\$) \$11,210,000,000 - total program cost (TY\$)
Total Program	4 170 missiles

Table 2. AGM-86 Air-Launched Cruise Missile (ALCM) Specifications [6]

Primary Function:	Air-to-surface strategic missile
Contractor:	Boeing Aerospace Co.
Guidance Contractors:	Litton Guidance and Control
Power Plant:	Williams Research Corp. F-107-WR-10 turbofan engine
Thrust:	600 pounds (270 kilograms)
Length:	20 feet, 9 inches (6.29 meters)
Weight:	3,150 pounds (1,417.5 kilograms)
Diameter:	24.5 inches (62.23 centimeter)
Wingspan:	12 feet (3.64 meters)
Range:	AGM-86B: 1,500-plus miles (1,305 nautical miles)
Speed:	About 550 mph (Mach 0.73)
Guidance System:	Litton inertial navigation element with terrain contour-matching updates
Warheads:	Nuclear capable
Sensors:	A terrain contour-matching guidance system that allows the missile to fly complicated routes to a target through use of maps of the planned flight route stored in on-board computers
Unit Cost:	\$1 million
Date Deployed:	December 1982
Inventory:	Active force, 1,628; ANG, 0; Reserve, 0

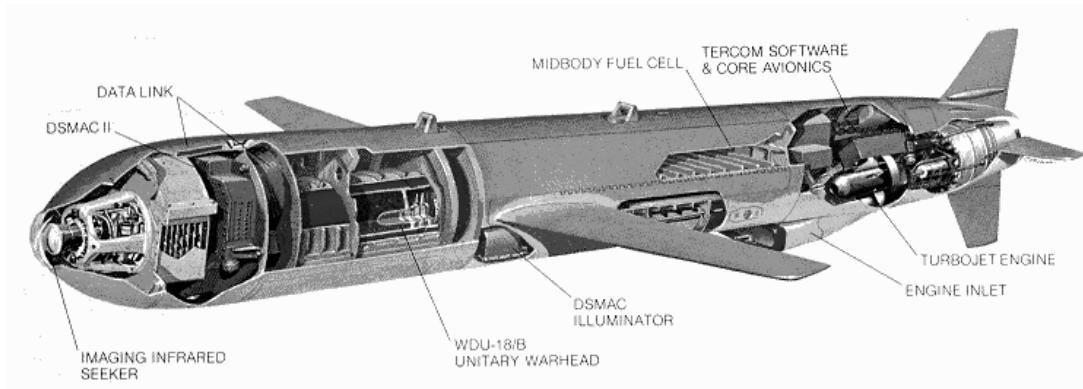


Figure 3. Main Components of Tomahawk Cruise Missile

#### 1.2.2.1. Airframe

The airframe is essentially that of a small (unmanned) aeroplane or a design based on a Remotely Piloted Vehicle (RPV). Early two-wing, three-surface tail aircraft designs were followed by four-wing, four-tail cruciform configurations. As an example, the body of a Tomahawk cruise which has two wings and a four-fan tail [7] is given in Figure 4 [5].

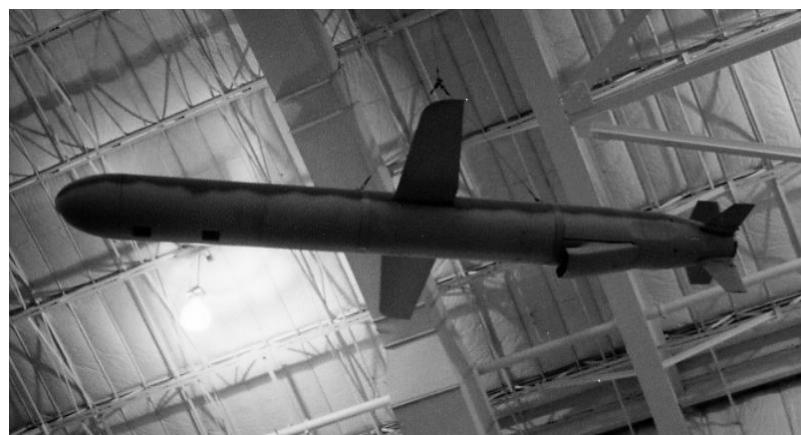


Figure 4. Tomahawk Cruise Missile Configuration

### **1.2.2.2. Propulsion System**

The propulsion system needs to maintain sufficient momentum to counter the force of gravity. Most cruise missiles are propelled by a small, highly specialized, air-breathing engine which thus needs to draw oxygen from the atmosphere into the engine for the bulk of the flight. Air-breathing engines are one of four types: pulsejet, ramjet, turbojet or turbofan, this last being a more efficient form of turbojet developed in the 1970's. Short to medium range systems tend to employ turbojets, which though less efficient, are usually less expensive than turbofans. Most long-range missiles, e.g. the Tomahawk use highly efficient turbofans propelling them at high subsonic speeds. The few long-range cruise missiles propelled by ramjets include the French ASMP and ASURA which are capable of Mach 2 and Mach 3 speeds respectively [7].

Many missiles are launched by rocket boosters and some missiles, especially short-range Anti Ship Cruise Missiles (ASCM's) like the Exocet, are powered throughout their flight by rocket motors. Older rocket-propelled models, such as the Styx and Silkworm ASCM's use liquid fuelled rocket engines, while newer ones, such as the Exocet, use solid fuel motors [7].

### **1.2.2.3. Guidance Systems**

Cruise missiles have at least two guidance systems: an in-flight guidance system to maintain its flight path and altitude, and a terminal guidance system for the final approach to the target. Depending upon the particular characteristics of the guidance system, the missile may be programmed: as autonomous (i.e. launch and leave); or for remote piloting by command (i.e. flow by a human operator over a remote communications link); or as semi-autonomous (a combination of the two, with remote manual input in the terminal stage) [7].

#### ***1.2.2.3.1. In-flight Guidance***

In-flight guidance relies on Inertial Navigation Systems (INS) using gyroscopes to ascertain the missile's position. Shorter-range cruise missiles may use only inertial and terminal guidance. Longer-range missiles require supplemental information to make up for inherent inertial guidance inaccuracies (or drift). One sophisticated supplemental system in current use is Terrain Contour Matching, known in the United States as TERCOM, a position fixing technique. A digital terrain map of the missile's planned route has first to be made, and it is then stored in the weapon's guidance system. Updates received from a radar altimeter determine the missile's altitude and this information is then compared with terrain heights in the pre-stored digital map. Once the updates are received, the missile can correct its flight back to the planned route [7].

Due to the high cost and complexities of obtaining the satellite data needed to create the digitized maps for TERCOM, the US is the only nation that currently incorporates this technology widely in its cruise missiles. However, the Soviet SS-N-21 Sampson, a long-range Submarine-Launched Cruise Missile (SLCM) dubbed "Tomahawksi" due to its similarity to the US system, is believed to be able to incorporate a TERCOM-like guidance system. The French ASMP and Apache missiles also use terrain matching and several other Western nations, including the UK and Sweden, are believed to have the capability to incorporate TERCOM into cruise missiles [7].

Global Positioning System (GPS) data is another supplementary guidance system, which has yet to be fully exploited. It uses a constellation of 24 continuously transmitting navigation satellites provided by the US Department of Defense. Military users can receive positional data accurate down to 5 meters or less. However, the same signal 'degraded' for civilian users (and potential adversaries) is less precise, but accurate enough for most purposes. A similar system, GLONASS, is being deployed by Russia [7].

GPS is now used in US cruise missiles, such as the Tomahawks used to attack Bosnian Serb targets in September 1995. France intends to use GPS for its Apache series of cruise missiles, and other countries are also expected to do so. Thus, with the ever-widening availability of this technology, all proliferators can significantly enhance the accuracy of their cruise missiles. For them GPS will be preferable to TERCOM because it does not require such elaborate and cost-intensive pre-programming of data [7].

However both systems have their limitations. TERCOM navigates by identifying distinctive terrain features. On the other hand, the GPS system utilizes long-range satellite systems whose transmissions can be jammed using shorter range, more powerful signals, and the civilian signals can also be switched off if necessary, as was the case during Operation Desert Storm in 1991 [7].

#### ***1.2.2.3.2. Terminal Guidance***

Terminal guidance systems help the missile to home in on the target in the final stages of flight. These systems may make use of active or semi-active radar, infrared, television, or “home-on-jam” (i.e. on a jamming signal) techniques [7].

The Tomahawk uses an additional set of precise terminal navigation updates known as the Digital Scene Matching Area Correlator (DSMAC), a two-dimensional, map-matching concept that employs an onboard sensor to obtain a sequence of images of the ground directly below the missile. The images are compared to reference data stored in the missile’s navigational computer, and route changes are made accordingly, prior to final target acquisition [7].

In Figure 5, TERCOM and DSMAC guidance principles are shown [5].

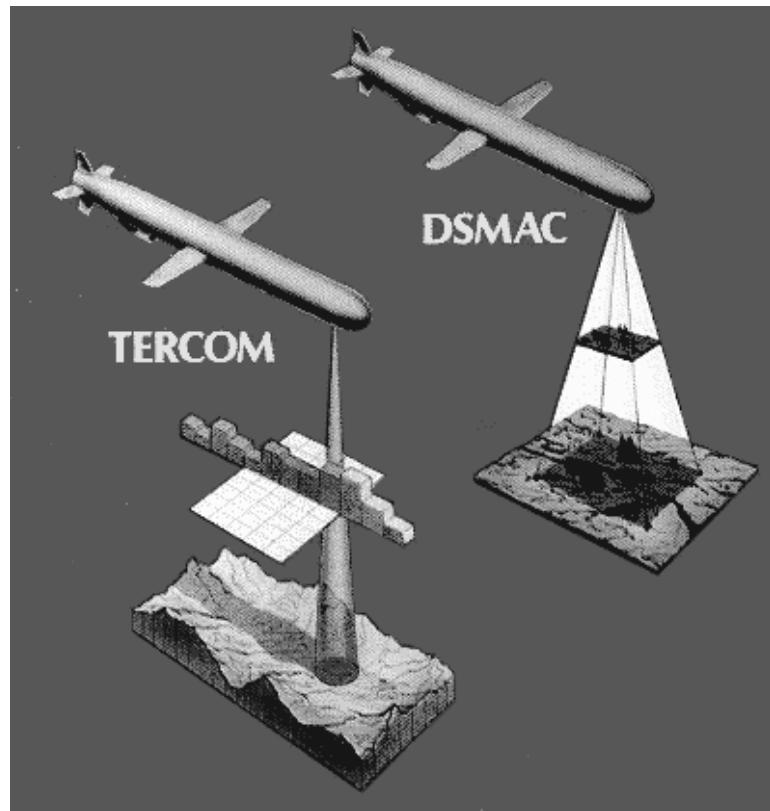


Figure 5. TERCOM and DSMAC Guidance

#### 1.2.2.4. Warhead

In some respects the most significant component of a cruise missile is its warhead. Unlike a ballistic missile, which places enormous stresses on its warhead as it accelerates and as it re-enters the Earth's atmosphere, a cruise missile flies much like an aircraft. Its warhead can therefore be based upon munitions originally designed for manned aircraft, making the development of Chemical and Biological (CW and BW) payloads for cruise missiles a comparatively simple matter [7].

Currently, most cruise missiles are armed with conventional, high-explosive warheads. However several countries are known to have available blast fragmentation warheads for use with their cruise missiles (e.g. ASCM's such as the

Chinese Silkworm, the Iraqi FAW series, and the Israeli Gabriel) or sub-munitions (e.g. the French Exocet ASCM and Apache TLACM and the German Kormoran ASCM) [7].

Some ex-Soviet systems are dual-capable and can be fitted with either a conventional or nuclear warhead. Only the US, Russia and France are known to deploy nuclear-armed cruise missiles at present. The Chinese were also reported some time ago to be nearing completion of a nuclear warhead for their Silkworm ASCM. No nation is currently known to possess a CW or BW cruise missile warhead. However there have been media reports suggesting that Syria, Iran, and China are attempting to develop these [7].

Different variants of the Tomahawk may be nuclear or conventionally armed. Due partly to arms control constraints and partly to improvements in conventional payloads, the US is concentrating on the development and deployment of conventionally armed missiles. The Tomahawk TLAM-D for example carries a sub-munitions dispenser that allows it to deliver bomblets on three different targets, before diving into a fourth [7].

### **1.2.3. Low-Cost Cruise Missiles**

Advances in new commercial technologies make the development of low cost guided weapons possible. US authorities developed the Low Cost Cruise Missile Defense (LCCMD) program in order to defeat a threat consisting of unsophisticated air vehicles attempting to overwhelm their defensives by attacking in large numbers or by attacking over wide geographic areas [8].

It is claimed that 82 countries (including third world countries like Egypt, Chile, and Singapore) possess cruise missiles where 75 systems are in service and 42 are in development [9].

Advanced low cost interceptor seekers, using commercial hardware, and matching seeker performance as Noise Radar Seeker, the Micro Electromechanical Machine System (MEMS) Electronically Steerable Antenna (ESA) Seeker, Laser Seeker, Infrared (IR) Seeker, Optical ESA, Ultra High Frequency (UHF) Seeker and advanced navigation algorithms can be examples for low cost systems which can be used in cruise missiles in the future.

As an example, in reference [10], an amateur researcher claims to build a cruise missile in his own garage with a budget of just 5,000 US dollars. He also subscribes all the work he does in his site.

### **1.3. Terrain Aided Navigation (TAN)**

Terrain Aided Navigation (TAN) is a technique to estimate the position of a moving vehicle by comparing the measured terrain profile under the vehicle to a stored elevation map. TAN has been operational for unmanned vehicles for some time. Although this operational system has proven to be reliable and cost effective, it is desirable to develop enhancements which can either reduce the pre-planning effort or increase the operational envelope, i.e., reliable operation in terrain with less reliable or with stored elevation data with larger errors. It is anticipated that terrain aided navigation will be in use for many years to come due to the long term stability of the terrain profile of earth, the relative ease of mapping and maintaining maps of large operational areas, the ease and reliability with which on-board measurements can be made and the relatively low computational burden of computing navigation updates in an embedded vehicle processor [11].

TAN is an important part of “Integrated Navigation Systems” in military and civil avionics. TAN provides position fixes, which can be used to aid a central navigation system. Especially, if other sources for position aids, like the Global Positioning System (GPS), are not available, TAN can provide reliable position information in low level flights over significant terrain. The outage of the GPS in hostile jammed environments or due to shadowing effects caused by low level flights in valleys is always possible and has to be expected. Therefore, TAN, which is independent from external information sources, is predestinated for additional position aiding [3].

### 1.3.1. TAN Techniques

A number of TAN techniques have been developed and tested. These fall into two general algorithmic categories [11]:

1. Batch Algorithms,
2. Recursive Algorithms.

In addition, there are two general map storage techniques: small, high fidelity maps which are used at specific points along the intended route of the vehicle; and a single, large, low fidelity map which encompasses the entire operating area of the vehicle. These techniques are shown in Figure 6 and are associated with the two most widely understood TAN implementations [11]:

1. Terrain Contour Matching (TERCOM)
2. Sandia Inertial Terrain Aided Navigation (SITAN) [12]

The most widely known form of TAN is TERCOM. With TERCOM a strip of terrain elevation measurements are collected while the vehicle flies along the intended route and the measurements are post processed by a batch algorithm to provide a correlation with a high fidelity map. In the operational missile systems employing TERCOM the stored map preparation and validation process includes extensive analysis to evaluate the probability of obtaining a strong and unambiguous correlation with candidate maps. The map size in the cross-track direction is determined by the “worst case” navigation uncertainty and in the down-track direction by the larger of “worst case” navigation uncertainty or the map length necessary to provide an unambiguous update opportunity. A sequence of maps are then developed to provide navigation update opportunities from the launch point to the target. The operational TERCOM applications use a mean absolute difference (MAD) algorithm which is only a modest computational requirement in an embedded flight processor. In addition, the map storage requirements are minimized by carefully selecting the minimum number and size of maps required for each mission [11].

In the late 1970’s TAN in the form of SITAN was proposed. SITAN uses an extended Kalman Filter (EKF) and a local terrain linearization technique to implement a recursive algorithm. This algorithm operates on individual terrain elevation measurements as they become available and for the entire duration of the mission. This requires a map for the entire mission. For missile applications the map could be for the length of the mission with the width determined by navigation uncertainty and terrain uniqueness or suitability. However, for manned aircraft applications map data for the entire operating area must be stored because the pilot may deviate from the preplanned route at any time. SITAN has been developed and evaluated for the manned aircraft application using Digital Terrain Elevation Data (DTED) which is a low fidelity Defense Mapping Agency (DMA) product readily available in most operational areas [11].

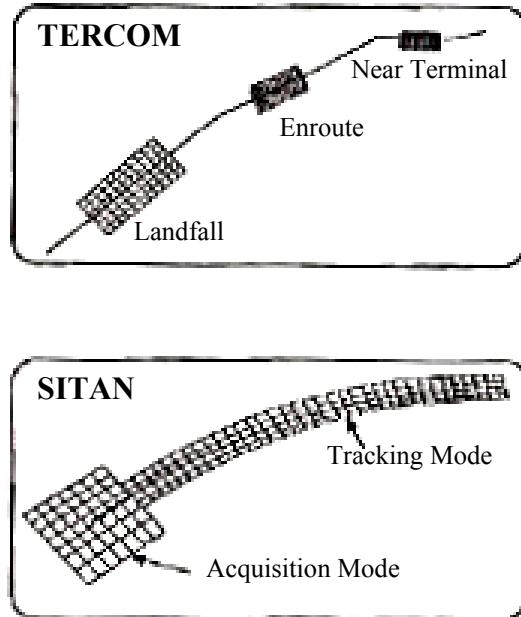


Figure 6. Terrain Aided Navigation (TAN) Techniques [11]

Other TAN techniques including TERPROM and SPARTAN have been developed and evaluated since then. In the late 1950's and throughout the 1960's when TAN concepts were originally developed and in the 1970's when TAN concepts were applied to missile applications, digital computer capabilities were limited. Within the past few years the computational, data storage and memory access capabilities of embedded vehicle computers have improved dramatically. Thus, the previously assumed computational constraints do not apply as techniques are developed to enhance the performance and to expand the operational envelope of TAN techniques [11].

The TERCOM and SITAN approaches both have attributes that are of interest. Although enhancements can be envisioned in a number of areas the approach here is to investigate algorithm techniques which would make more complete use of the information content of the stored elevation data, the a priori knowledge of the errors in the stored elevation data and the elevation

measurements. Batch processing algorithm (i.e. TERCOM) was generally selected for cruise missiles because: no linearization of the terrain profile is necessary; it does not require an acquisition process; and the algorithm techniques are applicable to both small discrete maps and large maps which can support continuous navigation updating [11].

### **1.3.2. TAN System Considerations**

In terrain aided navigation, position estimates are referenced to the terrain data and are insensitive to position bias errors in the terrain data. Because of this characteristics, terrain aided navigation systems are especially useful in applications that require accurate navigation relative to targets, obstacles, structures, and other features whose locations are derived from the same source as the stored elevation data [13].

Terrain aided navigation (TAN) consists of sensing a terrain elevation profile beneath an air vehicle and correlating the profile with stored digital terrain elevation data (DTED) to produce an estimate of vehicle position. An INS, usually with barometric altimeter aiding, provides the approximate trajectory. TAN systems provide three dimensional position updates to the navigation system by estimating INS trajectory errors. Radar or laser altimeter measures ground clearance and the DTED gives terrain elevation above mean sea level (MSL). Implementation requires an INS, an altimeter, DTED, and a flight computer for executing the TAN algorithm. In Figure 7, an illustration is given for TAN measurement process [13].

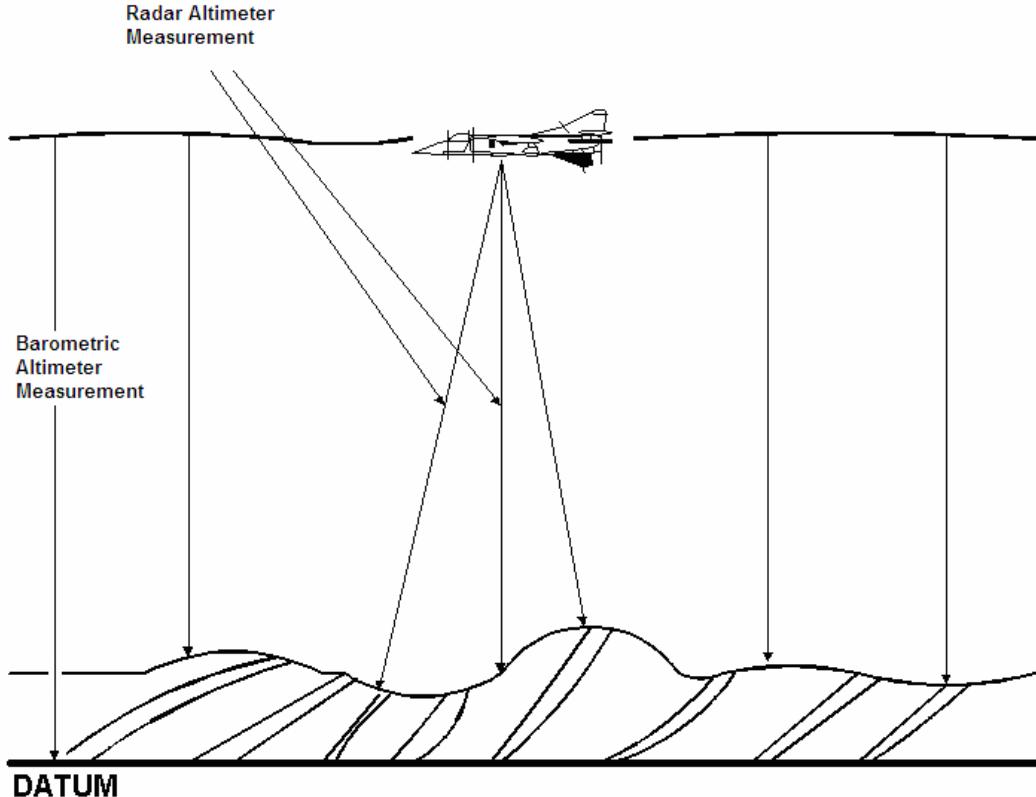


Figure 7. TAN Measurements

### 1.3.3. Digital Terrain Elevation Data (DTED)

As it is mentioned in the previous sections, the critical part of TAN is the elevation model used in the system. Generally, DTED is used for military purposes. The U. S. Department of Defense, through the National Geospatial Intelligence Agency, produces several kinds of digital cartographic data. One is digital elevation data, in a series called DTED. The data is available as 1-by-1 degree quadrangles at horizontal resolutions ranging from about 1 kilometer to 1 meter. The lowest resolution data is available to the public [14].

DTED Level 0 files have 121-by-121 points. DTED Level 1 files have 1201-by-1201. The edges of adjacent tiles have redundant records. DTED files are binary. No line ending conversion or byte-swapping is required when downloading a DTED file [14]. The data available to the public is called Level 0 and has a 30 arc second spacing. Other higher resolution data called Level 1 and Level 2 is not available to public. Performance specifications of DTED files are defined in a US military standard [15] and detailed information can be obtained from there.

For cruise missile mid-course navigation phase, because of its broad-area coverage, Level 1 DTED is used by most TAN systems. With very accurate and expensive-to-produce DTED, TAN system horizontal position accuracies rivaling those of GPS can be achieved. In TAN systems using Level 1 DTED over broad areas, accuracies in the range of 50–200 m CEP are typical for low-flying air vehicles like cruise missiles [13].

In Turkey, various levels of DTED are prepared by HGK (Harita Genel Komutanlığı - Turkish General Mapping Commandership) for all regions of Turkey from topographic maps and they are served to national institutions with protocols. The properties of DTED prepared for Turkey are given in Table 3 [16].

#### **1.4. Literature Survey on TAN**

TAN systems are generally used for military purposes. As a result of this, access to literature about TAN became very difficult. Especially for TERCOM, original famous report of Baker and Clem (1977), named “Terrain Contour Matching (TERCOM) Primer” could not be obtained. However, all the papers about TAN found in IEEE and AIAA are investigated and classified for the study. Moreover, US patents about TAN are also investigated.

Table 3. Properties of DTED Prepared by HGK

DTED Type	DTED Level 2	DTED Level 1
Map Scale	1/25,000	1/250,000
Map Datum	WGS84, ED50	WGS84, ED50
Map Coverage	1"x1"	3"x3"
Unit Map Coverage	7.5'x7.5'	1°x1°
Unit File Size	0.5 MB	3 MB
Resolution and Accuracy	±26 m horizontal ±20 m altitude	±130 m horizontal ±30 m altitude
DTED Preparation Source and Method	YÜKPAF25 Interpolation	YÜKPAF250 Interpolation
Confidentiality	Classified	Unclassified

TAN papers can be classified according to their subjects as follows:

1. Cruise Missile System Performance,
2. Terrain Models and Path Optimization,
3. TAN Applications,
4. TAN Algorithms.

As it can be seen from the survey results, TAN can be found in various subjects related with cruise missiles and navigation applications. The papers are investigated considering TAN point of view for the study.

### 1.4.1. Cruise Missile System Performance

In several papers, general system performances of TAN applications of cruise missiles are investigated. Henley [17] provides an overview of the SPARTAN technique and other techniques for improving navigation performance over very flat terrains. Navigation system performances of various TAN systems are investigated and terrain data requirements are defined. Details of the TAN algorithms are not given in the paper. In Table 4, navigation performances of various INS aided systems are compared.

Table 4. Navigation System Performances of Various INS Aiding Systems [17]

		Nominal Performance (order of magnitude figures)	Performance under conditions of				Operational Area
			Rough Terrain	Smooth Terrain	Water	Jamming	
INS		1 nm/hr	Good	Good	Good	Good	Total Coverage
I	N	TAN	< 100 m	Good	Medium	Poor	All Mapped Land
A	I	TERCOM	< 50 m	Poor	Good	Poor	Way Patches
D	E	GPS	30 m	Limited by Screening	Good	Good	Medium to Poor
w	i	TAN + GPS	30 m	Good	Good	Good	Total Coverage
t	h	TAN + GPS + TERCOM	< 30 m	Good	Good	Good	Total Coverage
							Total Coverage

Nielson [18] investigates the Conventional Air Launched Cruise Missile (CALCM) performance. Advantages of integrating GPS navigation into the missile in place of TERCOM are stated in this paper. Results and benefits of the GPS integrated cruise missile are given considering the applications in the Gulf War. However, it is known that jamming is a very important problem for GPS integrated systems. In Iraq War, several GPS aided cruise missiles have been jammed by Iraq military forces. Therefore, besides ease of using GPS, reliability problems should also be taken into consideration.

Hicks [19] provides a functional description of the navigation and guidance system in the Advanced Cruise Missile (ACM) and discusses some of the areas of improvements over the ALCM. From the paper, it can be seen that ACM has a very complex navigation and flight control system. The paper is helpful for understanding the navigation and guidance of a cruise missile.

Bennett [20] investigates the use of digital terrain map data for airborne operations. The fundamental uses of digital map data for TAN and simulators are given in the paper. Moreover, information about mission planning and simulation is also given. From his work, Bennett [20] concludes that GPS and TAN are complementary navigation sensors, and when properly integrated with INS, they provide the essential correlation of aircraft position with respect to the actual ground contours.

#### **1.4.2. Terrain Models and Path Optimization**

Terrain models used in TAN systems are very critical. They should be modeled as accurate as possible in order to obtain better navigation solutions. Chen and Yu [21] improve the models used for TAN. Actually, they improve the terrain model by considering horizontal position noises of the terrain model as colored noises. The TAN algorithm used in the paper is SITAN which will be discussed in the following sections in detail.

A very similar paper is presented by Wang and Chen [22]. In their paper, possible error sources related to the elevation model is added to the SITAN equations in order to improve navigation solutions. Yu, et al [23] also propose various terrain linearization techniques required for SITAN implementation and present the improvements in navigation solutions.

Terrain models are also critical for mission planning of the airborne vehicles. Therefore, selection of the optimal path when using terrain models is one of the major problems of TAN. Paris and Le Cadre [24] investigate the planification of a mobile trajectory in order to use its own motion for improving its position estimation. In other words, an optimal trajectory is aimed to be planned which minimizes the localization error along the path or at the arrival area. In the paper of McFarland, et al [25], techniques originally developed for robot motion planning are applied to compute paths for autonomous air vehicles, such as cruise missiles or UAV's. This approach is said to be particularly useful in multi-objective optimization problems such as intercepting a target while also maneuvering to minimize observability to ground-based tracking stations.

Improvement of TAN using optimization is also one of the subjects of TAN. Bar-Gill, et al [26] propose a new method for improving the accuracy of TAN algorithms. They minimize the navigation errors which propagate along the flight path by designing airframe trajectories in a priori mission planning. The method uses information theory-based conditional entropy mapping and synthesizes minimum-entropy trajectories. Hence, by selecting optimal flight paths, navigation accuracies of the used TAN algorithms are improved.

In the paper of Li, et al [27], optimal control methodology is adopted to design a terrain following controller for cruise missile. In this methodology, both tracking errors and control increments are considered in a quadratic penalty function. This paper is different from others; because, terrain following flight is investigated from the control point of view. Here, the TAN algorithms used and errors due to navigation system are not considered.

### **1.4.3. TAN Applications**

TAN applications can be found in literature especially for military purposes. Here, instead of developing new TAN algorithms, the applications of the known algorithms to real systems are given. In order to concentrate on TAN algorithms in detail, TAN applications are presented in a separate section. TAN algorithms found in literature will be investigated in the following section.

Baird and Snyder [28] describe the design, mechanization and preliminary flight testing of a new altitude channel implementation, referenced primarily to the SITAN altitude estimates for AFTI/F-16 aircraft. Their paper is a typical SITAN algorithm application to a real system. In a similar way, Hollowell [29] presents the application of SITAN algorithm to US Army UH-1 Helicopter. In the paper, Multiple Model Adaptive Estimation (MMAE) techniques are employed for SITAN algorithm using a bank of single state Kalman filters to ensure that reliable position estimates are obtained even in the face of large initial position errors.

Another example of a TAN application is the paper of Nordlund and Gustafsson [30]. They estimate the position of an aircraft using a terrain aided positioning algorithm based on a Rao-Blackwellisation technique. This technique uses recursive Monte Carlo methods, also known as particle filters and provides a favorable approximate solution. The TAN algorithm used here will be investigated in the following section considering the original paper of Bergman, et al [31].

TAN can be used in applications not only for air vehicles but for underwater and land vehicles as well. Newman and Durrant-Whyte [32] describe and investigate autonomous navigation of an underwater vehicle which uses inertial and sonar based sensors. In their paper, they associate inertial and sonar based world frame feature information in order to form a robust navigation algorithm. They do not use ready terrain map information for navigation; but, they form feature information around the underwater vehicle by using sonar measurements.

TAN applications for land vehicles are found in the papers of Madhavan, et al [33] and Bruder, et al [34]. Madhavan, et al [33] describe a TAN system which employs points of maximum curvature extracted from laser scan data as primary landmarks. On the other hand, Bruder, et al [34] present the development and implementation of a new sensor integration algorithm employing a terrain map to reduce INS errors. The algorithm used in this paper is very similar to classical TAN applications which use terrain height data for navigation correction.

Terrain model improvement can also be considered as applications of TAN. Accurate terrain models not only improve the accuracy of the navigation system but they are also required for accurate height profile of the concerned areas. Morisue and Ikeda [35] demonstrate a navigation system which is used for high level of location accuracy. They achieve it by using various map-matching techniques. On the other hand, McLellan and Schleppe [36] describe an integrated real-time differential GPS and barometry system, with the prime aim of significantly changing and improving the method of positioning and layout of Shell Canada's land seismic surveys. They also state that the system had provided horizontal positioning better than 5 meters and height accuracy of better than 2 meters at 2 sigmas. The system proposed is actually an integrated GPS system instead of a TAN algorithm application. However, since terrain height information is obtained accurately, the paper can be considered as an example for batch process terrain modeling application.

#### **1.4.4. TAN Algorithms**

As stated in the previous sections above, the heart of TAN is the algorithms. A number of TAN techniques have been developed and tested. These fall into two general algorithmic categories of batch and recursive algorithms [11] as explained in the previous sections.

Literature survey on TAN algorithms is done considering these algorithmic categories.

#### 1.4.4.1. Batch TAN Algorithms

The famous batch TAN algorithm found in literature is TERCOM. As it was mentioned before, original famous report of Baker and Clem (1977), named “Terrain Contour Matching (TERCOM) Primer” could not be obtained. However, detailed information about TERCOM is found in the book of Siouris [37]. TERCOM is a form of correlation guidance based on a comparison between the measured and the pre-stored features of the profile of the ground (i.e., terrain) over which a missile or aircraft is flying. Generally, terrain height forms the basis of this comparison [37]. There are a number of correlation algorithms (e.g., mean squared difference (*MSD*), mean absolute difference (*MAD*), the normalized *MAD*, the normalized *MSD*, and the product method) of varying complexity and accuracy that can be used to correlate the measured data with the reference data. Furthermore, the *MAD* algorithm provides the best combination of accuracy and computational efficiency for performing real-time terrain contour matching in an onboard computer environment [37]. Actually, TERCOM is a maximum likelihood estimator which uses only terrain height information for determining the vehicle’s actual position. TERCOM is a batch process. Therefore, information about the position of the vehicle is post processed in order to have a navigation solution. TERCOM will be investigated in detail in the following chapter as one of the major TAN algorithms.

Johnson, et al [11] improve the performance of TERCOM and SITAN by using maximum a posteriori estimator (MAP). Their technique makes more complete use of the information content of the stored elevation data, the a priori knowledge of the errors in the stored elevation data and the elevation

measurements. Actually, the technique is a batch algorithm which uses past information for TAN algorithm. However, this past information improves the results of both batch and recursive algorithms as explained in the paper. The theory behind the algorithm is straight forward; however, in order to apply MAP algorithm, extra computations are required.

Erhui, et al [38] propose a new TAN algorithm based on the probability distribution differences of terrain height samples. They call their technique as Probability-Based Terrain Aided Navigation (PTAN) approach. The technique proposed is a batch algorithm and instead of correlating the height data collected by the radar and the barometer as in TERCOM, the proposed PTAN algorithm computes the probability distribution difference between them. The minimum probability distribution difference gives the best matching and the position of the air vehicle is determined accordingly. Again, the theory behind the algorithm is straight forward; however, in order to find probability distributions, considerable computational load is required.

Zhou and Zhang [39] propose a scheme of TAN based on principle of computer vision. Being different from the conventional terrain matching technique, i.e. TERCOM, the scheme uses CCD camera rather than barometer and radio altimeter as sensing element. The technique proposed is a batch algorithm and it is claimed in the paper that shorter flight time is sufficient for successful terrain matching. Since, the original paper is in Chinese, details of the algorithm can not be obtained; only abstract of the paper is investigated. However, since CCD camera is used for correlation, environmental constraints should be considered. In other words, CCD camera can not be used in all weather conditions. This is thought to be the major drawback of the algorithm proposed.

Quintang, et al [40] propose a new TAN approach using probabilistic data association filter (PDAF) to overcome irresolvable ambiguities in the correlation function used in TERCOM. The basic idea of the approach is to convert correlation

function value to the probability of position estimate being actual position of the air vehicle. It is shown via set of simulations that the method can improve the performance of TAN compared to TERCOM. The approach proposed is a batch algorithm. The interesting point is it uses one of the modern radar tracking algorithms. Actually, TAN is a data association problem, especially for the acquisition mode where INS position errors are very large. Here, it was thought whether the algorithm could be used for real-time applications. Therefore, this paper gave inspiration for implementing modern data association algorithms to TAN in the Ph.D. study.

#### 1.4.4.2. Recursive TAN Algorithms

The major recursive TAN algorithm found in literature is SITAN proposed by Hostetler and Andreas [12]. They investigate the application of nonlinear Kalman filtering techniques to the continuous updating of an INS using individual radar terrain clearance measurements in their paper. First order Extended Kalman Filter (EKF) is used in order to model the slopes of the terrain surface. Hence, real-time TAN solution can be obtained. Moreover, for large initial position uncertainties, a parallel Kalman filter technique which uses a bank of reduced order filters is used. The technique is the first EKF implementation to TAN. However, due to highly nonlinear structure of the terrain profiles, the filter solutions can diverge especially for large position errors. Moreover, linearization of the terrain slopes is the critical point for SITAN algorithm. Especially for mountainous terrains, modeling of the terrain slopes is a considerable problem. SITAN will be investigated in detail in the following chapter as one of the major TAN algorithms.

Pei, et al [41] propose BITAN algorithm for navigation solution in their paper. BITAN is a type of TAN algorithm using the Kalman filtering theory to estimate position and velocity errors of the INS. The algorithm has been developed

for both acquisition and tracking modes of operation. Actually, the algorithm is an improved SITAN algorithm especially for acquisition mode. Moreover, again bank of Kalman filters are used for navigation solution. Hence, same problems for SITAN algorithm exist also for this algorithm.

TAN is a nonlinear estimation problem. Bergman, et al [31] derive the optimal Bayesian solution for TAN. The implementation is grid based, calculating the probability of a set of points on an adaptively dense mesh. Actually, Bayes formula is a well-known formula in estimation. However, direct application of the formula is very restricted due to computational problems. As a result of this, Bergman, et al [31] propose the Cramer-Rao bound for Bayesian solution implementation. The major disadvantage of the algorithm proposed is originated from computational problems.

One of the most interesting algorithms is proposed by Enns and Morrell [42]. They propose a new TAN algorithm called VATAN which uses the Viterbi algorithm for navigation solution. The Viterbi algorithm is a dynamic programming algorithm used for data association problem. From the simulation results, it is shown that VATAN algorithm overcomes divergence problems associated with the EKF in SITAN and provides position estimates with smaller average squared errors. Actually, navigation accuracy is improved with VATAN compared to SITAN especially for flat and mountainous terrains. In order to implement the algorithm, conditional probabilities of the measurements and INS states should be calculated recursively. Actually, the algorithm is different from other real-time TAN algorithms and it has better results than SITAN.

Dezert [43] proposes a new application of PDAF for improving the accuracy of autonomous strapdown INS. The method proposed is a TAN algorithm based on landmark detection combined with a classical strapdown INS. It is also stated that the algorithm can be integrated with relatively low cost in existing operational TAN systems. Actually, the algorithm does not use elevation data for navigation solution.

However, it is a real-time application of PDAF and relation with the former paper of Quintang, et al [40] can be obtained where batch implementation of PDAF is used. Therefore, this paper also gave inspiration for implementing real-time PDAF to TAN in the Ph.D. study.

Some recursive TAN algorithms are also proposed which use images for navigation solution. Hongbo, et al [44] and Bevington, et al [45] can be examples to image based TAN. Hongbo, et al [44] propose a TAN algorithm which use range images from imaging laser radar. On the other hand, Bevington, et al [45] use images of Synthetic Aperture Radar (SAR) for navigation solution. Both methods require detection of land marks since images are used. As it was stated in the previous section, the major drawback of the image based TAN comes from environmental constraints.

There exist also some hybrid TAN methods which use both batch and recursive algorithms together. Metzger, et al [3] propose a hybrid TAN system which uses a bank of Kalman filters and a comparison technique. Actually, the proposed algorithm is a mixture of TERCOM and SITAN algorithms. Using the advantages of both algorithms, better navigation solutions can be obtained.

#### **1.4.5. TAN Patents**

United States patents related with TAN are also investigated in literature survey. Since patents are practical applications, detailed information can be obtained from them. Several patents are investigated related with TAN algorithms.

Chan and Snyder [46] propose a system for correlation and recognition of terrain elevation. They use correlation function in frequency domain in order to

improve navigation solutions. The method is the improvement of the correlation technique.

Baird [47] integrates TERCOM and SITAN algorithms with a modified Kalman filter processor. Hence, the operation of the SITAN processing is effectively continuously optimized. Actually, the system is the application of a hybrid TAN algorithm as explained in the previous section.

Lerche [48] improves TERCOM method by scanning a larger area for correlation process. The method is applied for the navigation of an aircraft. Raymer, et al [49] proposes a method for Schuler cycle error reduction for use in a TAN system. By detecting Schuler cycles, TAN system errors are degraded.

Finally, Goebel, et al [50] propose a terrain correlation system for TAN. Actually, the correlation system is an improved TERCOM algorithm which uses MAD correlation. Detailed information including application methods are given in the related patent.

It is known that TAN algorithms are used generally for military purposes. Due to the confidentiality of the subject, related patents about TAN are limited. In fact, the patents found are taken many years later than their technology developed.

## 1.5. Target Tracking

TAN is a nonlinear estimation problem; since, terrain height information is used for navigation solution. Actually, TAN can be considered as a data association problem, especially for the acquisition operation mode where INS position errors are considerably large. From the literature survey of Quintang, et al [40] and Dezert [43], it has been thought that modern data association algorithms can be

implemented for real-time TAN algorithms. Therefore, radar tracking, especially data association subject is investigated. In this section, general information about radar tracking and data association algorithms will be given.

### 1.5.1. Background

The modern need for tracking algorithms began with the development of radar during World War II. By the 1950's, radar was a relatively mature technology. Systems were installed aboard military ships and aircraft and at airports. The tracking of radar targets, however, was still performed manually by drawing lines through blips on a display screen. The first attempts to automate the tracking process were modeled closely on human performance. For the single-target case, the resulting algorithm was straight forward; the computer accumulated a series of positions from radar reports and estimated the velocity of the target to predict its future position [51].

Even single-target tracking presented certain challenges related to the uncertainty inherent in position measurements. A first problem involves deciding how to represent this uncertainty. A crude approach is to define an error radius surrounding the position estimate. This practice implies that the probability of finding the target is uniformly distributed throughout the volume of a three-dimensional sphere. Unfortunately, this simple approach is far from optimal. The error region associated with many sensors is highly non-spherical; radar, for example, tends to provide accurate range information but has relatively poorer radial resolution. Furthermore, one would expect the actual position of the target to be closer on average to the mean position estimate than to the perimeter of the error volume, which suggests, in turn, that the probability density should be greater near the center [51].

A second difficulty in handling uncertainty is determining how to interpolate the actual trajectory of the target from multiple measurements, each with its own error allowance. For targets known to have constant velocity (e.g., they travel in a straight line at constant speed), there are methods for calculating the straight-line path that best fits, by some measure, the series of past positions. A desirable property of this approach is that it should always converge on the correct path, as the number of reports increases, the difference between the estimated velocity and the actual velocity should approach zero. On the other hand, retaining all past reports of a target and recalculating the entire trajectory every time a new report arrives is impractical. Such a method would eventually exceed all constraints on computation time and storage space [51].

A near-optimal method for addressing a large class of tracking problems was developed in 1960 by R.E. Kalman. His approach, referred to as Kalman filtering, involves the recursive fusion of noisy measurements to produce an accurate estimate of the state of a system of interest. A key feature of the Kalman filter is its representation of state estimates in terms of mean vectors and error covariance matrices, where a covariance matrix provides an estimate (usually a conservative over-estimate) of the second moment of the error distribution associated with the mean estimate. The square root of the estimated covariance gives an estimate of the standard deviation. If the sequences of measurement errors are statistically independent, the Kalman filter produces a sequence of conservative fused estimates with diminishing error covariances [51].

Kalman's work had a dramatic impact on the field of target tracking in particular and data fusion in general. By the mid-1960's, Kalman filtering was a standard methodology. It has become as central to multiple-target tracking as it has been to single-target tracking; however, it addresses only one aspect of the overall problem [51].

### 1.5.2. Data Association Algorithms

In tracking targets with less-than-unity probability of detection in the presence of false alarms (clutter), data association, deciding which of the received multiple measurements to use to update each track is crucial. A number of algorithms have been developed to solve this problem. Two simple solutions are:

1. Strongest Neighbor Filter (SNF), and
2. Nearest Neighbor Filter (NNF).

In the SNF, the signal with the highest intensity among the validated measurements (in a gate) is used for track update and the others are discarded. In the NNF, the measurement closest to the predicted measurement is used. While these simple techniques work reasonably well with benign targets in sparse scenarios, they begin to fail as the false alarm rate increases or with low observable (low probability of target detection) maneuvering targets [52].

The NNF is perhaps the simplest approach for determining which tracked object produced a given sensor report. When a new position report arrives, all existing tracks are projected forward to the time of the new measurement. Then, the distance from the report to each projected position is calculated, and the report is associated with the nearest track. More generally, the distance calculation is computed to reflect the relative uncertainties (covariances) associated with each track and report [51].

In Figure 8, NNF implementation is shown. The idea of the rule is to estimate each object's position at the time of a new position report, and then assign the report to the nearest such estimate. This intuitively plausible approach is especially attractive because it decomposes the multiple-target tracking problem into a set of single-target problems [51].

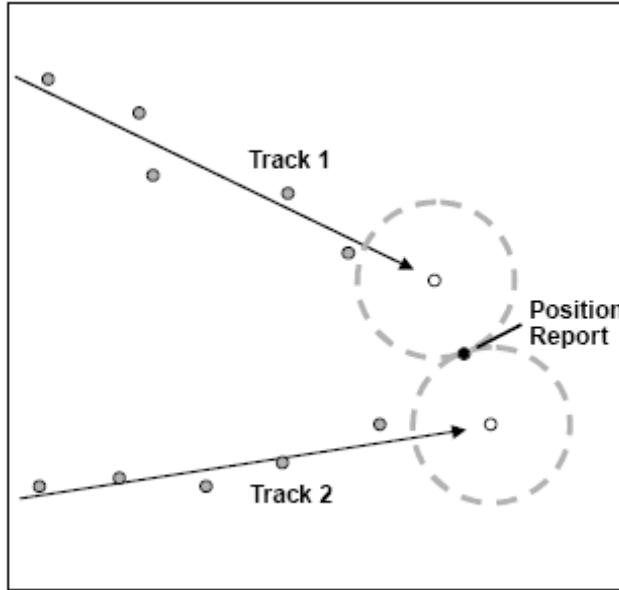


Figure 8. NNF Implementation [51]

Data association becomes more difficult with multiple targets where the tracks compete for measurements. Here, in addition to a track validating multiple measurements as in the single target case, a measurement itself can be validated by multiple tracks (i.e., contention occurs among tracks for measurements). Several algorithms are developed to handle this contention:

1. Track Splitting (TS),
2. Multiple Hypothesis Tracking (MHT),
3. Probabilistic Data Association (PDA),
4. Joint Probabilistic Data Association (JPDA).

Actually, there are various data association algorithms proposed for multiple target tracking in literature. However, the algorithms listed above can be considered as modern data association algorithms.

In track splitting and MHT, a robust solution to the problem of assignment ambiguities is found by creating multiple hypothesis tracks. Under this scheme, the tracking system does not have to commit immediately or irrevocably to a single assignment of each report. If a report is highly correlated with more than one track, an updated copy of each track can be created; subsequent reports can be used to determine which assignment is correct. As more reports come in, the track associated with the correct assignment will rapidly converge on the true target trajectory, whereas the falsely updated tracks are less likely to be correlated with subsequent reports [51].

This basic technique is called track splitting. One of its worrisome consequences is a proliferation in the number of tracks upon which a program must keep tabs. The proliferation can be controlled with the same track deletion mechanism used in the nearest-neighbor algorithm, which scans through all the tracks from time to time and eliminates those that have a low probability of association with recent reports. A more sophisticated approach to track splitting, called multiple-hypothesis tracking, maintains a history of track branchings, so that as soon as one branch is confirmed, the alternative branches can be pruned away [51]. MHT is a more powerful (but much more complex) algorithm that handles the multi-target tracking problem by evaluating the likelihood that there is a target given a sequence of measurements.

In PDA, instead of using only one measurement among the received ones and discarding the others, an all of the validated measurements with different weights (probabilities) are used. The standard PDA and its numerous improved versions have been shown to be very effective in tracking a single target in clutter [52].

JPDA algorithm is used to track multiple targets by evaluating the measurement-to-track association probabilities and combining them to find the state estimate [52]. Actually, JPDA is the developed version of PDA algorithm for multiple targets.

PDA, TS and MHT will be investigated in detail in the following chapters for TAN implementation.

## 1.6. Outline of the Thesis

In this section, the outline of the Ph.D. study will be given. The thesis is composed of five chapters:

1. Introduction,
2. Major TAN Methods,
3. Implementation of Radar Tracking Algorithms to TAN,
4. Case Study,
5. Discussion and Conclusion.

In the first chapter, an introduction to the study was done. First, the scope of the study was presented. Then, general information about cruise missiles and TAN was given. A detailed literature survey was performed about TAN and was presented in this chapter. Finally, general information about data association algorithms was given as fundamental knowledge of the study.

In Chapter 2, major TAN methods are investigated. First, INS errors of the cruise missiles and need for TAN systems are discussed. Then, major TAN methods including TERCOM, SITAN and VATAN are presented in detail. Fundamentals of the major methods are discussed in this chapter in order to make comparisons for the implemented TAN algorithms in the Ph.D. study.

In Chapter 3, implementation of data association algorithms to TAN is presented. This chapter contains the original Ph.D. work. First, general information about modern target tracking algorithms are given. PDAF and TSF algorithms and their general implementations are investigated. Then, PDAF and TSF implementations to TAN are presented. At the end of the chapter, a simple simulation model is developed for the mid-course flight of the cruise missile. Finally, simulations are performed with the implemented TAN algorithms and the results are compared with the major TAN methods.

In Chapter 4, case studies are performed. A 6 DOF simulation tool is developed for the simulation of the mid-course flight of a cruise missile. Implemented TAN algorithms are used with the 6 DOF simulation model and their performances are investigated.

In Chapter 5, the results obtained from the study are discussed. Advantages and disadvantages of the new implemented TAN algorithms are compared with the major TAN algorithms. Finally, conclusions of the study are presented.

## **CHAPTER 2**

### **MAJOR TERRAIN AIDED NAVIGATION METHODS**

In this chapter, first, general INS errors and TAN INS errors in cruise missiles will be discussed. Then, the need for TAN in cruise missiles will be investigated. Next, major TAN algorithms and their implementations will be presented. TERCOM, SITAN and VATAN will be investigated in detail in this chapter. Eventually, navigation performance of these major TAN algorithms will be discussed; and, conclusions obtained will be presented.

#### **2.1. Cruise Missile INS Errors**

##### **2.1.1. INS Only Errors**

The development of inertial navigation technology took place primarily in Germany, the United States and the former Soviet Union. The gyro compass indicating true north on a moving base as on ships can be regarded as the beginning of inertial navigation. At the end of World War I the allies had in the Treaty of Versailles imposed restrictions to Germany for the maximum size of ships to be built. These restrictions promoted in this country gun stabilization and inertial technology in general, which culminated at the end of World War II in a

functioning air-supported gyrocompass with electronic Schuler tuning for the “One-Man Submarines”, in the V2 guidance system and a true concept for an INS. After the war the development of this technology was taken over by the superpowers, the United States and the former Soviet Union [53].

Inertial navigation systems (INS) are sophisticated autonomous, electromechanical systems that supply the position, velocity and attitude of the vehicle on which they are mounted. INS is basically a measuring system; therefore, the outputs of an INS will contain errors due to its sensors (accelerometers and gyroscopes) and mechanization. Inertial navigation sensor component errors create error in the navigation system’s computed position, velocity, and attitude. Accelerometer and gyroscope errors can be represented in a general form, including some significant environment dependent errors as [54]:

1. Biases and drifts,
2. Scale factor and misalignments,
3. White noise,
4. Time correlated short-term errors,
5. Other environment sensitive errors.

A lot of experience has been gained on the behavior of INS errors from the accumulated experience of INS users and analysts. Various linear models were developed that describe accurately the behavior of these errors as given in references [54], [55], and [56]. These models were used in the implementation of Kalman filters for estimating the INS error outputs and error sources [55].

There are two approaches to the derivation of INS error models. One of them is known as the  $\phi$ -angle (perturbation or true frame) approach, and the other is known as  $\psi$ -angle (or computer frame) approach. When deriving the perturbation error model, the nominal non-linear navigation equations are perturbed in the local-level north-pointing Cartesian coordinate system that corresponds to the true geographic location of the INS. The  $\psi$ -angle error model, on the other hand, is obtained when the nominal equations are perturbed in the local-level north-pointing coordinate system that corresponds to the geographic location indicated by the INS. It has been shown that both models are equivalent and yield, therefore, identical results. The differential equations that describe the error behavior of the INS are divided into equations describing the propagation of the attitude errors. Both the translatory and the attitude error equations can be expressed in two different ways that yield two versions of the translatory error equations and two versions of the attitude error equations. The two versions of the translatory equations depend on whether the equation variables are position error components or velocity error components. The two versions of the attitude equations depend on whether the equation variables are components of the platform to computer frame attitude difference, or components of the platform to true frame attitude difference. All these versions are, of course, identical. In order to obtain a complete set of INS error equations, the analyst has to decide whether to adopt the perturbation or  $\psi$ -angle approach. Once this choice is made, the analyst has to decide which of the two corresponding versions of the translatory equations to use and which of the two versions of the attitude equations to use. (These two choices are independent.) [56].

Most of the published work on INS errors adopt the  $\psi$ -angle approach and use the velocity error version of the translatory error equation. This model is also used in the present analysis. In addition, the components of the platform to computer frame attitude differences are used as the variables of the attitude error equations. Although this angular difference is imaginary and cannot be measured, it possesses the advantage that the translatory error is not coupled into the attitude

error equations. The physical attitude difference between the platform and the local-level north-pointing coordinate system is calculable using the position and attitude errors obtained from the solution of these INS error equations. Then, a complete terrestrial INS error model, expressed by the following equations is obtained [55]:

$$\delta\dot{\vec{v}} + (\vec{\Omega} + \vec{\omega}) \times \delta\vec{v} = \vec{\nabla} - \delta\vec{\Psi} \times \vec{f} + \Delta\vec{g} \quad (2.1)$$

$$\delta\dot{\vec{r}} + \vec{\rho} \times \delta\vec{r} = \delta\vec{v} \quad (2.2)$$

$$\delta\dot{\vec{\Psi}} + \vec{\omega} \times \delta\vec{\Psi} = \vec{\varepsilon} \quad (2.3)$$

where  $\delta\vec{v}$ ,  $\delta\vec{r}$  and  $\delta\vec{\Psi}$  are, respectively, the velocity, position, and attitude error vectors;  $\vec{\Omega}$  is the Earth rate vector;  $\vec{\omega}$  is the angular rate vector of the true coordinate system with respect to inertial frame;  $\vec{\nabla}$  is the accelerometer error vector;  $\vec{f}$  is the specific force (accelerometer readings) vector;  $\Delta\vec{g}$  is the error in the computed gravity vector;  $\vec{\rho}$  is the vector of the rate of turn of the true frame with respect to Earth; and finally  $\vec{\varepsilon}$  is the gyro drift vector. From geometric relations, it can be shown that in the local north, east and down coordinate system (i.e. in the geographic frame) [55]:

$$\vec{\Omega} = \begin{bmatrix} \Omega \cdot \cos \lambda \\ 0 \\ -\Omega \cdot \sin \lambda \end{bmatrix} \quad (2.4)$$

where  $\lambda$  is the local latitude. In the same manner, the vector  $\vec{\omega}$  is computed as follows [55]:

$$\vec{\omega} = \vec{\Omega} + \vec{\rho} \quad (2.5)$$

where  $\vec{\rho} = \begin{bmatrix} \dot{\mu} \cdot \cos \lambda \\ -\dot{\lambda} \\ -\dot{\mu} \cdot \sin \lambda \end{bmatrix}$  and  $\mu$  is the local longitude.

When INS position, velocity and attitude error equations are resolved in true frame (i.e. geographic frame), nine scalar differential equations are obtained, which can be put in a state-space model. If the expressions for  $\vec{\Omega}$ ,  $\vec{\omega}$  and  $\vec{\rho}$  are used, the resulting state-space model is obtained as follows [55]:

$$\frac{d}{dt} \begin{bmatrix} \delta r_N \\ \delta r_E \\ \delta r_D \\ \delta v_N \\ \delta v_E \\ \delta v_D \\ \delta \psi_N \\ \delta \psi_E \\ \delta \psi_D \end{bmatrix} = \begin{bmatrix} 0 & -\dot{\mu} \cdot s\lambda & \dot{\lambda} & 1 & 0 & 0 & 0 & 0 & 0 \\ \dot{\mu} \cdot s\lambda & 0 & \dot{\mu} \cdot c\lambda & 0 & 1 & 0 & 0 & 0 & 0 \\ -\dot{\lambda} & -\dot{\mu} \cdot c\lambda & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -g/R & 0 & 0 & 0 & -(2 \cdot \Omega + \dot{\mu}) \cdot s\lambda & \dot{\lambda} & 0 & -f_D & f_E \\ 0 & -g/R & 0 & (2 \cdot \Omega + \dot{\mu}) \cdot s\lambda & 0 & (2 \cdot \Omega + \dot{\mu}) \cdot c\lambda & f_D & 0 & -f_N \\ 0 & 0 & 2 \cdot g/R & -\dot{\lambda} & -(2 \cdot \Omega + \dot{\mu}) \cdot c\lambda & 0 & -f_E & f_N & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -(2 \cdot \Omega + \dot{\mu}) \cdot s\lambda & \dot{\lambda} \\ 0 & 0 & 0 & 0 & 0 & 0 & (\Omega + \dot{\mu}) \cdot s\lambda & 0 & (\Omega + \dot{\mu}) \cdot c\lambda \\ 0 & 0 & 0 & 0 & 0 & 0 & -\dot{\lambda} & -(\Omega + \dot{\mu}) \cdot c\lambda & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta r_N \\ \delta r_E \\ \delta r_D \\ \delta v_N \\ \delta v_E \\ \delta v_D \\ \delta \psi_N \\ \delta \psi_E \\ \delta \psi_D \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \nabla_N \\ \nabla_E \\ \nabla_D \\ \varepsilon_N \\ \varepsilon_E \\ \varepsilon_D \end{bmatrix} \quad (2.6)$$

where;

$\delta r$ : Scalar position errors

$\delta v$ : Scalar velocity errors

$\delta \psi$ : Scalar attitude errors

$\Omega$ : Earth's inertial angular velocity

$\mu$ : Longitude of the true frame with respect to Greenwich meridian

$\dot{\mu}$ : Longitude rate of the true frame

$\lambda$ : Latitude of the true frame with respect to Equator

$\dot{\lambda}$ : Latitude rate of the true frame

$g$ : Earth's gravity

$R$ : Radius of Earth

$f$ : Specific forces sensed by the accelerometers

$\nabla$ : Scalar accelerometer biases

$\varepsilon$ : Scalar gyro drifts

$N, E, D$ : Subscripts denoting north, east and down components respectively

$s$ : Sine of the defined angle

$c$ : Cosine of the defined angle

The error model given in equation (2.6) can be used in simulations for predicting INS errors of the system. Another way of determining INS errors is the direct application of the real error sources (from both sensors and mechanization) in the navigation equations. Actually, the error model obtained above is used for

integrating various navigation systems with INS. However, for simulations, INS error model will be sufficient and its implementation will be much easier.

INS error model is applied for the mid-course phase of a cruise missile. The cruise missile is assumed to be moving with constant velocity. Moreover, acceleration changes during mid-course flight are assumed smaller. INS quality is taken as 1.0 nm/hr for simulations.

INS quality is the major parameter in order to achieve the required navigation solutions. It is mainly determined by sensor quality and initial alignment errors. In Table 5, various INS qualities and corresponding sensor and initial alignment errors are presented.

Table 5. INS Sensor Error Sources [57]

	INS Quality (All errors except random walk are $1\sigma$ biases)			
Error Source	10 nm/hr	1.0 nm/hr	0.5 nm/hr	0.2 nm/hr
Accelerometer Bias	223 $\mu$ g	37 $\mu$ g	19 $\mu$ g	4.2 $\mu$ g
Accel. Scale Factor	223 ppm	179 ppm	90 ppm	21 ppm
Input Axis Misalign.	22 arcsec	3 arcsec	1.5 arcsec	0.4 arcsec
Random Walk	56 $\mu$ g/ $\sqrt{\text{hz}}$	56 $\mu$ g/ $\sqrt{\text{hz}}$	7.5 $\mu$ g/ $\sqrt{\text{hz}}$	4.2 $\mu$ g/ $\sqrt{\text{hz}}$
Gyro Bias	0.11 deg/hr	4.5e-3 deg/hr	2.2e-3 deg/hr	8.4e-4 deg/hr
Gyro Scale Factor	112 ppm	112 ppm	7.5 ppm	1.67 ppm
Input Axis Misalign.	22 arcsec	2.2 arcsec	1.1 arcsec	0.4 arcsec
Random Walk	0.078 deg/ $\sqrt{\text{hr}}$	2.2e-3 deg/ $\sqrt{\text{hr}}$	1.1e-3 deg/ $\sqrt{\text{hr}}$	5e-4 deg/ $\sqrt{\text{hr}}$
Initial Misalignment (Vertical/Horizontal)	2089 arcsec/ 59 arcsec	606 arcsec/ 59 arcsec	600 arcsec/ 29 arcsec	600 arcsec/ 29 arcsec

INS quality is generally expressed by the total position error divided by time. For example, 10 nm/hr INS quality will be sufficient for ballistic missiles. However, for military aircrafts and cruise missiles 1.0 nm/hr INS quality is required. In the same manner, as operation time and required range increases, INS quality will be also increased. Intercontinental Ballistic Missiles (ICBM) and space vehicles use very accurate INS. The major problem of using very accurate INS is its cost. Moreover, due to large space requirements of very accurate INS, they can not be used in most of the military systems.

INS error model simulations are performed in Simulink [58]. Considering 1.0 nm/hr INS quality, horizontal position and velocity errors and attitude errors are obtained. It is known that, an initial altitude error ( $\Delta h_0$ ) or altitude-rate error ( $\Delta \dot{h}_0$ ) or an accelerometer error will grow exponentially with time, thus making the indicated altitude and altitude-rate indications useless after a few minutes. The instability of the vertical channel for INS will result, no matter how carefully the vertical component of gravity is mechanized as a function of computed altitude [59]. In real systems, vertical channel of INS is generally aided by barometric altimeters. Since, horizontal position errors are critical for TAN, altitude errors are not investigated. Simulation results are shown in Figure 9, Figure 10, and Figure 11.

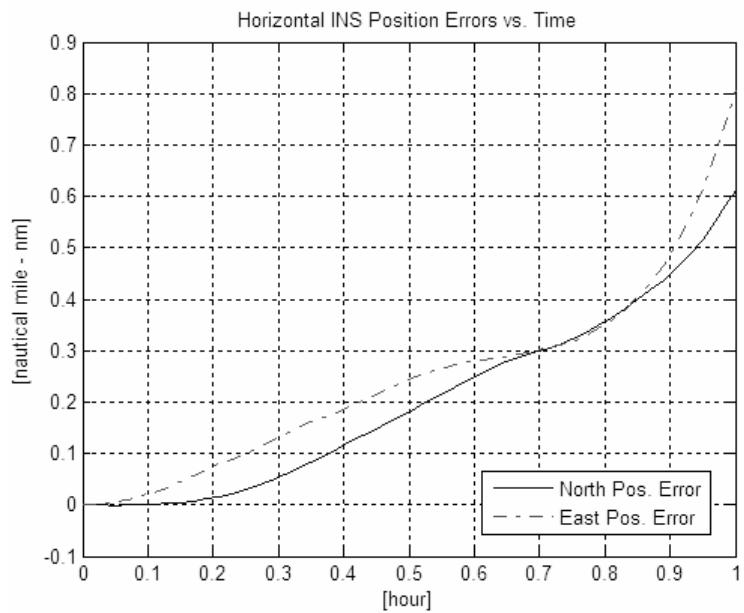


Figure 9. Horizontal Position Errors of the INS Error Model

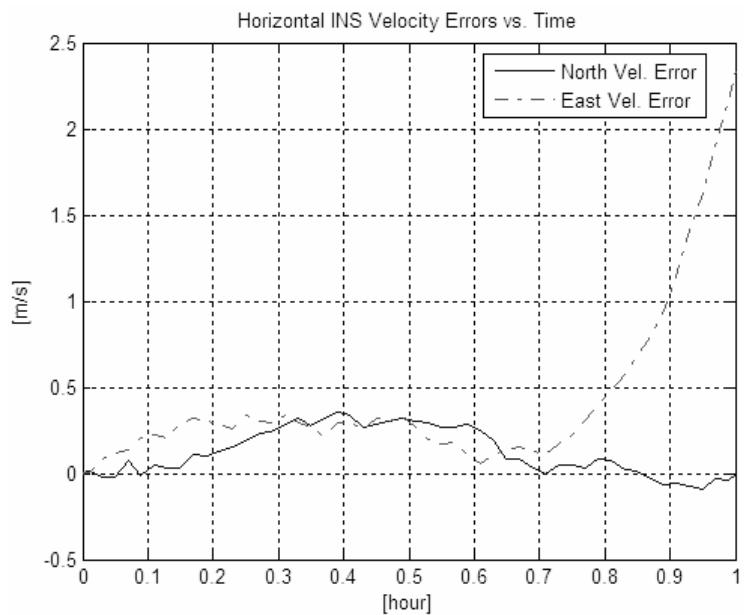


Figure 10. Horizontal Velocity Errors of the INS Error Model

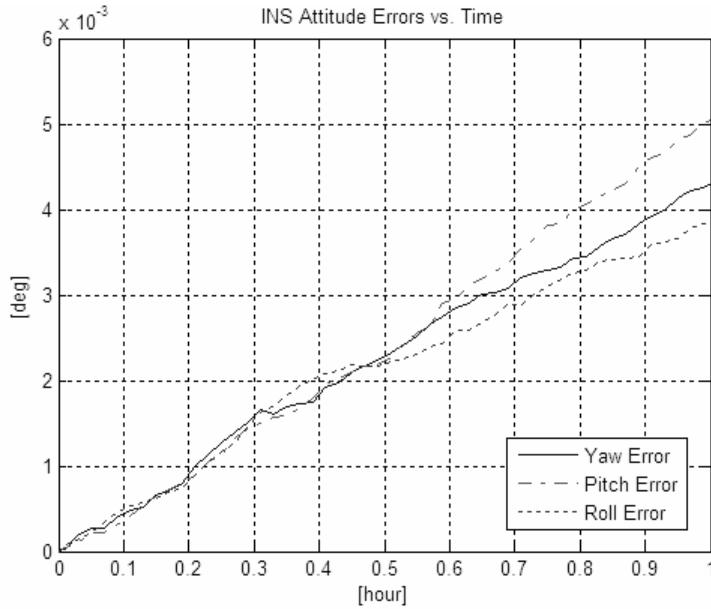


Figure 11. Attitude Errors of the INS Error Model

As it can be seen from the simulation results, quadratic increase of position errors due to double integration dominates the INS only navigation solution. On the other hand, the validity of the error model can be seen from Figure 9. Here, using sensor errors defined in Table 5, 1.0 nm/hr quality INS is achieved.

### 2.1.2. TAN INS Errors

INS only errors of a navigation system are discussed in the previous section. As it can be seen from the simulation results, due to large navigation times of cruise missiles, INS should be aided with other navigation systems. TAN is the well-known method for improving navigation solution.

The accuracy of the TAN position estimate for a simple case is derived from application of linear estimation theory. Using horizontal INS position errors

modeled as independent random walks (uncorrelated white noises), following expression can be obtained for the circular error probable (CEP) of horizontal position updates [13].

$$CEP_{ss} = 0.57 \cdot \delta V^{1/4} \cdot (\Delta d / s)^{3/8} \cdot (\sigma_n / h)^{3/4} \quad (2.7)$$

where;

$CEP_{ss}$  : The steady state CEP of horizontal position updates (m),

$\sigma_n$  : Standard deviation of the profile measurement errors (m),

$h$  : Deterministic local terrain slope at the measurement locations in both down-range and cross-range directions (unitless),

$\Delta d$  : Distance between profile measurements (m),

$s$  : Vehicle ground speed (m/s),

$\delta V$  : Maximum INS velocity error (m/s).

The primary value of the equation above is that it shows the sensitivities of accuracy to implementation parameters. Steady state CEP is most sensitive to the ratio  $\sigma_n / h$ , least sensitive to  $\delta V$ , and nominally sensitive to the time between profile measurements  $\Delta d / s$ . Using typical values of;

$$\delta V = 1 \text{ m/s (1 nm/hr-class INS)}$$

$$s = 250 \text{ m/s}$$

$$\Delta d = 100 \text{ m}$$

$$h = 0.05 \text{ (moderately rough terrain)}$$

$$\sigma_n = 15 \text{ m}$$

results in a  $CEP_{ss}$  of 29 m. Because of the assumptions in the equation given above, predictions should be treated as approximations, a conservative lower bound for TAN accuracy [13].

As it can be seen from linear TAN estimation results, INS error growth in time is limited using TAN algorithms. By correlating terrain profiles with INS solutions a few times during operation or recursively, position estimates are obtained. Then, INS is updated according to the estimated navigation solutions.

## 2.2. TERCOM

### 2.2.1. Background

Terrain Contour Matching (TERCOM) can be defined as a technique for determination of the position location of an airborne vehicle with respect to the terrain over which the vehicle is flying. More specifically, TERCOM is a form of correlation guidance based on a comparison between the measured and the pre-stored features of the profile of the ground (i.e., terrain) over which a missile or aircraft is flying. Generally, terrain height forms the basis of this comparison. Reference terrain elevation source data descriptive of the relative elevations of the terrain in the fix point areas are stored in the air vehicle's onboard computer.

Obtaining the reference data requires prior measurement of the ground contours of interest. These data are in the form of a horizontally arranged matrix of digital elevation numbers. A given set of these numbers describes a terrain profile. The length of contour profile necessary for a unique fit is a function of terrain roughness, but is in the range of 6 to 10 km and can be a curved path [37].

As the vehicle flies over the matrix area, data describing the actual terrain profile beneath the vehicle are acquired. That is, the actual profile is acquired using a combination of radar and barometric altimeter outputs sampled at specific intervals, and when compared against the stored matrix profiles provide the position location. This type of guidance is used for updating a mid-course guidance system on a periodic basis, and has been applied to the guidance of cruise missiles, which usually fly at subsonic speeds and fairly constant altitude. With regard to mid-course guidance, it is well known that the simplest mid-course guidance is the explicit guidance method. The guidance algorithm has the capability to guide the missile to a desired point in the air while controlling the approach angle and minimizing an appropriate cost function. Furthermore, the guidance gains of the explicit guidance law are usually selected to shape the trajectory for the desired conditions [37].

The TERCOM technique, first patented in 1958, relies for its operating principle on the simple fact that the altitude of the ground above sea level varies as a function of location. Historically, TERCOM has evolved from several R&D programs that developed certain areas of the overall process. These programs perfected the technology as it is known today [37]. In Table 6, a chronological overview of this development is summarized.

In the following sections, TERCOM method will be investigated in detail. Fundamentals of the TERCOM concept are taken from Siouris [37]. At the end of TERCOM section, simulations performed will be discussed.

Table 6. Chronological Overview of TERCOM Development [37]

PROGRAM	YEAR	OBJECTIVES
Fingerprint	1958	Guidance package for SLAM missile TERCOM concept first proposed.
TERCOM	1960-1961	Feasibility study of terrain contour matching.
LACOM (Low Altitude Contour Matching)	1963-1965	Design and development of a complete fix-taking subsystem.
RACOM (Rapid Contour Matching)	1963-1966	Improve TERCOM computation procedures and increase accuracy.
SAMSO (USAF's Space and Missiles Systems Organization) Programs (a) TPLS (Terminal Position Location System) (b) TERSE (Terminal Sensing Experiment) (c) TERF (Terminal Fix). (d) TSOFT (Terminal Sensor Overland Flight Test).	1963-1971	Application of terrain correlation techniques for ballistic missiles.
Avionics Update	1972-1975	Study and define a TERCOMI drone system capable of operational deployment.
TAINS (Terrain Aided INS) TERCOM	1972-1974	Feasibility study for incorporation in cruise missile and evaluation of snow coverage effects on terrain profile acquisition.
Competitive Flyoff	1975	McDonnell – Douglas Astrodynamics awarded a contract for TERCOM system.
RACOM (Recursive All Weather Contour Matching)	1975	Improve terrain correlation update accuracy.

### 2.2.2. TERCOM Concept

TERCOM system uses an airborne altimeter and a data processor to correlate the measured terrain contours to obtain the best estimate of position. The TERCOM system relies on a set of digital maps stored in the memory of the missile's onboard computer. These maps consist of rectangular arrays of numbered squares representing the variation of ground elevation above sea level as a function of location. Consequently, as the missile approaches an area for which the computer memory has a map, the onboard radar altimeter starts providing a stream of ground elevation data. Furthermore, the computer, by comparing these data with the information it has in its memory, can accurately determine the actual trajectory of the missile and instruct the autopilot to return the missile to its planned trajectory. Four such corrective maneuvers are shown in the vertical overhead view in Figure 12 [37].

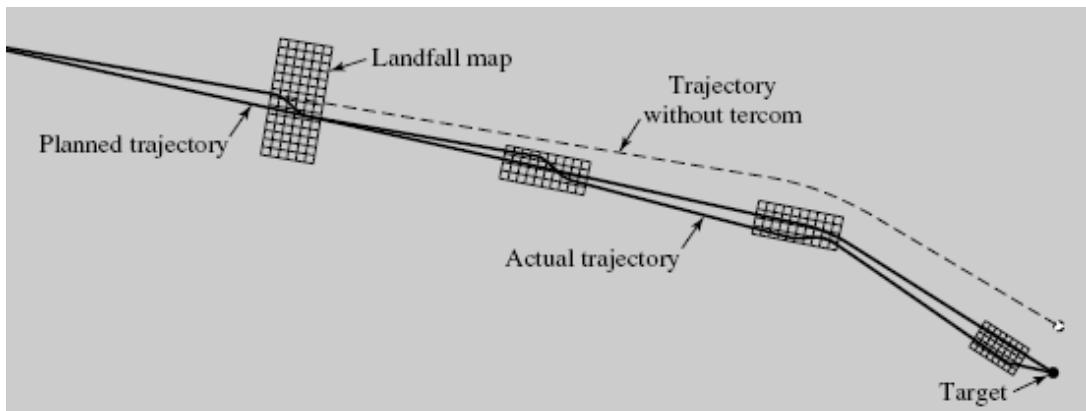


Figure 12. TERCOM Maps in Use [37]

The map types used in TERCOM differ in length, width, and cell size. The cell size determines, in part, the accuracy of the TERCOM fix. The TERCOM maps

become smaller and are spaced closer together as the missile approaches the target. As a result, because of the decreasing cell size, the updates become more accurate. A terminal accuracy on the order of 100 meters (i.e. DTED Level 1) is considered feasible for the TERCOM system [37].

The process of determining air vehicle position by the use of terrain contour matching can generally be described as consisting of three basic steps; data preparation, data acquisition, and data correlation. In Figure 13, TERCOM concept is illustrated [37].

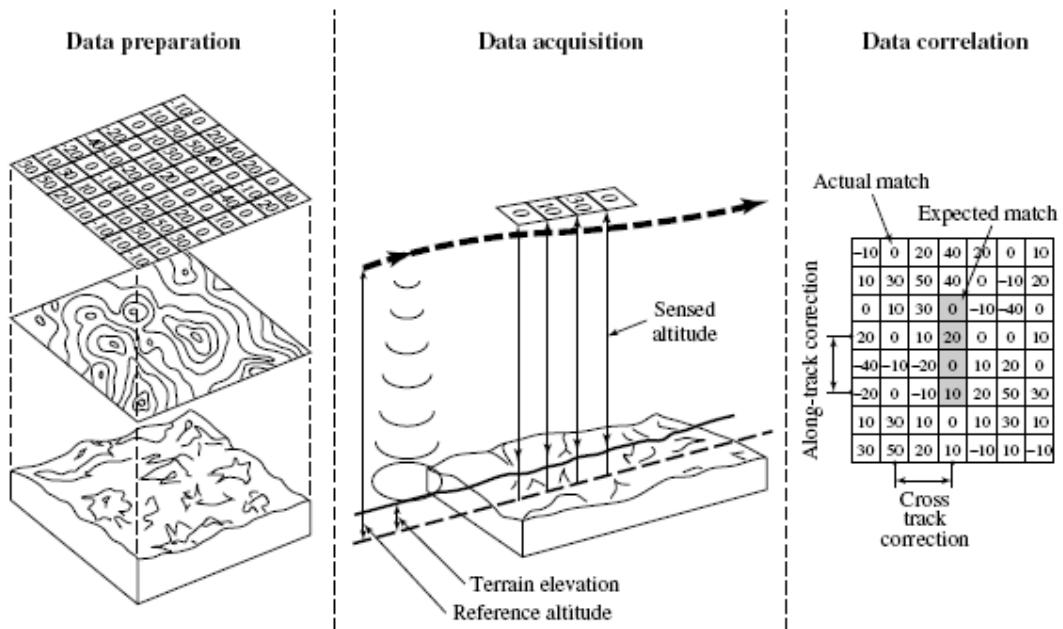


Figure 13. TERCOM Concept [37]

The critical part of TERCOM process is the data correlation where navigation solution is performed. There are several data correlation algorithms like MAD and MSD and they will be discussed in the following section.

TERCOM measurement process is illustrated in Figure 14 in block diagram form. The radar altimeter acquires altitude estimates above terrain. Then, the radar altimeter output is differenced with the system's reference altitude. Various arithmetic operations (e.g. mean removal and quantization) are then performed on the differenced data. Finally, the correlation between the stored and acquired data is performed with the MAD function, and a position fix is determined [37].

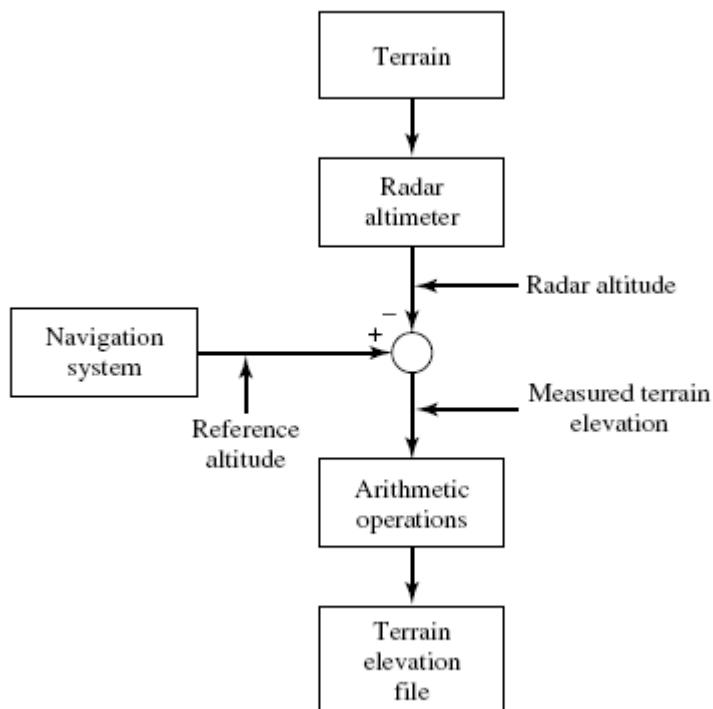


Figure 14. TERCOM Measurements [37]

### 2.2.3. TERCOM Data Correlation Techniques

There are a number of correlation algorithms (e.g., mean squared difference (MSD), mean absolute difference (MAD), the normalized MAD, the normalized

MSD, and the product method) of varying complexity used in TERCOM. Furthermore, the MAD algorithm provides the best combination of accuracy and computational efficiency for performing real-time terrain contour matching in an onboard computer environment. Therefore, here only the MAD and MSD correlation algorithms will be discussed [37].

The MAD algorithm is applied considering the first  $N$  height differences to be acquired. Then, these differences are removed, so that the sample profile is its mean value. Next, this profile is compared with each row of matrix data in the following manner. Let  $h_n$  ( $1 \leq n \leq N$ ) denote any row of matrix data and  $H_n$  the sequence of required data. Consequently, the MAD algorithm, which is used for correlating the measured terrain elevation file with each down-track column of the reference matrix, is defined as follows [37]:

$$MAD_{k,m} = (1/N) \sum_{i=1}^N |h_{k,m} - H_{m,n}| \quad (2.8)$$

where;

$MAD_{k,m}$ : The value of the mean absolute difference between the  $k$ 'th terrain elevation file and the  $m$ 'th reference matrix column,

$N$ : The number of samples in the measured terrain elevation file and usually it is also equal to the number of rows in the reference matrix,

$M$ : The number of reference matrix columns,

$K$ : The number of measured terrain elevation files used in the correlation process,

$| | =$  The absolute value of the argument,

$n, m, k :$  Row, column, and terrain elevation file indices,

$H_{m,n} :$  The stored reference matrix data,  $1 \leq m \leq M, 1 \leq n \leq N,$

$h_{k,m} :$  The  $k$ 'th measured terrain elevation file,  $1 \leq k \leq K.$

The MSD algorithm can be expressed in terms of the profile in question. Mathematically, the expression for MSD is [37],

$$MSD_{jk} = (1/N) \sum_{i=1}^N (S_{ij} - S_{ik})^2 \quad (2.9)$$

where,

$S_j, S_k :$   $j$ 'th and  $k$ 'th profiles,

$N :$  Length of each profile.

Note that for uniformity, the MAD algorithm can also be expressed as in the expression for the MSD. Thus,

$$MAD_{jk} = (1/N) \sum_{i=1}^N |S_{ij} - S_{ik}| \quad (2.10)$$

Examination of the expressions for the MAD and MSD processors indicates that both of these correlators can be viewed as distance measures, where the dimensions of the space for which these distances are defined correspond to the number of elements in the profiles. From (2.9) and (2.10), it is noted that the ambiguity between any two profiles is defined as the probability “ $P$ ” that sensed data corresponding to one of the profiles will be closer (in terms of the distance measure) to the other profile than to the one from which it was taken [37].

Mathematically, the ambiguity  $\xi$  can be expressed as:

$$\xi_{jk} = \begin{cases} P[C_{jk} < C_{jj}], \text{ where a minimum of } C_{jk} \text{ is sought,} \\ P[C_{jk} > C_{jj}], \text{ where a maximum of } C_{jk} \text{ is sought.} \end{cases} \quad (2.11)$$

For a MAD processor,  $C_{jk}$  is given by the following expression:

$$C_{jk} = (1/N) \sum_{i=1}^N |S_{ij} - R_{ik}| \quad (2.12)$$

where,

$S_j$ : j'th measured profile,

$R_k$ : k'th reference profile.

A more detailed account of the terrain correlation processing for a single map is conceptually shown in Figure 15.

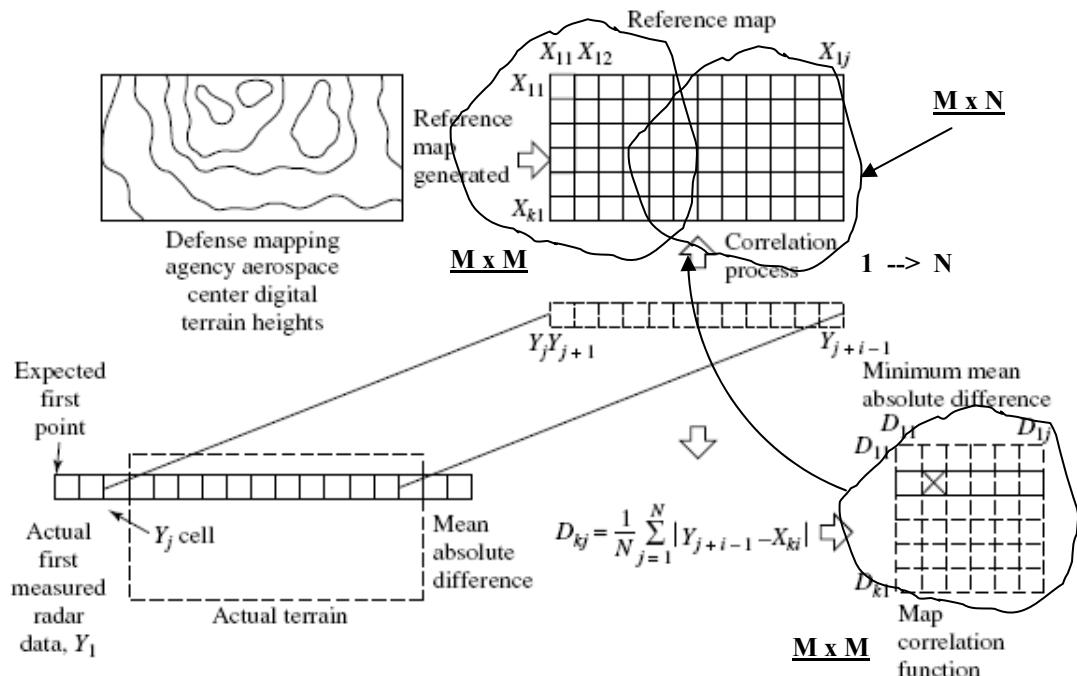


Figure 15. Terrain Correlation Processing [37]

#### 2.2.4. Terrain Roughness Characteristics

For TERCOM correlation process, roughness and uniqueness of the selected terrain is very critical. It should be noted that the TERCOM concept will not work over all types of terrain. For instance, the rougher the terrain, the better TERCOM works. However, good terrain must be more than just rough, it must be unique (i.e., a given profile out of the TERCOM map must not resemble any other map [37]).

Terrain roughness is defined as the standard deviation of the terrain elevation samples as shown in Figure 16. It is usually referred to as “sigma-T” (or  $\sigma_T$ ) [37].

Sigma- T is defined by the equation:

$$\sigma_T = \sqrt{(1/N) \sum_{i=1}^N (H_i - \bar{H})^2} \quad (2.13)$$

where,

$$\bar{H} = (1/N) \sum_{i=1}^N H_i : \text{ Mean Elevation}$$

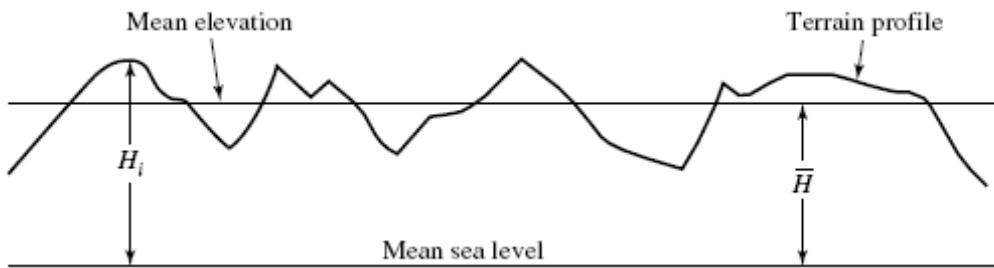


Figure 16. Terrain Standard Deviation (Sigma-T) [37]

Thus,  $\sigma_T$  is a measure of the variation of the terrain elevation about its average elevation. Note that the minimum value of  $\sigma_T$  required to support TERCOM operation is approximately 25 ft (7.62 m). Areas that have sigma-T values of fifty or greater are usually considered as good candidates for TERCOM fix areas. Obviously, lakes and very flat or smooth areas have low values of sigma-T. Therefore, they are not suitable as fix areas. However, sigma-T is not the only criterion for determining whether a given area is suitable for TERCOM operation [37].

In particular, there are three parameters that are used to describe TERCOM-related terrain, and their values can give an indication of the terrain's ability to support a successful TERCOM fix. These parameters are sigma-T, sigma-Z ( $\sigma_Z$ ),

and the terrain correlation length ( $X_T$ ). It is usually assumed that parallel terrain elevation profiles that are separated by a distance greater than  $X_T$  are independent of each other [37].

Sigma-Z is defined as the standard deviation of the point-to-point changes in terrain elevation (i.e., the slope) as shown in Figure 17. Like sigma-T, the value of sigma-Z provides a direct indication of terrain roughness. Sigma-Z has also been shown to be a valid indicator of TERCOM performance. The expression for sigma-Z, assuming a Gaussian autocorrelation function, can be obtained from Figure 17. Mathematically, sigma-Z is given by the equation [37]:

$$\sigma_Z = \sqrt{[1/(N-1)] \sum_{i=1}^N (D_i - D)^2} \quad (2.14)$$

where,

$$D_i = H_i - H_{i+1}$$

$$D = (1/(N-1)) \sum_{i=1}^{N-1} D_i$$

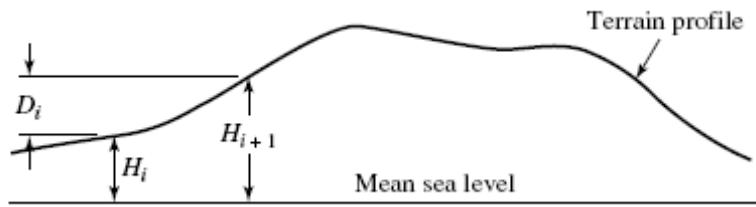


Figure 17. Definition of Sigma-Z [37]

The two parameters sigma-T and sigma-Z are related to the third parameter  $X_T$  according to the relation [37]:

$$\sigma_z^2 = 2 \cdot \sigma_T^2 \cdot [1 - \exp(-\Delta d / X_T)^2] \quad (2.15)$$

where;

$\Delta d$ : Cell size (or distance between elevation samples).

### 2.2.5. Simulations and Discussion

In order to investigate TERCOM performance, a simulation model is developed with Matlab [60]. The sample map considered for the simulations has the size of M=21 by N=100 where the cells are 100 x 100 meters approximately. The TERCOM procedure is as follows:

1. Map is selected considering CEP of the INS.
2. M x M (21 x 21) for N=1 is considered.
3. Height measurements are considered then.
4. Absolute differences,  $|h_{l,m} - H_{m,n}|$  n= 1 to M and m= 1 to M are calculated for MAD process. (21 x 21 operations)
5. Square differences,  $(h_{l,m} - H_{m,n})^2$  n= 1 to M and m= 1 to M are calculated for MSD process. (21 x 21 operations)

6. Steps “1” to “3” are repeated for N=1 to 100. (21 x 21 x 100 operations)
7.  $MAD_{k,m} = (1/N) \sum_{i=1}^N |h_{k,m} - H_{m,n}|$ , is calculated for MAD process.  
(Extra sum and averaging operations for 21 x 21 x 100 elements)
8.  $MSD_{jk} = (1/N) \sum_{i=1}^N (S_{ij} - S_{ik})^2$ , is calculated for MSD process. (Extra sum and averaging operations for 21 x 21 x 100 elements)
9. Minimum of MAD and MSD functions are sought in order to determine the indices “i” and “j” of the horizontal position fixes for both MAD and MSD processes. (Determination of the minimum points)

For the simulations, DTED Level 1 data were required and they have been obtained from HGK. The properties of DTED prepared for Turkey were given in Table 3 [16]. Horizontal accuracy of Level 1 DTED is defined as  $\pm 130$  m, and vertical accuracy as  $\pm 30$  m. In fact, especially horizontal accuracy of the DTED Level 1 data for Turkey is not sufficient for navigation purposes. Actually, DTED Level 2 data which have horizontal accuracy of  $\pm 26$  m can be resampled to DTED Level 1 and used for practical applications. However, for the Ph.D. study, DTED Level 1 data are used considering the horizontal accuracies of DTED Level 2.

In order to perform the simulations, first selection of the areas is performed using the mapping software OziExplorer [61]. In order to select the areas for simulation, roughness of the surfaces is investigated using elevation property of the software which uses DTED Level 1 files obtained from HGK. Sample area selection using OziExplorer is shown in Figure 18.

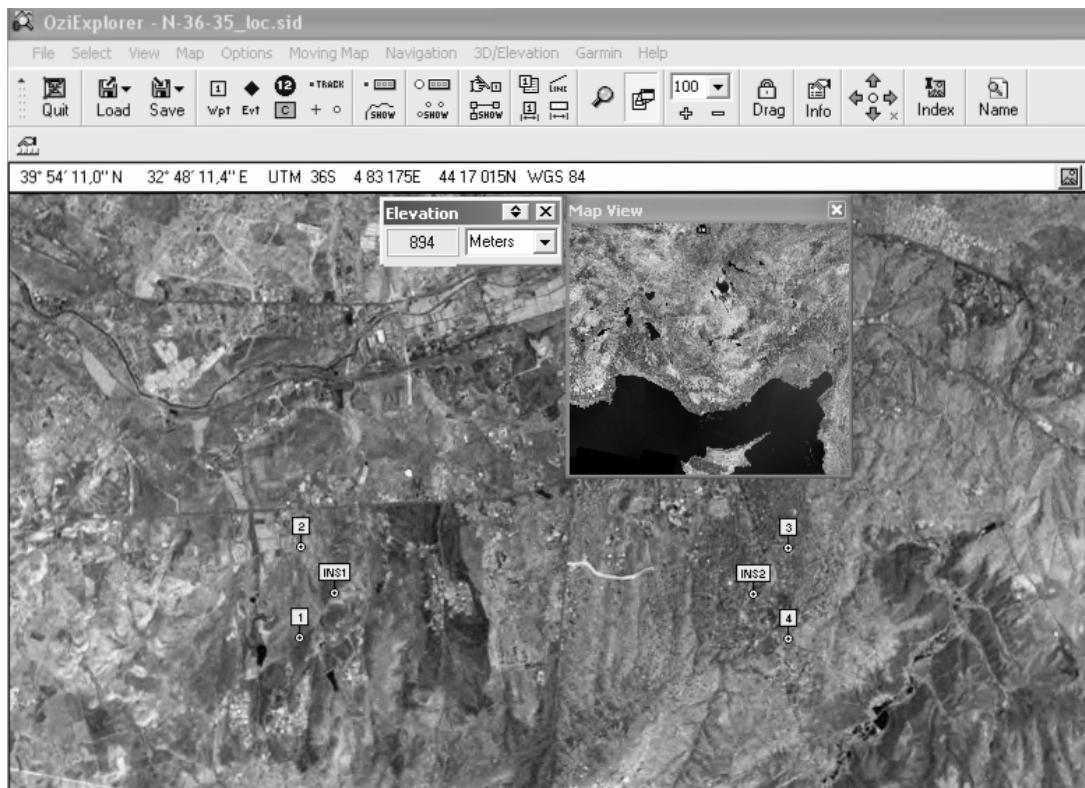


Figure 18. Sample Area Selection from OziExplorer Software [61]

Then, using OziExplorer3D [62] software, optional add-on to the OziExplorer software which allows map images to be viewed in 3D, selected areas were rendered as matrix grids. For the simulations, three special areas were selected:

1. Area with rough surface,
2. Area with smooth surface,
3. Area with having uniqueness (i.e. a single mountain).

Selected areas rendered using OziExplorer3D are shown in Figure 19 and Figure 20.

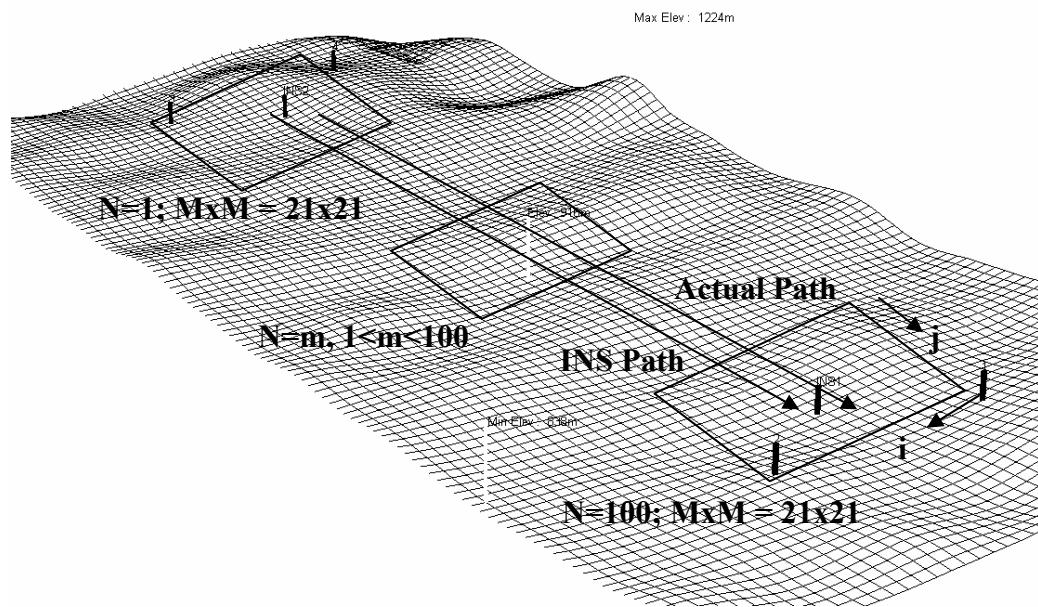


Figure 19. Rendered Rough Surface Area

As it can be seen from Figure 19,  $M \times N$  area is totally concerned for the TERCOM process. Therefore, at least  $M \times M \times N$  calculations are required. Actually, it is obvious that “ $M$ ” depends on the accuracy of the INS. If the INS quality is worse, the area considered should be larger (i.e. large  $M$ ). On the other hand, TERCOM algorithms (both MAD and MSD) are simple and straight forward. However, unnecessary calculations can be performed during the process since the whole area is concerned.

Next, the matrix cells are formed as seen in the figures above considering SSLM (short sample long matrix) map selection method for TERCOM [37]. The required sample size for the along track is approximately 7.78 km. For the

simulations, it is selected nearly 10 km depending on the latitude of the area (Actually the numbers of the cells are taken to be constant.) Cross track errors depend on the accuracy of the INS. Depending on the typical 1 nmi/hr class INS which is generally used for cruise missiles, cross track sample size is selected approximately 1.85 km (1 nmi) assuming the worst case for cross track errors.

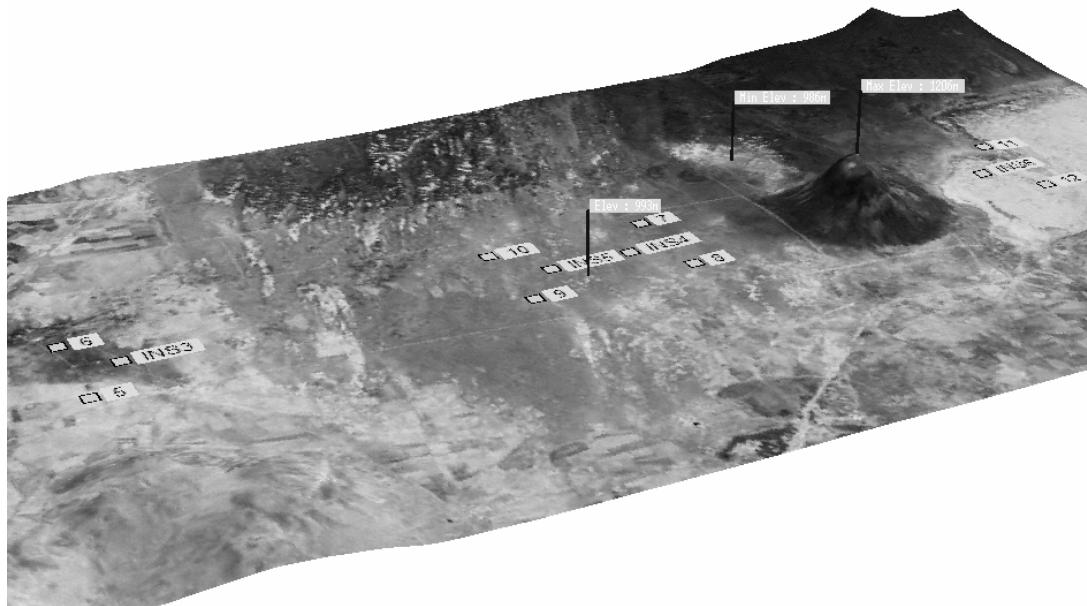


Figure 20. Rendered Smooth Surface Area and Area with Uniqueness

On the other hand, velocity of the cruise missile is considered to be constant for the simulations moving from west to east direction (or vice versa) in order to investigate TERCOM concept. Before performing TERCOM algorithms, sigma-T and sigma-Z values are calculated in order to validate the roughness of the surfaces selected.

First, considering perfect measurement and no error sources, selected profiles were determined. Here, the ambiguity term,  $\xi$  in equation (2.11) exactly

becomes zero as expected. Then, errors are added to the selected profile measurement values considering white noise.

Monte Carlo simulations of 100 runs are performed for TERCOM simulations. Since, TERCOM is a batch process; true position fixes for navigation solutions are sought. Simulation results are given in Table 7 to Table 9 for different terrain types.

Table 7. TERCOM Simulation Results for Rough Terrain

Initial INS Position Error (One axis, approximate)	400 m	
Height Measurement Standard Deviation (One sigma)	10 m	
TERCOM Map Grid Size (MxM)	21x21	
Number of Height Measurements for Correlation (N)	100	
Sigma-T of the Area Concerned	47.07 m	
Sigma-Z of the Area Concerned	1.88 m	
Correlation Method	MAD	MSD
Percentage of False Fix	6 %	9 %
Maximum False Fix Error (Total approximate error)	200 m	200 m

Table 8. TERCOM Simulation Results for Smooth Terrain

Initial INS Position Error (One axis, approximate)	400 m	
Height Measurement Standard Deviation (One sigma)	10 m	
TERCOM Map Grid Size (MxM)	21x21	
Number of Height Measurements for Correlation (N)	100	
Sigma-T of the Area Concerned	7.22 m	
Sigma-Z of the Area Concerned	0.38 m	
Correlation Method	MAD	MSD
Percentage of False Fix	100 %	100 %

Table 9. TERCOM Simulation Results for Terrain with Uniqueness

Initial INS Position Error (One axis, approximate)	400 m	
Height Measurement Standard Deviation (One sigma)	10 m	
TERCOM Map Grid Size (MxM)	21x21	
Number of Height Measurements for Correlation (N)	100	
Sigma-T of the Area Concerned	46.4	
Sigma-Z of the Area Concerned	6.37	
Correlation Method	MAD	MSD
Percentage of False Fix	1 %	1 %
Maximum False Fix Error (Total approximate error)	150 m	150 m

From the results, it is seen that best position fix results are obtained with terrain with uniqueness. However, it should be noted that the critical parameter is the sigma-Z value of the area concerned where standard deviation of the point-to-point changes in terrain elevation (i.e., the slope) are calculated. Eventhough, rough surface has larger sigma-T (standard deviation of height of the area) value, having a larger value of sigma-Z terrain with uniqueness gives better correlation results than rough surface. Moreover, for smooth terrain, correlation algorithms do not give position fixes as expected.

For the TERCOM process, several conclusions are achieved from the concept study and simulations performed. They are summarized as follows:

1. Navigation solutions can be obtained for rough and unique surfaces as expected.
2. Correlation algorithm is simple but not smart. Many calculations should be performed in order to have a position fix.

3. It is thought that the algorithm was derived considering the capability of the computers of 1950's, performing only matrix calculations and simple mathematical operations.
4. Physical meaning of MAD and MSD processes is the minimization of the area difference between the measured and the reference areas along the route of the missile.
5. In the simulations, it was shown that MAD process shows better position fix than MSD process. For a terrain with small terrain height changes, MSD process neglects the small height difference terms and exaggerate the larger height difference terms. On the other hand, in MAD process absolute height difference terms are taken into account with same weights.
6. The critical parameter for best terrain correlation is sigma-Z value of the area concerned where standard deviation of the point-to-point changes in terrain elevation (i.e. the slope) are calculated instead of sigma-T value where standard deviation of height of the area is calculated. In other words, the slopes of the area concerned are more critical than the roughness of the area for correlation.
7. TERCOM process is independent of the target model where cruise missile is the target. Possible tracks for the missile are selected where tracks are the missile path formed by the terrain elevation file (DTED). Since, the target motion is not modeled, kinematical behavior of the system is not known.
8. TERCOM process is actually a Maximum Likelihood Estimator (MLE) which uses "Least Squares Estimation (LSE)" technique.

Here, minimum error of the height measurements are sought for position fixes using least squares (LS) estimation:

$$z(j) = h(j, x) + \varepsilon(j) \quad j = 1, \dots, k \quad (2.16)$$

$$\hat{x}^{LS}(k) = \arg_{x \in X} \min \sum_{j=1}^k [z(j) - h(j, x)]^T [z(j) - h(j, x)] \quad (2.17)$$

where;

$z(j)$ : Measurement of the terrain taken at “j”

$h(j, x)$ : DTED value of the related points with respect to taken measurement,  $z(j)$

$\hat{x}^{LS}(k)$ : Minimum “MSD value times k” of the terrain height differences

## 2.3. SITAN

### 2.3.1. SITAN Fundamentals

As it was stated in the first chapter, the major recursive TAN algorithm found in literature is SITAN which is proposed by Hostetler and Andreas [12]. In order to investigate SITAN in detail, first original work of Hostetler and Andreas [12] will be investigated in detail.

The basic configuration for optimal terrain aided navigation is shown in Figure 21. This structure is typical of Kalman filtering in which nonlinear auxiliary

measurements are iteratively processed to estimate and compensate for the errors in a navigation system. At each measurement update time the current state estimate in conjunction with stored topographical data (i.e. terrain elevation data), is used to obtain a prediction of what the radar ground clearance measurement should be. The actual radar measurement is then compared with this predicted measurement, and their difference is processed by the Kalman filter to generate estimates of the navigation system's error states. The measurement matrix in this case is related to the downrange and cross range terrain slopes calculated from the stored data. The error estimates are then fed back to compensate the navigation system and thus provide an improved estimate of the actual state (position, velocity, etc.) of the system. This process is iterated many times e.g. every 30-50 m of distance traveled, as the system maneuvers along its trajectory, thus providing essentially continuous updating to the navigation system [12].

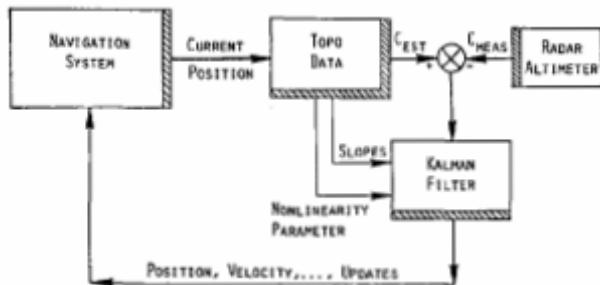


Figure 21. SITAN Process [12]

System equations for extended Kalman filtering (EKF) are derived considering navigation equations. True navigation state vector  $\bar{x}$  is defined as [12]:

$$\bar{x} = \begin{bmatrix} x \\ y \\ h \\ v_x \\ v_y \end{bmatrix} \quad (2.18)$$

where;

$x$ : Horizontal coordinates along eastward direction,

$y$ : Horizontal coordinates along northward direction,

$h$ : Height above sea level,

$v_x$ : Velocity along  $x$  direction,

$v_y$ : Velocity along  $y$  direction.

Let  $\tilde{x}$  be the measured state vector for  $\bar{x}$  from INS with the help of barometric altimeter,  $\hat{x}$  the estimated state vector for  $\bar{x}$  after updating,  $\delta\hat{x}$  the optimal estimation of error vector  $\delta\bar{x}$  for  $\bar{x}$  from the outputs of Kalman filter.

For a constant sampling period  $T$ , the recursion error state vector equation is as [12]:

$$\delta\bar{x}(k+1) = \Phi(k) \cdot \delta\bar{x}(k) + \bar{w}(k) \quad (2.19)$$

where;

$$\delta \bar{x} = \begin{bmatrix} \delta x \\ \delta y \\ \delta h \\ \delta v_x \\ \delta v_y \end{bmatrix} : \quad \text{Error state vector,}$$

$$\Phi(k) = \begin{bmatrix} 1 & 0 & 0 & T & 0 \\ 0 & 1 & 0 & 0 & T \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} : \quad \text{Transition matrix,}$$

$$\bar{w}(k) = \begin{bmatrix} w_x(k) \\ w_y(k) \\ w_h(k) \\ w_{v_x}(k) \\ w_{v_y}(k) \end{bmatrix} : \quad \text{Process noise vector (White noises).}$$

In order to implement EKF, 1-D measurement  $\delta h$  is needed, which is the difference between estimated relative height  $C_{est}$ , and measured relative height  $C_{meas}$ .  $C_{meas}$  comes from the measurement of radar altimeter;  $C_{est}$  is the difference between estimated height above sea level from barometric altimeter (or INS),  $\hat{h}_{baro}$  and terrain height  $\hat{h}_{DTED}$  from digital terrain elevation data based on the estimated position of  $(\hat{x}, \hat{y})$  from INS. Thus  $\delta h$  is expressed by [12]:

$$\delta h = C_{est} - C_{meas} \quad (2.20)$$

where;

$$C_{est} = \hat{h}_{baro} - \hat{h}_{DTED},$$

$\hat{h}_{baro}$ : Estimated height above sea level from barometric altimeter,

$$\hat{h}_{DTED} = \hat{h}_{ter} + \varepsilon_{DTED},$$

$\hat{h}_{ter}$ : DTED (Terrain) height at the estimated position  $(\hat{x}, \hat{y})$ ,

$\hat{h}_{ter}$ : Actual terrain height at the estimated position  $(\hat{x}, \hat{y})$ .

$$C_{meas} = h_{radar} + w_{radar},$$

$h_{radar}$ : Radar altimeter measurement at the actual position  $(x, y)$ ,

$w_{radar}$ : Radar altimeter white noise measurement error.

SITAN measurement process is shown in Figure 22.

Now, expand terrain height difference measurement given in equation (2.20).

$$\delta h = C_{est} - C_{meas} = [h_{baro}(\hat{x}, \hat{y}) - (h_{ter}(\hat{x}, \hat{y}) + \varepsilon_{DTED})] - [h_{radar}(x, y) + w_{radar}] \quad (2.21)$$

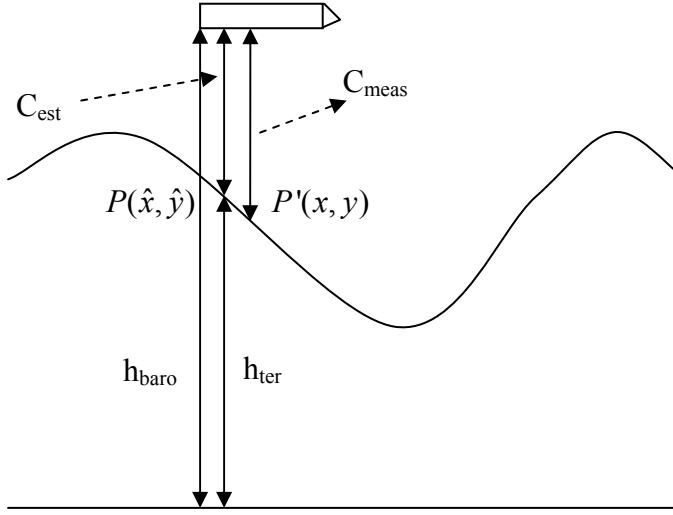


Figure 22. SITAN Measurement Process

Here, using Taylor series expansion, actual position is assumed to be near the estimated position. Therefore,

$$h_{baro}(\hat{x}, \hat{y}) = h_{baro}(x, y) + \delta h_{baro} \quad (2.22)$$

Then,

$$\delta h = [h_{baro}(x, y) + \delta h_{baro}] - [h_{ter}(\hat{x}, \hat{y}) + \varepsilon_{DTED}] - [h_{radar}(x, y) + w_{radar}] \quad (2.23)$$

The correlation between the estimated and the actual positions is the key point of the SITAN process [12]. Consider a fitted function  $f(x, y)$  to the terrain profile being expanded near  $(\hat{x}, \hat{y})$  as shown in Figure 23. Then,

$$f(x, y) = f(\hat{x}, \hat{y}) + \frac{\partial f}{\partial x}(x - \hat{x}) + \frac{\partial f}{\partial y}(y - \hat{y}) \quad (2.24)$$

where;

$$h_x = \frac{\partial f}{\partial x} : \quad \text{Terrain slopes along eastward direction,}$$

$$h_y = \frac{\partial f}{\partial y} : \quad \text{Terrain slopes along northward direction,}$$

$$\delta x = \hat{x} - x ,$$

$$\delta y = \hat{y} - y .$$

Here, terrain profiles  $h_x$  and  $h_y$  are needed for the EKF.

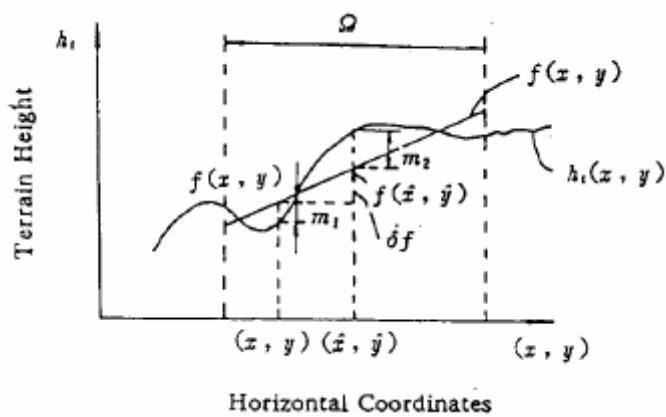


Figure 23. Terrain Stochastic Linearization (TSL) [12]

From the figure, following relations can be defined [12]:

$$f(\hat{x}, \hat{y}) = f(x, y) + \delta f = f(x, y) + h_x \cdot \delta x + h_y \cdot \delta y \quad (2.25)$$

$$h_{ter}(x, y) = f(x, y) + m_1 \quad (2.26)$$

$$h_{ter}(\hat{x}, \hat{y}) = f(\hat{x}, \hat{y}) + m_2 \quad (2.27)$$

where;

$m_1$ : TSL error at the actual point  $(x, y)$ ,

$m_2$ : TSL error at the estimated point  $(\hat{x}, \hat{y})$

Considering the actual position, following relation is valid [12]:

$$h_{baro}(x, y) - h_{ter}(x, y) - h_{radar}(x, y) = 0 \quad (2.28)$$

Then using equations (2.23), (2.25), (2.26), and (2.27),

$$\delta h = \delta h_{baro} - h_x \cdot \delta x - h_y \cdot \delta y - \underbrace{(\varepsilon_{DTED} + w_{radar} + m_1 + m_2)}_{w_{meas}} \quad (2.29)$$

Here, terrain linearization errors  $m_1$  and  $m_2$  are included in the measurement error. Moreover, DTED error  $\varepsilon_{DTED}$  was modeled as white. Finally system measurement equation in discrete form can be written as follows [12]:

$$z(k) = H(k) \cdot \delta\bar{x}(k) + w_{meas}(k) \quad (2.30)$$

where;

$$H(k) = \begin{bmatrix} -h_x & -h_y & 1 & 0 & 0 \end{bmatrix}: \text{ Measurement matrix,}$$

$$w_{meas}(k): \text{ Measurement white noise}$$

Using linearization methods to the nonlinear system equations, the system is linearized. Therefore, EKF is implemented. Following part is the application of the standard EKF equations. EKF equations for SITAN are presented in Table 10 considering standard EKF equations given in Gelb [63].

Implementation of the SITAN process is straight forward after the terrain slopes are modeled. However, the main problem of the process is the divergence of the KF. Due to highly nonlinear nature of terrain surfaces, filter divergence can occur especially when the linearization error is comparable to the measurement error. In these cases the standard EKF may yield unsatisfactory performance, and divergence can occur in which the actual estimation errors become orders of magnitude larger than the filter's own computation of their covariance [12]. Figure 24 demonstrates this phenomenon for a simulation test case in which the initial position error standard deviations were 75 m and all other conditions were the same as in the prior simulation.

Table 10. Kalman Filter Equations for SITAN Process

SITAN System Error Model:	
$\delta\bar{x}(k+1) = \Phi(k) \cdot \delta\bar{x}(k) + \bar{w}(k)$	given in (2.19)
$\bar{w}(k) = N(\bar{0}, Q(k))$	
System Noise Covariance Matrix:	
$Q(k) = Cov\{\bar{w}(k)\bar{w}(k)^T\}$	(2.31)
SITAN Measurement Model:	
$z(k) = H(k) \cdot \delta\bar{x}(k) + w_{meas}(k)$	given in (2.30)
$w_{meas}(k) = N(\bar{0}, R(k))$	
Measurement Noise Covariance Matrix:	
$R(k) = Cov\{\bar{w}_{meas}(k)\bar{w}_{meas}(k)^T\}$	(2.32)
Initial Conditions:	
$\delta\bar{x}_0 = N(\hat{\delta\bar{x}}_0, P_0)$	(2.33)
Other Assumptions:	
$E[\bar{w}(k) \cdot \bar{w}_{meas}(k)^T] = 0$ for all $k$ (Measurements are independent)	(2.34)
State Estimate Propagation:	
$\hat{\delta\bar{x}}(k   k-1) = \Phi(k-1) \cdot \hat{\delta\bar{x}}(k-1   k-1)$	(2.35)
Error Covariance Propagation:	
$P(k   k-1) = \Phi(k-1) \cdot P(k-1   k-1) \cdot \Phi(k-1)^T + Q(k-1)$	(2.36)
Gain Matrix:	
$K(k) = P(k   k-1) \cdot H(k)^T \cdot [H(k) \cdot P(k   k-1) \cdot H(k)^T + R(k)]^{-1}$	(2.37)
State Estimate Update:	
$\hat{\delta\bar{x}}(k   k) = \hat{\delta\bar{x}}(k   k-1) + K(k) \cdot [z(k) - H(k) \cdot \hat{\delta\bar{x}}(k   k-1)]$	(2.38)
Error Covariance Update:	
$P(k   k) = [I - K(k) \cdot H(k)] \cdot P(k   k-1)$	(2.39)

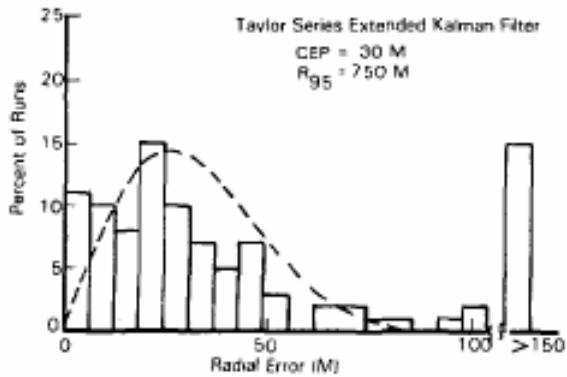


Figure 24. Standard EKF Divergence Problem [12]

In order to improve EKF performance, modified stochastic linearization approach is used. However, single EKF for large errors actually can not perform good results. Therefore, parallel Kalman filters are used in order to estimate large position errors, especially large initial errors as shown in Figure 25. After initial errors are estimated within the accepted CEP values (i.e.  $\sim 30$  m), single KF becomes sufficient for navigation purposes.

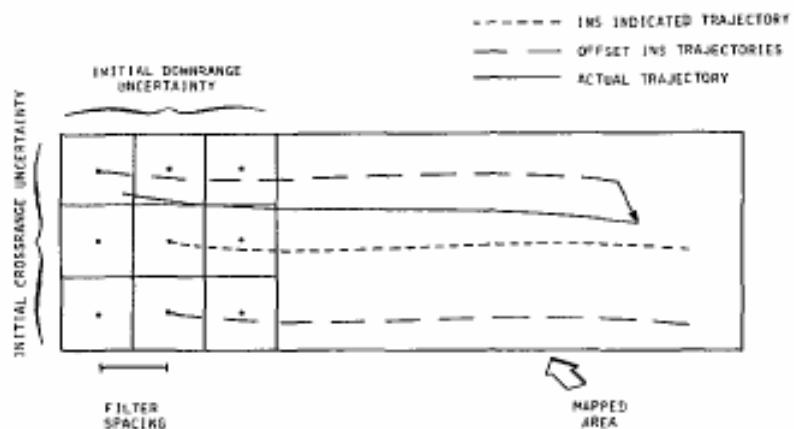


Figure 25. Parallel KF Configuration [12]

The selection of the convergent filter can be done quite easily by examining the residuals (i.e. estimated minus measured values of height)  $\Delta_i$  for each filter. A selection algorithm based upon the assumed whiteness property of the filter residuals that worked well in practice is to choose the filter with the smallest value of [12];

$$AWRS_{j\text{th filter}} = \frac{1}{N} \left[ \sum_{i=1}^N \frac{\Delta_i}{H_i P_i H_i^T + R_i} \right]_{j\text{th filter}} \quad (2.40)$$

where;

$AWRS_{j\text{th filter}}$  : Average Weighted Residual Squared of the j'th filter,

$H_i$  : Measurement vector containing the terrain slopes at the i'th time interval,

$P_i$  : Error covariance matrix,

$R_i$  : Measurement noise covariance matrix,

$N$  : Number of measurements processed,

This AWRS value is the average weighted residual squared between the predicted ground clearance for each filter and the ground clearance measured by the radar altimeter for each time  $t_i$ . The weighting factor is inherently calculated by each Kalman filter and is simply the expected variance of  $\Delta_i$  at each measurement. By examining the minimum AWRS values for each filter after a sufficiently large

number of measurements have been processed, the correct filter and its associated state error estimates can be chosen.

### 2.3.2. Simulations and Discussion

Simulations for SITAN are performed for both tracking and acquisition modes. In order to perform simulations, Simulink [58] is used. Mathematical models described in the previous section are used for trajectory and INS models in order to obtain 1.0 nm/hr INS quality by adding white noise terms to horizontal positions, altitude and horizontal velocities. Terrain slopes are derived considering the gradients of the height values of the related DTED files.

For the simulations, three special terrain types are selected:

1. Rough terrain,
2. Smooth terrain,
3. Mountainous terrain.

Some properties of these selected terrains for TAN are given in Table 11. It should be noted that, these properties satisfy terrain requirements for the simulations.

Simulation model details will be presented in the following section. In this section, SITAN characteristic simulation results will be presented. First, horizontal position errors for tracking mode are performed for three different terrain types. Simulation parameters for tracking mode are given in Table 12. Here, it should be

noted that initial position error is less than the grid size of the DTED considered (i.e. less than 100 meters for DTED Level 1).

Table 11. Terrain Parameters for SITAN Simulations

Terrain Type	Rough	Smooth	Mountainous
Mean height of the terrain profile	1093 m	1104 m	1177 m
Sigma-T	77.9 m	34.1 m	212.9 m
Sigma-Z	16.3 m	3.7 m	23.1 m
$X_T$	670.2 m	1309 m	1302 m

Table 12. SITAN Simulation Parameters for Tracking Mode

Initial INS position deviation (one axis)	80 m
Initial vehicle velocity	240 m/s
Initial INS east velocity bias	0.5 m/s
Initial INS north velocity bias	0.5 m/s
INS horizontal position standard deviation	5 m
INS altitude position standard deviation	3 m
Radar altimeter standard deviation	3 m
INS velocity standard deviation	0.3 m/s

SITAN filter works at 1 Hz. In other words, it gives updates at every 1 second. Simulations are performed for 100 seconds of operation time. In actual systems, INS is updated recursively considering SITAN position corrections. Hence, INS errors become zero at discrete SITAN updates. However, in the simulations, in order to show SITAN characteristics, INS error model is not

updated; and only first 100 seconds of operation is considered. Simulation results of tracking mode for different terrain types are shown from Figure 26 to Figure 31.

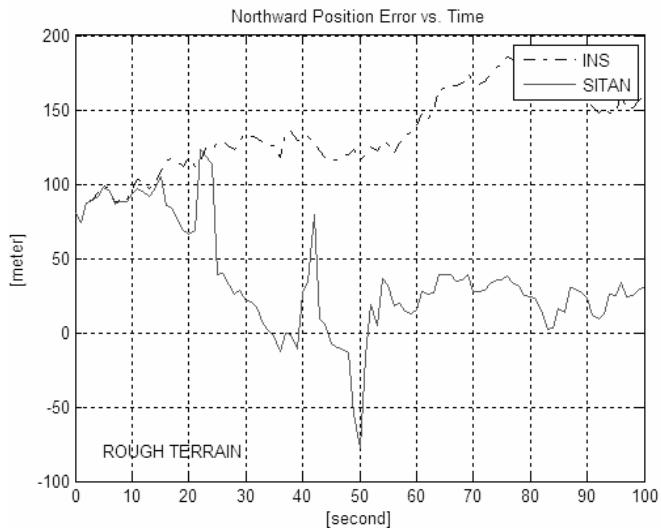


Figure 26. Rough Terrain Northward Position Error vs. Time

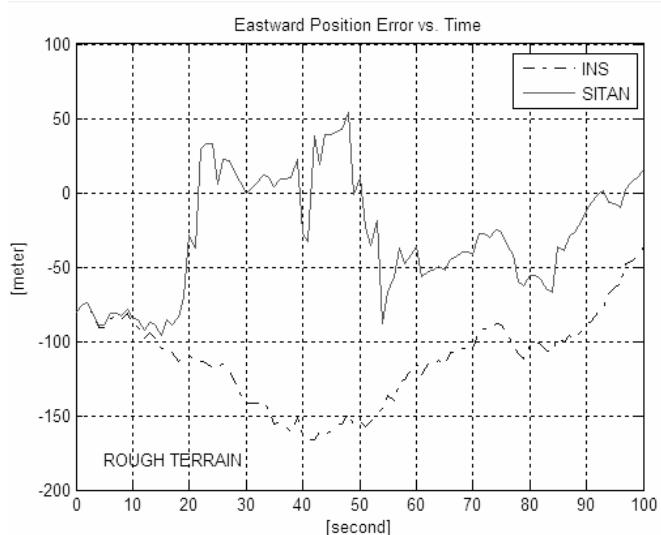


Figure 27. Rough Terrain Eastward Position Error vs. Time

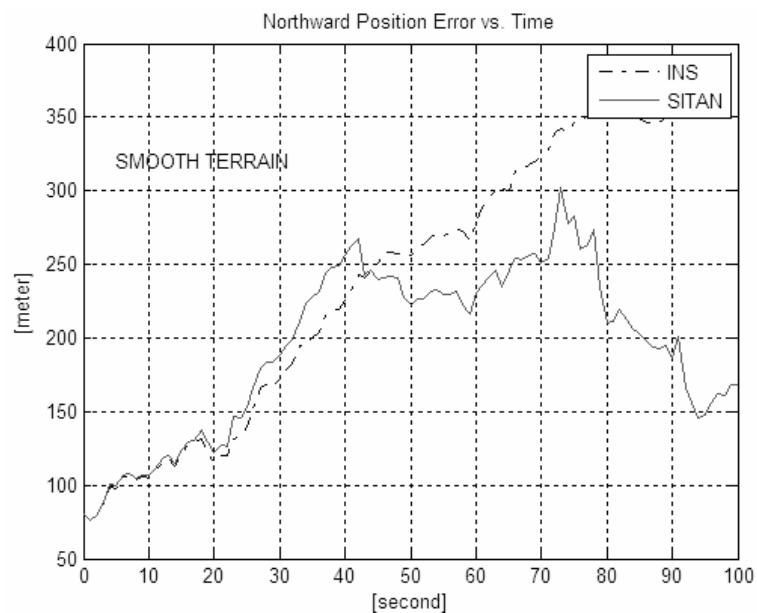


Figure 28. Smooth Terrain Northward Position Error vs. Time

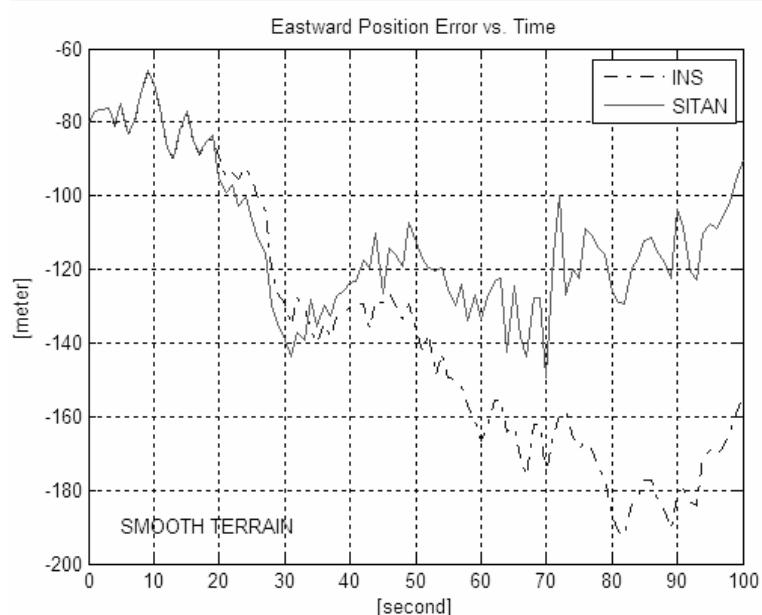


Figure 29. Smooth Terrain Eastward Position Error vs. Time

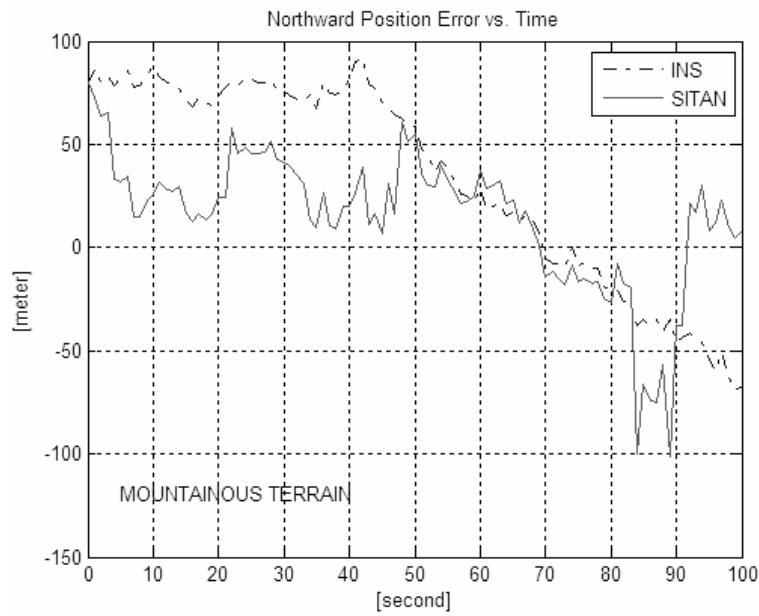


Figure 30. Mountainous Terrain Northward Position Error vs. Time

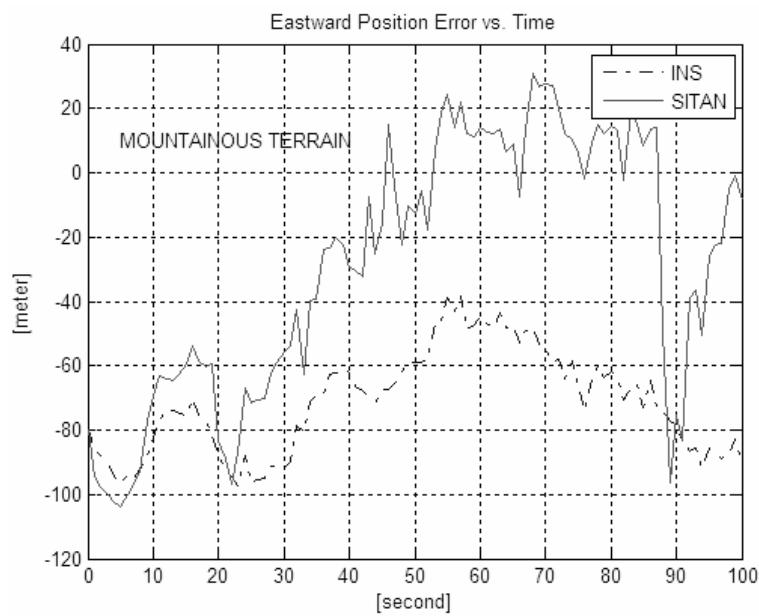


Figure 31. Mountainous Terrain Eastward Position Error vs. Time

As it can be seen from the figures above, SITAN improves position errors for rough and mountainous terrain types. On the other hand, due to slope determination process in SITAN, solutions have serious jumps for mountainous terrain type. This can be explained by the severe slope changes in the mountainous terrain modeling. As a result of this, SITAN works better for rough terrains. However, by TSL or other linearization methods as explained in the original paper [12], navigation solutions can be improved. In fact, these techniques depend on the terrain selected; and, extra work is required for terrain linearization.

Next, SITAN simulations for acquisition mode are performed. Here, initial position error is assumed to be greater than DTED grid size. 25 parallel KF's are used in the simulations as shown in Table 13. Here, the index “ $i$ ” indicates the related grid for initial position. For example, if the initial position error was  $2''$  short along longitude and  $2''$  long along latitude considering INS outputs, actual initial position would be at  $i = 5$  where  $i = 13$  was the INS index.

Table 13. Parallel KF Structure for SITAN Acquisition Mode

$\lambda = \lambda_{INS} + 2*3''$	$i = 5$	$i = 10$	$i = 15$	$i = 20$	$i = 25$
$\lambda = \lambda_{INS} + 3''$	$i = 4$	$i = 9$	$i = 14$	$i = 19$	$i = 24$
$\lambda = \lambda_{INS}$	$i = 3$	$i = 8$	$i = 13$	$i = 18$	$i = 23$
$\lambda = \lambda_{INS} - 3''$	$i = 2$	$i = 7$	$i = 12$	$i = 17$	$i = 22$
$\lambda = \lambda_{INS} - 2*3''$	$i = 1$	$i = 6$	$i = 11$	$i = 16$	$i = 21$

Time:  $t = k$     $\mu = \mu_{INS} - 2*3''$     $\mu = \mu_{INS} - 3''$     $\mu = \mu_{INS}$     $\mu = \mu_{INS} + 3''$     $\mu = \mu_{INS} + 2*3''$

Note: Index  $i=13$  gives  $\delta h(k)$  at position  $(\lambda_{INS}, \mu_{INS})$  of INS, at time “ $t = t_0$ ”.

Index  $i=1$  gives  $\delta h(k)$  at position  $(\lambda_{INS} - 2*3'', \mu_{INS} - 2*3'')$  of INS for DTED Level 1, at time  $k$ .

Hence, the simulations are performed for acquisition mode. Simulations are performed only for rough terrain in order to show acquisition performance. Simulation parameters for acquisition mode are given in Table 14. Here, initial position errors are given for both axes in order to determine initial position index. From the simulations, determination of the initial position index is required.

Table 14. SITAN Simulation Parameters for Acquisition Mode

Initial INS position deviation (northward axis)	-200 m
Initial INS position deviation (northward axis)	-180 m
Initial position index (according to Table 13)	25
Initial vehicle velocity	240 m/s
Initial INS east velocity bias	0.5 m/s
Initial INS north velocity bias	0.5 m/s
INS horizontal position standard deviation	5 m
INS altitude position standard deviation	3 m
Radar altimeter standard deviation	3 m
INS velocity standard deviation	0.3 m/s

Simulations are performed for rough terrain for both minimum AWRS filter and the central filter. Horizontal position errors are given in Figure 32 and Figure 33. Minimum AWRS filter index versus time is given in Figure 34. Here, central SITAN filter results are also presented in order to show filter divergence. From the simulations, it can be seen that in order to obtain correct navigation solutions for large initial position errors, parallel KF's should be used. Moreover, from Figure 34 initial position index obtained is exactly the same with the simulation initial condition which means that the correct initial position is found from the parallel filter structure SITAN simulations.

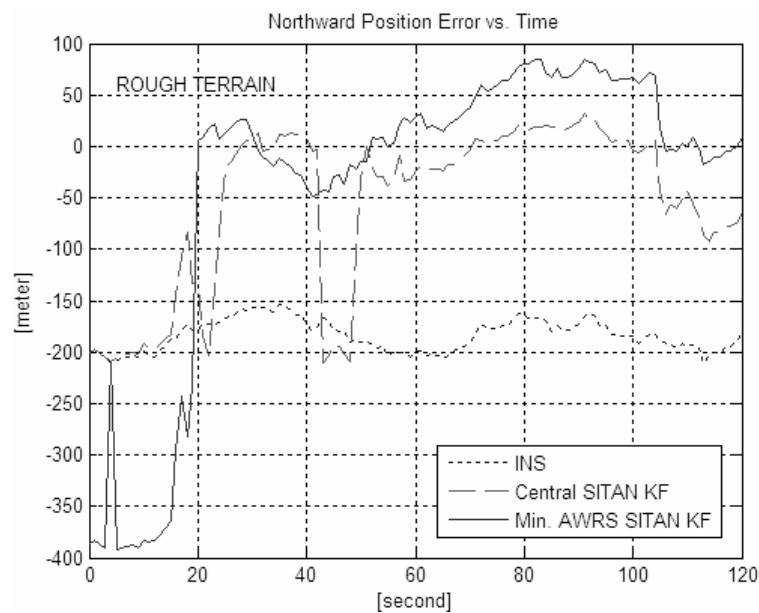


Figure 32. Northward Position Error vs. Time for Acquisition Mode

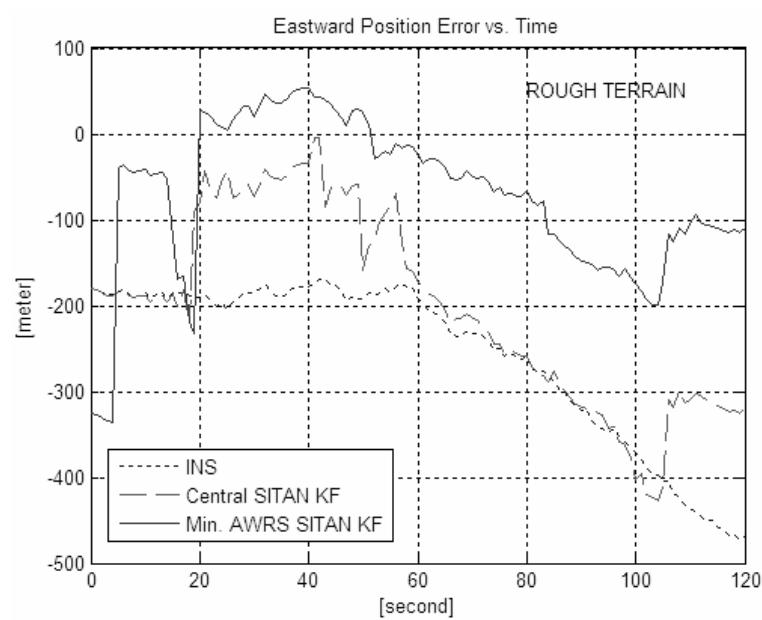


Figure 33. Eastward Position Error vs. Time for Acquisition Mode

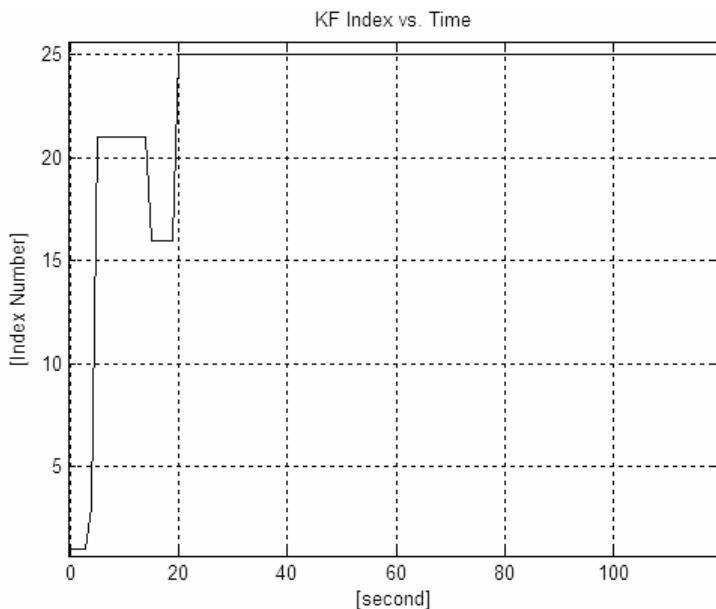


Figure 34. Minimum AWRS KF Index vs. Time for Acquisition Mode

For the SITAN process, several conclusions are achieved from the concept study and simulations performed. They are summarized as follows:

1. SITAN is a recursive TAN technique which uses EKF unlike TERCOM which is a batch process.
2. SITAN performance depends on the linearization of the terrain profiles since terrain slopes are required for the KF measurements. For large position errors, divergence can occur due to linearization errors in the EKF. In order to get rid of this, modified terrain linearization techniques and parallel KF structure are used.
3. SITAN improves position errors for rough and mountainous terrain types. However, due to slope determination process in SITAN, solutions have sometimes serious jumps for mountainous terrain type. This can be explained by the severe slope changes in the

mountainous terrain modeling. Therefore, linearization of the terrain profiles is very critical especially for mountainous terrains in SITAN.

4. SITAN performance is better than both INS and terrain grids unlike TERCOM. In TERCOM, error can not be better than the terrain grid dimensions.
5. SITAN performs better for smaller position errors due to terrain linearization. Due to this fact, for large initial position errors TERCOM or SITAN with parallel KF structure must be used.
6. SITAN is a tracking process (i.e. it tracks the actual path with minimum errors) where TERCOM is an acquisition process (i.e. it estimates the initial position of the target).

## 2.4. VATAN

### 2.4.1. VATAN Fundamentals

As it was stated in the first chapter, one of the interesting TAN algorithms found in literature is VATAN which is proposed by Enss and Morrell [42]. In order to investigate VATAN in detail, first original work of Enss and Morrell [42] will be investigated in detail.

VATAN is a recursive TAN technique which uses Viterbi Algorithm (VA). VA is a maximum a posteriori (MAP) estimator that estimates a sequence of system states from a sequence of observation values [42]. Viterbi algorithm is actually a dynamic programming technique for estimation which uses past information.

The state and observation sequences are denoted by [42]:

$$\bar{x} = (\bar{x}_0, \dots, \bar{x}_n) \quad (2.41)$$

$$z = (z_0, \dots, z_n) \quad (2.42)$$

where;

$$\bar{x}_k = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix}: \text{ Vehicle's position at time } t_k,$$

$$z_k = [z(k)]: \text{ Measured terrain elevation at time } t_k.$$

The VA consists of the computation of a metric function  $L_k$  that is a measure of the likelihood of each state value being the true state at time  $k$ ;  $L_k$  can be computed recursively using conditional probability density functions  $p(\bar{x}_{k+1} | \bar{x}_k)$  and  $p(z_k | \bar{x}_k)$  as follows [42] based on the assumption that the system dynamics are Markov; that is, the state at  $k+1$  is conditionally independent, given the state at time  $k$ , of the state at any previous time:

$$L_k(\bar{x}_k) = \max_{\bar{x}_{k-1}, \dots, \bar{x}_0} \left[ \sum_{i=1}^k \ln p(\bar{x}_i | \bar{x}_{i-1}) + \ln p(\bar{x}_0) + \sum_{i=1}^k \ln p(z_i | \bar{x}_i) \right] \quad (2.43)$$

$$= \ln p(z_k | \bar{x}_k) + \max_{\bar{x}_{k-1}} \left[ \sum_{i=1}^k \ln p(\bar{x}_k | \bar{x}_{k-1}) + L_{k-1}(\bar{x}_{k-1}) \right] \quad (2.44)$$

where;

$$L_0(\bar{x}_0) = \ln p(\bar{x}_0) : \text{ Initial condition of } L_k.$$

The optimal estimate  $\hat{\bar{x}}_k$  is that  $\bar{x}_k$  for which  $L_k$  is maximum. The value  $\bar{x}_{k-1}$  that maximizes equation (2.44) for each  $\bar{x}_k$  is termed the survivor. Denoted  $S_k(\bar{x}_k)$ , it is used to generate MAP state sequence estimates. The MAP state sequence estimate  $\hat{\bar{x}} = (\hat{\bar{x}}_k, \dots, \hat{\bar{x}}_0)$  can be generated via the following recursive procedure [42]:

$$\hat{\bar{x}}_k = \arg \max_{\bar{x}_k} \{L_k(\bar{x}_k)\} \quad (2.45)$$

$$\hat{\bar{x}}_{k-1} = S_k(\bar{x}_k)$$

$$\begin{aligned} \hat{\bar{x}}_{k-2} &= S_{k-1}(\hat{\bar{x}}_{k-1}) = S_{k-1}(S_k(\hat{\bar{x}}_k)) \\ \hat{\bar{x}}_{k-j} &= S_{k-j+1}(\hat{\bar{x}}_{k-j+1}) = S_{k-j+1}(S_{k-j}(\dots(S_k(\hat{\bar{x}}_k)))) \end{aligned} \quad (2.46)$$

The recursion in equation (2.44) is a filter, providing state estimates based on the system dynamics and observations. For an observable linear system model with Gaussian noises, equation (2.44) is functionally equivalent to a Kalman filter and equation (2.46) is equivalent to a fixed interval smoother (e.g., a Rauch-Tung-Striebel smoother). These equivalences suggest that the VA is a suitable replacement in applications that use Kalman filtering [42].

For the TAN problem, the VA has two significant advantages over the EKF used in SITAN [42]:

1. Metric function is computed for all possible state values makes the VA much more robust than the EKF in situations where the observations do not strongly support a single estimate of the state value.
2. The nonlinear relationship between vehicle position and measured terrain elevation can be represented exactly with the VA but must be approximated for the EKF.

In VATAN, the VA generates optimal MAP vehicle position estimates using the terrain elevation beneath the vehicle as its observation. In order to implement VATAN, conditional observation and; state transition densities in equation (2.44) are needed as well as an initial value of the metric  $L_0$ . In the original VATAN paper [42], very simple models are used to obtain the required densities.

Parameters required for the VATAN technique are given as follows [42]:

Nominal terrain height (Actual terrain height with zero measurement errors):

$$h_{ter}(\bar{x}_k) = h_{INS}(\bar{x}_k) - h_{radar}(\bar{x}_k) \quad (2.47)$$

Measured terrain elevation (Observation used in VATAN):

$$z_k = \delta h(k) = h_{ter}(\hat{\bar{x}}_k) = h_{INS}(\hat{\bar{x}}_k) - h_{radar}(\hat{\bar{x}}_k) \quad (2.48)$$

The measurements are assumed to be unbiased, independent and Gaussian. Therefore;  $E[h_{INS}(\hat{\bar{x}}_k)] = h_{INS}(\bar{x}_k)$  and  $E[h_{radar}(\hat{\bar{x}}_k)] = h_{radar}(\bar{x}_k)$  with variances  $\sigma_{h_{INS}}^2(k)$  and  $\sigma_{h_{radar}}^2(k)$ . Then [42]:

$$E[z_k | \bar{x}_k] = E[h_{INS}(\hat{\bar{x}}_k)] - E[h_{radar}(\hat{\bar{x}}_k)] = h_{ter}(\bar{x}_k) \quad (2.49)$$

$$\sigma_z^2(k) = \sigma_{h_{INS}}^2(k) + \sigma_{h_{radar}}^2(k) \quad (2.50)$$

Thus the conditional observation probability density function is:

$$p(z_k | \bar{x}_k) = \frac{1}{\sqrt{2\pi\sigma_z^2(k)}} \cdot \exp\left(-\frac{[z_k - h_{ter}(\bar{x}_k)]^2}{2\sigma_z^2(k)}\right) \quad (2.51)$$

The state transition density  $p(\bar{x}_{k+1} | \bar{x}_k)$  describes the states' evolution with time. Given a known velocity vector  $\dot{\bar{x}}_k$  that is constant over the  $T$  second sample interval from  $t_k$  to  $t_{k+1}$ , the state's evolution is [42]:

$$\bar{x}_{k+1} = \bar{x}_k + \dot{\bar{x}}_k \cdot T \quad (2.52)$$

Since the vehicle's velocity is provided by the INS, it is not known precisely. This uncertainty is dealt with by modeling the INS velocity as a Gaussian random variable with mean  $\dot{\bar{x}}_k^{INS}$  and variance  $\sigma_{\dot{\bar{x}}_k^{INS}}^2$ , indicative of the INS

precision. Thus the random variable  $\bar{x}_{k+1}$  conditioned on  $\bar{x}_k$  is a random variable with mean and variance [42]:

$$E[\bar{x}_{k+1}] = \bar{x}_k + E[\dot{\bar{x}}_k^{INS}] \cdot T \quad (2.53)$$

$$\sigma_{\bar{x}_{k+1}}^2 = \sigma_{\dot{\bar{x}}_k^{INS}}^2 \cdot T^2 \quad (2.54)$$

Since  $\bar{x}_0$  is assumed to be Gaussian, by equation (2.52),  $\bar{x}_k$  is Gaussian for  $k > 0$ . Moreover, the metric function  $L_k$  is initialized considering equation (2.43) by [42]:

$$L_0(\bar{x}_0) = \ln p(\bar{x}_0) = \ln p(\bar{x}_0^{INS}) \quad (2.55)$$

Here, it should be noted that VATAN models are derived considering a simplified INS model. Actually, INS error model used in SITAN can also be used for VATAN implementation.

#### 2.4.2. Simulations and Discussion

Simulation results for VATAN are presented from the original work of Enss and Morrell [42]. They performed simulations for VATAN using four different terrain types:

1. Typical rough terrain,

2. Flat terrain,
3. Mountainous terrain,
4. Sloped and flat terrain.

Flight paths regarding the terrain types are shown in Figure 35.

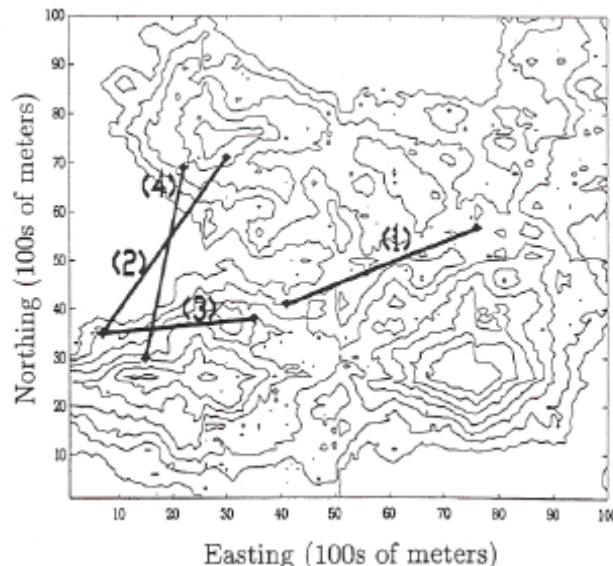


Figure 35. Contour Plot for VATAN Simulation Terrain Types [42]

Simulations are performed for horizontal position errors (mean errors performed with Monte Carlo simulations and deviation errors simulated in tracking mode) and the results are compared with SITAN. Simulation results for different terrain types show that VATAN consistently performs as well as or better than SITAN implementation. VATAN performs as well as SITAN in moderately rough,

sloped terrain and it exceeds SITAN's performance in very flat or very rough terrains [42].

However, there exist some drawbacks of the VATAN technique. VATAN's major limitation is the increased computational capacity necessary to implement the VA when compared to an EKF [42]. In the original paper, the two-dimensional VA has only been implemented in a discrete state space; a continuous state-space implementation of the two-dimensional VA would improve VATAN's accuracy and could result in a substantial reduction of the computational capacity necessary to implement VATAN [42].

For the VATAN process, several conclusions are achieved from the investigation of the original paper of Enss and Morrell [42]. They are summarized as follows:

1. VATAN is a recursive TAN technique like SITAN. However, since the past measurements are stored and used its performance is said to be better than SITAN.
2. From the paper, it is shown that VATAN performs better results for all terrain conditions (both very rough and flat terrains).
3. VATAN uses VA which is a maximum a posteriori (MAP) estimator that estimates a sequence of system states from a sequence of observation values. VA is actually a dynamic programming technique for estimation which uses past information.
4. The major disadvantage of VATAN is the limitation of the increased computational capacity necessary to implement the process. Actually, VA is also used as a radar tracking algorithm. By

investigating this paper, implementation of modern radar tracking algorithms to TAN has been inspired.

In this chapter, major TAN algorithms have been investigated in detail with their fundamentals described in original references and the simulations performed. Simulation model details are not presented in this chapter; since, they will be given in the following chapter. Several conclusions have been obtained from the detailed study of the major algorithms and they have been discussed in the chapter.

## **CHAPTER 3**

### **IMPLEMENTATION OF TARGET TRACKING ALGORITHMS TO TERRAIN AIDED NAVIGATION**

In this chapter, implementation of target tracking algorithms to TAN is presented. First, general information about modern target tracking algorithms are given. Next, PDAF and TSF data association algorithms and their general implementations are investigated. Then, PDAF and TSF implementations to TAN are presented. At the end of the chapter, a simple simulation model is developed for the mid-course flight of the cruise missile. Finally, simulations are performed with the implemented TAN algorithms and the results are compared with the major TAN methods.

#### **3.1. Target Tracking Background**

In the first section of the chapter, a historical background about target tracking will be introduced. Major developments in multi-target tracking over the past four decades and how algorithms developed primarily for tracking air targets will be discussed. Then, target state estimation algorithms like Kalman filtering and association algorithms fundamentals will be investigated.

Eventhough tracking problems can be found in many applications, e.g., ocean surveillance and submarine tracking, most tracking algorithms have been developed for air targets [64].

A tracking problem is defined by the targets of interest, the sensors that collect the measurements, and the environment in which the targets move and sensors observe the targets. The basic functions in multi-target tracking consist of prediction, association, and estimation and they are shown in Figure 36. When measurements are received, the current tracks are predicted to the time of the measurements and associated with the measurements. Then the associated measurements are used to update the state estimates of the tracks. Although these functions are not always performed sequentially, they are present in most tracking algorithms [64].

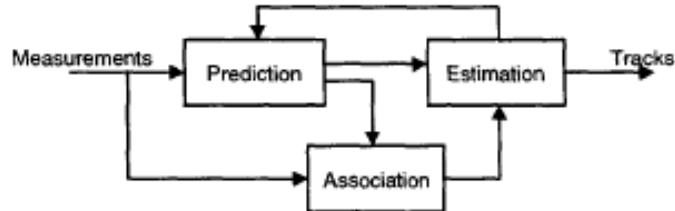


Figure 36. Basic Tracking Functions [64]

Prediction and estimation are single target state estimation functions in the absence of measurement uncertainty. Prediction difficulty depends on target dynamics and sensor revisit time. On the other hand, when the origins of the measurements are uncertain, e.g., when clutter or multiple targets are present, the measurements have to be associated with other measurements or tracks before the target state estimates can be generated. Therefore, data association establishes

tracking as a separate discipline from traditional state estimation in target tracking [64].

Chong, et al [64] investigated target estimation and data association algorithms in detail for ground target tracking. However, they discussed the subject from the historical point of view and investigated the algorithms with the related references in detail. In the following sections, target estimation and data association algorithms will be summarized considering the helpful reference of Chong, et al [64].

### **3.1.1. Target State Estimation**

Target state estimation is an important component of any multi-target tracking algorithm. The association of measurements to tracks requires the prediction of the target state of each track to the time of the measurements so that the measurement to track likelihood can be computed. Accurate state prediction is a key to good association performance. Once an association decision has been made, the output of the tracker consists of updated state estimates of the tracks using the associated measurements [64].

Target state algorithms can be grouped according to the algorithms applied as follows:

1. Linear Estimation Algorithms:

These algorithms assume linear target motion and observation models and provide estimates of the target state by means of linear transformations [64].

a. Alpha-Beta-Gamma Filters:

These constant coefficient filters estimate the target position and velocity from position measurements only. The alpha-beta filter assumes a second order model driven by white noise for the target dynamics while the alpha-beta-gamma filter assumes a third order model. In either case, the filters can be considered as steady state Kalman filter [64].

b. Kalman Filter:

The Kalman filter has been the standard approach to filtering for linear systems since its development in the earlier sixties [64]. Details of Kalman filtering have been discussed in several chapters of the study.

2. Adaptive Filters:

When a target maneuvers, the model no longer matches the dynamics and performance will degrade. Several approaches have been developed to detect maneuvers and adapt the filter to the target dynamics in real-time [64].

a. Parameter Adjustment:

The structure of the filter is fixed. However, the filter will monitor its own performance (such as the size of the residuals) and adapt parameters (such as the process noise covariance or the Kalman filter gain) when a target maneuver is detected [64].

b. State Augmentation:

This approach uses different dynamics models when a maneuver has been detected. For example, before maneuver, a constant velocity model is used. When a maneuver has been detected, the filter switches to an acceleration model with higher state dimension and switches back to the original model when the maneuver is determined to have ended [64].

3. Multiple Models:

When the measurement does not contain sufficient information, an incorrect decision may be made, resulting in poor performance. Therefore, algorithms that maintain multiple target dynamic models have been developed. These algorithms compute the probability of each model being true given the measurements and generate a target state estimate as a weighted sum of the estimates given the individual models [64].

a. Static Multiple Models:

These models assume that the true target motion model is static and contained in a fixed set of models. Because the target model does not change with time, this approach is not appropriate for maneuvering targets [64].

b. Model Sequence Pruning:

The optimal multiple model estimator requires a filter for each possible model sequence hypothesis. Since the number of model sequences and thus the number of filters increases exponentially with

time, the optimal estimator is not practical. An obvious sub-optimal approach is to prune the least likely model sequences according to their probabilities [64].

c. Generalized Pseudo Bayesian Estimator:

The Generalized Pseudo Bayesian (GPB) method is a suboptimal approach that reduces the number of filters by merging model sequences that end up with the same fixed length sub-sequences [64].

d. Interacting Multiple Models:

The Interacting Multiple Model (IMM) algorithm is one of the most popular algorithms for tracking maneuvering targets because of its relatively simple implementation and its ability to handle complicated dynamics [64].

e. Variable Structure Interacting Multiple Models:

While IMM has been successfully used in several applications, having a fixed model set has its disadvantages. Variable Structure Interacting Multiple Model (VSIMM) approach is used to track ground targets moving over roads and open field. The target motion models reflect the mobility of a target for different conditions [64].

4. Nonlinear Estimation:

Many dynamic models or observation models do not satisfy the linear assumptions. Therefore, approaches for estimating the state of nonlinear systems have been developed.

a. Extended Kalman Filter (EKF):

The non-linearity of the dynamic and observation models can be linearized about a nominal trajectory, and then a Kalman filter can be developed with the linearized model which is called EKF [64].

b. Gaussian Sum Approximations:

The EKF assumes that the conditional probability distribution can be approximated reasonably accurately by a Gaussian distribution. When this approximation is not valid, the conditional probability distribution of the states given the cumulative measurements can be approximated by a sum of Gaussian distribution [64].

c. Nonlinear Filtering:

This optimal nonlinear filtering algorithm has nice features such as the ability to update the probability distribution of the states due to non-detections. However, implementation is computationally intensive since it requires discretization of the state space and performing the integration by a summation. Thus, even though the algorithm has been known for many years, it has seldom been used [64].

### **3.1.2. Data Association**

When the origins of the measurements are uncertain, e.g., when clutter or multiple targets are present, the measurements have to be associated with other measurements or tracks before the target state estimates can be generated. Association is what distinguishes target tracking from traditional state estimation and establishes tracking as a separate discipline [64].

Data association algorithms can be classified according to whether they focus on single targets or consider explicitly the presence of multiple targets and whether association decisions are made using single or multiple scans of data. The early algorithms tend to focus on single scan and single targets, while the recent algorithms deal with multiple scans of data and multiple targets. In general, algorithms that consider multiple targets and use multiple scans of data perform better but require more computations [64].

Data association algorithms can be grouped according to the algorithms applied as follows:

1. Single Target Track Formation:

These track formation algorithms initiate tracks from sequences of measurements without considering competition from other tracks [64].

- a. “M” out of “N” Test:

A track is tentatively initiated from a single measurement. A validation gate is then established around this measurement and a measurement falling inside this gate becomes part of the track. When there are “M” detections out of “N” scans of measurements, then the

track is formed or confirmed. This method is very simple but does not provide a score on the confidence of the track [64].

b. Likelihood (Ratio) Test:

In the likelihood tests, tracks are declared as confirmed (or deleted) when the likelihood or ratio exceeds (or falls below) a certain threshold [64].

2. Single Target Track Maintenance:

These algorithms associate measurements with the existing tracks without considering the presence of other tracks. Thus a measurement may be associated with multiple tracks [64].

a. Nearest Neighbor:

In this method, the measurement that is closest (according to some distance measure) to the track is associated with the track from the multiple measurements. This approach makes a hard decision based on a single scan and is very easy to implement. However, it does not perform well in high density situations [64].

b. Track Splitting:

This is basically applying the likelihood function (or ratio) approach to track maintenance. For every measurement that falls in the validation gate, the track is split. Each track is scored using a likelihood function as discussed before. The track is pruned when the likelihood falls below a threshold. This approach makes soft decisions based upon multiple scans of data. Because of its

computational requirements and limited performance, this approach is no longer popular [64].

c. Probabilistic Data Association (PDA):

Instead of associating a single measurement with a track, this approach probabilistically associates all measurements in the validation gate. The PDAF is an all-neighbors association algorithm. It is fairly easy to implement and has been shown to perform better than the nearest neighbor approach in high clutter [64].

d. Optimal Bayesian Approach:

The PDAF is a suboptimal approach since the association event only considers the current measurements. On the other hand, the optimal Bayesian approach will consider all possible association hypotheses up to the current time [64].

3. Multiple Target Track Maintenance:

Association performance can be improved when the algorithms consider explicitly the presence of multiple targets and recognize that a single measurement cannot belong to multiple tracks [64].

a. Optimal Assignment:

The optimal assignment approach, also sometimes called global nearest neighbor, is the coordinated version of nearest neighbor. Instead of selecting the measurement that is closest to a track, this approach selects the set of measurements that is closest to the set of tracks according to some global distance measure subject to the

constraint that two tracks do not share a single measurement, and two measurements do not appear in the same track [64].

b. Joint Probabilistic Data Association (JPDA):

This is the extension of PDA to multiple targets. The tracks for a known number of targets are assumed to have been initiated and the problem is to associate the measurements to the tracks [64].

4. Multiple Scan Coordinated Association:

Both the measurement and the target motion models have uncertainty. Therefore, the single scan decisions may not be the correct associations. Thus association performance can be improved by using multiple scans of data. The core of all multiple scan algorithms is the evaluation of track likelihoods, which can be used for both track formation and maintenance. Thus multiple scan algorithms generally can be used for both track formation and association [64].

a. Integer Programming:

This approach was the first multiple scan algorithm and integer programming problem can be solved by branch and bound or other methods. However, this algorithm was improved to other multiple scan algorithms such as multiple hypothesis tracking [64].

b. Multiple Hypothesis Tracking (MHT):

Multiple hypothesis tracking delays making hard decisions when there is not sufficient information to make a good decision. Alternative hypotheses are formed to represent the ambiguities and

each hypothesis is evaluated. The MHT is conceptually simple but computationally intensive since the number of hypotheses grows exponentially [64].

c. Multi-Dimensional Assignment:

Traditional MHT requires the explicit expansion and evaluation of many hypotheses. Successful implementation requires the use of sophisticated hypothesis management techniques to handle the combinations. During the last decade, alternative optimization based methods that do not require the explicit expansion and evaluation of hypotheses have been developed. Such algorithms are easier to implement and computationally more efficient [64].

5. Tracking Without Data Association:

Several approaches have been proposed to perform tracking without an explicit association function. Instead of dealing with individual target states and individual measurements, these approaches treat all targets and measurements as components of one system, and estimate the system state directly without explicitly forming association hypotheses [64].

a. Symmetric Measurement Equations:

In this approach the original measurements on the targets are converted into a new set of measurements that are symmetric functions of the original measurements [64].

b. Multi-target Nonlinear Filtering:

The individual target motion and measurement models can be aggregated into a multi-target motion model given by the conditional probabilities and a measurement model given by the likelihood function. Then, the same nonlinear filtering method developed for a single target can be used (conceptually at least) for tracking multiple targets [64].

As it can be seen from the historical point of view, target tracking is a comprehensive subject. In this study, implementation of some of these algorithms to TAN is done. As a result of this, some of the algorithms summarized above will be discussed in detail in the following sections. Then, they will be implemented for TAN applications.

## **3.2. Probabilistic Data Association Filter (PDAF)**

### **3.2.1. Theory**

The PDA algorithm calculates in real-time the probability that each validated measurement is attributable to the target of interest. This probabilistic (Bayesian) information is used in a tracking filter, the PDA filter (PDAF) which accounts for the measurement origin uncertainty [52].

The following assumptions are made to obtain the recursive PDAF state estimator (tracker) [52]:

1. There is only one target of interest whose state evolves according to a dynamic equation driven by process noise.
2. The track has been initialized.
3. The past information about the target is summarized approximately by;

$$p[x(k) | Z^{k-1}] = N[x(k); \hat{x}(k | k-1), P(k | k-1)] z \quad (3.1)$$

where,

$N[x(k); \hat{x}(k | k-1), P(k | k-1)]$ : Normal probability density function.

$x(k)$ : Argument,

$\hat{x}(k | k-1)$ : Mean,

$P(k | k-1)$ : Covariance matrix.

4. At each time, a validation region is set up.
5. Among the possibly several validated measurements, at most one of them can be target-originated, if the target was detected and the corresponding measurement fell into the validation region.
6. The remaining measurements are assumed to be false alarms or clutter and are modeled as independent identically distributed measurements with uniform spatial distribution.

7. The target detections occur independently over time with known probability.

These assumptions enable a state estimation scheme to be obtained, which is almost as simple as the Kalman filter, but much more effective in clutter [52].

The probabilistic data association algorithm associates all valid observations with a track. For each validated observation, an updated estimate  $\hat{x}_i(k|k)$  is computed. A probability of correct association  $\beta_i$  is computed for each such track. Then a combined track is formed from the weighted average of these tracks:

$$\hat{x}(k|k) = \sum \beta_i \cdot \hat{x}_i(k|k) \quad (3.2)$$

For multiple targets, the same process occurs although the probability calculations are more complex which is called Joint Probabilistic Data Association Filter (JPDAF) [65].

In Figure 37, general PDAF implementation is shown. Then, the following approach can be implemented in order to perform PDAF algorithm [65]:

1. The set of validated measurements is computed.
2. For each validated measurement an updated track is computed.
3. For each updated track an association probability  $\beta_i$  is computed.

The calculation of this probability can be quite complex and dependent on the assumed clutter densities. However, it is normally

adequate to set  $\beta_i$  proportional to the normalized innovation for the association.

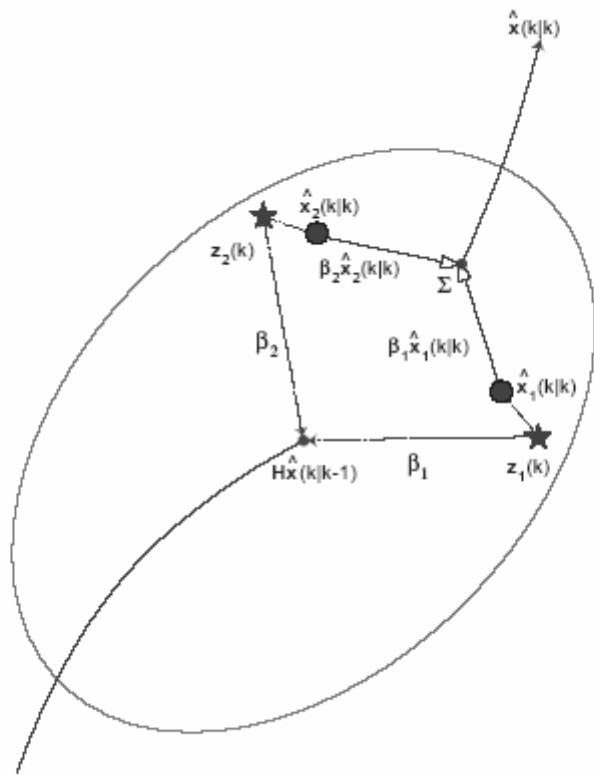


Figure 37. PDAF Implementation [65]

4. A combined (average) track is computed.
5. A combined average covariance can also be computed although this can become quite complex.

6. A single prediction of the combined track is then made to the next scan time and the process is repeated.

The PDAF and JPDAF methods are appropriate in situations where there is a high degree of clutter. The great advantage with the PDAF method is that you are never wrong. The problem is you are also never right [65].

PDA procedure is summarized in Figure 38 and detailed PDAF equations are given in Appendix.

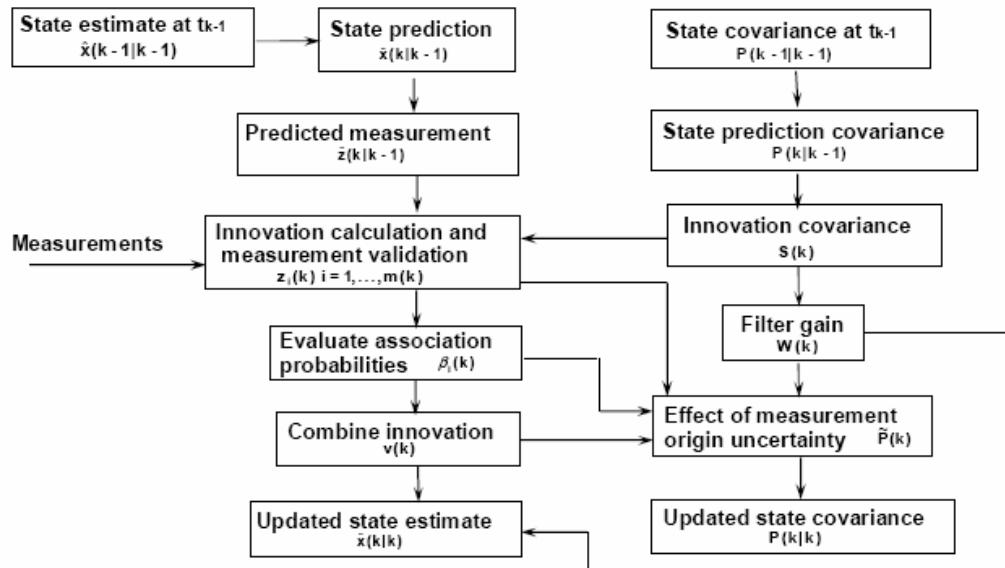


Figure 38. PDAF Procedure [66]

### **3.2.2. Implementation of PDAF to TAN**

#### **3.2.2.1. Implementation Methods of PDAF for TAN**

Terrain Aided Navigation (TAN) algorithms estimate the position of a moving vehicle by comparing the measured terrain profile under the vehicle to a stored elevation map. Therefore, the critical part of the TAN is the terrain elevation database, i.e. DTED for the common military applications. Batch and recursive algorithms are used for TAN as explained before. For batch algorithm, i.e. TERCOM, only DTED is used in order to estimate the position of the vehicle. On the other hand, for recursive algorithms, like SITAN, VATAN, etc. besides DTED, the system dynamics should also be modeled.

Application of the target tracking algorithms to navigation problems have been investigated in several papers given in Qingtang, et al [40], Dezert [43] and Maksarov and Durrant-Whyte [67]. In the paper of Qingtang, et al [40], TAN using PDAF was investigated for the batch algorithm. Association probabilities have been derived using the MSD function of the TERCOM process and performance of the TAN using PDA and TERCOM has been compared. In the paper of Dezert [43], PDAF has been used in order to improve the accuracy of a strapdown INS using landmark detections. Maksarov and Durrant-Whyte [67] used multiple hypothesis technique (MHT) algorithm for an autonomous mobile vehicle with multiple sonar sensors for range measurements.

As it can be seen from the papers investigated, Qingtang, et al [40] is directly related with the Ph.D. study. The main difference of the Ph.D. study from Qingtang, et al [40] is the real-time application of PDAF to TAN which will be discussed in detail.

First application of a target tracking algorithm to TAN problem should be discussed. Consider the assumptions to obtain a recursive PDAF estimator again:

1. There is only one target of interest whose state evolves according to a dynamic equation driven by process noise.
  - Target of interest is the cruise missile and its motion can be modeled. In MHT and JPDA multiple targets are considered with more complex algorithms.
2. The track has been initialized.
  - Initial conditions of the vehicle's motion can be modeled.
3. The past information about the target is summarized.
  - Cruise missile dynamics model gives information about its motion.
4. At each time, a validation region is set up.
  - CEP of the vehicle is determined by the quality of the INS used. The validation region for the measurements is set considering  $3\sigma$  position error bound of the INS horizontal positions.
5. Among the possibly several validated measurements, at most one of them can be target-originated, if the target was detected and the corresponding measurement fell into the validation region.
  - DTED area considering the  $3\sigma$  position error bound of the vehicle can be used considering only one of the height measurements rely on the target.

6. The remaining measurements are assumed to be false alarms or clutter and are modeled as independent identically distributed measurements with uniform spatial distribution.
  - False measurements are modeled considering the height channel covariances due to radar and INS height measurements.
  
7. The target detections occur independently over time with known probability densities.
  - Radar height measurements can be modeled as time independent occurrences.

From the assumptions of PDAF, it is seen that PDAF can be applied to TAN. Data association problem comes from the DTED height differences used for TAN. It is known that one of the grid of the DTED batch considered gives the correct position of the vehicle. Now, the implementation of PDAF algorithm to TAN can be discussed.

First, consider other TAN algorithms used for both batch and recursive algorithms. In Figure 39, batch algorithm concept is summarized.

In the batch algorithm, only the height measurements and their relations with the related DTED are considered. For a period of time, measurements are taken and correlation can be obtained using TERCOM process which is actually a maximum likelihood estimator. For the batch algorithm, the model of the vehicle motion is not required. Therefore, procedure is simple and larger DTED area should be used for the calculations. As a result of this, it can be concluded that batch algorithm can be successfully used in cruise missiles with considerably accurate INS. In fact, INS quality used in cruise missiles is around 1 nm/hr. Then, a few position updates

during the mid-course phase of the operation with the help of TERCOM algorithm will be sufficient for the navigation solution.

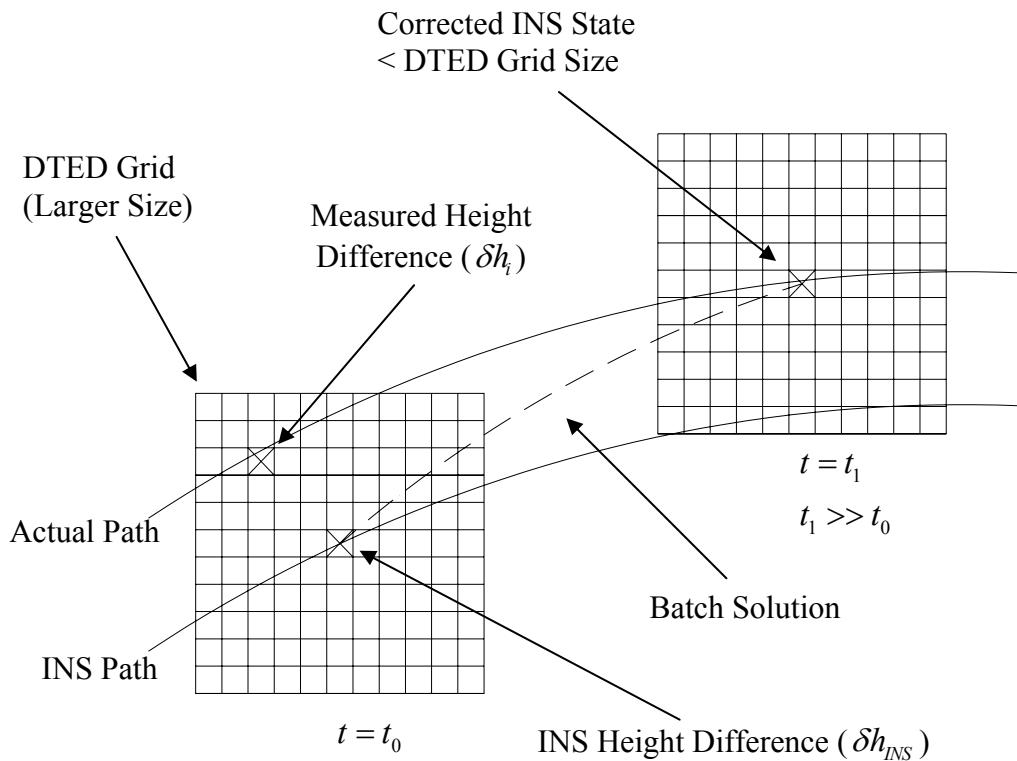


Figure 39. Batch Algorithm for TAN Solution (Acquisition Mode)

In recursive algorithms, TAN solution is done continuously. In order to achieve a complete navigation solution, the motion of the vehicle should also be modeled. In Figure 40, recursive algorithm for TAN concept is shown.

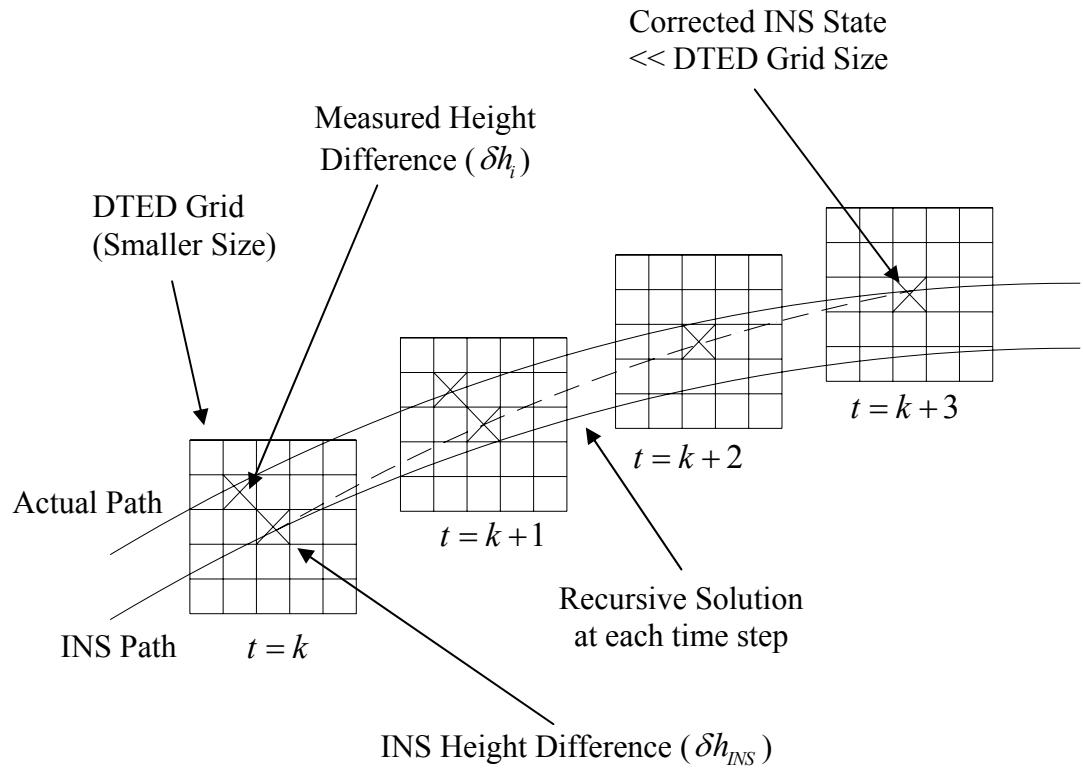


Figure 40. Recursive Algorithm for TAN Solution (Tracking Mode)

As it was discussed in earlier chapters, there are various recursive TAN algorithms proposed in literature. The well-known recursive algorithm is SITAN. Recursive TAN algorithms generally use estimation theory in order to solve the navigation problem. In SITAN, extended Kalman filtering, where in VATAN, Viterbi algorithm (a maximum a posteriori state sequence estimator) is used. Various recursive algorithms are also proposed as using maximum a posteriori estimation and optimal Bayesian estimation.

Due to the nonlinear dynamics of the navigation system using terrain information, algorithms for recursive TAN solutions require generally complex

calculations. In SITAN, linearization is done using EKF for the terrain. Several disadvantages of the recursive TAN algorithms can be summarized as follows:

1. TAN requires terrain information for the navigation solution and the dynamics of the system is highly nonlinear.
2. Equations derived for recursive algorithms are generally complex and needs considerable calculation work.
3. Real-time application for the TAN solution is generally impractical for high velocity vehicles like cruise missile due nonlinear characteristics of the system.
4. In SITAN, terrain linearization and terrain slopes are required in order to apply extended Kalman filter equations which are actually critical stages for TAN solution.

In this study, a new recursive TAN algorithm is investigated, which uses PDA, a target tracking algorithm. In Figure 41, real-time PDAF application for TAN is shown. Consider the PDA approach given in the previous section for TAN again:

1. The set of validated measurements is computed.
  - Measurement gate is taken as  $3\sigma$  position error bound of the INS and the invalid possibilities for the height differences  $\delta h_i(k)$  are discarded.
2. For each validated measurement an updated track is computed.

- Here, INS error model is used due to its linear characteristics. The updated tracks  $\delta\hat{x}_i(k)$  are computed for all valid points in the gate using PDA filtering.
3. For each updated track an association probability  $\beta_i$  is computed.
- Using height differences for each valid element of the grid, association probabilities  $\beta_i$  are calculated.
4. A combined (average) track is computed.
5. A combined average covariance can also be computed although this can become quite complex.
6. A single prediction of the combined track is then made to the next scan time and the process is repeated.
- Estimated error state  $\delta\hat{x}(k|k)$  is computed considering equation (3.2). Then the estimated error state becomes:

$$\delta\hat{x}(k|k) = \sum \beta_i \cdot \delta\hat{x}_i(k|k) \quad (3.3)$$

Next, derivation of the PDAF equations for TAN is done. PDAF equations for TAN are implemented considering standard real-time PDAF equations which are given in Appendix.

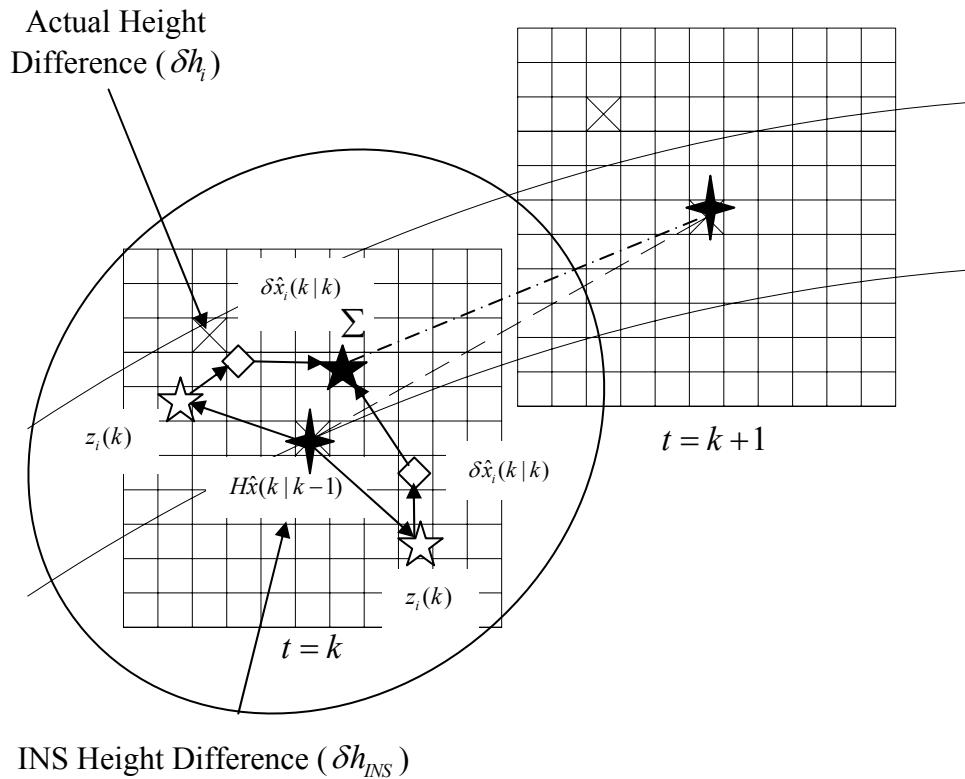


Figure 41. PDAF Implementation for TAN Solution

### 3.2.2.2. PDAF Equations Implemented for TAN

#### 3.2.2.2.1. Past Measurement Information

In the TAN algorithm, the only measurement is the height difference  $\delta h$  given in equation (2.23) which was given for SITAN in the previous chapter. However, since the  $3\sigma$  horizontal error bound of the vehicle is estimated considering the quality of the INS used, a batch of height differences around the INS position can be obtained. Rewrite SITAN equations considering the barometric height given as the INS height. Then, barometric height definition can be rewritten as follows:

$$h_{INS}(x, y) = h_{baro}(x, y) \quad (3.4)$$

Since DTED are used in the simulations where the heights are given with respect to related longitudes and latitudes, instead of using eastward and northward positions  $(x, y)$ , longitudes and latitudes  $(\mu, \lambda)$  are selected in the equations. Then, estimated and trajectory (real) positions can be defined as:

$$(\hat{x}, \hat{y}) = (\mu_{INS}, \lambda_{INS}) \quad (3.5)$$

$$(x, y) = (\mu_{traj}, \lambda_{traj}) \quad (3.6)$$

where;

$(\hat{x}, \hat{y})$ : Estimated eastward and northward position,

$(\mu_{INS}, \lambda_{INS})$ : Longitude and latitude of the INS position,

$(x, y)$ : Trajectory eastward and northward position,

$(\mu_{traj}, \lambda_{traj})$ : Longitude and latitude of the trajectory position.

Considering the measurements to be taken at discrete time steps  $k$ , SITAN measurement equations derived according to Figure 22 can be rewritten as follows:

$$\begin{aligned} h_{INS}(k) &= h_{INS}(\mu_{INS}(k), \lambda_{INS}(k)) \\ &= h_{DTED}(\mu_{INS}(k), \lambda_{INS}(k)) + w_{INS}(k) + C_{est}(k) \end{aligned} \quad (3.7)$$

$$h_{INS}(k) = h_{DTED}(\mu_{traj}(k), \lambda_{traj}(k)) + w_{radar}(k) + C_{meas}(k) \quad (3.8)$$

$$h_{radar}(k) = C_{meas}(k) + w_{radar}(k) \quad (3.9)$$

$$\delta h(k) = C_{meas}(k) - C_{est}(k) \quad (3.10)$$

$$\begin{aligned} \delta h(k) &= [h_{DTED}(\mu_{INS}(k), \lambda_{INS}(k)) + w_{INS}(k)] \\ &\quad - [h_{DTED}(\mu_{traj}(k), \lambda_{traj}(k)) + w_{radar}(k)] \end{aligned} \quad (3.11)$$

Now, consider the DTED batch for each height difference which is shown in Figure 42. Then, for each grid node, equation (3.11) can be written as follows considering the grid index  $i$ :

$$\begin{aligned} \delta h_i(k) &= [h_{DTED}(\mu_i(k), \lambda_i(k)) + w_{INS}(k)] \\ &\quad - [h_{DTED}(\mu_{traj}(k), \lambda_{traj}(k)) + w_{radar}(k)] \end{aligned} \quad (3.12)$$

$$i = 1, \dots, m(k)$$

where;

$i$ : Index of the DTED grid node,

$m(k)$ : DTED grid size (selected as square of an odd number for INS position to be at the center of the DTED grid)

Here, it should be noted that INS height difference is at  $i=13$  for a 5x5 DTED grid size.

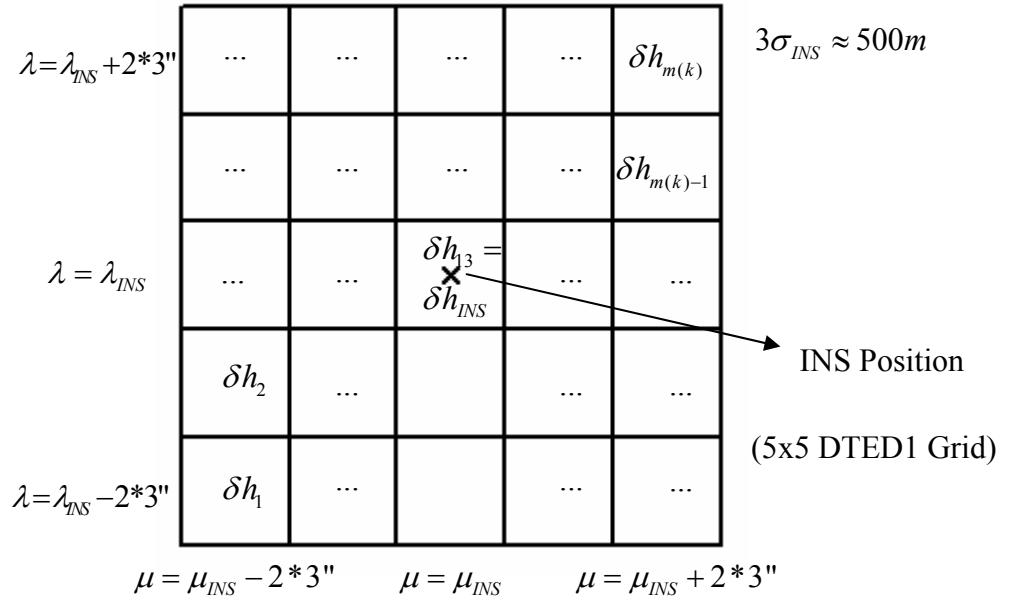


Figure 42. Height Differences Batch Used in PDAF

Height difference at the INS position can be found for a DTED grid size of  $m(k)$  as:

$$\delta h_{INS}(k) = \delta h_{\frac{1}{2}[m(k)+1]}(k) \quad (3.13)$$

From Figure 42, it can be seen that the only actual measurement is  $\delta h_{i=13} = \delta h_{INS}$  at time step  $k$ . Other  $\delta h_i$ ,  $i=1,...,m(k)$  values are derived using the

defined DTED grid. Using the DTED heights given at the positions around the given INS position, height difference batch at time step  $k$  is obtained.

For the TAN algorithm, past height difference measurements are averaged in order to smooth the effects of the past measurements. In order to achieve this, height measurements are put in a buffer and then, the average is used as the measurement batch to the PDA filter. For the study, buffer size of 20 to 30 (i.e., 20-30 seconds of measurements) was sufficient. As the new measurements come, the oldest measurements are eliminated.

Consideration of the past measurement information can be summarized as follows:

$$\delta h_{i_{ave}}(s+k) = \frac{1}{s} \cdot \sum_k^{s+k} \delta h_i(k) \quad s > k \geq 0 \quad (3.14)$$

where;

$\delta h_{i_{ave}}$  : Average of the height difference at position related to index  $i$ ,

$s$  : Buffer size.

In the same manner, height difference batch matrix can be written as:

$$\hat{\delta h}_{ave}(s+k) = \begin{bmatrix} \delta h_{1_{ave}}(s+k) & \dots & \dots & \dots & \dots \\ \delta h_{2_{ave}}(s+k) & \dots & \dots & \dots & \dots \\ \delta h_{3_{ave}}(s+k) & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \delta h_{m(k)-1_{ave}}(s+k) \\ \dots & \dots & \dots & \dots & \delta h_{m(k)_{ave}}(s+k) \end{bmatrix}_{m(k)^{1/2} \times m(k)^{1/2}} \quad s > k \geq 0 \quad (3.15)$$

### 3.2.2.2. Measurement Validation

In order to determine validation region for measurements, consider standard PDAF equations in the Appendix. In the TAN algorithm, validation region given in equation (A.3) is directly taken as the  $3\sigma$  error bound of the vehicle. However, height difference measurements in the batch matrix must be valid for the calculations. In the equation considered, measurements are the height differences  $\delta h_i, i = 1, \dots, m(k)$ . Innovation covariance  $S(k)$  given in equation (A.4) contains the system height state covariance  $P(k | k-1)$ , and the radar measurement noise covariance matrix  $R(k)$ . Therefore:

$$P(k | k-1) = \sigma_{h_{INS}}^2; R(k) = \sigma_{radar}^2 \quad (3.16)$$

Then, equation (A.3) becomes:

$$\begin{aligned} \delta h_i(k)^T \cdot [P(k | k-1) + R(k)]^{-1} \cdot \delta h_i(k) &\leq \gamma \\ \delta h_i(k)^2 &\leq \gamma \cdot (\sigma_{h_{INS}}^2 + \sigma_{radar}^2) \\ \delta h_i(k) &\leq \left[ \gamma \cdot (\sigma_{h_{INS}}^2 + \sigma_{radar}^2) \right]^{1/2} \end{aligned} \quad (3.17)$$

Gate threshold  $\gamma$  is taken as 16 ( $4\sigma$  error bound) considering 99.9989% of the measurements to be in the gate as in the original reference of Kirubarajan and BarShalom [52]. Height difference measurement values,  $\delta h_i(k)$ , which are not

valid according to equation (3.17) are eliminated by assigning very large values of  $\delta h_i(k)$ . Hence, in the data association process for the measurements given in equation (A.33), probability of the invalid measurements become zero (i.e.  $e_i \equiv 0$ ) and have no effect in the TAN solution.

### **3.2.2.2.3. State & Covariance Estimation, Update and Prediction**

For the TAN algorithm, generic PDAF equations are directly used using a modification for the definition of the states. Filter equations are given as follows:

INS Error Model:

$$\delta \bar{x}(k+1) = \Phi(k) \cdot \delta \bar{x}(k) + \bar{w}(k) \quad (3.18)$$

where;

$$\delta \bar{x}(k) = [\delta rN; \delta rE; \delta h; \delta vN; \delta vE]^T : \text{ Navigation error states vector,}$$

$\delta rN$  : Northward position error state,

$\delta rE$  : Eastward position error state,

$\delta h$  : Height position error state,

$\delta vN$  : Northward velocity error state,

$\delta vE$  : Eastward velocity error state,

$$\Phi(k) = \begin{bmatrix} 1 & 0 & 0 & T & 0 \\ 0 & 1 & 0 & 0 & T \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} : \text{ State transition matrix with sample time } T .$$

$\bar{w}(k) = [w_{\delta rN}(k), w_{\delta rE}(k), w_{\delta h}(k), w_{\delta vN}(k), w_{\delta vE}(k)]$ : INS error state white noises where  $w_i(k) = N(0, \sigma_i^2)$  with mean zero and variance  $\sigma_i^2$  to the related state.

PDAF Measurement Model:

$$z_i(k) = H_m(k) \cdot \delta \bar{x}_i(k) + \bar{w}_{meas}(k) \quad (3.19)$$

where;

$z_i(k) = \delta h_{i_{ave}}(k)$ : Average of the height difference at position related to index  $i$ ,

$H_m(k) = [0 \ 0 \ 1 \ 0 \ 0]$ : Height measurement matrix,

$w_{meas}(k) = N(0, \sigma_{radar}^2)$ : Measurement White Noise with mean zero and variance  $\sigma_{radar}^2$ .

After defining system and measurement models, PDAF equations can be implemented for TAN. PDAF equations contain Kalman filter equations with

association probabilities. For PDAF equations, innovation form of Kalman filter equations are used.

Propagation (Prediction) Equations:

$$\delta\hat{x}(k | k-1) = \Phi(k-1) \cdot \delta\hat{x}(k-1 | k-1) \quad (3.20)$$

$$P(k | k-1) = \Phi(k-1) \cdot P(k-1 | k-1) \cdot \Phi(k-1)^T + Q(k-1) \quad (3.21)$$

where;

$P(k)$ : State covariance matrix,

$$Q(k) = \begin{bmatrix} \sigma_{rN}^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{rE}^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{h_{INS}}^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{vN}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{vE}^2 \end{bmatrix} : \text{System noise covariance matrix}$$

State and Covariance Update:

$$S(k) = H_p(k) \cdot P(k | k-1) \cdot H_p(k)^T + R(k) \quad (3.22)$$

$$K(k) = P(k | k-1) \cdot H_p(k)^T \cdot S(k)^{-1} \quad (3.23)$$

$$\delta\hat{x}(k | k) = \delta\hat{x}(k | k-1) + K(k) \cdot v_p(k) \quad (3.24)$$

where;

$S(k)$ : Innovation covariance matrix,

$K(k)$ : PDAF filter gain,

$$H_p(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}; \quad \text{PDAF measurement matrix,}$$

$$v_p(k) = \sum_{i=0}^{m(k)} \beta_i(k) \cdot v_{P_i}(k); \quad \text{Combined PDAF innovation,}$$

$$v_{P_i}(k) = z_{P_i}(k) - \hat{z}_{P_i}(k | k-1); \quad \text{PDAF innovation states,}$$

$$z_{P_i}(k) = [\delta rN_i; \delta rE_i; \delta h_{t_{ave}}]; \quad \text{PDAF measurement states,}$$

$$\hat{z}_{P_i}(k | k-1) = H_p(k) \cdot \hat{\delta x}(k | k-1); \quad \text{Measurement state estimation,}$$

$$R(k) = \begin{bmatrix} \sigma_{rN}^2 & 0 & 0 \\ 0 & \sigma_{rE}^2 & 0 \\ 0 & 0 & \sigma_{radar}^2 \end{bmatrix}; \quad \text{PDAF measurement noise covariance matrix}$$

Here, it should be noted that PDAF measurement states  $z_{P_i}(k)$  are different from the actual measurement states  $z_i(k)$ . Since, only height difference measurements are taken into account, position updates are not available with the height measurement matrix  $H_m(k)$ . As a result of this, PDAF measurement matrix  $H_p(k)$  is defined in order to make position corrections with the same filter gains of height corrections.

PDAF measurement states  $z_{P_i}(k)$  are defined such that northward and eastward positions are calculated according to the index  $i$  as shown in Figure 42. Moreover, due to position inaccuracies along horizontal coordinates, horizontal position white noises  $\sigma_{rN}^2$  and  $\sigma_{rE}^2$  terms are added in the measurement noise covariance matrix. Height difference measurements  $z_i(k)$  are used for the determination of the conditional probability of the event  $\beta_i(k)$  instead of  $z_{P_i}(k)$ ; since, positions are not actually measured. With the use of  $z_{P_i}(k)$ , filter gains obtained for height difference states are directly used for position states.

For the position error definitions, consider Figure 42 again. As it was given in equation (3.13), INS position index is at  $i_{INS} = \frac{1}{2}[m(k)+1]$  for a DTED grid size of  $m(k)$  where the grid is selected as a square. At the INS position index  $i_{INS}$ ,  $\delta rN_{INS} = 0$  and  $\delta rE_{INS} = 0$ ; since no position correction exists for the INS position. Using geometrical relations for the DTED grid index and horizontal positions, following definitions can be done for horizontal position errors:

$$\delta rN_i = \left( i + \sqrt{m(k)} \cdot \left[ 1 - \text{ceil}\left(\frac{i}{\sqrt{m(k)}}\right) \right] - \frac{\sqrt{m(k)}+1}{2} \right) \cdot d_y \quad (3.25)$$

$$\delta rE_i = \left( \text{ceil}\left(\frac{i}{\sqrt{m(k)}}\right) - \frac{\sqrt{m(k)}+1}{2} \right) \cdot d_x \quad (3.26)$$

where;

$\delta rN_i$ : Northward position error at index  $i$ ,

$\delta rE_i$ : Eastward position error at index  $i$ ,

$m(k)$ : DTED grid size,

$\text{ceil}(X)$ : Function which rounds the element of  $X$  to the nearest integer towards infinity.

$d_y$ : DTED spacing along latitude direction,

$d_x$ : DTED spacing along longitude direction.

Horizontal position error definitions are shown in Figure 43.

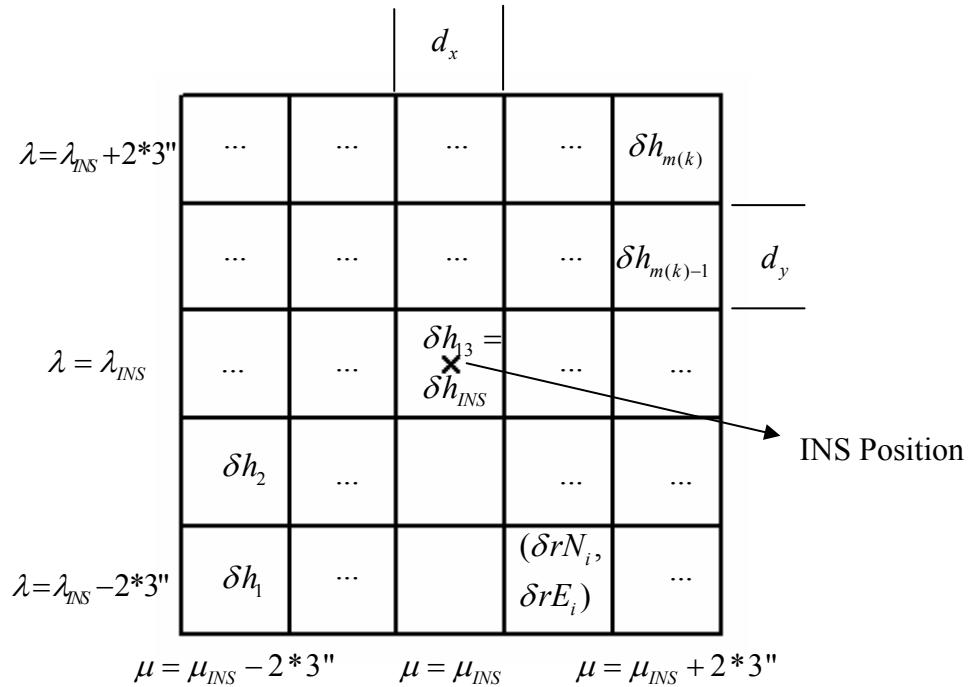


Figure 43. Horizontal Position Error Definitions

As an example, consider the horizontal position errors at index  $i = 2$  for 5x5 DTED grid size as shown in Figure 43. Using equations (3.25) and (3.26), horizontal position errors can be found as follows:

$$\delta rN_{i=2} = \left( 2 + \sqrt{25} \cdot \left[ 1 - \text{ceil}\left(\frac{2}{\sqrt{25}}\right) \right] - \frac{\sqrt{25}+1}{2} \right) \cdot d_y = -1 \cdot d_y$$

$$\delta rE_{i=2} = \left( \text{ceil}\left(\frac{2}{\sqrt{25}}\right) - \frac{\sqrt{25}+1}{2} \right) \cdot d_x = -2 \cdot d_x$$

As it can be seen from the calculated results, horizontal position errors, which are determined from equations (3.25) and (3.26), can be directly used for INS position updates. Therefore, by determining the index of the position from TAN algorithms, INS error model can be updated. Moreover, as it was stated earlier, horizontal position white noises  $\sigma_{rN}^2$  and  $\sigma_{rE}^2$  terms are added in the measurement noise covariance matrix in order to model position inaccuracies along horizontal coordinates.

For the state covariance update, equations (A.15) to (A.17) are used. However, the conditional probability of the false events is zero (i.e.  $\beta_0(k) \equiv 0$ ) for the TAN application which will be explained in the following section. Moreover, for the spread of the innovations term  $\tilde{P}(k)$ , only the measured state  $z_i(k)$  will be considered. Hence:

$$P(k | k) = P(k | k-1) - K(k) \cdot S(k) \cdot K(k)^T + \tilde{P}(k) \quad (3.27)$$

$$\tilde{P}(k) \cong K(k) \cdot \left[ \sum_{i=0}^{m(k)} \beta_i(k) \cdot v_i(k) \cdot v_i(k)^T - v(k) \cdot v(k)^T \right] \cdot K(k)^T \quad (3.28)$$

where;

$$v(k) = \sum_{i=0}^{m(k)} \beta_i(k) \cdot v_i(k) : \quad \text{Combined height difference innovation,}$$

$$v_i(k) = z_i(k) - \hat{z}_i(k | k-1)$$

$$v_i(k) = \delta h_{i_{ave}}(k) - H(k) \cdot \delta \hat{x}(k | k-1) : \text{Height difference innovation states}$$

Here, it should be noted that  $\tilde{P}(k)$  term will be effected only from the height channel. Therefore, in state covariance matrix  $P(k | k)$ ,  $\tilde{P}(k)$  term will be added only to the height channel.

#### **3.2.2.2.4. The Probabilistic Data Association**

Association probabilities are calculated considering the height differences used in parametric PDA equation (A.32). In this equation, determination of  $P_D$  and  $P_G$  parameters are critical. Probability of detection of a target originated measurement  $P_D$  must be one; since the height difference measurements grid is formed virtually from the related DTED within the  $3\sigma$  horizontal error bound of the vehicle position. If no measurements are taken, then the height measurement grid could not be formed. Probability of measurements in the gate  $P_G$  is also taken one

considering the used DTED grid is the  $3\sigma$  horizontal error bound of the vehicle obtained from the quality of the INS. Actually measurement gating process eliminates the impossible height difference solutions according to equation (3.17) derived. Therefore:

$$P_D = 1, P_G = 1 \quad (3.29)$$

Then from equation (A.34),  $b \leq 0$  and equation (A.32) becomes:

$$\beta_i(k) = \begin{cases} \frac{e_i}{\sum_{j=1}^{m(k)} e_j} & i = 1, \dots, m(k) \\ 0 & \text{otherwise} \end{cases} \quad (3.30)$$

where;

$$e_i \equiv e^{-\frac{1}{2}v_i(k)^T S(k)^{-1} v_i(k)} : \quad \text{Given in equation (A.33)}$$

### 3.2.2.2.5. Summary of PDAF Equations for TAN

PDAF equations derived in the previous sections for real-time TAN application is summarized in Table 15. Initialization of Kalman filters is generally done by setting state covariance matrix as a coefficient of system covariance matrix:

$$P_0 = \alpha^2 \cdot Q(k), \quad (\text{typically } \alpha = 10) \quad [65] \quad (3.31)$$

Table 15. PDAF Equations for TAN Process

INS Error Model:	
$\delta\bar{x}(k+1) = \Phi(k) \cdot \delta\bar{x}(k) + \bar{w}(k)$	given in (3.18)
$\bar{w}(k) = N(\bar{0}, Q(k))$	
System Noise Covariance Matrix:	
$Q(k) = \text{Cov}\{\bar{w}(k)\bar{w}(k)^T\}$	given in (3.21)
PDAF Measurement Model:	
$z_i(k) = H_m(k) \cdot \delta\bar{x}_i(k) + \bar{w}_{\text{meas}}(k)$	given in (3.19)
$w_{\text{meas}}(k) = N(0, \sigma_{\text{radar}}^2)$	
Association Probabilities:	
$\beta_i(k) = \begin{cases} \frac{e_i}{\sum_{j=1}^{m(k)} e_j} & i = 1, \dots, m(k) \\ 0 & \text{otherwise} \end{cases}$	given in (3.30)
Initial Conditions:	
$\delta\bar{x}_0 = N(\hat{\delta\bar{x}}_0, P_0)$	(3.32)
Other Assumptions:	
$E[\bar{w}(k) \cdot \bar{w}_{\text{meas}}(k)^T] = 0 \text{ for all } k$ (Measurements are independent)	(3.33)
State Estimate Propagation:	
$\hat{\delta\bar{x}}(k   k-1) = \Phi(k-1) \cdot \hat{\delta\bar{x}}(k-1   k-1)$	given in (3.20)
Error Covariance Propagation:	
$P(k   k-1) = \Phi(k-1) \cdot P(k-1   k-1) \cdot \Phi(k-1)^T + Q(k-1)$	given in (3.21)
PDAF Gain Matrix:	
$K(k) = P(k   k-1) \cdot H_p(k)^T \cdot S(k)^{-1}$	given in (3.23)
State Estimate Update:	
$\hat{\delta\bar{x}}(k   k) = \hat{\delta\bar{x}}(k   k-1) + K(k) \cdot v_p(k)$	given in (3.24)

Table 15. PDAF Equations for TAN Process (Continued)

Error Covariance Update:	
$P(k   k) = P(k   k-1) - K(k) \cdot S(k) \cdot K(k)^T + \tilde{P}(k)$	given in (3.27)
$\tilde{P}(k) \cong K(k) \cdot \left[ \sum_{i=0}^{m(k)} \beta_i(k) \cdot v_i(k) \cdot v_i(k)^T - v(k) \cdot v(k)^T \right] \cdot K(k)^T$	given in (3.28)

### 3.2.2.3. Discussion of Real-time PDAF Implementation for TAN

In this section, implemented PDA method for TAN will be discussed. The advantages of the PDA approach for TAN solution can be summarized as follows:

1. Real-time TAN solution can be obtained with a single PDA filter.
2. PDA filter can be used for both batch and recursive TAN solution.  
For batch solution, larger grid size is selected for navigation solution.  
For recursive solution, horizontal positions are calculated recursively in relatively small DTED grids.
3. Since past measurements are taken into account, smoothing of the measurements in the filter is achieved which decreases errors.
4. Since INS error model is used for navigation solution, application of the filter is simple and the filter is linear.
5. Batch size of the DTED area concerned can be changed. Both larger DTED areas for acquisition mode or smaller DTED areas for tracking modes can be selected using the same filter.

6. Results of the filter are good for both recursive and batch algorithms. The results compared with SITAN and TERCOM algorithms will be discussed in the simulations section.

The difference of the PDA approach from Qingtang, et al [40] is also summarized as follows:

1. In the paper of Qingtang, et al [40], TAN using PDAF was investigated for the batch algorithm. The motion of the vehicle is not modeled.
2. In the paper of Qingtang, et al [40], the batch algorithm obtained using PDAF actually uses maximum likelihood approach as used for TERCOM. Therefore, association probabilities  $\beta_i$  are calculated with the help of the MSD function used in TERCOM.
3. In the paper of Qingtang, et al [40], performance of the TAN using PDA and TERCOM has been compared. It is stated that PDA was used in order to improve the performance of TAN compared to TERCOM.
4. In the Ph.D. study, real-time PDAF implementation is done. By using the error model of the INS used in the vehicle, system dynamics is modeled. Using PDAF, error states of the system are estimated.
5. In the Ph.D. study, PDAF equations are directly implemented for the TAN solution. Association probabilities obtained from height difference measurements for each element of the DTED grid concerned are used for position updates considering the index of the DTED grid.

Simulation results of the implemented PDA filter will be presented at the “Simulations” section of the chapter.

### **3.3. Multiple Hypothesis Tracking (MHT) and Track Splitting Filter (TSF)**

#### **3.3.1. Theory**

In classical multiple-target tracking, the problem is divided into two steps, association and estimation. Step 1 associates contacts with targets. Step 2 uses the contacts associated with each target to produce an estimate of that target’s state. Complications arise when there is more than one reasonable way to associate contacts with targets. The classical approach to this problem is to form association hypotheses and to use MHT. In this approach, alternative hypotheses are formed to explain the source of the observations. Each hypothesis assigns observations to targets or false alarms. For each hypothesis, MHT computes the probability that it is correct. This is also the probability that the target state estimates that result from this hypothesis are correct. Most MHT algorithms display only the estimates of target state associated with the highest probability hypothesis [68].

The model used for the MHT problem is a generalization of the recursion for general multiple-hypothesis tracking. This recursion applies to problems that are nonlinear and non-Gaussian as well as to standard linear Gaussian situations. In this general case, the distributions on target state may fail to be independent of one another (even when conditioned on an association hypothesis) and may require a joint state space representation. This recursion includes a conceptually simple Bayesian method of computing association probabilities [68].

Numerous books and articles on multiple-target tracking examine in detail the many variations and approaches to MHT problem. Many of these discuss the practical aspects of implementing multiple target trackers and compare approaches. In addition to the full or classical MHT as defined by “Reid” and “Mori et al.”, a number of approximations are in common use for finding solutions to tracking problems. Examples include joint probabilistic data association and probabilistic MHT [68].

Multiple hypotheses tracking (MHT) is a deferred decision logic in which alternative data association hypotheses are formed whenever there are observation to track conflict situations. Then, rather than combining these hypotheses, as in the JPDA method, the hypotheses are propagated in anticipation that subsequent data will resolve the uncertainty [69].

The original MHT method, denoted Reid’s algorithm, was first presented by Reid [70]. There are two basic approaches to MHT implementation. The first (hypothesis-oriented) approach follows the original work of Reid [70]. It maintains the hypothesis structure from scan to scan and continually expands and cuts back (prunes) the hypotheses as new data are received. At each scan, a set of hypotheses will be carried over from the previous scan and composed of one or more tracks that are compatible with all other tracks in the hypothesis. Compatible tracks are defined to be tracks that do not share any common observations. Then, on the receipt of new data, each hypothesis is expanded into a set of new hypotheses by considering all observation-to-track assignments for the tracks within the hypothesis. Again, as new hypotheses are formed, the compatibility constraint for tracks within a hypothesis is maintained [69].

An alternative (track-oriented) approach [71] does not maintain hypotheses from scan to scan. The tracks formed on each scan are reformed into hypotheses and the tracks that survive pruning are predicted to the next scan where the process continues [69].

In Figure 44 and Figure 45, the operations of MHT that are required by both implementation methods are summarized.

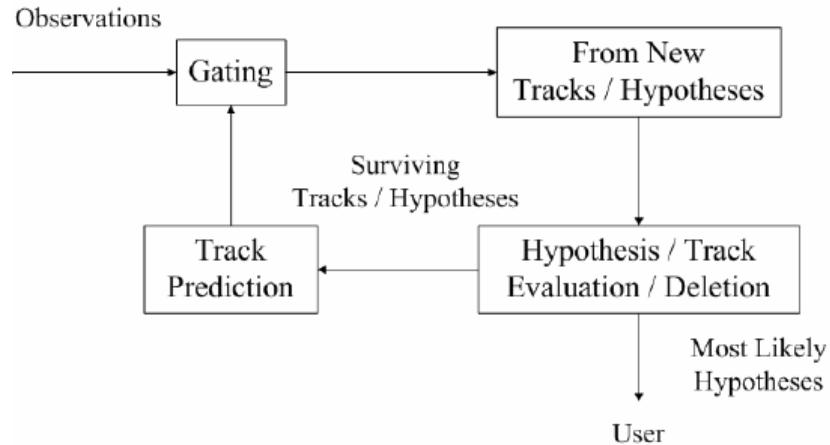


Figure 44. MHT Logic Overview [69]

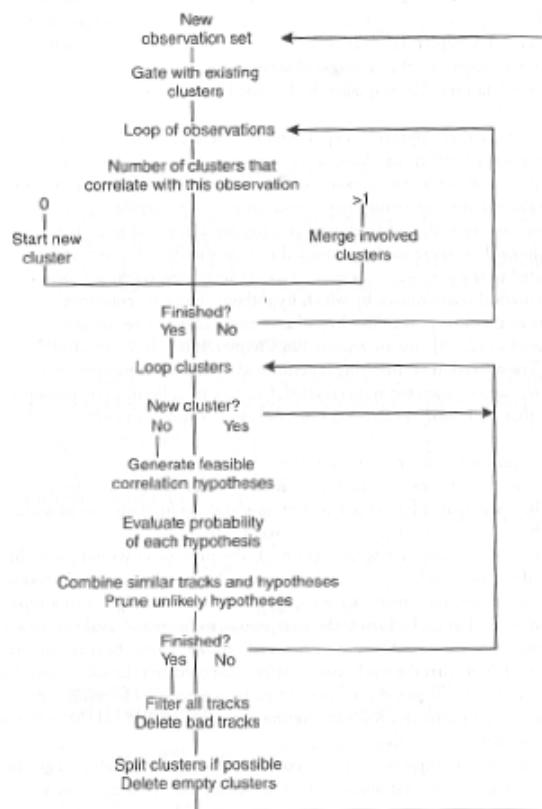


Figure 45. High-level Flow Chart of MHT Algorithm [69]

The Multiple Hypothesis Tracking (MHT) filter maintains separate tracks for each possible associated observation. At each time step, the predicted observation is used to establish a validation gate and for each measurement that is found in this validation gate, a new hypothesis track is generated. Thus a single track is split into “n” tracks, one associated with each valid measurement, plus one track (usually denoted 0) for the no-association hypothesis [65].

Each of these new tracks is then treated independently and used to generate new predictions for the next time step. Since the number of branches into which the track is split can grow exponentially, the likelihood function of each split track is computed and the unlikely ones are discarded. The MHT algorithm works on complete sequences of observations [65].

In MHTF, every validated observation  $z_p(k)$  is used to establish a new track,  $\hat{x}_p(k|k)$ . In addition the “false alarm” and/or “missed observation” hypothesis also generates a track,  $\hat{x}_0(k|k)$ . These tracks are propagated forward to the next gate and again each track is associated with each valid observation,  $z_q(k+1)$  and the tracks are again split into tracks associated with each possible pair-wise association,  $\hat{x}_{pq}(k+1|k+1)$ . Probabilities or likelihoods,  $\lambda_{pq}$  of correct track histories are maintained to prune the resulting hypothesis tree [65].

In Figure 46, MHTF implementation is shown.

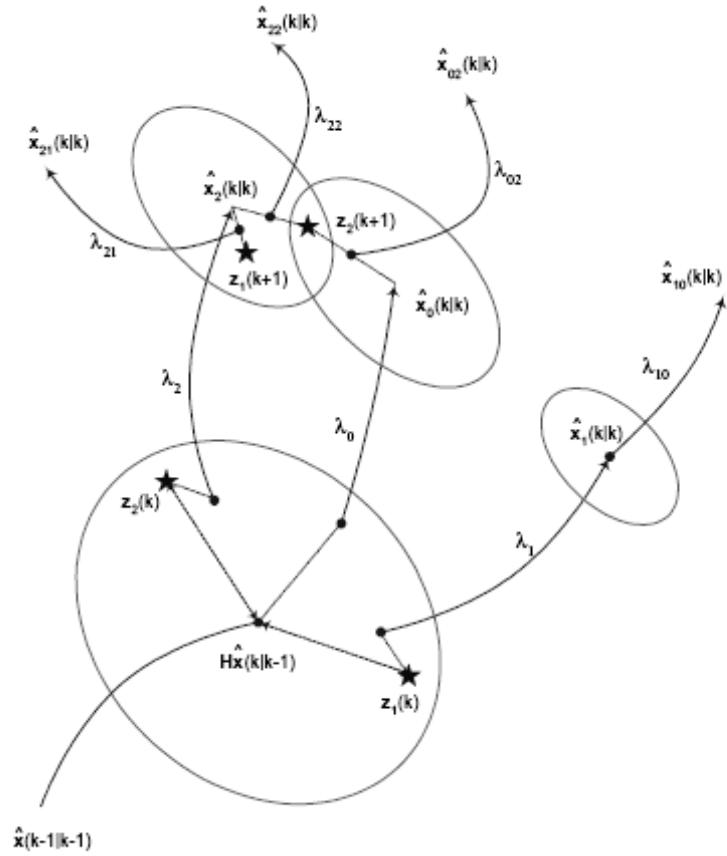


Figure 46. MHTF Implementation [65]

Then, the following approach can be implemented in order to perform MHTF algorithm [65]:

1. A predicted observation and validation gate are computed.
2. All validated observations are associated plus no-association hypothesis.
3. The track is updated separately with each validated hypothesis.
4. A likelihood associated with correct association is computed.

5. The likelihood of the each entire track sequence is computed.
6. Some pruning of the hypothesis tree may take place.
7. Each track hypothesis is now independently predicted forward to the next time-step.
8. Producing as many new tracks as associated measurements (plus no track solution).
9. The process repeats.

There are three points to note about the MHT and TSF algorithm [65]:

1. A unity detection probability (no missing data) is assumed.
2. Likelihood pruning method does not work well with long measurement sequences as it becomes dominated by old measurements. One “hack” around this is to use a fading-memory window.
3. Method is dominated by computational and memory requirements of the splitting algorithm.

MHT algorithm is good in situation with low clutter rates but high track uncertainty (crossing tracks, maneuvering targets, etc). Practically, the algorithm is dominated by the approach used for pruning unlikely target hypotheses [65].

TSF is proposed by Smith and Buechler [72] and older than the original MHT method presented by Reid [70]. In TSF, a tree of hypotheses is kept for each target individually, and a maximum likelihood criterion is used to prune the tree. On the other hand, Reid's MHT constructs a tree of all possible hypotheses, including all possible new track initiations at every time step. Reid discusses a number of strategies to prune the tree in order to achieve reasonable computation times. In the Ph.D. study, TSF is implemented for TAN due to INS error model characteristics. Since, horizontal INS error bound is estimated for the cruise missile and errors do not change rapidly, implementation of TSF for TAN became sufficient for navigation solution.

### 3.3.2. Implementation of TSF to TAN

As it was stated in the previous section, TSF is an older method than the original MHT method. TSF is a recursive branching algorithm for multiple-object discrimination and tracking consists of a bank of parallel filters of the Kalman form, each of which estimates a trajectory associated with a certain selected measurement sequence. The measurement sequences processed by the algorithm are restricted to a tractable number by combining similar trajectory estimates, by excluding unlikely measurement/ state associations, and by deleting unlikely trajectory estimates. The measurement sequence selection is accomplished by threshold tests based on the innovations sequence and state estimates of each filter [72].

TSF and MHT methods are similar except for hypotheses formation. Consider the TSF and MHT approach given in the previous section for TAN again [65]:

1. A predicted observation and validation gate are computed.

- Measurement gate is taken as the  $3\sigma$  horizontal error bound of the INS and the invalid possibilities for the height differences  $\delta h_i(k)$  are discarded.
2. All validated observations are associated plus no-association hypothesis.
    - Every grid position (i.e. index) in the  $3\sigma$  horizontal error bound of the INS is considered to be one of the possible navigation solutions.
  3. The track is updated separately with each validated hypothesis.
  4. A likelihood associated with correct association is computed.
  5. The likelihood of the each entire track sequence is computed.
    - Navigation solution is assumed to be one of the grid positions in the  $3\sigma$  horizontal error bound of the INS. According to the index of the grid position, there exist “n x n” possible tracks (i.e. hypothesis) for each time step where “n x n” denotes the batch size of the DTED considered. The likelihood of the each possible track sequence is computed.
  6. Some pruning of the hypothesis tree may take place.
    - Number of possible tracks is limited considering INS error characteristics. According the small position error changes of the INS for small periods of time where the TAN algorithm is applied, it is assumed that possible tracks are in the  $3\sigma$  horizontal error bound of the INS where each

track follows the grid position from INS position to all grid positions in the INS horizontal error bound. Then, tracks with minimum likelihoods are selected for the navigation solution. Hence, hypotheses are pruned.

7. Each track hypothesis is now independently predicted forward to the next time-step.

- Navigation solution is found for each possible track using standard Kalman filter equations as given in the reference papers for MHT/ TSF procedure. Using a definite number of minimum likelihood values of the entire track sequences, navigation solution is achieved.

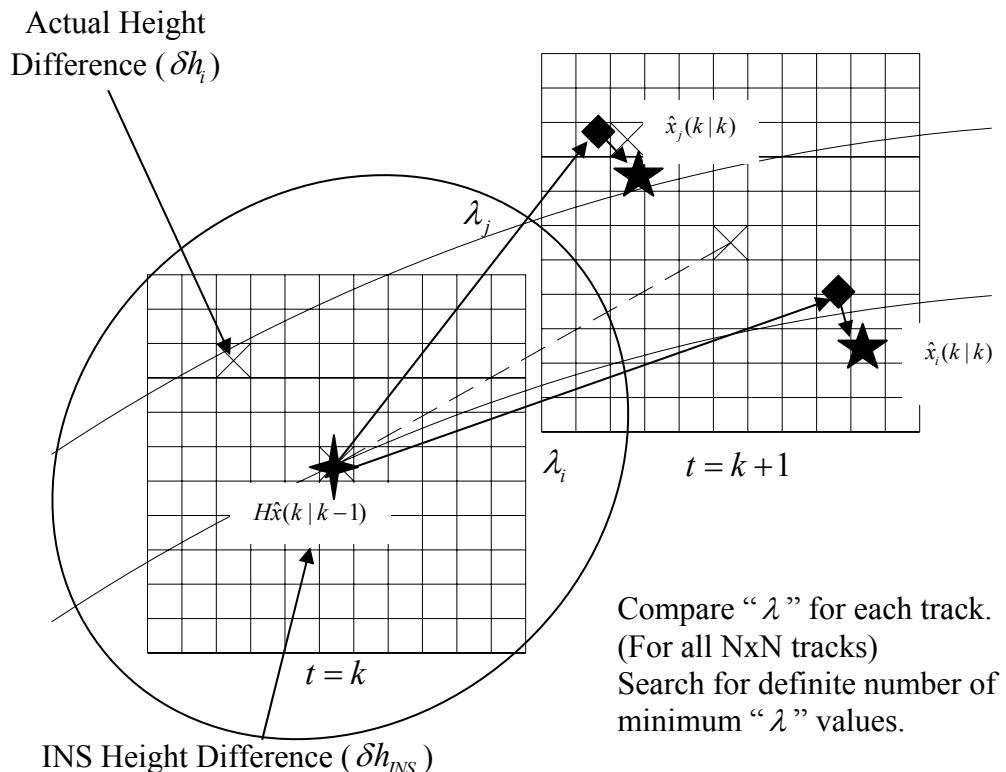


Figure 47. MHTF Implementation for TAN for a Single Time Step

Details of the TSF procedure applied for TAN and derivation of the TSF equations are discussed in detail in the following section. In Figure 47, TSF application for TAN for a single time step is shown.

### 3.3.3. TSF Equations Implemented for TAN

TSF maintains separate tracks for each possible associated observation. At each time step, the predicted observation is used to establish a validation gate and for each measurement that is found in this validation gate, a new hypothesis track is generated. Thus a single track is split into “n” tracks, one associated with each valid measurement, plus one track (usually denoted 0) for the no-association hypothesis [65].

Each of these new tracks is then treated independently and used to generate new predictions for the next time step. Since the number of branches into which the track is split can grow exponentially, the likelihood function of each split track is computed and the unlikely ones are discarded [65].

The TSF procedure works as follows [65]:

1. The TSF algorithm works on complete sequences of observations.
2. The probability that a given branch sequence of observations (from root to leaf) is correct.
3. The l'th sequence of measurements up to time k:

$$Z^{kl} \equiv \{z_{i_{l,l}}(1), \dots, z_{i_{k,l}}(k)\} \quad (3.34)$$

4.  $\Theta^{k,l}$  is the event that the sequence  $Z^{kl}$  is a correct track.

5. Then the likelihood function for this event is clearly:

$$\Lambda(\Theta^{k,l}) = P(Z^{k,l} | \Theta^{kl}) = P(z_{i_{1,l}}(1), \dots, z_{i_{k,l}}(k) | \Theta^{kl}) \quad (3.35)$$

6.  $Z^k$  the cumulative set of all measurements up to time k:

$$\Lambda(\Theta^{k,l}) = \prod_{j=1}^k P(z_{i_{j,l}} | Z^{j-1} \Theta^{k,l}) \quad (3.36)$$

7. Linear and Gaussian distribution for the likelihood function is assumed:

$$\Lambda(\Theta^{k,l}) = c_k \cdot \exp \left[ -\frac{1}{2} \sum_{j=1}^k v^T(j) \cdot S^{-1}(j) \cdot v(j) \right] \quad (3.37)$$

where,

$v(j) = z(j) - \hat{z}(j | j-1)$ : Innovation between track and measurement

8. Modified log-likelihood function is defined as:

$$\lambda(k) \equiv -2 \cdot \log \left[ \frac{\Lambda(\Theta^{k,l})}{c_k} \right] = \sum_{j=1}^k v^T(j) \cdot S^{-1}(j) \cdot v(j) \quad (3.38)$$

9. Modified log-likelihood function is recursively computed from:

$$\lambda(k) = \lambda(k-1) + v^T(k) \cdot S^{-1}(k) \cdot v(k) \quad (3.39)$$

10. Each track is updated using standard Kalman filter equations.
11. A “goodness of fit” and test for accepting a track is that  $\lambda(k) < d$ .
12. Definite numbers of tracks are accepted for each time step.

As it can be seen from the TSF procedure, for each track, standard Kalman filter equations are used. Hence, PDAF state estimation, state and covariance update and prediction equations which are given in the previous sections are used in order to include horizontal position inaccuracies to the navigation solution.

The critical part of the TSF procedure implemented for TAN is the track formation and track pruning steps. These steps are summarized as follows:

1. Navigation solution is assumed to be one of the grid index followed by some of the tracks in the  $3\sigma$  horizontal error bound of the INS. According to the index of the grid position in the  $3\sigma$  horizontal error bound, there exist “n x n” possible navigation solutions where “n x n” denotes the batch size of the DTED considered.
2. Possible tracks are different from grid indices. At the initial time step, there exist “n x n” possible tracks from INS position grid to all possible grid positions as shown in Figure 48. The modified log-likelihood of the each possible track sequence is computed from equation (3.38) as follows:

$$\lambda_i(k) \equiv -2 \cdot \log \left[ \frac{\Lambda_i(\Theta^{k,l})}{c_k} \right] = \sum_{j=1}^k v_{M_i}^T(j) \cdot S^{-1}(j) \cdot v_{M_i}(j)$$

for  $i = 1 \dots n \times n$

where,

$m(k)$ : DTED Grid Size (DTED grid is taken as a square.)

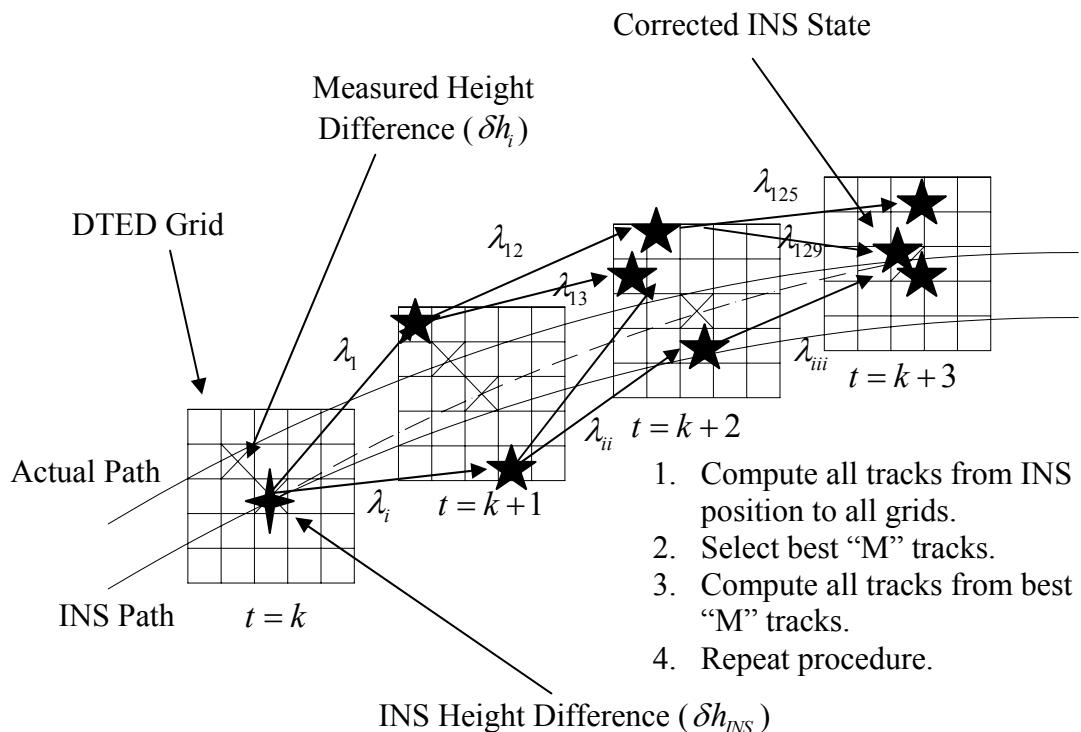


Figure 48. TSF Track Formation and Pruning

3. At the following time step, best "M" tracks of the existing " $n \times n$ " possible tracks are selected. Hence, hypothesis is pruned. In order to apply pruning, several methods can be applied. Using  $\lambda_i(k) < d$  for

accepting tracks or selecting minimum  $\lambda_i(k)$  values for  $i=1\dots M$  can be applied. Then, “n x n” possible tracks from “M” accepted tracks to all possible grid positions are selected. Modified log-likelihood function is recursively computed from equation (3.39) for each formed new track as follows:

$$\lambda_{ij}(k+1) = \lambda_{ij}(k) + v_{M_{ij}}^T(k+1) \cdot S^{-1}(k+1) \cdot v_{M_{ij}}(k+1)$$

for  $i = 1\dots M, j = 1\dots "n \times n"$

4. Procedure defined at step 3 is done recursively in order to obtain navigation solution.
5. For real-time navigation solution, accepted tracks can be used in several methods. Selecting and using the results of the best track which gives minimum likelihood is actually a dynamic programming method which was discussed for Viterbi algorithm. On the other hand, using mean value of the selected tracks’ results can also be used which is actually a kind of data association process discussed for PDA algorithm. However, from the simulations which will be discussed in the following sections, all tracks converge to the same index of the DTED grid. Actually, for sufficiently rough surfaces this is the expected result of the TSF.
6. In order to decrease the effects of the old measurements, modified log-likelihood function defined in equation (3.39) can be used by a weighting factor as follows:

$$\lambda_{ij}(k+1) = K_{WF} \cdot \lambda_{ij}(k) + v_{M_{ij}}^T(k+1) \cdot S^{-1}(k+1) \cdot v_{M_{ij}}(k+1)$$

for  $i = 1\dots M, j = 1\dots "n \times n"$  (3.40)

TSF equations derived for TAN application is summarized in Table 16. Initialization of Kalman filters is again done by setting state covariance matrix as a coefficient of system covariance matrix given in equation (3.31).

As it can be seen from Table 16, standard Kalman filter equations are used for each track considering horizontal position error inaccuracies. Track formation, pruning and real-time solutions are also summarized.

Table 16. TSF Equations for TAN Process

INS Error Model:	
$\delta\bar{x}(k+1) = \Phi(k) \cdot \delta\bar{x}(k) + \bar{w}(k)$	given in(3.18)
$\bar{w}(k) = N(\bar{0}, Q(k))$	
System Noise Covariance Matrix:	
$Q(k) = Cov\{\bar{w}(k)\bar{w}(k)^T\}$	given in (3.21)
TSF Measurement Model:	
$z_i(k) = H_m(k) \cdot \delta\bar{x}_i(k) + \bar{w}_{meas}(k)$	given in (3.19)
$w_{meas}(k) = N(0, \sigma_{radar}^2)$	
Measurement Noise Covariance Matrix:	
$R(k) = Cov\{\bar{w}_{meas}(k)\bar{w}_{meas}(k)^T\}$	given in (3.30)
Initial Conditions:	
$\delta\bar{x}_0 = N(\hat{\delta\bar{x}}_0, P_0)$	given in (3.32)
Other Assumptions:	
$E[\bar{w}(k) \cdot \bar{w}_{meas}(k)^T] = 0$ for all $k$ (Measurements are independent)	given in (3.33)
State Estimate Propagation (for each track):	
$\hat{\delta\bar{x}}(k   k-1) = \Phi(k-1) \cdot \hat{\delta\bar{x}}(k-1   k-1)$	given in (3.20)

Table 16. TSF Equations for TAN Process (Continued)

Error Covariance Propagation (for each track):	
$P(k   k-1) = \Phi(k-1) \cdot P(k-1   k-1) \cdot \Phi(k-1)^T + Q(k-1)$	given in (3.21)
TSF Gain Matrix (for each track):	
$K(k) = P(k   k-1) \cdot H_p(k)^T \cdot S(k)^{-1}$	given in (3.23)
State Estimate Update (for each track):	
$\hat{\delta\bar{x}}(k   k) = \hat{\delta\bar{x}}(k   k-1) + K(k) \cdot v_p(k)$	given in (3.24)
Error Covariance Update (for each track):	
$P(k   k) = [I - K(k) \cdot H(k)] \cdot P(k   k-1)$	given in (2.39)
Modified Log-likelihood Function (Initial Track Formation):	
$\lambda_i(k) \equiv -2 \cdot \log \left[ \frac{\Lambda_i(\Theta^{k,l})}{c_k} \right] = \sum_{j=1}^k v_{M_i}^T(j) \cdot S^{-1}(j) \cdot v_{M_i}(j)$ for $i = 1 \dots "n \times n"$	given in (3.38)
Recursively Computed Modified Log-likelihood Function:	
$\lambda_{ij}(k+1) = K_{WF} \cdot \lambda_{ij}(k) + v_{M_{ij}}^T(k+1) \cdot S^{-1}(k+1) \cdot v_{M_{ij}}(k+1)$ for $i = 1 \dots M, j = 1 \dots "n \times n"$	given in (3.39)
Track Pruning:	
Select best "M" tracks from "M x n x n" tracks such that; $\lambda_{ij}(k+T) = \min \lambda_{ij}(k+T)$ for $ij = 1 \dots M$	(3.41)
State Estimate Propagation (for all tracks):	
$\hat{\delta\bar{x}}(k   k-1) = \text{average} [\hat{\delta\bar{x}}_i(k   k-1)]$ for $i = 1 \dots M$	(3.42)

### 3.3.4. Discussion of TSF Implementation for TAN

In this section, implemented TSF method for TAN will be discussed. The advantages of the TSF approach for TAN solution can be summarized as follows:

1. Real-time TAN solution can be obtained with parallel TSF's.

2. Application of the filter is more complex than PDAF but the filter is again linear since INS error model is used.
3. Batch size of the DTED area concerned can be changed. Both larger DTED areas for acquisition mode or smaller DTED areas for tracking modes can be selected using the same TSF structure.
4. TSF gives solutions for various tracks selected. Actually, all tracks converge to the same index of the DTED grid (i.e. solution grid). However, for smooth terrains, there exist more than one position solution index and the tracks can be investigated separately in order to give more than one but finite number of navigation solutions.
5. Results of the filter are good for both recursive and batch algorithms. The results compared with SITAN and TERCOM algorithms will be discussed in the simulations section.

Simulation results of the implemented TSF will be presented at the “Simulations” section of the chapter.

### **3.4. Simulations**

#### **3.4.1. Simulation Model Development**

After the implementation of PDAF and TSF for TAN, the algorithms are tested using simple kinematic models. Kinematic models are prepared considering

the mid-course flight of a cruise missile with constant heading and velocity motion at constant altitude.

Simulations for the TAN models are done using Simulink [58]. In order to perform simulations, first trajectory and INS kinematic models are formed. Then, DTED database model is prepared. Finally, SITAN, TERCOM, PDAF and TSF models are formed. Finally, the overall architecture is formed in order to perform simulations for position errors along east and north directions of the vehicle motion.

In Figure 49, a general Simulink model for the studied TAN models is given. Loosely coupled integration structure is used for TAN models where INS is not updated at each TAN correction step but updated at a greater period. This is done in order not to influence INS results from possible fault corrected TAN solutions. Details of the simulation sub-models will be given in the following sections.

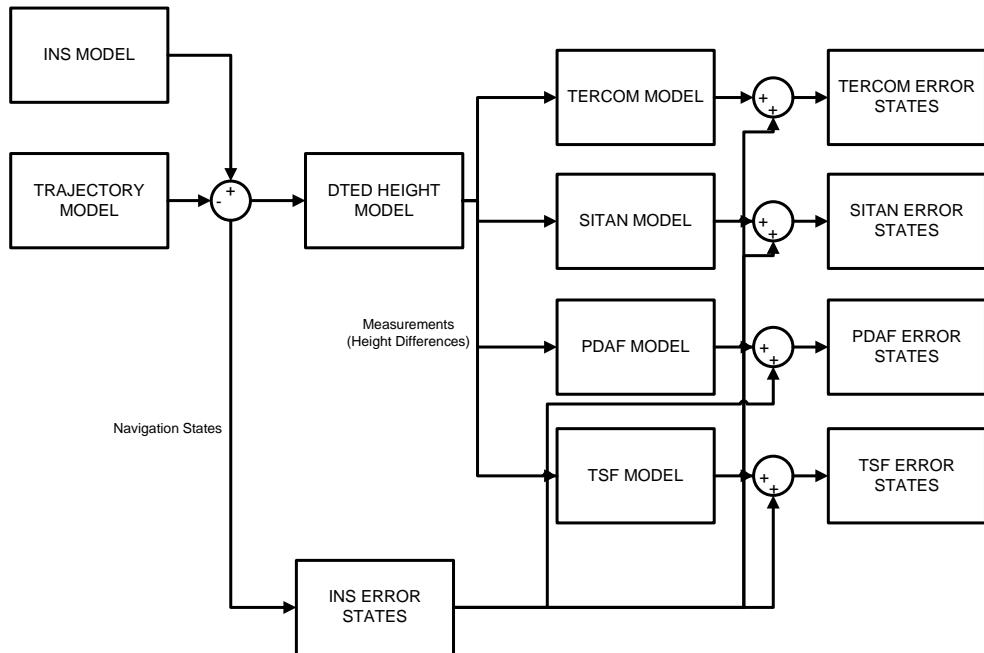


Figure 49. General Simulink Model for TAN Models

### 3.4.1.1. Trajectory and INS Model

For the simulations, the motion of the vehicle is modeled considering the mid-course flight of a cruise missile with constant heading and velocity motion at constant altitude. Actually, this assumption is almost valid for mid-course flight of a generic cruise missile. For the INS model, white noise terms are added to velocity and position terms. Since the height terms will be taken from the DTED database according to vehicle's latitude and longitude (i.e. horizontal positions), height is not considered in the vehicle's state.

Trajectory model of the vehicle considering continuous states can be modeled as follows:

$$\dot{\bar{x}}_{traj}(t) = F(t) \cdot \bar{x}_{traj}(t) \quad (3.43)$$

where,

$$\bar{x}_{traj}(t) = [rN_{traj}; rE_{traj}; h_{traj}; vN_{traj}; vE_{traj}; vD_{traj}]^T$$

$rN_{traj}$  : Northward position of the vehicle

$rE_{traj}$  : Eastward position of the vehicle

$h_{traj}$  : Altitude of the vehicle

$vN_{traj}$  : Northward velocity of the vehicle

$vE_{traj}$  : Eastward velocity of the vehicle

$vD_{traj}$ : Down velocity of the vehicle

$$F(t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

For the simulations, vehicle horizontal velocity and altitude are taken as constant values. These assumptions can be summarized as follows:

$$vN_{traj} = Const. \quad (3.44)$$

$$vE_{traj} = Const. \quad (3.45)$$

$$h_{traj} = Const. \quad (3.46)$$

$$vD_{traj} = 0 \quad (3.47)$$

In the same manner, for INS model white noise terms are added to the trajectory model:

$$\dot{\bar{x}}_{INS}(t) = F(t) \cdot \bar{x}_{INS}(t) + \bar{w}(t) \quad (3.48)$$

where,

$$\bar{x}_{INS}(t) = [rN_{INS}; rE_{INS}; h_{INS}; vN_{INS}; vE_{INS}; vD_{INS}]^T$$

$$\bar{w}(t) = [w_{rN_{INS}}(t); w_{rE_{INS}}(t); w_{h_{INS}}(t); w_{vN_{INS}}(t); w_{vE_{INS}}(t); w_{vD_{INS}}(t)]^T$$

$w_i(t) = N(0, \sigma_i^2)$ : Zero mean normal distribution with variance  $\sigma_i^2$  corresponding to related position and velocity

For simulations with TAN algorithms, DTED are used in order to determine height differences at each time step as measurements which are given in equation (3.11). DTED heights are given as a function of longitude and latitude. However, simulations are performed considering Cartesian coordinates in order to visualize navigation errors better in the simulations. Actually, latitudes and northward positions and longitudes and eastward positions are correlated. Correlations are defined as follows [73]:

$$\frac{d}{dt} \lambda = \frac{vN}{R_N + h} \quad (3.49)$$

$$\frac{d}{dt} \mu = \frac{vE}{(R_E + h) \cdot \cos \lambda} \quad (3.50)$$

where,

$\lambda$ : Latitude of the vehicle

$\mu$ : Longitude of the vehicle

$h$ : Height of the vehicle above ground

$R_N$ : Earth's polar radius

$R_E$ : Earth's equatorial radius

The geometry of the earth is considered as an ellipsoid in WGS-84 coordinate system. The earth's polar radius  $R_N$  and equatorial radius  $R_E$  are defined as [74]:

$$R_N = \frac{a \cdot (1 - e^2)}{(1 - e^2 \cdot \sin^2 \lambda)^{3/2}} \quad (3.51)$$

$$R_E = a / (1 - e^2 \cdot \sin^2 \lambda)^{1/2} \quad (3.52)$$

where,

$a$ : Semi-major axis

$e^2$ : First eccentricity squared

WGS-84 values of these parameters are [74]:

$$a = 6378137 \pm 2 \text{ m} \quad (3.53)$$

$$e^2 = 0.00669437999013 \quad (3.54)$$

$R_N$  and  $R_E$  terms vary with changing latitude. However, in the simulations, changes in latitude will be less than one degree. Therefore, these terms are taken as constant values with their initial conditions taken in the simulations.

Then, correlations for the white noise terms for longitudes and latitudes can be expressed as follows:

$$\sigma_{\lambda}^2 = \frac{\sigma_{rN}^2}{(R_N + h)^2} \quad (3.55)$$

$$\sigma_{\mu}^2 = \frac{\sigma_{rE}^2}{[(R_E + h) \cdot \cos \lambda]^2} \quad (3.56)$$

In the simulations, discrete models are used considering the actual case. Height measurements will be taken at discrete intervals. Therefore, the models are discretized as follows:

$$\Phi(k) = I + F(t) \cdot T \quad (3.57)$$

where,

$\Phi(k)$ : State transition matrix

$T$ : Sample time (In the simulations,  $T = 1$  second taken.)

Finally, trajectory model becomes:

$$\bar{x}_{traj}(k+1) = \Phi(k) \cdot \bar{x}_{traj}(k) \quad (3.58)$$

$$\begin{bmatrix} rN_{traj}(k+1) \\ rE_{traj}(k+1) \\ h_{traj}(k+1) \\ vN_{traj}(k+1) \\ vE_{traj}(k+1) \\ vD_{traj}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} rN_{traj}(k) \\ rE_{traj}(k) \\ h_{traj}(k) \\ vN_{traj}(k) \\ vE_{traj}(k) \\ vD_{traj}(k) \end{bmatrix} \quad (3.59)$$

Here, it should be noted that  $\Phi(k)$  term becomes constant considering constant velocity motion with the given sample time. Hence, linear discrete model is formed. In the same manner, for the INS model:

$$\bar{x}_{INS}(k+1) = \Phi(k) \cdot \bar{x}_{INS}(k) + \bar{w}(k) \quad (3.60)$$

where,

$$\bar{x}_{INS}(k) = [rN_{INS}; rE_{INS}; h_{INS}; vN_{INS}; vE_{INS}; vD_{INS}]^T$$

$$\bar{w}(k) = [w_{rN_{INS}}(k); w_{rE_{INS}}(k); w_{h_{INS}}(k); w_{vN_{INS}}(k); w_{vE_{INS}}(k); w_{vD_{INS}}(k)]^T$$

$w_i(k) = N(0, \sigma_i^2)$ : Zero mean normal distribution with variance  $\sigma_i^2$  corresponding to related position and velocity

White noise terms can be written as follows then:

$$w_\lambda(k) = N(0, \frac{\sigma_{rN}^2}{(R_N + h)^2}) \quad (3.61)$$

$$w_\mu(k) = N(0, \frac{\sigma_{rE}^2}{[(R_E + h) \cdot \cos \lambda]^2}) \quad (3.62)$$

$$w_{rN}(k) = N(0, \sigma_{rN}^2) \quad (3.63)$$

$$w_{rE}(k) = N(0, \sigma_{rE}^2) \quad (3.64)$$

$$w_h(k) = N(0, \sigma_h^2) \quad (3.65)$$

$$w_{vN}(k) = N(0, \sigma_{vN}^2) \quad (3.66)$$

$$w_{vE}(k) = N(0, \sigma_{vE}^2) \quad (3.67)$$

$$w_{vD}(k) = N(0, \sigma_{vD}^2) \quad (3.68)$$

Here, it should be noted that,  $\sigma_h^2$  term is the variance of the barometric altimeter of the INS used. Radar altimeter errors will be added as measurement errors in the TAN algorithms.

### 3.4.1.2. DTED Database Model

As stated in the previous section, height of the vehicle is determined from the DTED maps according to the related latitude and longitude of the vehicle. For this purpose, Simulink “Lookup Table” blocks are used [58]. Then, measurement height differences are taken for the TERCOM, SITAN, PDAF and TSF models by adding INS white noises as system noise and radar white noises as measurement noise considering Figure 50.

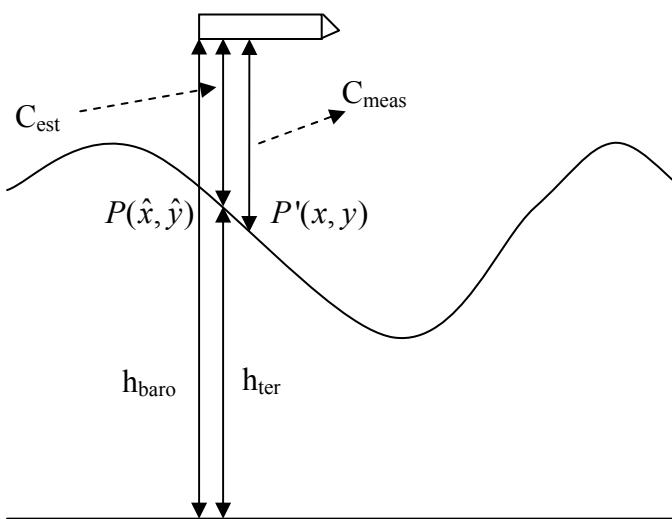


Figure 50. DTED Height Measurement Difference

DTED height difference for the INS position was derived in equation (3.11). Recall equation (3.11) in order to investigate DTED model:

$$\delta h(k) = \underbrace{\left[ h_{DTED}(\mu_{INS}(k), \lambda_{INS}(k)) + w_{INS}(k) \right]}_{\text{INS Height Model}} - \underbrace{\left[ h_{DTED}(\mu_{traj}(k), \lambda_{traj}(k)) + w_{radar}(k) \right]}_{\text{Radar Height Measurement Model}}$$

Here, it should be noted that first part of the equation is the model of the INS height and second part is the radar height measurement model. Hence, the difference gives height difference term,  $\delta h(k)$ . Using longitude and latitude values of the trajectory and INS models, height difference term can be obtained for simulations.

Height difference,  $\delta h(k)$  parameter is used in SITAN model directly. However, batch height difference values are required for the applications of TERCOM, PDAF, TSF and “SITAN with Bank of Kalman Filters”. This is done by calculating all height difference terms in the  $3\sigma$  horizontal error bound of the INS and was derived in equation (3.12). Recall equation (3.12):

$$\delta h_i(k) = \underbrace{\left[ h_{DTED}(\mu_i(k), \lambda_i(k)) + w_{INS}(k) \right]}_{\text{Height Model for All Positions in the DTED Batch}} - \underbrace{\left[ h_{DTED}(\mu_{traj}(k), \lambda_{traj}(k)) + w_{radar}(k) \right]}_{\text{Radar Height Measurement Model}}$$

$$i = 1, \dots, m(k)$$

where;

$i$ : Index of the DTED grid node,

$m(k)$ : DTED grid size (selected as square of an odd number for INS position to be at the center of the DTED grid)

Considering equation (3.12), formation of a batch of height differences for a 5x5 DTED grid size is shown in Table 17.

Table 17. Batch of Height Differences Formation for 5x5 DTED Grid

$\lambda = \lambda_{INS} + 2 * 3''$	$i = 5$	$i = 10$	$i = 15$	$i = 20$	$i = 25$
$\lambda = \lambda_{INS} + 3''$	$i = 4$	$i = 9$	$i = 14$	$i = 19$	$i = 24$
$\lambda = \lambda_{INS}$	$i = 3$	$i = 8$	$i = 13$	$i = 18$	$i = 23$
$\lambda = \lambda_{INS} - 3''$	$i = 2$	$i = 7$	$i = 12$	$i = 17$	$i = 22$
$\lambda = \lambda_{INS} - 2 * 3''$	$i = 1$	$i = 6$	$i = 11$	$i = 16$	$i = 21$

Time:  $t = k$     $\mu = \mu_{INS} - 2 * 3''$     $\mu = \mu_{INS} - 3''$     $\mu = \mu_{INS}$     $\mu = \mu_{INS} + 3''$     $\mu = \mu_{INS} + 2 * 3''$

Note: Index  $i=13$  gives  $\delta h(k)$  at position  $(\lambda_{INS}, \mu_{INS})$  of INS, at time “ $t = t_0$ ”.

Index  $i=1$  gives  $\delta h(k)$  at position  $(\lambda_{INS} - 2 * 3'', \mu_{INS} - 2 * 3'')$  of INS for DTED Level 1, at time  $k$ . In the same manner, batch of  $\delta h(k)$ ’s are obtained.

DTED database model formed in Simulink is shown in Figure 51. Here, latitudes and longitudes of the INS and trajectory model are determined first. Using “Lookup Table” blocks of Simulink, DTED heights for the INS and trajectory are found. Then height differences  $\delta h(k)$  are determined in order to use in the TAN models. Moreover, for SITAN process, terrain slopes along longitude and latitude directions,  $(h_x, h_y)$  are also determined using the same procedure above. For “SITAN with Bank of Kalman Filters” model, batch of slopes are also formed considering batch formation explained in Table 17.

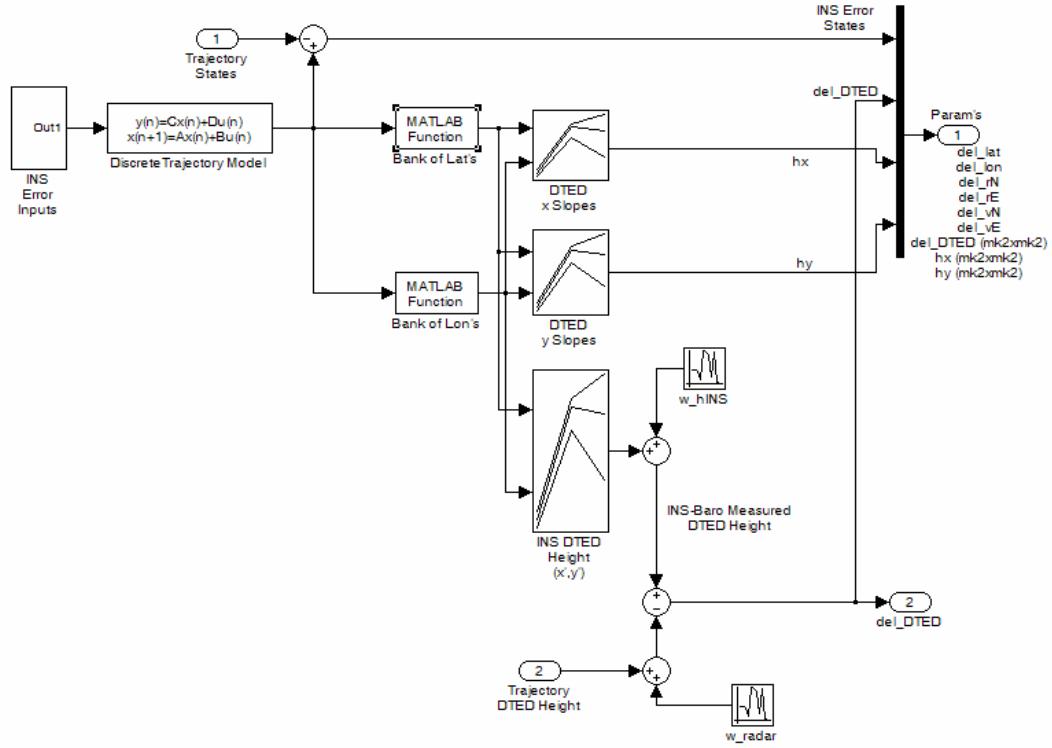


Figure 51. Simulink DTED Database Model

For the simulations, DTED Level 1 data were required and they have been obtained from HGK. The properties of DTED prepared for Turkey were given in Table 3 [16]. Horizontal accuracy of Level 1 DTED is defined as  $\pm 130$  m, and vertical accuracy as  $\pm 30$  m. On the other hand, horizontal accuracy of DTED Level 2 data is  $\pm 26$  m. In the simulations, DTED Level 1 data are used considering the horizontal accuracies of DTED Level 2. Moreover, simulations with real DTED Level 2 data are also performed which were also obtained from HGK for a few places of Turkey.

In order to read DTED files for simulations, Matlab “Mapping Toolbox” is used [75]. Hence, binary DTED files are directly used as text files in the simulations with the help of “Mapping Toolbox”.

### 3.4.1.3. TERCOM Model

TERCOM was studied in detail in Chapter 2 with the simulations performed and the results were discussed. For TAN algorithm simulations, the models for TERCOM are formed considering the well-known MAD and MSD processes derived for TERCOM.

Recall equations derived for MAD and MSD processes given from equation (2.9) to (2.11):

$$MSD_{jk} = (1/N) \sum_{i=1}^N (S_{ij} - S_{ik})^2$$

$$MAD_{jk} = (1/N) \sum_{i=1}^N |S_{ij} - S_{ik}|$$

$$\xi_{jk} = \begin{cases} P[C_{jk} < C_{jj}], \text{ where a minimum of } C_{jk} \text{ is sought,} \\ P[C_{jk} > C_{jj}], \text{ where a maximum of } C_{jk} \text{ is sought.} \end{cases}$$

Examination of the expressions for the MAD and MSD processors indicates that both of these correlators can be viewed as distance measures, where the dimensions of the space for which these distances are defined correspond to the number of elements in the profiles.

In order to form TERCOM model in Simulink, “Signal Processing Blockset” is used [76]. In this blockset, RMS, mean and minimum selection operations defined for MAD and MSD processes can be done in real-time.

TERCOM model formed in Simulink is shown in Figure 52. Here, at the upper part MAD process, at the lower part MSD process occurs. Using batch of height differences “ $\delta h(k)$ ” obtained from DTED database model, minimum of MAD and MSD functions are determined. It should be noted that, TERCOM Simulink model works in real-time by calculating MAD and MSD functions at each time step. In actual applications, these functions are calculated once after some time of the operation begins. However, considering calculated MAD and MSD functions at some definite times of operation, TERCOM processes can be understood.

On the other hand, position corrections are done considering calculated position indices from TERCOM. As a result of this, position accuracies for TERCOM are within the limit of the DTED grid size. In other words, TERCOM horizontal position accuracies can not be better than the DTED grid accuracy used.

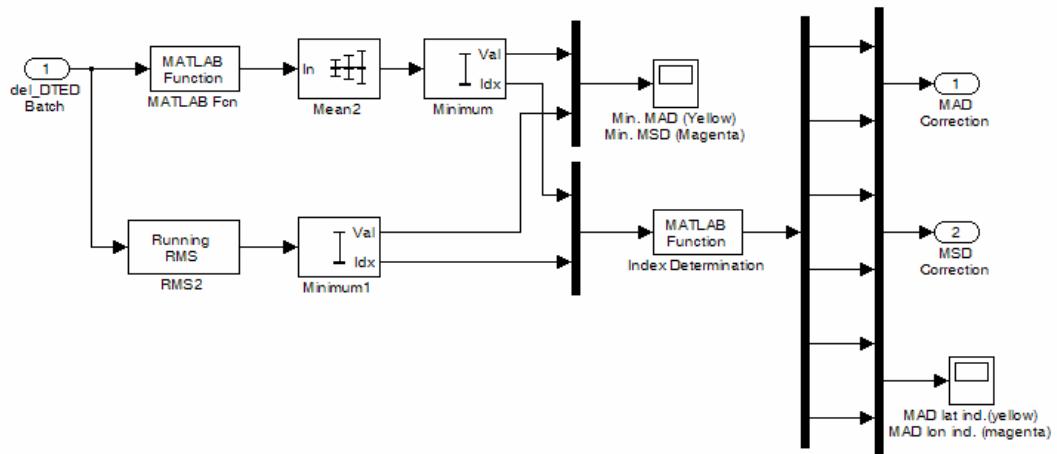


Figure 52. Simulink TERCOM Model

### **3.4.1.4. SITAN Model**

SITAN was studied in detail in Chapter 2 with the simulations performed and the results were discussed. For TAN algorithm simulations, the models formed for SITAN are directly used. Two models are used for SITAN:

1. Standard EKF for SITAN for tracking mode (Errors less than 100 meters for DTED Level 1 maps)
2. Bank of EKF's for acquisition mode (Errors greater than 100 meters for DTED Level 1 maps)

For tracking mode, equations derived for SITAN given in Table 10 are used. EKF equations are written in Simulink using S-functions [58]. An S-function is a computer language description of a Simulink block. S-functions use a special calling syntax that enables the user to interact with Simulink equation solvers. This interaction is very similar to the interaction that takes place between the solvers and built-in Simulink blocks. The form of an S-function is very general and can accommodate continuous, discrete, and hybrid systems [14].

Simulink model with standard EKF for SITAN for tracking mode is shown in Figure 53. Here, EKF uses the height difference between the INS and the trajectory and the slopes at the INS position.

For acquisition mode, parallel EKF structure with 3x3 and 5x5 grid size is used. In order to select position fix, equation (2.40) is used. Recall equation (2.40) which gives AWRS value of the selected filter:

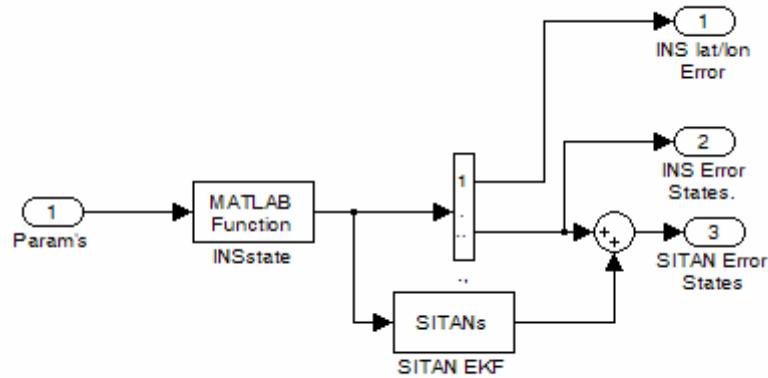


Figure 53. Simulink Single EKF SITAN Model

$$AWRS_{j\text{th filter}} = \frac{1}{N} \left[ \sum_{i=1}^N \frac{\Delta_i}{H_i P_i H_i^T + R_i} \right]_{j\text{th filter}}$$

where;

$AWRS_{j\text{th filter}}$  : Average Weighted Residual Squared of the j'th filter,

This AWRS value is the average weighted residual squared between the predicted ground clearance for each filter and the ground clearance measured by the radar altimeter for each time  $t_i$ . By examining the minimum AWRS values for each filter after a sufficiently large number of measurements have been processed, the correct filter and its associated state error estimates are chosen.

Formation of the SITAN model for acquisition mode is quite complex. In Figure 54 and Figure 55, SITAN models with bank of EKF's are shown.

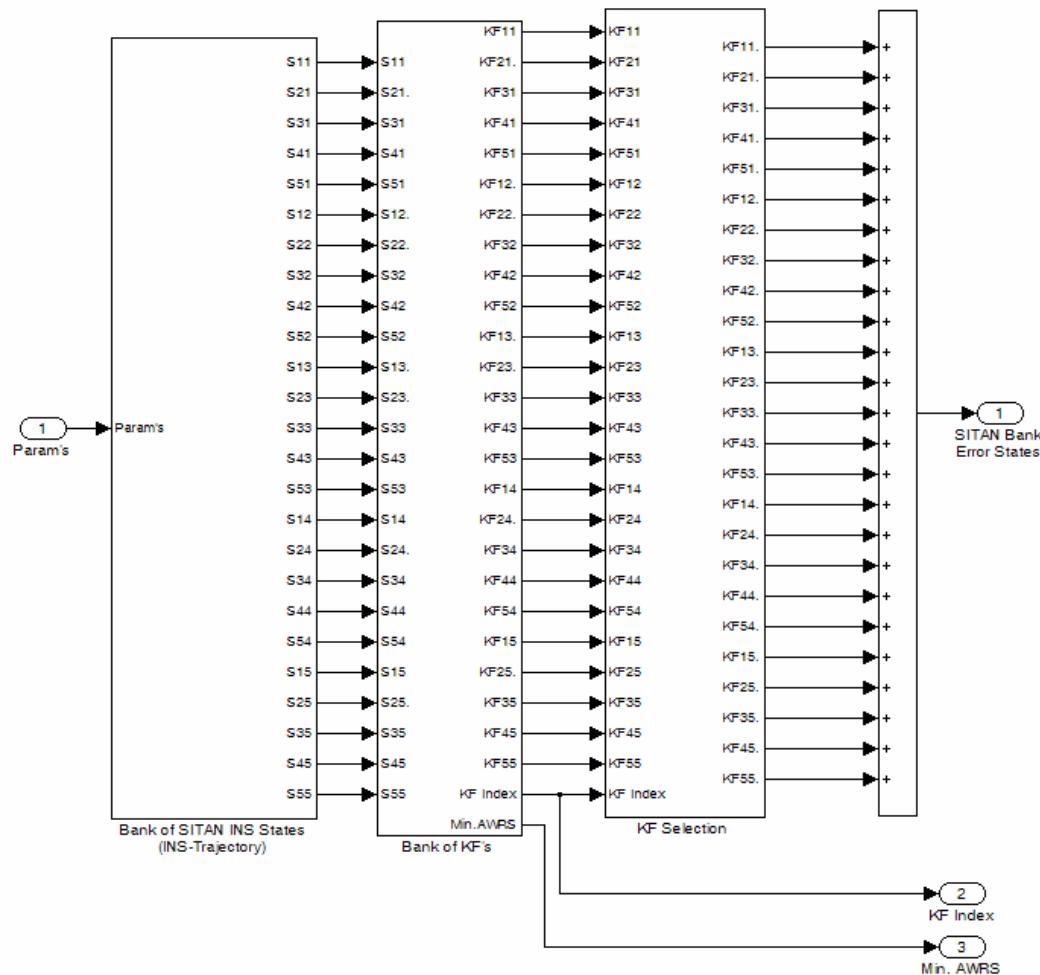


Figure 54. Simulink Bank of EKF SITAN Model for 5x5 Grids

In these models, parallel Kalman filters run with different initial conditions and slopes. Single EKF model is used for tracking mode simulations, where bank of EKF's models are used in the acquisition mode simulations. Simulation results of SITAN filters for tracking and acquisition modes were also given in Chapter 2.

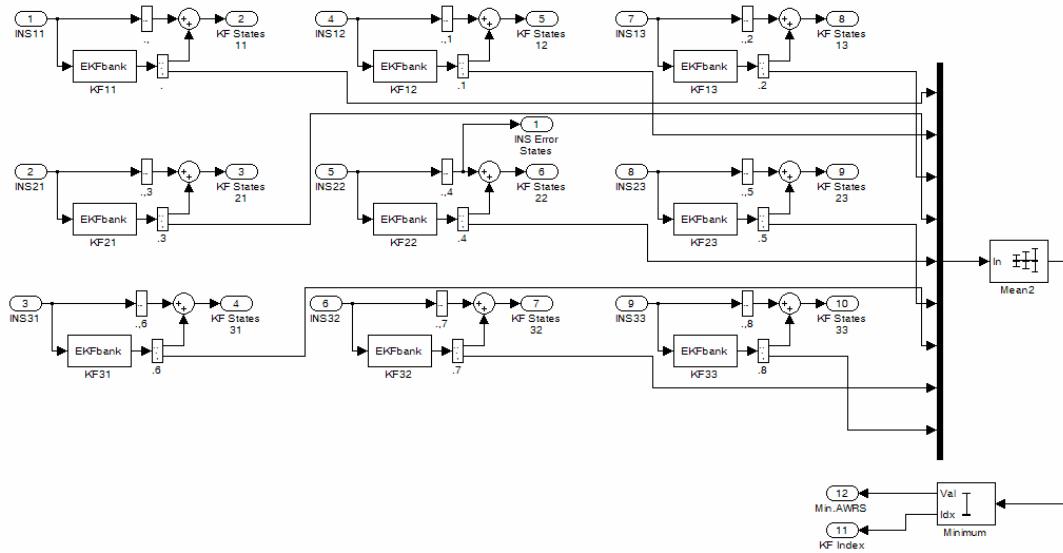


Figure 55. Simulink Bank of EKF SITAN Model for 3x3 Grids

### 3.4.1.5. PDAF Model

PDAF model is formed in Simulink considering the equations in section 3.2.2.2 which were summarized in Table 15. Here, it should be noted that Simulink architecture is very simple. Only a single PDAF is used in all of the simulations. Batch size of the filter (i.e. considered grid size) can be changed independent of the PDA filter. PDAF model can be used for both acquisition and tracking modes of the TAN solution.

In the PDAF model, following processes are performed:

1. Height differences are taken from the DTED database model.
2. Gating process is done in order to extract impossible position solutions.

3. Height differences are averaged in order to form past measurement information.
4. Averaged height differences are used in the PDAF in order to determine PDAF error states.

Simulink PDAF model is shown in Figure 56.

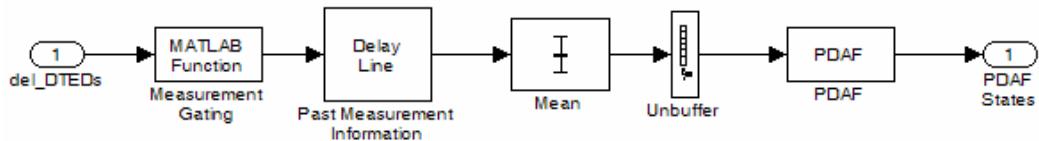


Figure 56. Simulink PDAF Model

#### 3.4.1.6. TSF Model

TSF model is formed in Simulink considering the equations in section 3.3.3 which were summarized in Table 16. Here, it should be noted that again Simulink architecture is very simple like PDAF model. However, more complex operations are done in TSF model, since bank of Kalman filter operations are performed. Bank of Kalman filter operations are performed using an S-function. Batch size of the filter (i.e. considered grid size) can be changed independent of the filter. TSF model can be used for both acquisition and tracking modes of the TAN solution as explained in the related sections before.

In the TSF model, following processes are performed:

1. Height differences are taken from the DTED database model.
2. Gating process is done in order to extract the impossible position solutions.
3. Height differences are used in the TSF in order to determine TSF error states.

Simulink TSF model is shown in Figure 57.

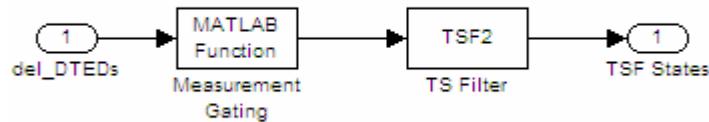


Figure 57. Simulink TSF Model

### 3.4.2. Case Studies

Case studies with simple kinematic models are performed for three different cases:

1. Simulations with DTED Level 1
2. Simulations with DTED Level 2

### 3. Simulations with Various DTED Grid Sizes

For various DTED level simulations, different terrain types are selected. Then, simulations for tracking and acquisition modes are performed simultaneously using the implemented TAN algorithms compared with the well-known algorithms.

#### 3.4.2.1. Simulations With DTED Level 1

##### 3.4.2.1.1. *Terrain Selection*

Simulations are performed for three different types of terrains:

1. Rough terrain
2. Smooth terrain
3. Mountainous terrain

In order to determine the required terrains, Microdem/ TerraBase II Software [77] is used. Terrain contours of 50 meters and the trajectory paths obtained from the software is shown in Figure 58. Then, terrain heights versus time plots for the selected terrains are given in Figure 59, Figure 60 and Figure 61. Finally, terrain parameters are calculated for the selected terrains and summarized in Table 18.

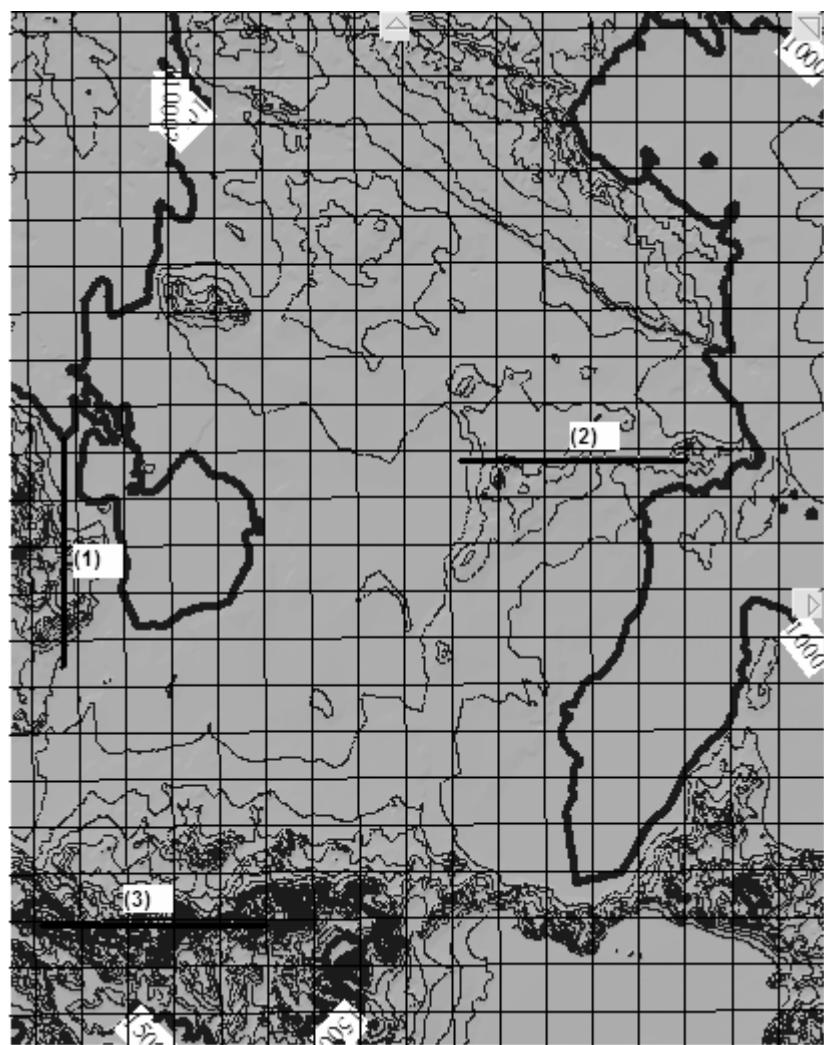


Figure 58. Terrain Contours for TAN Simulations

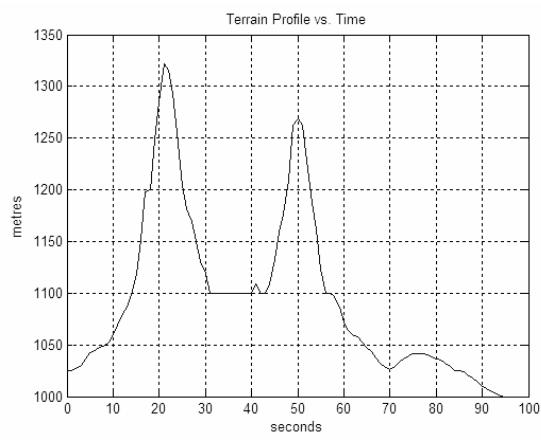


Figure 59. Terrain Height vs. Time for Rough Terrain

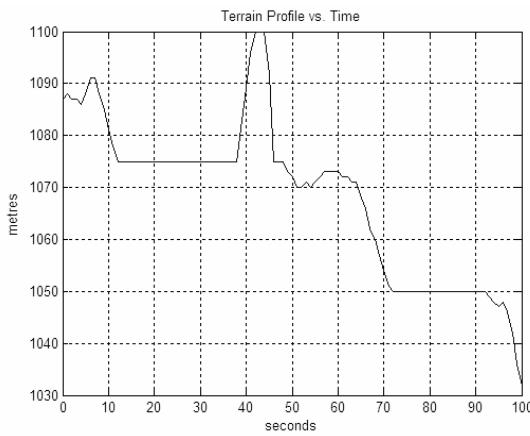


Figure 60. Terrain Height vs. Time for Smooth Terrain

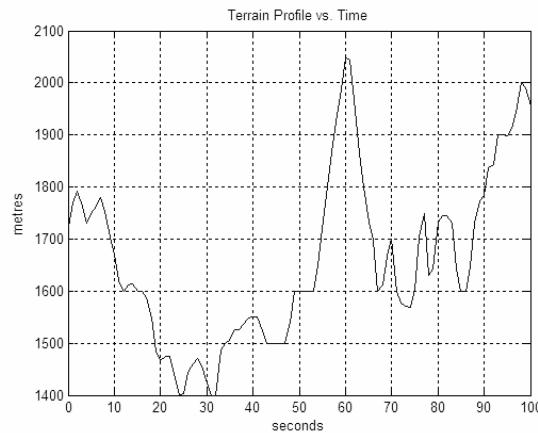


Figure 61. Terrain Height vs. Time for Mountainous Terrain

Table 18. Terrain Parameters for TAN Simulations

Terrain Type	Rough	Smooth	Mountainous
Mean height of the terrain profile	1093 m	1104 m	1177 m
Sigma-T	77.9 m	34.1 m	212.9 m
Sigma-Z	16.2 m	3.7 m	23.1 m
$X_T$	674.6	1309 m	1302 m

### **3.4.2.1.2. INS Model Verification**

In order to perform simulations, verification of the INS model used is required first. For cruise missiles, generally 1.0 nm/hr INS quality is required. In the simulations, simple INS model is formed considering equation (3.60). In this equation, white noise terms are added to positions and velocities considering constant velocity flight.

In Table 19, simulation parameters used for the INS model are given. Using the values provided, INS model is tested for horizontal position errors. Horizontal position and velocity errors are given in Figure 62 and Figure 63.

Table 19. INS Model Parameters for 1.0 nm/hr Quality

Initial vehicle velocity	240 m/s
Initial INS east velocity bias	0.5 m/s
Initial INS north velocity bias	0.5 m/s
INS horizontal position standard deviation ( $\sigma_{rN}$ , $\sigma_{rE}$ )	9 m
INS velocity standard deviation ( $\sigma_{vN}$ , $\sigma_{vE}$ )	0.05 m/s

As it can be seen from the horizontal position errors, INS quality is about 3.0 nm/hr. Actually, selecting worse INS quality than the real system used is preferred in order to test the performance of the TAN algorithms used. In the simulations, small operation times (like 100 seconds) are used for TAN applications where INS is not updated during the simulation period.

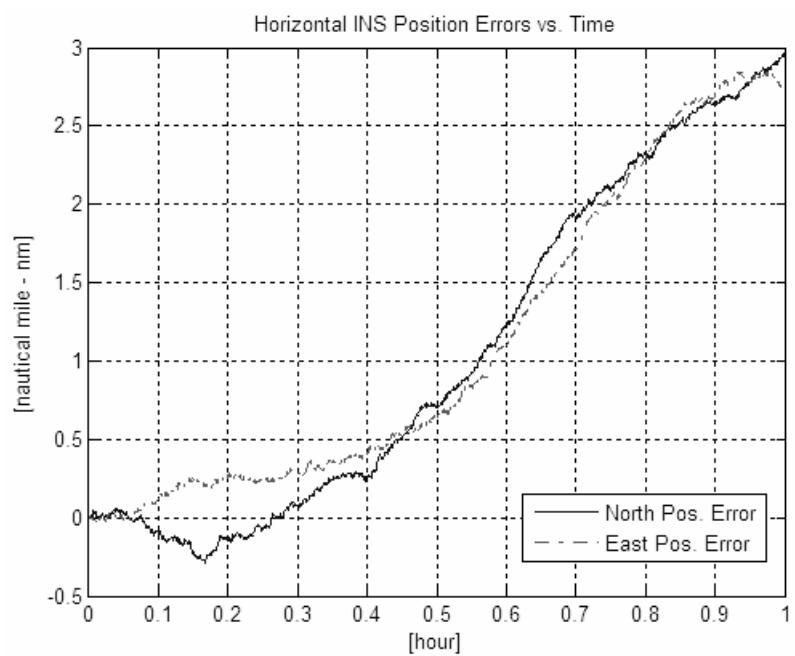


Figure 62. Horizontal Position Errors of the INS Model Used

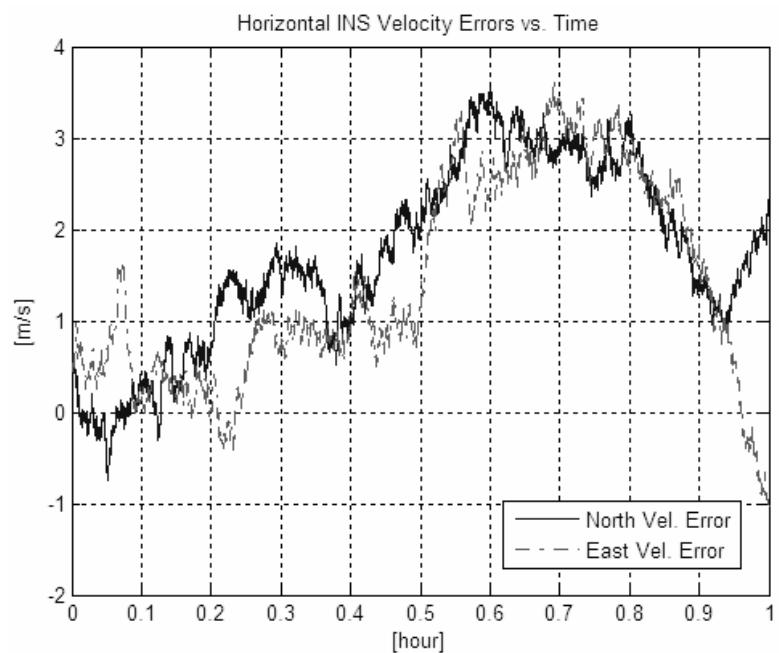


Figure 63. Horizontal Velocity Errors of the INS Model Used

### ***3.4.2.1.3. Simulation Results***

Simulations are performed for two modes of operation of TAN algorithms:

1. Tracking mode, where SITAN single filter, PDAF and TSF are compared with Monte Carlo simulations along the trajectory (Recursive Solution);
2. Acquisition mode, where PDAF, TSF and TERCOM are compared with Monte Carlo simulations for the position update at a defined time (Batch Solution).

#### ***3.4.2.1.3.1. Simulations for Tracking Mode***

First, simulations for tracking mode are done. Parameters used in the simulations are given in Table 20. Monte Carlo simulations of 100 runs are performed and the following plots are obtained for SITAN, PDAF and TSF.

1. Northward and eastward position errors;
2. RMS values of north and east positions.

For tracking mode simulations, TERCOM is not used. In order to apply TERCOM algorithm, larger DTED grid size and large initial position errors are required. When TERCOM algorithm is applied for small grid size, false position fixes occur with high percentages since INS error model is not used in TERCOM algorithm.

Table 20. Simulation Parameters for Tracking Mode

Initial INS position deviation (one axis)	60 m
Initial vehicle velocity	240 m/s
Initial INS east velocity bias	0.5 m/s
Initial INS north velocity bias	0.5 m/s
INS horizontal position standard deviation	9 m
INS altitude position standard deviation	3 m
Radar altimeter standard deviation	3 m
INS velocity standard deviation	0.05 m/s
DTED Grid Size (for PDAF and TSF)	3x3

### Terrain Type 1 (Rough Terrain):

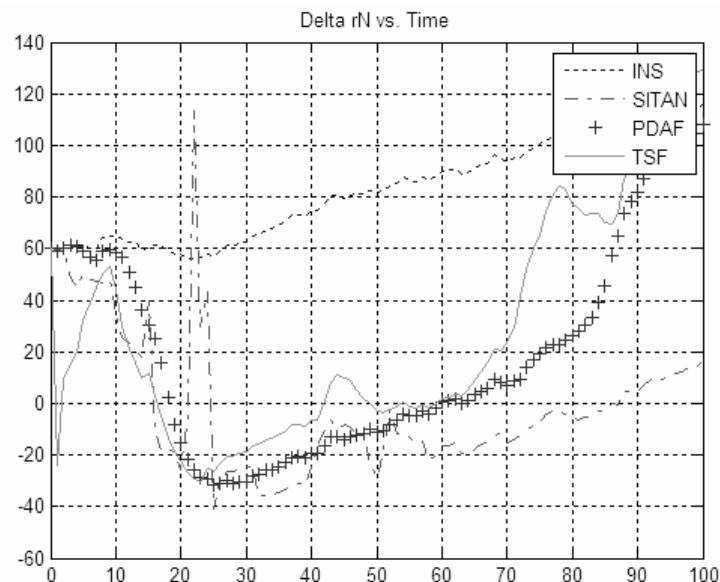


Figure 64. Northward Position Error vs. Time

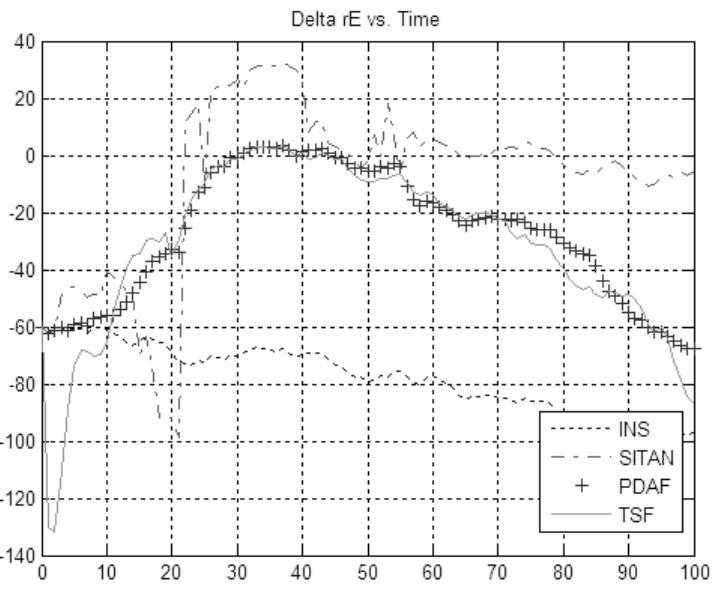


Figure 65. Eastward Position Error vs. Time

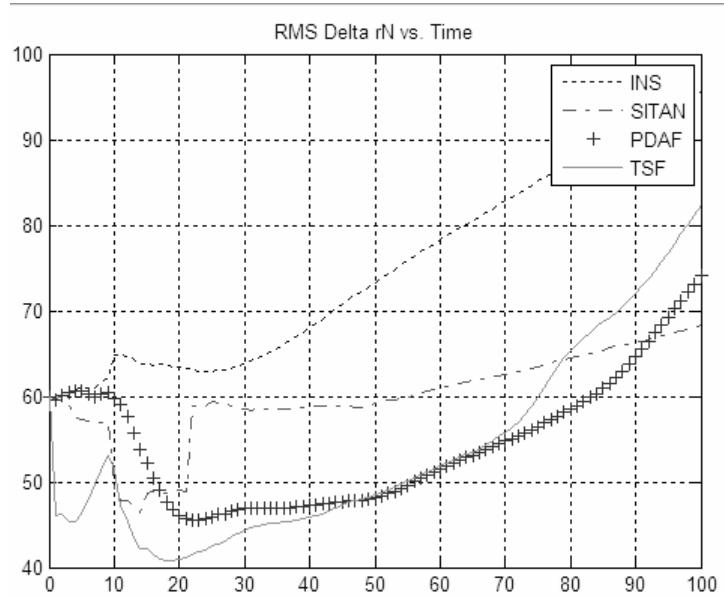


Figure 66. Northward Position RMS Error vs. Time

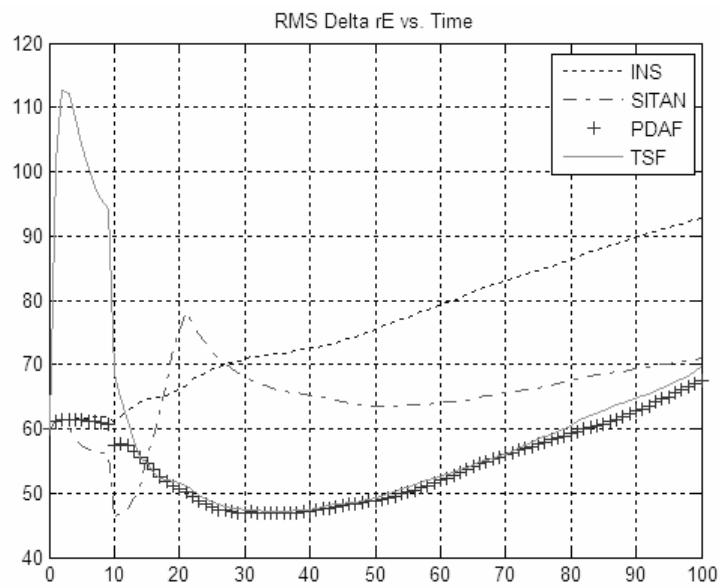


Figure 67. Eastward Position RMS Error vs. Time

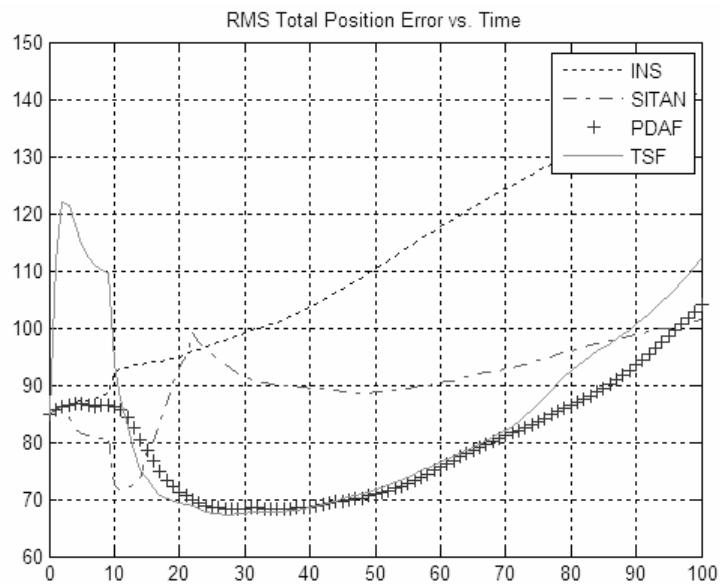


Figure 68. Total Position RMS Error vs. Time

### Terrain Type 2 (Smooth Terrain):

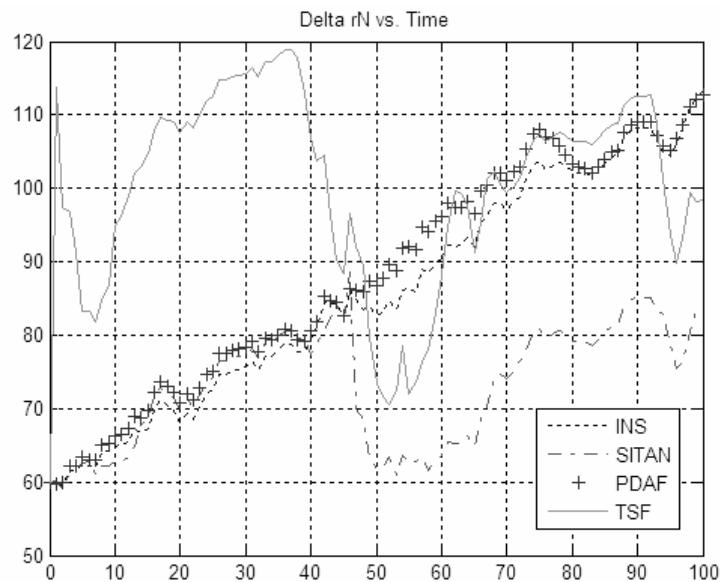


Figure 69. Northward Position Error vs. Time

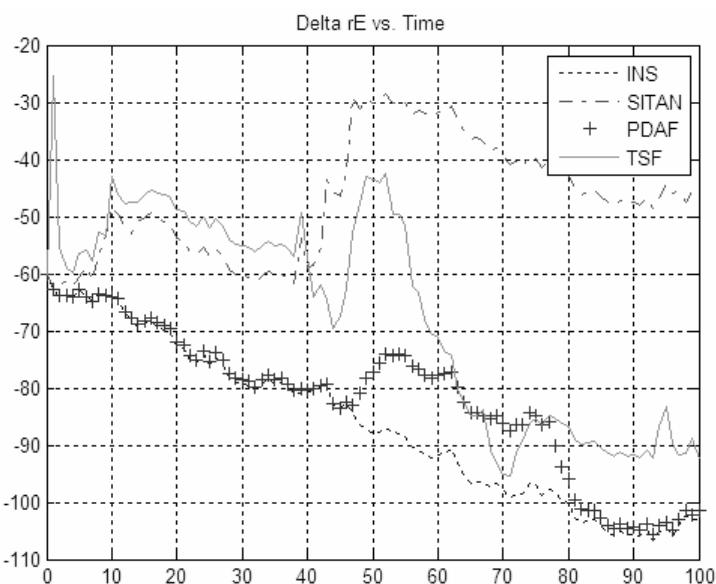


Figure 70. Eastward Position Error vs. Time

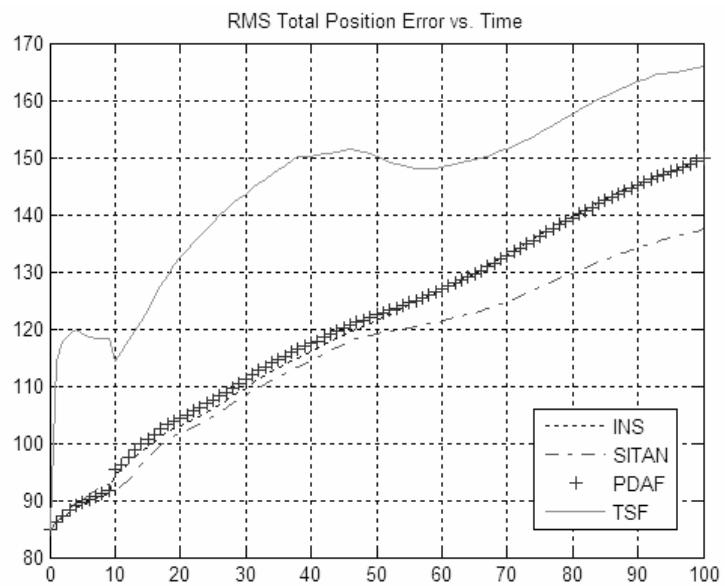


Figure 71. Total Position RMS Error vs. Time

### Terrain Type 3 (Mountainous Terrain):

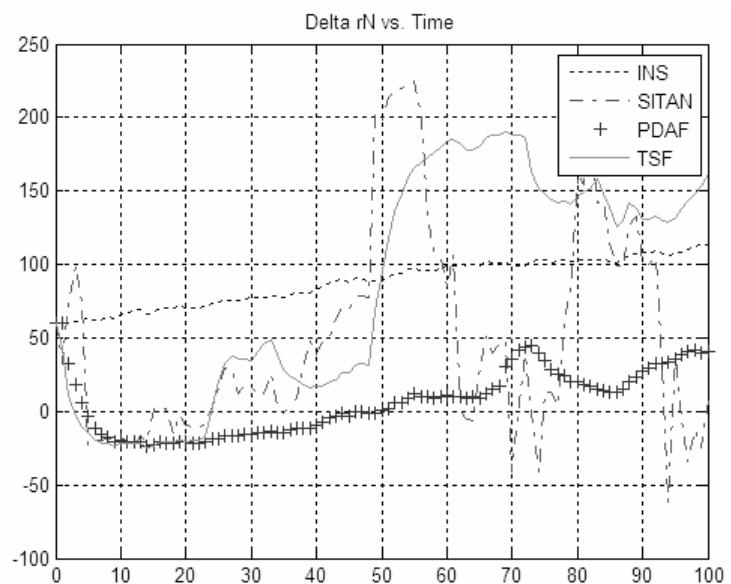


Figure 72. Northward Position Error vs. Time

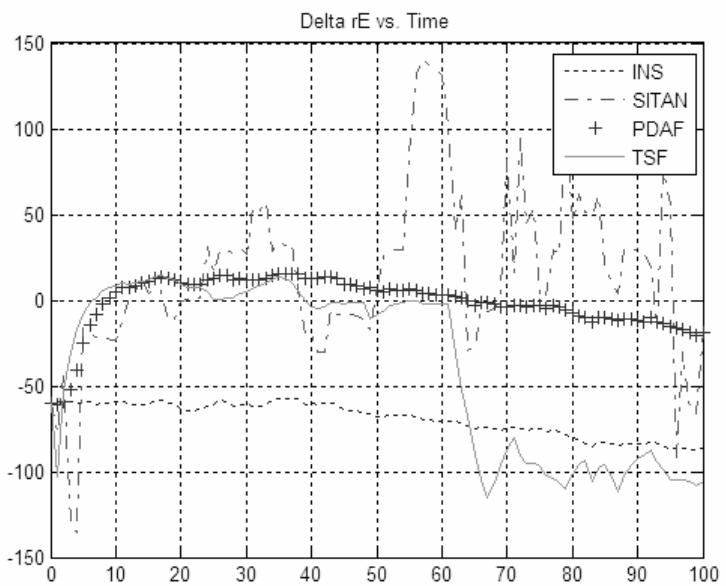


Figure 73. Eastward Position Error vs. Time

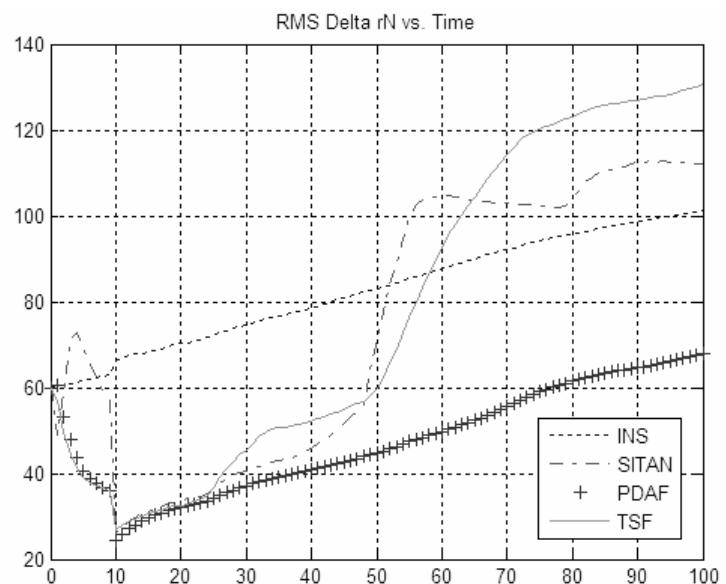


Figure 74. Northward Position RMS Error vs. Time

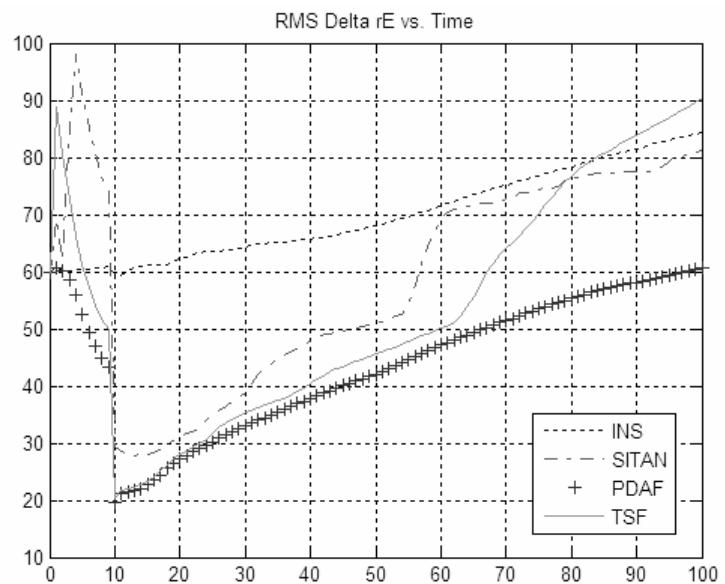


Figure 75. Eastward Position RMS Error vs. Time

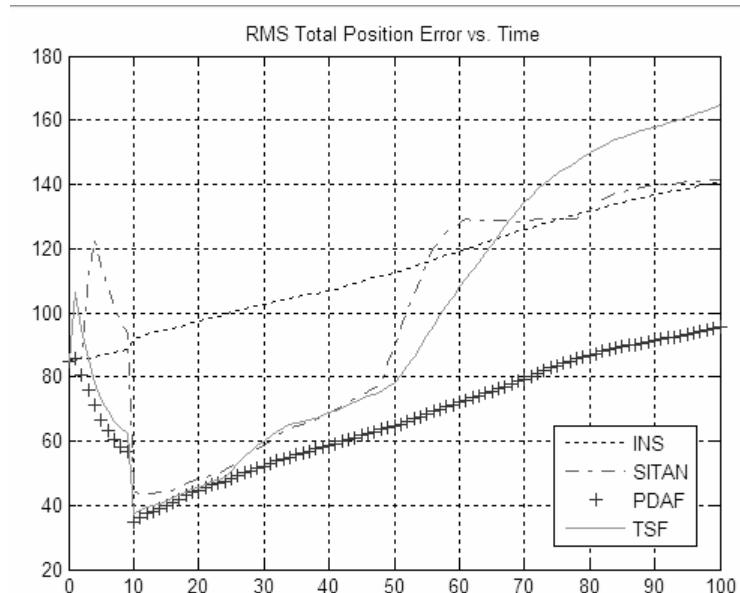


Figure 76. Total Position RMS Error vs. Time

Simulations for tracking mode are done for a small period of time (i.e. 100 seconds for tracking mode) in order to visualize the performance of the implemented TAN algorithms. In the actual navigation system of a cruise missile, INS will be updated at discrete time intervals according to the TAN algorithm used. In the simulations performed above, after 10 to 40 seconds of operation, since INS is not updated errors grow and TAN algorithm can not be used since small DTED grid size is selected. TAN performance depends on the terrain type. For mountainous terrains, implemented TAN algorithms find position fixes faster than rough terrains. Hence, INS update time can be determined from the selected terrain properties.

From the simulations, it is seen that better results than SITAN are obtained for rough and mountainous terrain types. For the smooth terrain, SITAN seems to show better results. However, response of the SITAN filter is also not stable and navigation solution cannot be obtained for smooth terrain. TSF and PDA filter results are considerably good, since if navigation solution does not exist, the filters follow INS error model which is actually a desired feature. From the Monte Carlo simulations, position RMS errors of the TSF and PDAF algorithms become less than 50 meters for mountainous terrains; in other words, a decreased navigation error is obtained. As it can be seen from the simulation results TSF behaves as a correction shift along the INS error model.

#### *3.4.2.1.3.2. Simulations for Acquisition Mode*

Parameters used in the simulations for acquisition mode are given in Table 21. Monte Carlo simulations of 100 runs are performed and northward and eastward position errors at the update time are obtained from the plots for TERCOM, PDAF and TSF.

Table 21. Simulation Parameters for Acquisition Mode

Initial INS position deviation (one axis)	400 m
Initial vehicle velocity	240 m/s
Initial INS east velocity bias	0.5 m/s
Initial INS north velocity bias	0.5 m/s
INS horizontal position standard deviation	9 m
INS altitude position standard deviation	3 m
Radar altimeter standard deviation	3 m
INS velocity standard deviation	0.05 m/s
DTED Grid Size (for PDAF and TSF)	11x11

### Terrain Type 1 (Rough Terrain):

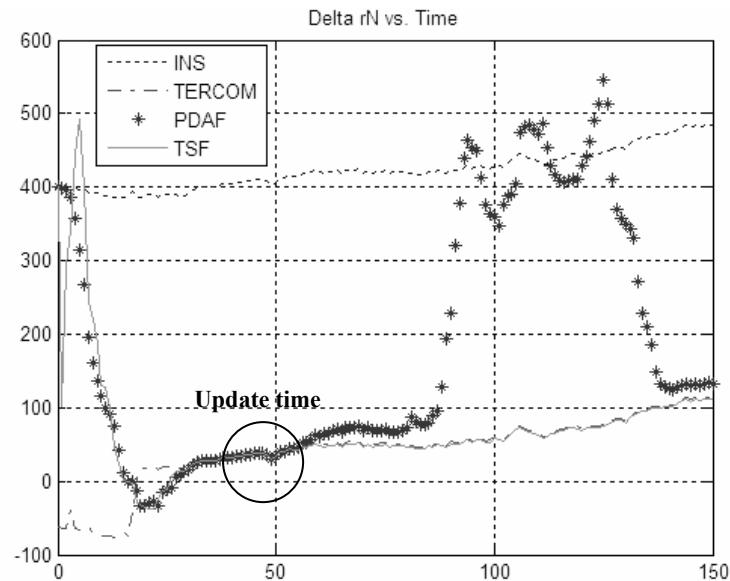


Figure 77. Northward Position Error vs. Time

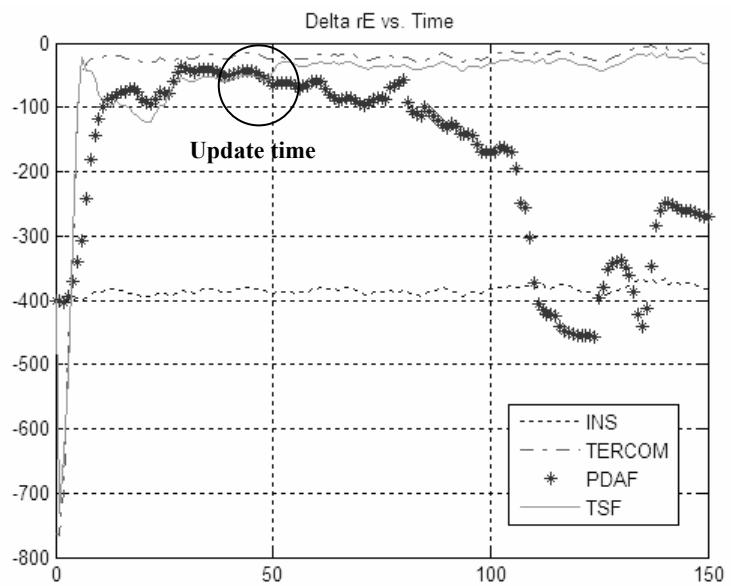


Figure 78. Eastward Position Error vs. Time

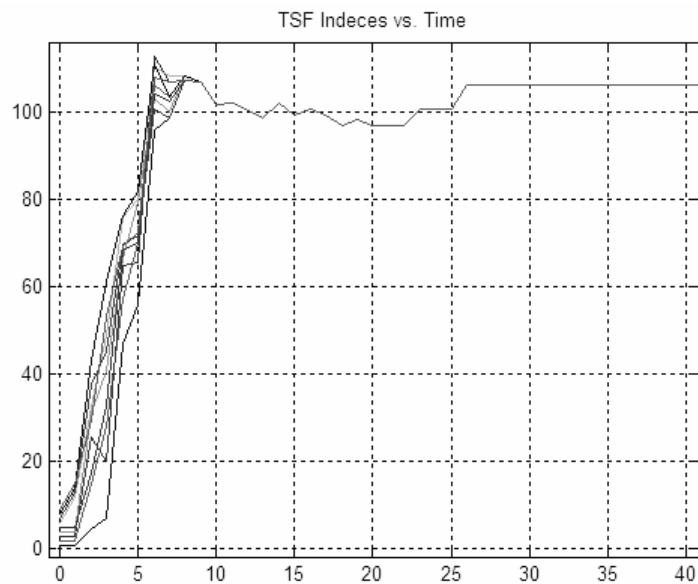


Figure 79. TSF Indices vs. Time

### Terrain Type 2 (Smooth Terrain):

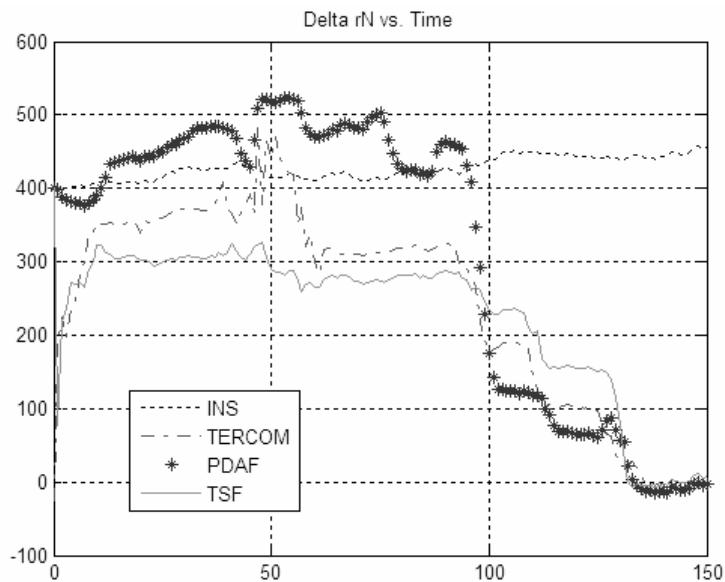


Figure 80. Northward Position Error vs. Time

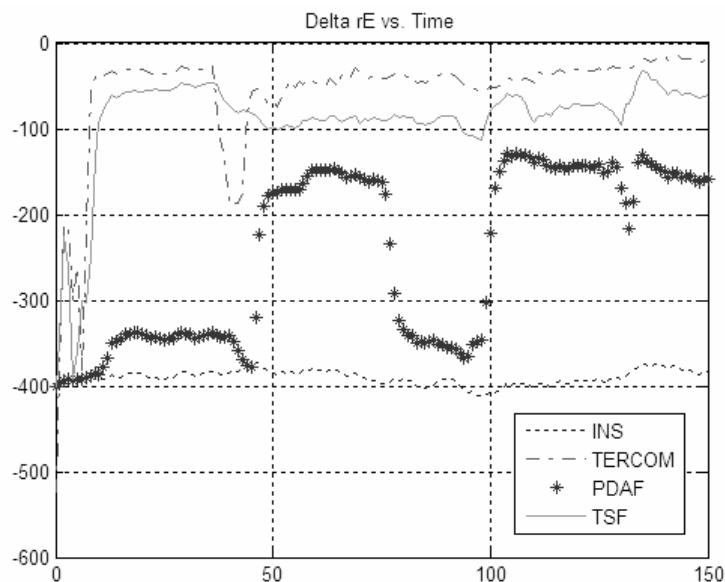


Figure 81. Eastward Position Error vs. Time

### Terrain Type 3 (Mountainous Terrain):

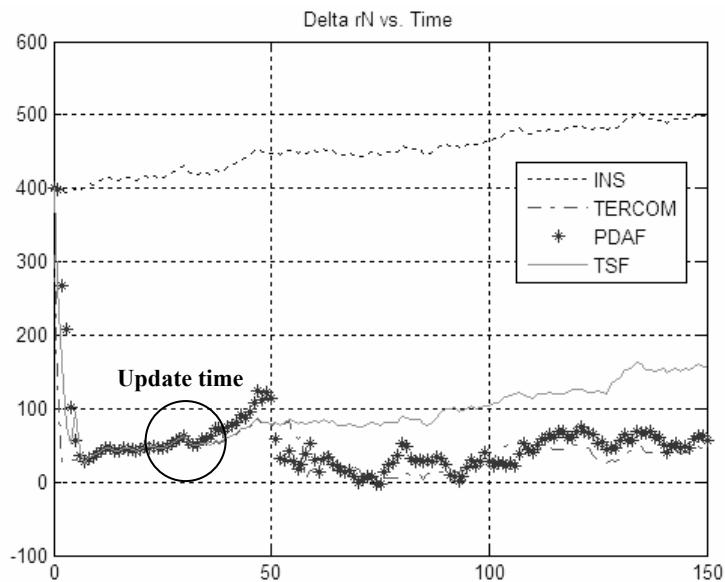


Figure 82. Northward Position Error vs. Time

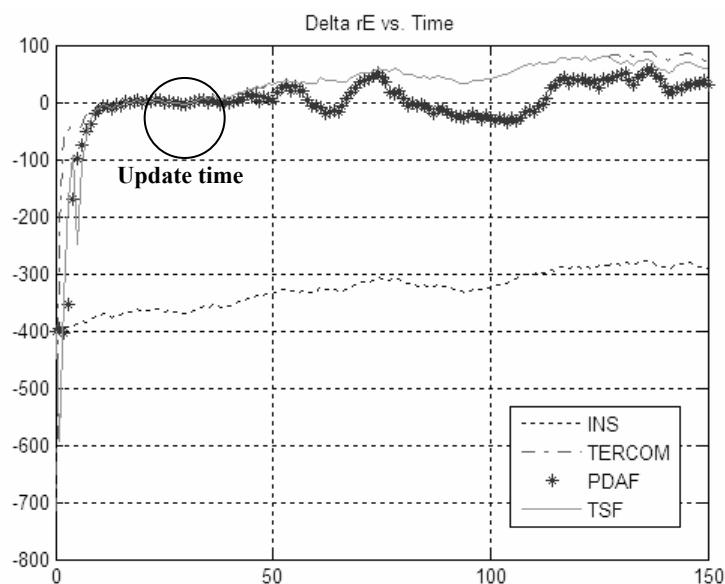


Figure 83. Eastward Position Error vs. Time

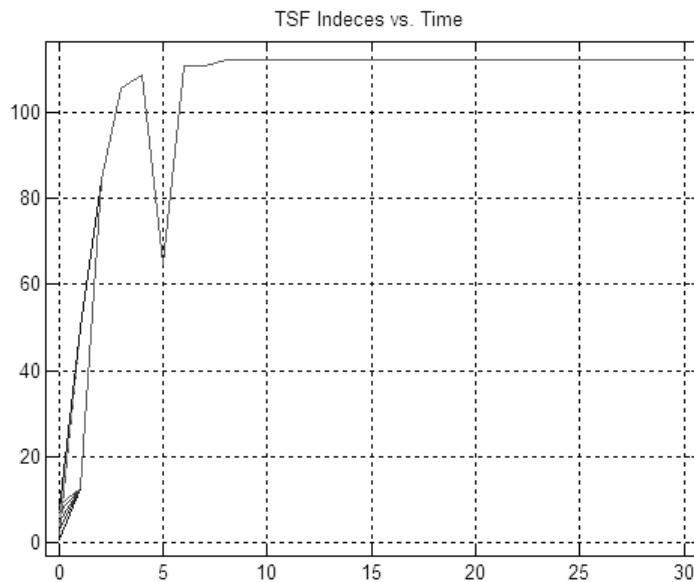


Figure 84. TSF Indices vs. Time

From the simulations, it is seen that similar results with TERCOM are obtained for rough terrain type. For the smooth terrain, both algorithms do not have navigation solution. As it can be seen from the results, TSF and PDA filter can be used also for acquisition mode of TAN solution using the considered DTED size. SITAN bank of Kalman filters is not considered for acquisition mode here; since, for the initial error given 121 Kalman filters should be run for the simulation. On the other hand, using TSF and PDAF, same solutions with TERCOM are obtained.

In the simulations, TSF indices for navigation solutions are also given. Here, best definite number of navigation solutions converges to the same position index in a few seconds especially for mountainous terrain. However, for smooth terrain, indices changes in time unlike other terrain types. On the other hand, pruning method is selected such that best definite number of tracks with minimum likelihood functions selected. Working on other pruning methods may improve navigation solutions for smooth terrains.

Another critical point in the simulations is the percentage of false fix for acquisition mode. In TERCOM, since INS error model is not considered for correlation process, there is always a probability of false fix in the position solutions. In the Monte Carlo simulations performed, a few false position fixes occurred for TERCOM for rough terrain type. On the other hand, with PDAF and TSF no false position fixes occurred.

### 3.4.2.2. Simulations With DTED Level 2

#### 3.4.2.2.1. Terrain Properties

Simulations with DTED Level 2 are performed for a single terrain type shown in Figure 85. Parameters of the selected terrain showed the area to be a rough terrain. Then, terrain heights versus time plot for the selected terrain is given in Figure 86.

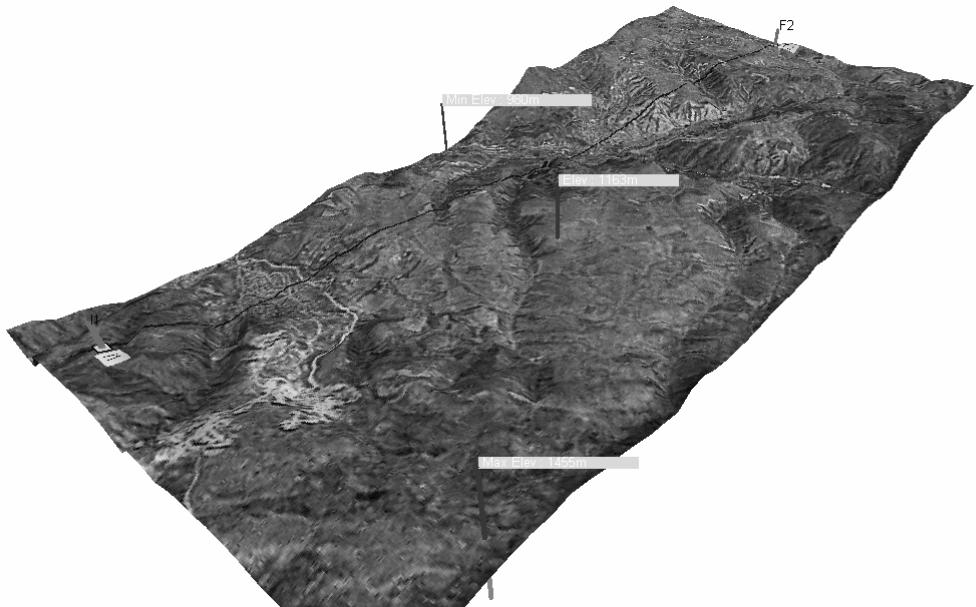


Figure 85. DTED Level 2 Terrain for TAN Simulations

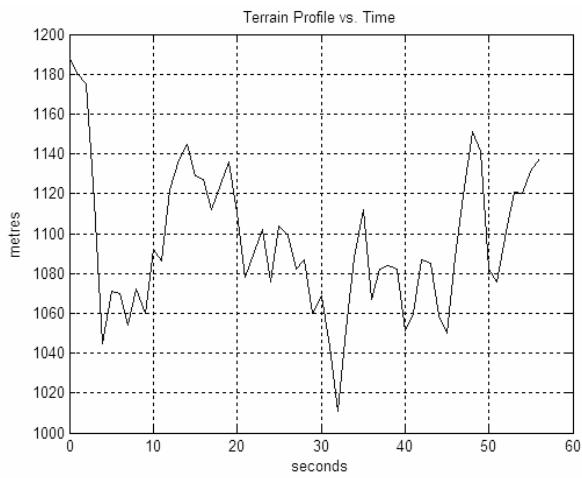


Figure 86. Terrain Height vs. Time for DTED Level 2 Terrain

### **3.4.2.2.2. *Simulation Results***

Simulations are performed for both tracking and acquisition modes of operation as in DTED Level 1 simulations.

#### *3.4.2.2.2.1. *Simulations for Tracking Mode**

Monte Carlo simulations of 100 runs are performed and position errors are obtained for SITAN, PDAF and TSF. Parameters used in the DTED Level 2 simulations are given in Table 22. It should be noted that since DTED Level 2 is used, initial INS error is selected smaller considering grid size of the DTED used.

Table 22. Simulation Parameters for DTED Level 2 Tracking Mode

Initial INS position deviation (one axis)	25 m
Other parameters	Same as in Table 20.

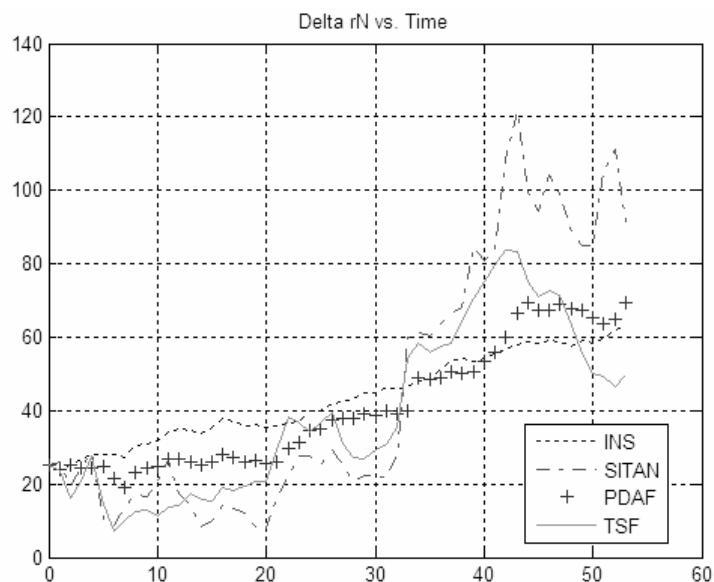


Figure 87. Northward Position Error vs. Time

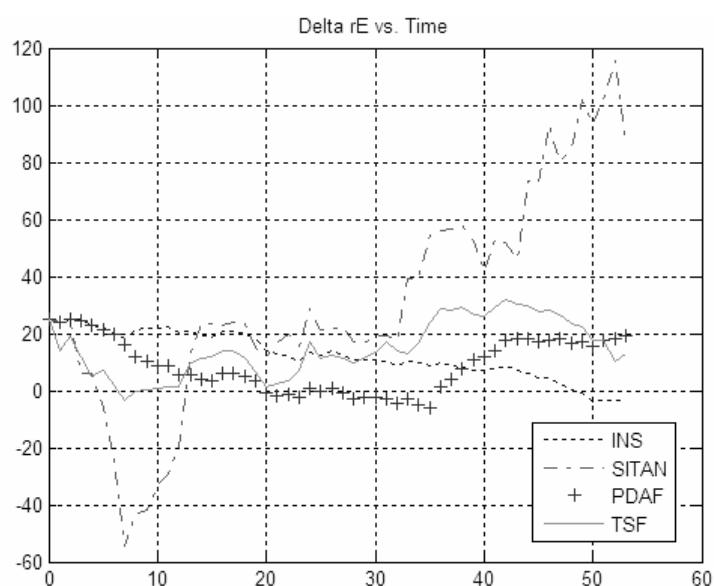


Figure 88. Eastward Position Error vs. Time

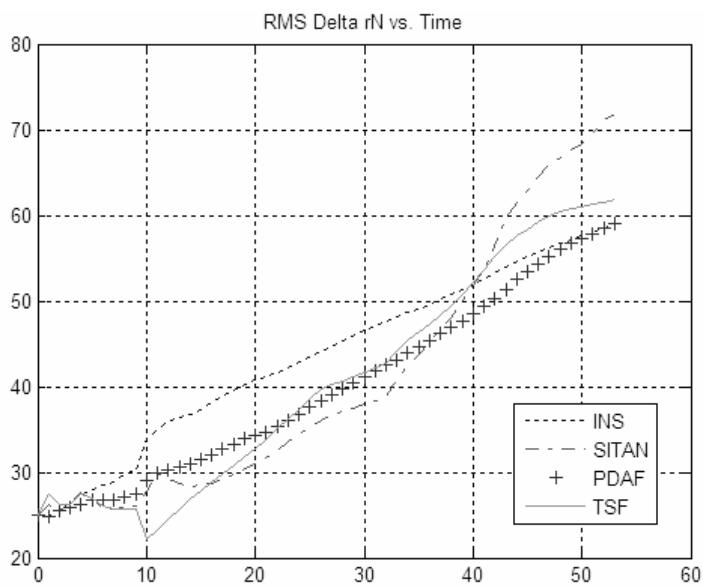


Figure 89. Northward Position RMS Error vs. Time

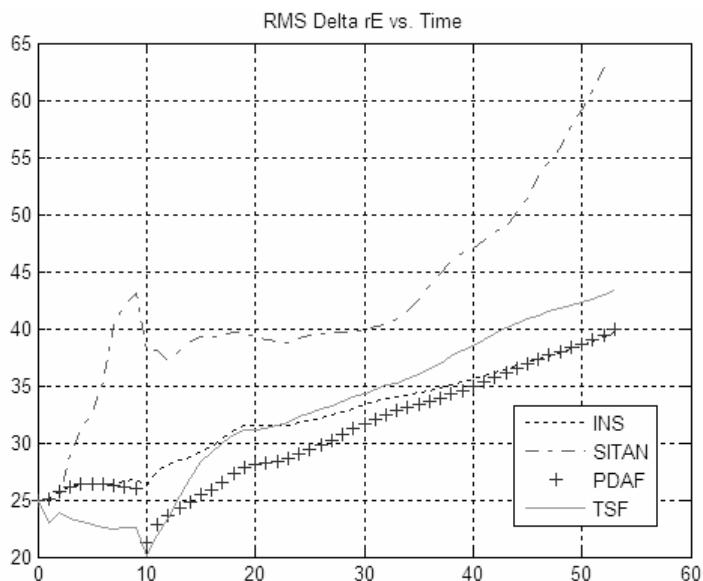


Figure 90. Eastward Position RMS Error vs. Time

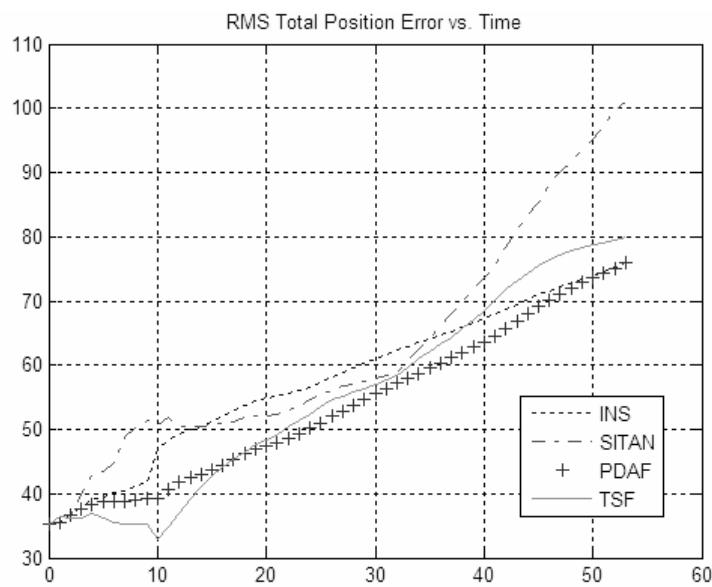


Figure 91. Total Position RMS Error vs. Time

From the simulations, it is seen that better results than SITAN are obtained for the selected terrain type. From the Monte Carlo simulations, position RMS errors of the TSF and PDAF algorithms become less than 25 meters for the selected terrain. It should be noted that DTED Level 2 grid size accuracy is about 30 meters. Hence, real-time accuracy is increased.

#### 3.4.2.2.2.2. Simulations for Acquisition Mode

Parameters used in the simulations for acquisition mode are given in Table 23. Monte Carlo simulations of 100 runs are performed and northward and eastward position errors at the update time are obtained from the plots for TERCOM, PDAF and TSF.

Table 23. Simulation Parameters for DTED Level 2 Acquisition Mode

Initial INS position deviation (one axis)	140 m
Other parameters	Same as in Table 21.

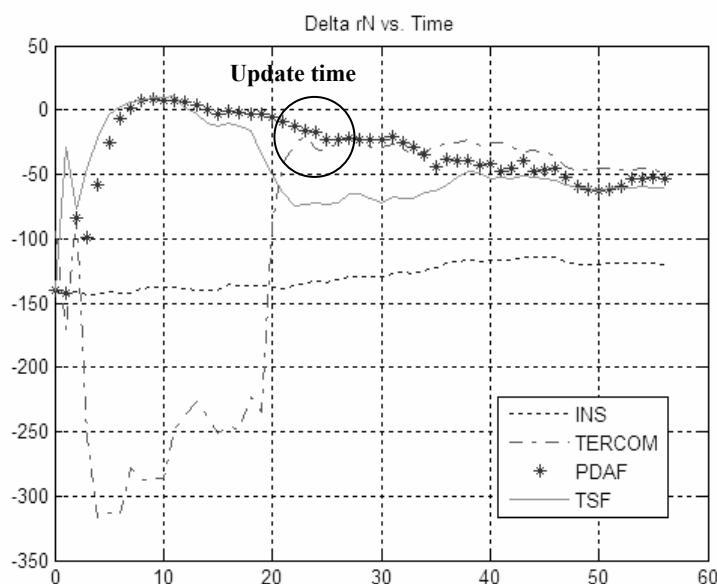


Figure 92. Northward Position Error vs. Time

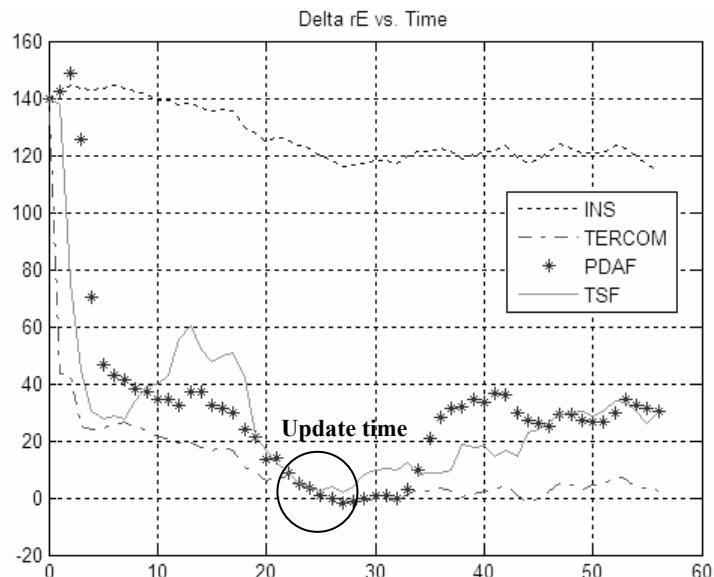


Figure 93. Eastward Position Error vs. Time

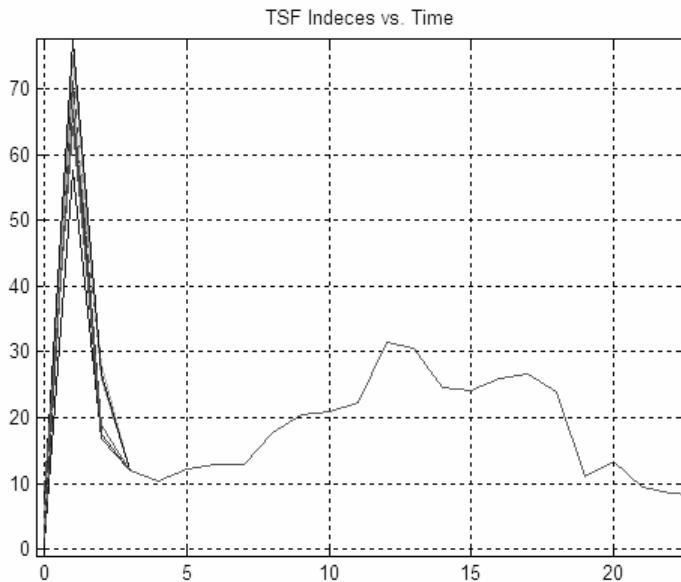


Figure 94. TSF Indices vs. Time

Simulation results show that TAN algorithms also work with DTED Level 2 for acquisition mode. However, DTED Level 1 results seem to be better than DTED Level 2 results. Unfortunately, there were not sufficient DTED Level 2 maps for simulations in order to compare simulation results in detail. Actually, vehicle velocity directly influences TAN performance. For cruise missiles, DTED Level 1 maps are sufficient for mid-course flight navigation solution where INS position fixes less than 50 meters can be obtained. For faster vehicles like cruise missiles, rapid changes in the terrain profile as in DTED Level 2 decreases TAN performance. As a result of this, use of DTED Level 1 maps for TAN acquisition mode seems to perform better solutions.

### 3.4.2.3. Simulations with Various DTED Grid Sizes

Final part of the case studies is done with various DTED grid sizes for PDAF and TSF. Here same initial position errors are taken for simulations in tracking mode along rough terrain. Parameters used in the simulations for acquisition mode are given in Table 24. Monte Carlo simulations of 100 runs are performed and northward and eastward position errors are obtained for PDAF and TSF.

Table 24. Simulation Parameters for Various DTED Grid Sizes

Initial INS position deviation (one axis)	80 m
DTED Grid Size (for PDAF and TSF)	3x3 5x5 7x7 9x9 11x11
Other parameters	Same as in Table 21.

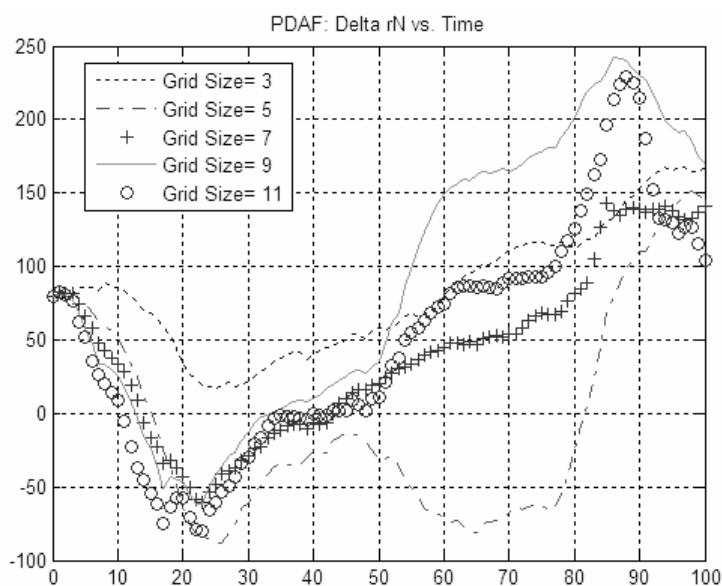


Figure 95. PDAF Northward Position Error vs. Time

It should be noted that initial position errors are taken small in order to have solutions with small DTED grid sizes. Hence, effects of selecting larger DTED grid sizes are examined in this section.

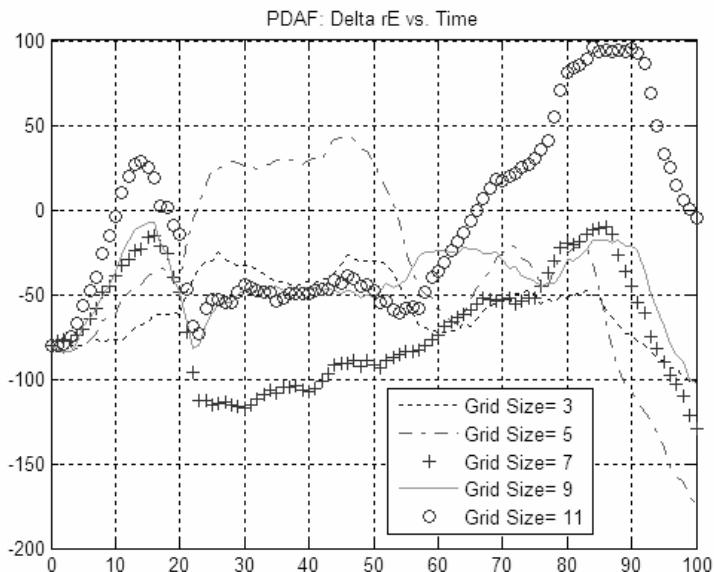


Figure 96. PDAF Eastward Position Error vs. Time

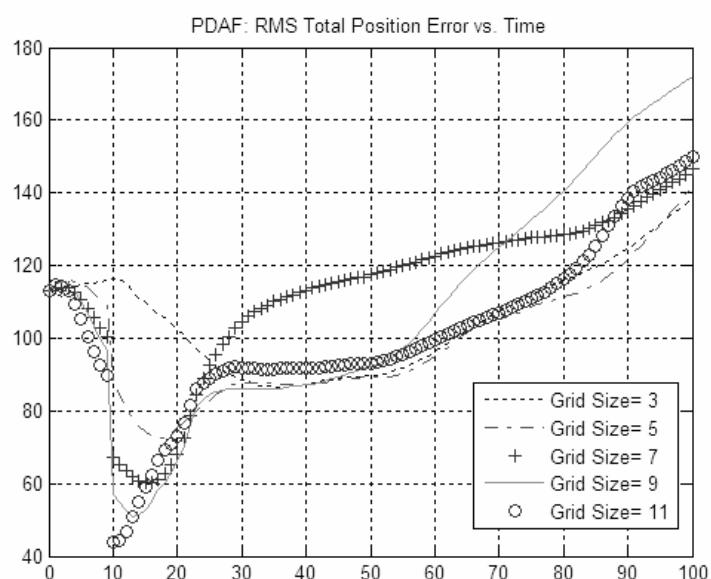


Figure 97. PDAF Total Position RMS Error vs. Time

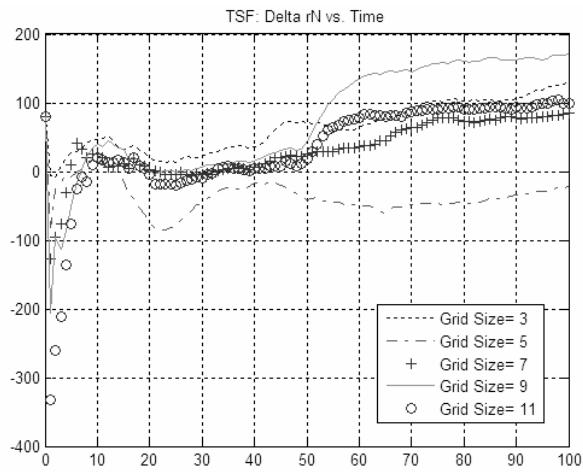


Figure 98. TSF Northward Position Error vs. Time

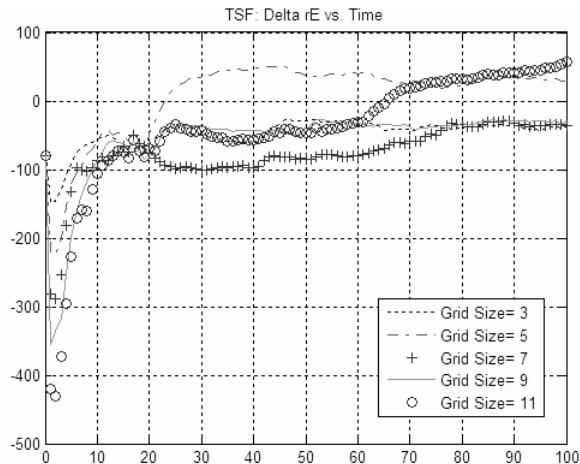


Figure 99. TSF Eastward Position Error vs. Time

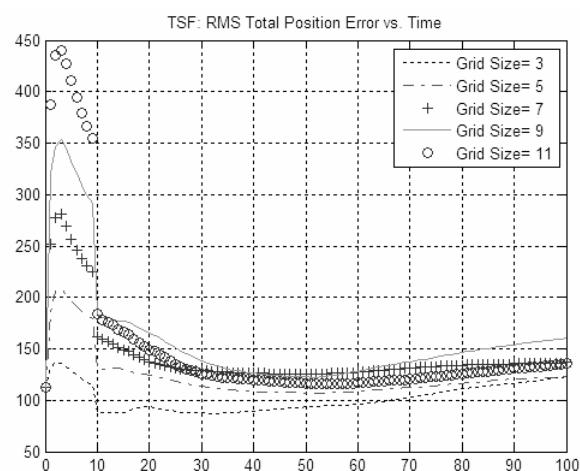


Figure 100. TSF Total Position RMS Error vs. Time

It is seen that changing grid size for the same initial position errors for TSF slightly changes simulation results. Again errors are bounded and limited with the related grid solution. However, selecting larger grid sizes for PDAF solutions generally increase position errors. This is due to PDA procedure where weighted averages of the all grid points are taken into account for navigation solution. Therefore, it can be concluded that PDAF DTED grid size should be selected in accordance with the position errors.

#### **3.4.2.4. Discussion**

From the simulations performed, several conclusions are achieved about the implemented TAN algorithms. The advantages of the new algorithms proposed can be summarized as follows:

1. Real-time TAN solution can be obtained with a single PDA filter. Since past measurements are taken into account, by changing the buffer size of the measurements the filter, measurements are smoothed.
2. Real-time TAN solution can be obtained with a single TSF structure. However, TSF operations are more complex than SITAN. On the other hand, in TSF, more than one track is selected in order to determine navigation solution. Hence, probability of false fix decreases unlike TERCOM.
3. Real-time TAN solution is obtained by considering horizontal position errors of DTED used in real-time PDA filter and TSF. Hence, horizontal position states are added to the Kalman filters used in PDAF and TSF.

4. Application of the filters is simple and the filters are linear, since INS error model is used.
5. Batch size of the DTED area concerned can be changed independent of the model used. Both larger DTED areas for acquisition mode or smaller DTED areas for tracking modes can be selected using the same filters.
6. Results of the filters are good for both recursive and batch algorithms. For tracking mode, position RMS error is less than 50 meters. Moreover, PDAF shows stable response. For smooth terrains where no navigation solution exists, PDAF follows the INS error model which is actually a desired feature.
7. TSF can be considered as a real-time TERCOM process for large position errors, i.e. large DTED batch size. Possibility of false position fixes decrease with TSF when compared with TERCOM. On the other hand, for small position errors, decreasing the weighting factor of the past measurements for TSF, better real-time solutions can be obtained.

## **CHAPTER 4**

### **CASE STUDY**

In this chapter, simulation results of the implemented TAN algorithms are presented for a cruise missile model. First, a 6 DOF simulation tool is developed in order to model cruise missile mid-course flight. Then, related sub-systems are modeled in order to reflect mid-course flight controls and cruise missile navigation system error models. Then, simulations are performed with PDAF and TSF TAN models with actual flight conditions. Finally, simulation results are compared with major TAN algorithms considering other flight parameters of the cruise missile model.

#### **4.1. Simulation Tool Development**

The simulation tool developed for the cruise missile is capable of performing full mid-course flight simulation of the cruise missile modeled. Actually, a generic simulation tool applicable to all air vehicles is considered except for guidance methods applied.

In order to investigate the performance of the TAN algorithms improved, a realistic 6 DOF simulation tool is required. 6 DOF simulation model is built in

Simulink [58] and mid-course flight of the cruise missile is simulated with the model developed. General 6 DOF simulation model architecture is shown in Figure 101 where TAN algorithms are used with the loosely coupled architecture for aiding INS.

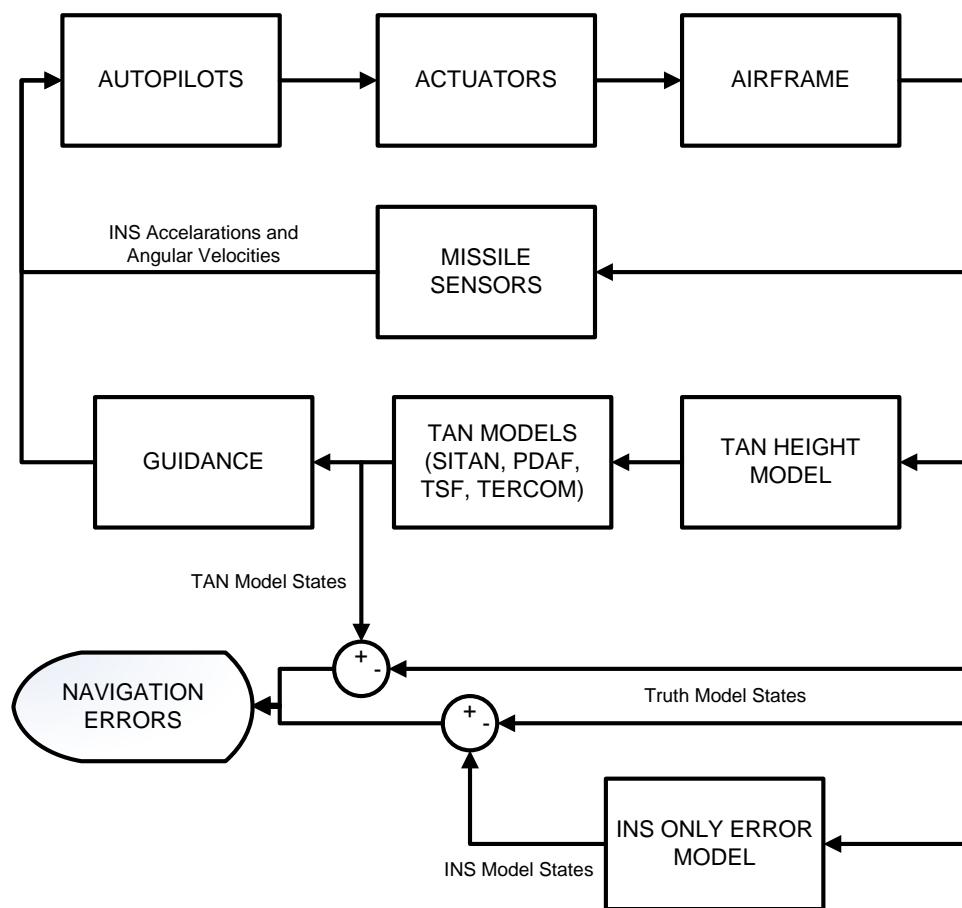


Figure 101. General 6 DOF Simulink Model with Implemented TAN Models

6 DOF simulation model is formed from the following main sub-systems:

1. Airframe: Airframe is composed of various models in order to simulate the dynamic behavior of the cruise missile. Dynamics, kinematics, aerodynamics, propulsion, force and environmental modules constitute airframe subsystem.
2. Autopilots: A variety of controllers are implemented for mid-course flight of the cruise missile: roll control, pitch acceleration controller for altitude hold, yaw stability augmentation, and BTT (bank-to-turn) heading angle tracker autopilots. All autopilots are derived by the pole placement techniques.
3. Actuators: Second-order actuators with rate and position limiters are used in the 6 DOF model in order to control the elevator, rudder and aileron.
4. Sensors: Strapdown INS sensor errors are modeled in order to reflect bias, drift, scale factor and misalignment errors for accelerometer and gyro outputs. Moreover, barometric and radar altimeter outputs are also modeled.
5. Guidance: For the selected waypoints along the missile path, heading correction is applied using the heading angle tracker.
6. INS Error Model: Strapdown INS error model given in equation (2.6) is used in the simulations in order to reflect INS velocity and position errors considering improved bias and drift models of

the INS sensors. However, altitude channel of the INS model is not used considering barometric altimeter measurements.

7. TAN Models: Derived TAN models in the previous chapters for TERCOM, SITAN, PDAF and TSF are directly used in the 6 DOF simulation model by tuning parameters of the derived filters.

6 DOF simulation model sub-systems will be discussed in the following sections except for INS error and TAN models which were investigated in detail in the previous chapters.

#### **4.1.1. Airframe**

Airframe sub-system models dynamic behavior of the cruise missile model. 6 DOF equations of motion are derived and used in this section with the related sub-systems described in the previous section.

Newton's law is applied for translational motion and Euler's law is applied for rotational motion in order to model cruise missile dynamics over an elliptical earth considering earth's rotation. First, reference frames are defined. Then, Newton's and Euler's equations are derived considering forces and moments on the system. Finally, kinematic equations are derived.

#### 4.1.1.1. Reference Frames

Coordinate frames are required in order to define the motion of the vehicle which is considered. Moreover, for kinematic equations, they have to be defined carefully. When the rotation of the earth and earth's geometry is considered, various coordinate frames have to be defined. Coordinate frames used for the simulations can be classified as follows:

1. Inertial Frame (Geocentric Inertial-J2000 Frame),  $\mathfrak{I}_i$
2. Earth Centered Earth Fixed (ECEF) Frame,  $\mathfrak{I}_e$
3. Geographic (North, East, Down – NED) Frame,  $\mathfrak{I}_g$
4. Body Frame,  $\mathfrak{I}_b$
5. Wind Coordinate Frame,  $\mathfrak{I}_w$

The position and attitude of the missile with respect to inertial frame is found using kinematic equations. Since time of flight for the cruise missiles are quite long, the effects of earth's curvature and earth's angular velocity should be added in the transformations.

In Figure 102 and Figure 103, these frames were presented. Transformation matrices are derived between these frames and using translational and rotational transformation equations, positions and attitudes are defined in the proper reference frames as follows considering tensor algebra [78]:

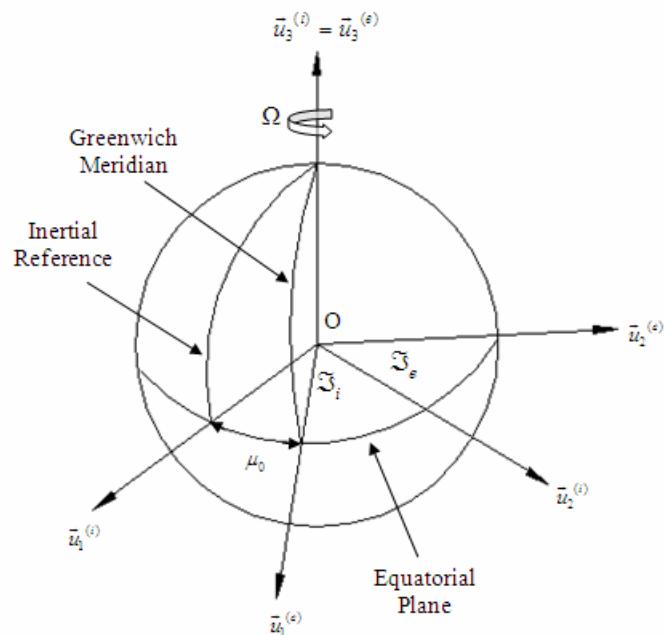


Figure 102. Inertial and ECEF Reference Frames

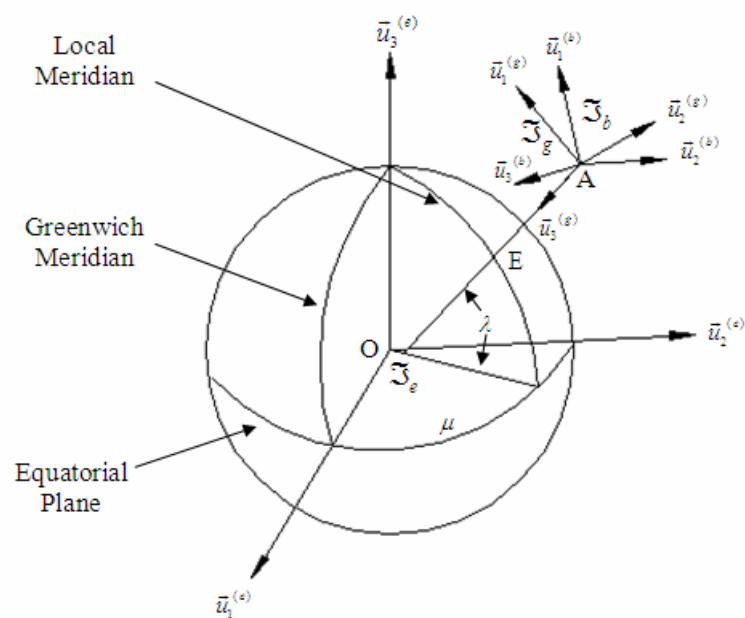


Figure 103. Geographic and Body Reference Frames

$$\bar{r}^{(a)} = \hat{C}^{(a,b)} \cdot \bar{r}^{(b)} \quad (4.1)$$

$$\dot{\hat{C}}^{(i,b)} = \hat{C}^{(i,b)} \cdot \tilde{\omega}_{b/i}^{(b)} \quad (4.2)$$

where,

$\bar{r}$  : Position vector

$\hat{C}$  : Transformation matrix (Direction Cosine Matrix, DCM)

$\tilde{\omega}$  : Skew symmetric matrix form of angular velocity vector,  $\bar{\omega}$

$\bar{\omega}_{b/i}^{(b)}$  : Angular velocity vector of the body frame with respect to inertial frame in body frame

$a, b, i$  : Indices of the related reference frames

Then, transformation procedures for the defined coordinate frames can be applied considering “Rotated Frame Based” (RFB) sequences [78]:

a. Transformation from Body to Geographic Frame

$$\begin{array}{c} \mathfrak{I}_g \xrightarrow[\psi]{\bar{u}_3^{(g)}} \mathfrak{I}_m \xrightarrow[\theta]{\bar{u}_2^{(m)}} \mathfrak{I}_n \xrightarrow[\phi]{\bar{u}_1^{(n)}} \mathfrak{I}_b \\ \hat{C}^{(g,b)} = \begin{bmatrix} c\psi \cdot c\theta & c\psi \cdot s\theta \cdot s\phi - s\psi \cdot c\phi & c\psi \cdot s\theta \cdot c\phi + s\psi \cdot s\phi \\ s\psi \cdot c\theta & s\psi \cdot s\theta \cdot s\phi + c\psi \cdot c\phi & s\psi \cdot s\theta \cdot c\phi - c\psi \cdot s\phi \\ -s\theta & c\theta \cdot s\phi & c\theta \cdot c\phi \end{bmatrix} \end{array} \quad (4.3)$$

( $\psi$ : Yaw angle,  $\theta$ : Pitch angle,  $\phi$ : Roll angle,  $c$ : Cos,  $s$ : Sin)

b. Transformation from Geographic to ECEF Frame

$$\mathfrak{J}_e \xrightarrow[\mu]{\bar{u}_3^{(e)}} \mathfrak{J}_g \xrightarrow[-(\lambda + \frac{\pi}{2})]{\bar{u}_2^{(g')}} \mathfrak{J}_g$$

$$\hat{C}^{(e,g)} = \begin{bmatrix} -s\lambda \cdot c\mu & -s\mu & -c\lambda \cdot c\mu \\ -s\lambda \cdot s\mu & c\mu & -c\lambda \cdot s\mu \\ c\lambda & 0 & -s\lambda \end{bmatrix} \quad (4.4)$$

( $\mu$ : Longitude,  $\lambda$ : Geocentric Latitude)

c. Transformation from ECEF to Inertial Frame

$$\mathfrak{J}_i \xrightarrow[\mu_0]{\bar{u}_3^{(i)}} \mathfrak{J}_e$$

$$\hat{C}^{(i,e)} = \begin{bmatrix} c\Omega \cdot t & -s\Omega \cdot t & 0 \\ s\Omega \cdot t & c\Omega \cdot t & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

( $\Omega$ : Earth's angular velocity)

d. Transformation from Body to Wind Frame

$$\mathfrak{J}_w \xrightarrow[-\beta]{\bar{u}_3^{(w)}} \mathfrak{J}_s \xrightarrow[\alpha]{\bar{u}_2^{(s)}} \mathfrak{J}_b$$

$$\hat{C}^{(w,b)} = \begin{bmatrix} c\alpha \cdot c\beta & s\beta & s\alpha \cdot c\beta \\ -c\alpha \cdot s\beta & c\beta & -s\alpha \cdot s\beta \\ -s\alpha & 0 & c\alpha \end{bmatrix} \quad (4.6)$$

( $\alpha$ : Angle of attack,  $\beta$ : Side slip angle)

Finally, between different coordinate frames successive transformations can be done as follows [78]:

$$\hat{C}^{(i,b)} = \hat{C}^{(i,e)} \cdot \hat{C}^{(e,g)} \cdot \hat{C}^{(g,b)} \quad (4.7)$$

$$\hat{C}^{(e,b)} = \hat{C}^{(e,g)} \cdot \hat{C}^{(g,b)} = \hat{C}^{(e,g)} \cdot \hat{C}^{(b,g)T} \quad (4.8)$$

#### 4.1.1.2. 6 DOF Equations of Motion

For translational motion, Newton's law is applied with respect to inertial frame considering the conventions in Figure 103 in vector notation.

$$m \cdot D_i \vec{v}_{A/O} = \vec{F}_{a,p} + m \cdot \vec{g} \quad (4.9)$$

where,

$m$  : Mass of the vehicle

$D_i$  : Differential operator in inertial frame

$\vec{v}_{A/O}$  : Velocity vector of the center of mass with respect to inertial frame

$\vec{F}_{a,p}$ : Aerodynamic and propulsion forces acting on the vehicle

$\vec{g}$ : Gravitational acceleration

Then, the equations are derived in geographic frame from equation (4.9) considering kinematic equations as follows:

$$D_i \vec{v}_{A/O} = D_e \vec{v}_{A/O} + \vec{\omega}_{e/i} \times \vec{v}_{A/O} \quad (4.10)$$

$$D_e \vec{v}_{A/O} = D_e \vec{v}_{A/E} + D_e (\vec{\omega}_{e/i} \times \vec{r}_{A/O})$$

$$D_e (\vec{\omega}_{e/i} \times \vec{r}_{A/O}) = \underbrace{D_e \vec{\omega}_{e/i}}_{=0} \times \vec{r}_{A/O} + \vec{\omega}_{e/i} \times D_e \vec{r}_{A/O}$$

$\vec{\omega}_{e/i} = \text{Const.}$

$$\vec{\omega}_{e/i} \times D_e \vec{r}_{A/O} = \vec{\omega}_{e/i} \times \left( D_e \vec{r}_{A/E} + \underbrace{D_e \vec{r}_{E/O}}_{=0} \right) = \vec{\omega}_{e/i} \times \vec{v}_{A/E}$$

$\vec{r}_{E/O} = \text{Const.}$

$$D_e \vec{v}_{A/O} = D_e \vec{v}_{A/E} + \vec{\omega}_{e/i} \times \vec{v}_{A/E}$$

$$D_i \vec{v}_{A/O} = D_e \vec{v}_{A/E} + \vec{\omega}_{e/i} \times \vec{v}_{A/E} + \vec{\omega}_{e/i} \times \vec{v}_{A/O}$$

$$\vec{v}_{A/O} = \vec{v}_{A/E} + \vec{\omega}_{e/i} \times \vec{r}_{A/O}$$

$$D_i \vec{v}_{A/O} = D_e \vec{v}_{A/E} + 2 \cdot \vec{\omega}_{e/i} \times \vec{v}_{A/E} + \vec{\omega}_{e/i} \times \vec{\omega}_{e/i} \times \vec{r}_{A/O} \quad (4.11)$$

After replacing equation (4.11) in (4.10), translational equations of motion can be written in geographic frame.

$$D_e \vec{v}_{A/E} = \frac{1}{m} \cdot \vec{F}_{a,p} + \bar{g} - 2 \cdot \vec{\omega}_{e/i} \times \vec{v}_{A/E} - \vec{\omega}_{e/i} \times \vec{\omega}_{e/i} \times \vec{r}_{A/O} \quad (4.12)$$

Equation (4.12) is written in tensor form in geographic frame because of its use in scalar manipulations.

$$\dot{\vec{v}}_{A/E}^{(g)} = \frac{1}{m} \cdot \hat{C}^{(g,b)} \cdot \bar{F}_{a,p}^{(b)} + \bar{g}^{(g)} - 2 \cdot \tilde{\omega}_{e/i}^{(g)} \times \vec{v}_{A/E}^{(g)} - \tilde{\omega}_{e/i}^{(g)} \cdot \tilde{\omega}_{e/i}^{(g)} \cdot \vec{r}_{A/O}^{(g)} \quad (4.13)$$

where,

$\hat{C}^{(g,b)} = \hat{C}^{(g,i)} \cdot \hat{C}^{(b,i)^T}$ : DCM from body to geographic frame

$\bar{F}_{a,p}^{(b)}$ : Aerodynamic and propulsion forces acting on the vehicle

$\bar{g}^{(g)} = [0 \ 0 \ g]^T$ : Gravitational acceleration vector

$\tilde{\omega}_{e/i}^{(g)} = [\Omega \cdot \cos \lambda \ 0 \ -\Omega \cdot \sin \lambda]$ : Earth's angular velocity in  $\mathfrak{I}_g$

$\vec{v}_{A/E}^{(g)} = [vN \ vE \ vD]^T$ : Velocity of the vehicle in geographic frame

Next, Euler's law is applied for rotational motion about the center of mass of the vehicle.

$$D_i \vec{H}_A = D_b \vec{H}_A + \vec{\omega}_{b/i} \times \vec{H}_A = \vec{M}_A \quad (4.14)$$

where,

$\vec{H}_A$ : Angular momentum vector about center of mass ( $\vec{H}_A = \bar{I}_b \cdot \vec{\omega}_{b/i}$ )

$\vec{M}_A$ : Moment vector about center of mass

Again equation (4.14) is written in tensor form in body frame in order to obtain angular velocities of the vehicle.

$$\hat{I}_b^{(b)} \cdot \dot{\tilde{\omega}}_{b/i}^{(b)} + \tilde{\omega}_{b/i}^{(b)} \cdot \hat{I}_b^{(b)} \cdot \bar{\omega}_{b/i}^{(b)} = \bar{M}_a^{(b)}$$

$$\dot{\tilde{\omega}}_{b/i}^{(b)} = \hat{I}_b^{(b)-1} \left( -\tilde{\omega}_{b/i}^{(b)} \cdot \hat{I}_b^{(b)} \cdot \bar{\omega}_{b/i}^{(b)} + \bar{M}_a^{(b)} \right) \quad (4.15)$$

where,

$$\hat{I}_b^{(b)} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} : \quad \text{Moment of inertia matrix of the vehicle}$$

$\bar{M}_a^{(b)}$ : Aerodynamic moments acting on the vehicle in body frame

$\bar{\omega}_{b/i}^{(b)} = [p \quad q \quad r]^T$ : Roll, pitch and yaw rate of the vehicle

#### 4.1.1.3. Kinematic Equations

Translational and angular velocities of the vehicle are obtained from 6 DOF equations of motion. Then, positions and attitudes are derived using kinematic equations.

$$\dot{\vec{r}}_{A/E}^{(g)} = D_e \vec{r}_{A/E} = \vec{v}_{A/E} \quad (4.16)$$

where,

$$\vec{r}_{A/E} = [rN \quad rE \quad rD]^T : \text{Northward, eastward and downward positions}$$

$$\dot{\hat{C}}^{(i,b)} = \hat{C}^{(i,b)} \cdot \tilde{\omega}_{b/i}^{(b)}$$

$$\dot{\hat{C}}^{(b,g)} = \hat{C}^{(i,b)T} \cdot \hat{C}^{(i,g)} \quad (4.17)$$

Using the elements of  $\hat{C}^{(b,g)}$  matrix given in equation (4.3) roll, pitch and yaw angles (i.e.  $\phi$ ,  $\theta$  and  $\psi$ ) are obtained.

#### 4.1.1.4. Aerodynamics and Propulsion

In the previous sections, dynamic and kinematic equations of motion are derived. In order to solve dynamic equations of motion, forces and moments acting on the body should be known. External forces and moments acting on the body are

due to aerodynamics and propulsion. Forces and moments acting on the body are summarized as follows [79]:

$$\bar{F}_{a,p}^{(b)} = \bar{F}_a^{(b)} + \bar{F}_p^{(b)} = \bar{q} \cdot S \cdot \begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix} + \begin{bmatrix} F_p \\ 0 \\ 0 \end{bmatrix} \quad (4.18)$$

$$\bar{F}_{a,p}^{(b)} = \hat{C}^{(b,w)} \cdot \bar{F}_a^{(w)} + \bar{F}_p^{(b)} = \hat{C}^{(b,w)} \cdot \left( \bar{q} \cdot S \cdot \begin{bmatrix} -C_D \\ C_Y \\ -C_L \end{bmatrix} \right) + \begin{bmatrix} F_p \\ 0 \\ 0 \end{bmatrix} \quad (4.19)$$

$$\bar{M}_a^{(b)} = \bar{q} \cdot S \cdot \begin{bmatrix} C_l \cdot b \\ C_m \cdot c \\ C_n \cdot b \end{bmatrix} \quad (4.20)$$

where,

$\bar{F}_a^{(b)}$ : Aerodynamic forces acting on the vehicle in body frame

$\bar{F}_a^{(w)}$ : Aerodynamic forces acting on the vehicle in wind (stability) frame

$\bar{F}_p^{(b)}$ : Propulsion force acting on the vehicle, ( $F_p$ : Scalar value)

$\bar{q}$ : Dynamic pressure ( $\bar{q} = \frac{1}{2} \cdot \rho \cdot V^2$ )

$\rho$ : Density of the ambient atmosphere

$V$ : Velocity of the vehicle

$S$ : Reference area (Theoretical wing area for aircraft type vehicle)

$b$ : Span

$c$ : Chord

$C_x, C_y, C_z$ : Aerodynamic force coefficients (Axial, side and normal force coefficients)

$C_d, C_y, C_l$ : Aerodynamic force coefficients (Drag, side and lift force coefficients)

$C_l, C_m, C_n$ : Aerodynamic moment coefficients (Rolling, pitching and yawing moment coefficients)

Aerodynamic force and moment coefficients are modeled generally by simple Taylor series expansion, including only the linear terms and making all derivatives tabular functions of Mach number and angle of attack considering small side slip angles as follows [79]:

$$C_L = C_{L_0}(M, \alpha) + C_{L_a}(M, \alpha) \cdot \alpha + C_{L_{\delta e}}(M, \alpha) \cdot \delta e + C_{L_q}(M, \alpha) \cdot q \cdot \frac{c}{2V} \quad (4.21)$$

$$C_D = C_{D_0}(M, \alpha) + C_{D_a}(M, \alpha) \cdot \alpha \quad (4.22)$$

$$C_Y = C_{Y_0}(M, \alpha) + C_{Y_\beta}(M, \alpha) \cdot \beta + C_{Y_{\delta a}}(M, \alpha) \cdot \delta a + C_{Y_{\delta r}}(M, \alpha) \cdot \delta r + C_{Y_p}(M, \alpha) \cdot p \cdot \frac{b}{2V} + C_{Y_r}(M, \alpha) \cdot r \cdot \frac{b}{2V} \quad (4.23)$$

$$C_l = C_{l_0}(M, \alpha) + C_{l_\beta}(M, \alpha) \cdot \beta + C_{l_{\delta a}}(M, \alpha) \cdot \delta a + \\ C_{l_{\delta r}}(M, \alpha) \cdot \delta r + C_{l_p}(M, \alpha) \cdot p \cdot \frac{b}{2V} + C_{l_r}(M, \alpha) \cdot r \cdot \frac{b}{2V} \quad (4.24)$$

$$C_m = C_{m_0}(M, \alpha) + C_{m_\alpha}(M, \alpha) \cdot \alpha + C_{m_{\delta e}}(M, \alpha) \cdot \delta e + \\ C_{m_q}(M, \alpha) \cdot q \cdot \frac{c}{2V} \quad (4.25)$$

$$C_n = C_{n_0}(M, \alpha) + C_{n_\beta}(M, \alpha) \cdot \beta + C_{n_{\delta r}}(M, \alpha) \cdot \delta r + \\ C_{n_{\delta a}}(M, \alpha) \cdot \delta a + C_{n_p}(M, \alpha) \cdot p \cdot \frac{b}{2V} + C_{n_r}(M, \alpha) \cdot r \cdot \frac{b}{2V} \quad (4.26)$$

The equations above represent a linear model for constant Mach numbers and angle of attack values. In force coefficients, the effect of body rates on the lift and side forces, i.e., the  $C_{L_q}$ ,  $C_{Y_p}$ , and  $C_{Y_r}$  derivatives can often be neglected [79]. For moment coefficients, rolling and yawing moment coefficients have negligible trim coefficients  $C_{l_0}$  and  $C_{n_0}$  [79].

For 6 DOF simulations, aerodynamic coefficients are obtained from USAF Digital DATCOM software [80]. First, a solid model of a generic cruise missile is formed using a CAD software. Next, mass and inertia properties are obtained from the solid model. Finally, USAF Digital DATCOM model of the cruise missile is prepared in order to obtain aerodynamic coefficients and coefficient derivatives. In Figure 104, a view of the cruise missile solid model is shown. In Table 25, some properties of the cruise missile used in the simulations are given.

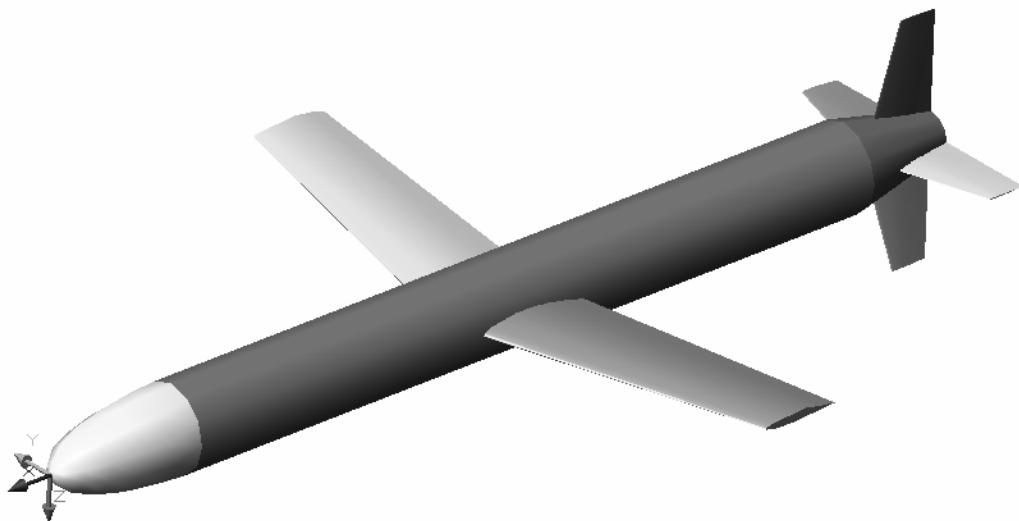


Figure 104. Cruise Missile Solid Model

Table 25. Cruise Missile Model Specifications

Length:	6.25 meters
Initial Missile Weight:	1200 kg
Fuel Weight:	450 kg
Diameter:	53.34 cm
Wing Span:	4.64 meters
Wing Chord:	0.689 meters
Speed:	0.4 – 0.8 Mach
Cruise Speed:	0.7 Mach
Engine Type:	600 lb-f (~2500 N) Type

Some of the coefficients required in equations (4.21) and (4.26) are not directly obtained from Digital DATCOM. Therefore, using the coefficients obtained from the Digital DATCOM manual [80], required coefficients are derived.

Propulsion force acting on the missile is considered from turbojet and acting along axial axis only. Turbojet force is obtained as tabulated results with respect to Mach number and altitude above sea level. The specific fuel consumption (SFC)  $b_F$  is an important indicator for the efficiency of the turbojet. It is defined by the ratio of fuel flow to thrust as [79]:

$$b_F = \frac{\dot{m}_F}{F_p} \quad (4.27)$$

where,

$\dot{m}_F$  : Fuel flow rate

Required thrust  $F_p$  is obtained from Mach control loop. Using look up tables, SFC and fuel flow rates are obtained. Hence, mass of the missile is modeled for simulations.

#### 4.1.1.5. Environmental Models

In order to obtain a generic 6 DOF model, environment is also modeled. Gravity and atmosphere is modeled for simulations and derivations for wind effects are taken into consideration.

For the gravity, WGS84 gravity model is used considering altitude of the vehicle above sea level [81]:

$$g_0 = 9.7803267714 \cdot \left( \frac{1 + 0.00193185138639 \cdot \sin^2 \lambda}{\sqrt{1 - 0.00669437999013 \cdot \sin^2 \lambda}} \right) \quad (4.28)$$

$$g(h) = g_0 \cdot \left( \frac{R^2}{(R+h)^2} \right) \quad (4.29)$$

where,

$\lambda$ : Geographic latitude

$g_0$ : Theoretical gravity

$R$ : Radius of the spherical earth

$h$ : Height of the vehicle above sea level

For the atmosphere model, the 1962 International Standard Atmosphere or ISO 2553 is used given in [79]. Actually, since mid-course flight of a cruise missile at constant speed and altitude is considered, only wind model is required for the simulations.

In 6 DOF simulations, winds and gusts alter the incidence angles and thus change the aerodynamic forces and moments. The incidence angles are calculated from the velocity vector of the vehicle's center of mass with respect to air. To determine velocity of the vehicle with respect to air  $\vec{v}_{A/air}$ , wind vector  $\vec{v}_{air/E}$  is subtracted from the geographic velocity  $\vec{v}_{A/E}$  considering the conventions given in Figure 103 [79].

$$\vec{v}_{A/air} = \vec{v}_{A/E} - \vec{v}_{air/E} \quad (4.30)$$

$$\bar{v}_{A/air}^{(b)} = \hat{C}^{(b,g)} \cdot (\bar{v}_{A/E}^{(g)} - \bar{v}_{air/E}^{(g)})$$

$$\bar{v}_{air/E}^{(g)} = [-V_w \cdot \cos \psi_w \quad -V_w \cdot \sin \psi_w \quad 0] \quad (4.31)$$

where,

$V_w$ : Wind magnitude

$\psi_w$ : Wind direction from north

In the simulations, a pre-specified wind profile with varying magnitude and direction is applied. Wind features will be given in the next section.

#### 4.1.2. Autopilots and Controls

A variety of controllers are implemented for the mid-course flight of the cruise missile:

1. Mach hold control
2. Roll position control
3. Heading angle control with bank-to-turn autopilot
4. Yaw stability augmentation

## 5. Altitude hold control with acceleration autopilot

All autopilots are derived by classical pole placement techniques summarized in Zipfel [79].

### 4.1.2.1. Mach Hold Control

Cruise missiles have to maintain Mach number under maneuvers and environmental effects. The thrust required  $F_r$  to maintain a certain Mach number is equal to drag force projected onto the centerline of the turbine. Considering the turbine axis parallel to the body's first axis, it is required that [79]:

$$F_r = \frac{\bar{q} \cdot S \cdot C_D}{\cos \alpha} \quad (4.32)$$

Mach hold control loop is shown in Figure 105. The time constant  $T_F$  of a generic turbojet engine is between 0.2 and 1.0 seconds. Gain  $G_M$  is calculated from second degree closed loop transfer function (TF) considering natural frequency and damping of the system as follows [79]:

$$G_M = \frac{m \cdot V_s}{4 \cdot T_F \cdot \zeta^2} \quad (4.33)$$

where,

$m$  : Mass of the missile

$V_s$  : Sonic speed ( $V_s = \sqrt{\gamma \cdot R \cdot T}$ )

$\zeta$  : Damping of the closed loop TF

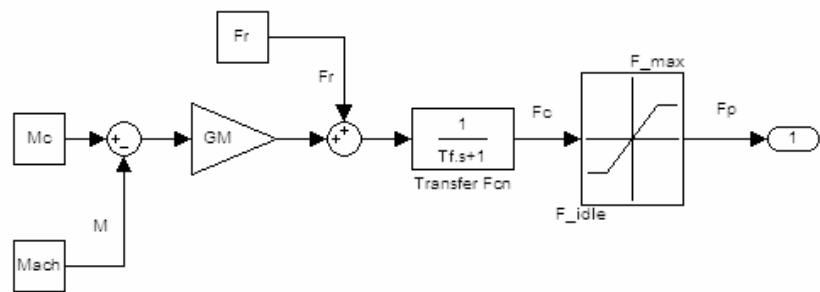


Figure 105. Mach Hold Control Loop [79]

#### 4.1.2.2. Roll Position Control

A dual feedback controller is built for roll position autopilot. The inner rate loop augments the aerodynamic damping and the outer loop executes the roll position command as shown in Figure 106. The transfer function between roll and aileron command is rather simple as follows [79]:

$$\frac{p(s)}{\delta a(s)} = \frac{LL_{\delta a}}{s - LL_p} \quad (4.34)$$

where,

$LL_{\delta a}$ : Roll control derivative,  $LL_{\delta a} = (\bar{q}Sb/I_1)C_{l_{\delta a}}$

$LL_p$ : Roll damping derivative,  $LL_p = (\bar{q}Sb/I_1)(b/2V)C_{l_p}$

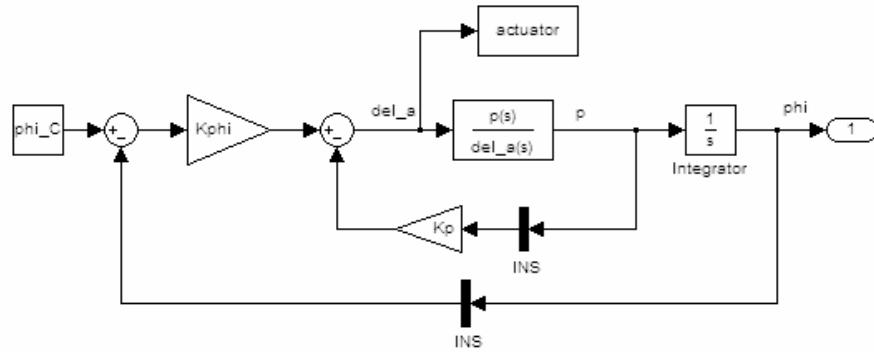


Figure 106. Roll Rate and Position Feedback Loops [79]

Using closed loop TF of the roll position against commanded roll position and setting parameters for second order TF, roll position autopilot gains  $K_\phi$  and  $K_p$  can be determined as follows [79]:

$$\frac{\phi(s)}{\phi_c(s)} = \frac{K_\phi \cdot LL_{\delta a}}{s^2 + \underbrace{(K_p \cdot LL_{\delta a} - LL_p)}_{2\zeta_{roll}\omega_{n_{roll}}} s + \underbrace{K_\phi \cdot LL_{\delta a}}_{\omega_{n_{roll}}^2}} \quad (4.35)$$

$$K_\phi = \frac{\omega_{n_{roll}}^2}{LL_{\delta a}} \quad (4.36)$$

$$K_p = \frac{2\zeta_{roll}\omega_{n_{roll}} + LL_p}{LL_{\delta a}} \quad (4.37)$$

#### 4.1.2.3. Heading Angle Control

Heading changes of the cruise missiles are executed by roll control since bank-to-turn control is used. As the lift vector is banked, a horizontal force component generates a lateral acceleration that turns the velocity vector horizontally [79].

Heading angle tracker is built by wrapping a heading loop around the roll position autopilot as shown in Figure 107. Again, the pole placement technique from root locus analysis is applied in order to determine heading gain  $K_\psi$  using the open loop TF derived from the figure and equation (4.35) [79]:

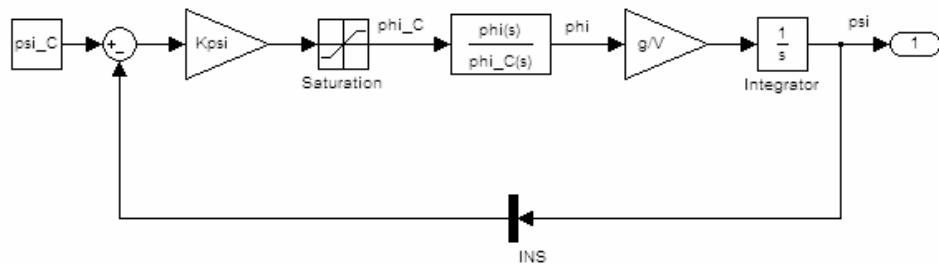


Figure 107. Heading Angle Tracker [79]

$$TF_{OL}(s) = K_\psi \cdot \frac{g}{Vs} \cdot \frac{\phi(s)}{\phi_c(s)} = K_\psi \cdot \frac{g}{V} \cdot \frac{\omega_{n_{roll}}^2}{s^3 + 2\zeta_{roll}\omega_{n_{roll}}s^2 + \omega_{n_{roll}}^2 s} \quad (4.38)$$

$$K_y = \frac{V}{g} \cdot \zeta_{roll} \cdot \omega_{n_{roll}} \cdot (1 - \zeta_{roll}^2) \quad (4.39)$$

#### 4.1.2.4. Yaw Stability Augmentation

Dynamic stability of the vehicle in yaw plane is improved by the yaw rate damping loop. Yaw rate feedback loop is shown in Figure 108. The transfer function between yaw rate and rudder command is found from the linear perturbation equations of the missile yaw plane which can be found in various references [79]:

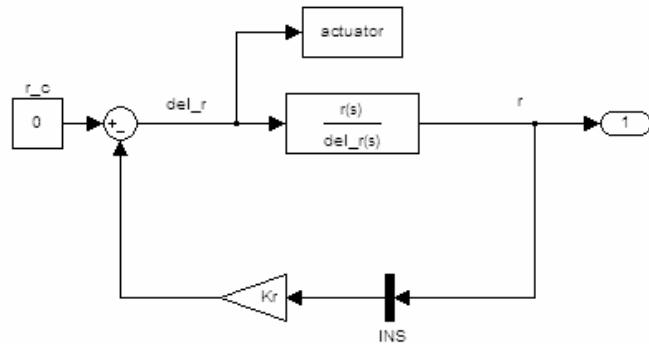


Figure 108. Yaw Rate Feedback Loop [79]

$$\frac{r(s)}{\delta r(s)} = \frac{LN_{\delta r} \left[ s - Y_\beta/V + (LN_\beta/LN_{\delta r})(Y_{\delta r}/V) \right]}{s^2 - (Y_\beta/V + LN_r)s + LN_\beta + LN_r Y_\beta/V} = \frac{G_r(s+z)}{s^2 + as + b} \quad (4.40)$$

where,

$Y_i$ : Dimensionalized derivatives of  $i$  for side force  $Y$

$LN_i$ : Dimensionalized yawing moment of  $i$

$$Y_\beta = (\bar{q}S/m)C_{Y_\beta}, \quad Y_{\delta r} = (\bar{q}S/m)C_{Y_{\delta r}},$$

$$LN_\beta = (\bar{q}Sb/I_3)C_{n_\beta}, \quad LN_{\delta r} = (\bar{q}Sb/I_3)C_{n_{\delta r}}, \quad LN_r = (\bar{q}Sb/I_3)(b/2V)C_{n_r}.$$

Using closed loop TF of the yaw rate against commanded yaw rate derived from Figure 108 and equation (4.40) and selecting the closed loop damping coefficient  $\zeta_r$ , yaw rate gain  $K_r$  is determined as follows [79]:

$$\frac{r(s)}{r_c(s)} = \frac{G_r(s+z)}{s^2 + (a + K_r G_r)s + b + K_r G_r z} = \frac{G_r(s+z)}{s^2 + 2\zeta_{yaw}\omega_{n_{yaw}}s + \omega_{n_{yaw}}^2} \quad (4.41)$$

$$K_r = \frac{1}{G_r} \left[ -\left( a - 2\zeta_{yaw}^2 z \right) + \sqrt{\left( a - 2\zeta_{yaw}^2 z \right)^2 - \left( a^2 - 4\zeta_{yaw}^2 b \right)} \right] \quad (4.42)$$

$$\omega_{n_{yaw}} = \sqrt{b + K_r G_r z} \quad (4.43)$$

#### 4.1.2.5. Altitude Hold Control

In order to build altitude hold autopilot, two feedback loops are wrapped around the normal acceleration autopilot with two gains  $G_H$  and  $G_v$  determining the dynamic response as shown in Figure 109. These gains are determined from the root locus analysis of the inner and outer altitude loops. Adaptive gain scheduling is not required for  $G_H$  and  $G_v$ , since altitude corridors are usually fixed and a constant

set of gains is sufficient. Actually, constant gain values give also good performance for terrain following [79].

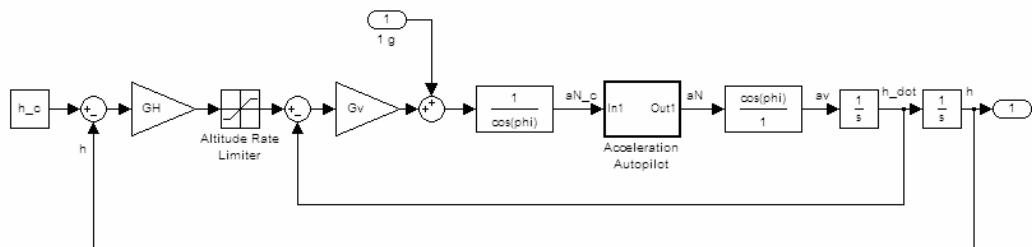


Figure 109. Altitude Hold Autopilot [79]

#### 4.1.2.6. Acceleration Autopilot

In cruise missiles, the normal load factor plane generally contains an acceleration feedback loop. Guidance systems of cruise missiles like terrain following and obstacle avoidance require rapid response that only an acceleration autopilot can provide [79].

Acceleration autopilot loop is shown in Figure 110. The transfer function between pitch rate, normal acceleration and elevator command is found from the linear perturbation equations of the missile pitch plane which can be found in various references [79]:

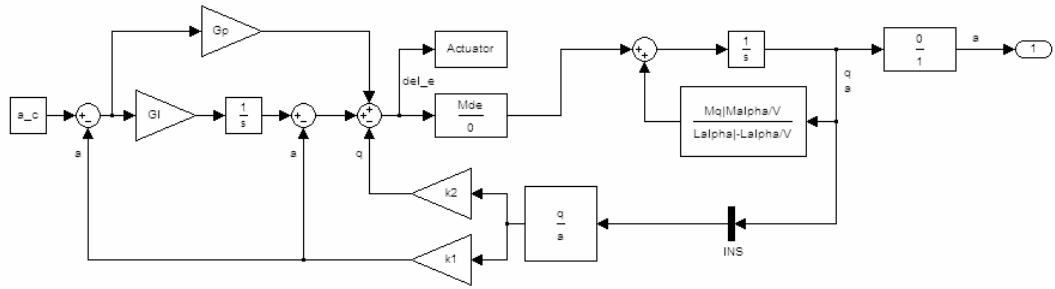


Figure 110. Acceleration Autopilot Loop [79]

$$\begin{bmatrix} \dot{q} \\ \dot{a}_N \end{bmatrix} = \begin{bmatrix} M_q & \frac{M_\alpha}{L_\alpha} \\ L_\alpha & -\frac{L_\alpha}{V} \end{bmatrix} \cdot \begin{bmatrix} q \\ a_N \end{bmatrix} + \begin{bmatrix} M_{\delta e} \\ 0 \end{bmatrix} \cdot \delta e \quad (4.44)$$

where,

$L_\alpha$ : Dimensionalized derivative of lift force  $L$  with respect to  $\alpha$ .

$M_i$ : Dimensionalized pitching moment of  $i$

$$L_\alpha = (\bar{q}S/m)C_{L_\alpha},$$

$$M_\alpha = (\bar{q}Sc/I_2)C_{m_\alpha}, \quad M_{\delta e} = (\bar{q}Sc/I_2)C_{m_{\delta e}}, \quad M_q = (\bar{q}Sc/I_2)(c/2V)C_{m_q}.$$

Using closed loop TF of the normal acceleration against commanded normal acceleration derived from Figure 110 and equation (4.44) and selecting the closed

loop damping coefficient  $\zeta_{pitch}$ , the natural frequency  $\omega_{pitch}$  and a pole location  $p$ , acceleration autopilot gains are determined as follows [79]:

$$\frac{a_N(s)}{a_c(s)} = \frac{M_{\delta e}VL_\alpha(G_p s + G_I)}{s^3 V + s^2(L_\alpha - M_q V + M_{\delta e}k_2 V) + \dots} \\ s \left( \begin{array}{l} -M_q L_\alpha - M_\alpha V + M_{\delta e} k_1 V L_\alpha + \dots \\ M_{\delta e} k_2 L_\alpha + M_{\delta e} G_p V L_\alpha \end{array} \right) + M_{\delta e} G_I V L_\alpha \quad (4.45)$$

$$G_I = \frac{\omega_{pitch}^2 p}{L_\alpha M_{\delta e}} \quad (4.46)$$

$$k_2 = \frac{1}{M_{\delta e}} \left( 2\zeta_{pitch} \omega_{pitch} + p + M_q - \frac{L_\alpha}{V} \right) \quad (4.47)$$

$$k_1 = \frac{1}{L_\alpha M_{\delta e}} \left( \omega_{pitch} + 2\zeta_{pitch} \omega_{pitch} + M_\alpha + \frac{M_q L_\alpha}{V} - k_2 \frac{M_{\delta e} L_\alpha}{V} \right) - G_p \quad (4.48)$$

As it can be seen from the equations, the position feed-forward gain  $G_p$  can not be determined from pole placement technique. Therefore, it must be determined from root locus analysis. Fixed value of gain  $G_p$  determined from root locus gave sufficient performance for acceleration autopilot designed.

#### 4.1.3. Actuators

An actuator is a device that actualizes steering inputs to motivators. These motivators are aileron, elevator and rudder for the cruise missile. For the

simulations, the response of the fin actuator is modeled by a second order TF as [79]:

$$\frac{\delta i(s)}{\delta i_c(s)} = \frac{\omega_{n_{act}}^2}{s^2 + 2\zeta_{act}\omega_{n_{act}}s + \omega_{n_{act}}^2} \quad (4.49)$$

where,

$\delta i$  : Actual control surface deflection

$\delta i_c$  : Fin command

$\omega_{n_{act}}$  : Natural frequency of the actuator

$\zeta_{act}$  : Damping ratio of the actuator

Although the TF models only the linearized dynamics, fin deflection and fin rate limiters are included in the actuator model as two important nonlinearities which are shown in Figure 111.

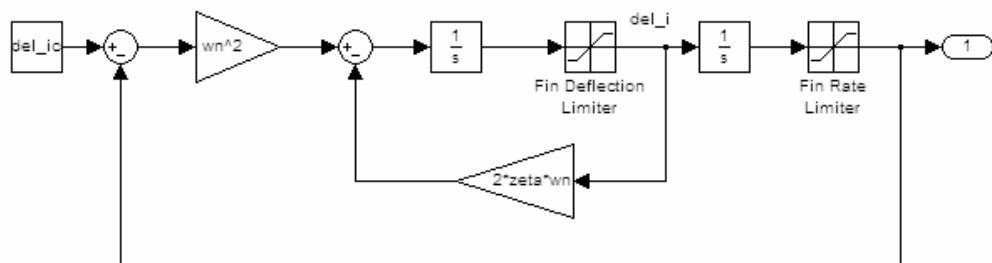


Figure 111. Second Order Actuator Model [79]

Actuator model derived is applied for all control channels (i.e. elevator, rudder and aileron) considering critical damping for the actuators by selecting large natural frequency values compared to autopilot natural frequencies. (For the simulations,  $\omega_{n_{act}} = 100$  rad/s is selected for all channels.)

#### 4.1.4. Sensors

Cruise missile model contain IMU and altimeter sensors. Accelerometers and gyros are used in the IMU of the missile. For height channel stability of the INS, barometric altimeter is used. For terrain clearance measurements in order to apply TAN algorithms, a radar altimeter is also required.

Accelerometers are modeled considering strapdown INS architecture with random bias and noise, scale factor and misalignment as follows [79]:

$$\bar{f}_{acc}^{(b)} = \bar{f}^{(b)} + \delta \bar{f}^{(b)} \quad (4.50)$$

$$\delta \bar{f}^{(b)} = \bar{\nabla} + (\hat{S}_a + \hat{M}_a) \cdot \bar{f}^{(b)} + \bar{w}_a \quad (4.51)$$

where,

$\bar{f}_{acc}^{(b)}$ : Output of the accelerometers

$\bar{f}^{(b)}$ : True acceleration values

$\delta \bar{f}^{(b)}$ : Accelerometer errors

$\bar{\nabla}$ ,  $\bar{w}_a$ : Random bias and white noise vector

$$\hat{S}_a = \begin{bmatrix} aSF_x & 0 & 0 \\ 0 & aSF_y & 0 \\ 0 & 0 & aSF_z \end{bmatrix}; \quad \text{Scale factor error matrix}$$

$$\hat{M}_a = \begin{bmatrix} 0 & aMA_{xy} & aMA_{xz} \\ -aMA_{xy} & 0 & aMA_{yz} \\ -aMA_{xz} & -aMA_{yz} & 0 \end{bmatrix}; \quad \text{Misalignment matrix}$$

Gyro errors are also modeled in the same manner [79]:

$$\bar{\omega}_{b/i_{gyro}}^{(b)} = \bar{\omega}_{b/i}^{(b)} + \delta\bar{\omega}_{b/i}^{(b)} \quad (4.52)$$

$$\delta\bar{\omega}_{b/i}^{(b)} = \bar{\varepsilon} + (\hat{S}_g + \hat{M}_g) \cdot \bar{\omega}_{b/i}^{(b)} + \bar{w}_g \quad (4.53)$$

where,

$\bar{\omega}_{b/i_{gyro}}^{(b)}$  : Output of the gyros

$\bar{\omega}_{b/i}^{(b)}$  : True angular rates

$\delta\bar{\omega}_{b/i}^{(b)}$  : Gyro errors

$\bar{\varepsilon}$ ,  $\bar{w}_g$  : Random drift and white noise vector

$\hat{S}_g$  : Scale factor error matrix

$\hat{M}_g$  : Misalignment matrix

Barometric and radar altimeters are modeled with scale factor error and random noise. Actually, there exist detailed error models in the literature especially for barometric altimeters [82]. However, simple error models are sufficient for simulation purposes, since the effects of other errors can be neglected. Then error models for the altimeters become as follows [83]:

$$\delta h_{baro} = S_{baro} \cdot h_{baro} + v_{baro} \quad (4.54)$$

$$\delta h_{radar} = S_{radar} \cdot h_{radar} + v_{radar} \quad (4.55)$$

where,

$\delta h$  : Altimeter errors

$S$  : Scale factors

$v$  : Measurement white noises

## 4.2. Simulations

In the previous section of the chapter, 6 DOF simulation models were developed. In this section, simulation results of a realistic cruise missile operation scenario with TAN will be discussed in detail.

The operation scenario of the cruise missile model used in the simulations can be summarized as follows:

1. 6 DOF cruise missile model with TAN simulations are performed with the loosely coupled architecture shown in Figure 101.

2. Simulation path is selected such that the missile had a constant heading through north for a constant time with large position errors. Then, TAN algorithms are applied during this time period in order to correct position errors. Next, the heading is corrected at discrete time intervals with the applied TAN algorithms in order to reach the desired waypoint. Simulation path used for the simulation is shown in Figure 112. Actually, selected terrain profile is a rough terrain.

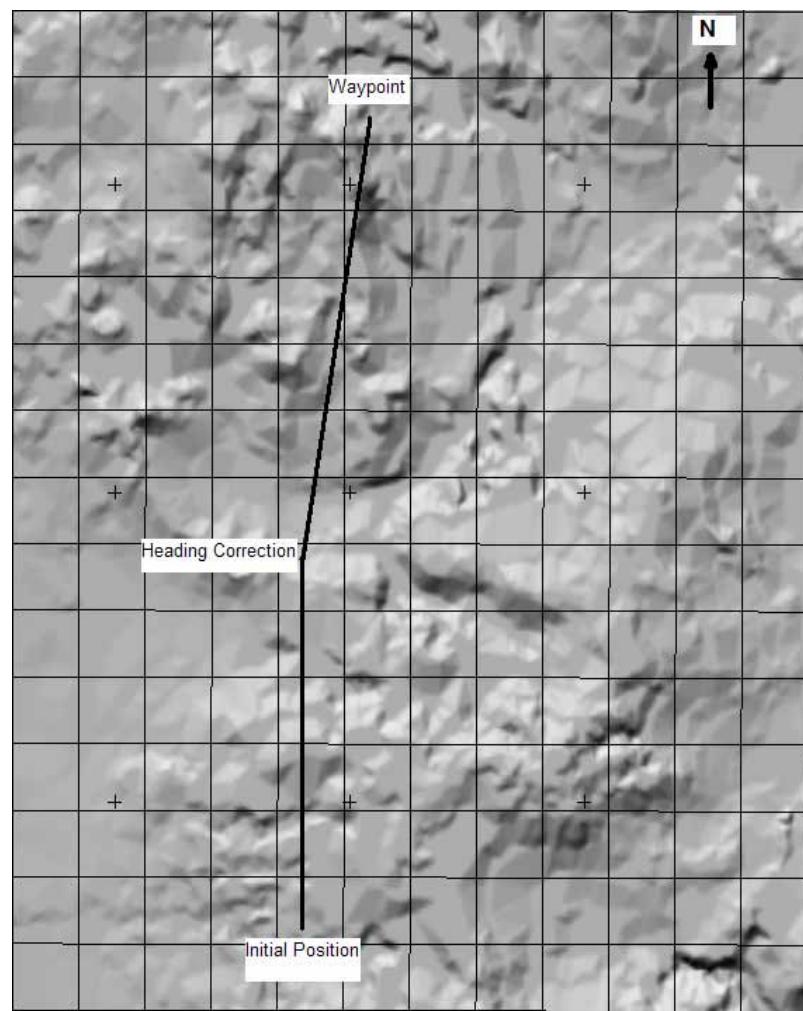


Figure 112. Simulation Path Used for 6 DOF Cruise Missile Simulation

3. Large DTED grid size is used in acquisition mode for TAN algorithms and after first INS update, DTED grid size is decreased in tracking modes of operation.
4. 10 nmi/hr class INS sensor model parameters are used in order to model the INS used in the simulations. Initial errors are used for the INS in order to simulate cruise missile model, such that simulations began after about half an hour of operation.
5. In order to accelerate simulation times, simulations are performed with constant aerodynamic parameters derived at the cruise Mach number. Hence, aerodynamic coefficients become function of angle of attacks only.
6. Control strategy used for the simulations is as follows:
  - a. Terrain following guidance is applied throughout the simulations. Altitude hold control is performed with pitch acceleration controller.
  - b. Constant speed is achieved by Mach hold control loop throughout the operation.
  - c. Heading angle tracker is used with bank-to-turn controller. Constant heading for initial period of operation is used. Then, heading updates are done at discrete times when INS is updated.
  - d. Yaw stability is obtained with yaw stability augmentation controller.

Table 26. 6 DOF Cruise Missile Model Simulation Parameters

Simulation Scenario	<ol style="list-style-type: none"> <li>1. 0 – 41 s. : Heading = 0°</li> <li>2. At 41 s. : Initial heading correction for the desired waypoint</li> <li>3. At 81 s. : Second heading correction</li> </ol>
Controls Applied	<ol style="list-style-type: none"> <li>1. Terrain following with altitude hold</li> <li>2. Heading angle tracker with BTT control</li> <li>3. Yaw stability controller</li> <li>4. Mach hold control</li> </ol>
Environmental Conditions	<ol style="list-style-type: none"> <li>1. ISO 2553 atmosphere model</li> <li>2. Pre-specified wind profile</li> <li>3. Universal gravity model</li> </ol>
Commanded Mach Number	0.7 M
Commanded Height	300 m AGL
INS Quality	10 nmi/hr Class (Parameters are given in Table 5)
Initial Horizontal INS Position Error (Total Horizontal Position Error)	~ 400 m
Initial Horizontal INS Velocity Error Standard Deviation (Each axis, 1 $\sigma$ )	0.5 m/s
Initial Vertical INS Position Error Standard Deviation (1 $\sigma$ )	5 m
Barometric Altimeter Standard Deviation (1 $\sigma$ )	3 m
Radar Altimeter Standard Deviation (1 $\sigma$ )	3 m
Initial INS Attitude Errors Standard Deviation (1 $\sigma$ )	0.05°
TAN Algorithms Update Interval	0.5 s
INS Update Interval	40 s
TERCOM Update Interval	55 s
Initial DTED Grid Size (Acquisition Mode)	11 x 11 (for PDAF and TSF)
Tracking Mode DTED Grid Size	5 x 5 (for PDAF and TSF)

Parameters used in the simulations are summarized in Table 26. Simulations are performed in real-time with these parameters for three INS position updates. Next, simulation results are presented. First, 6 DOF simulation results of the system are given as follows:

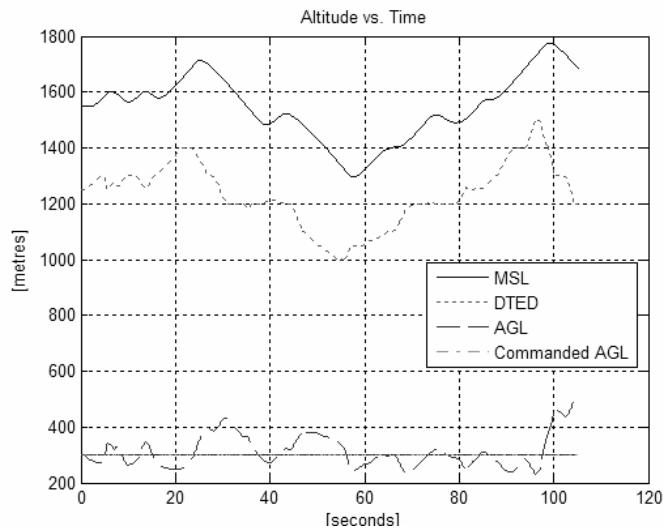


Figure 113. Altitude vs. Time

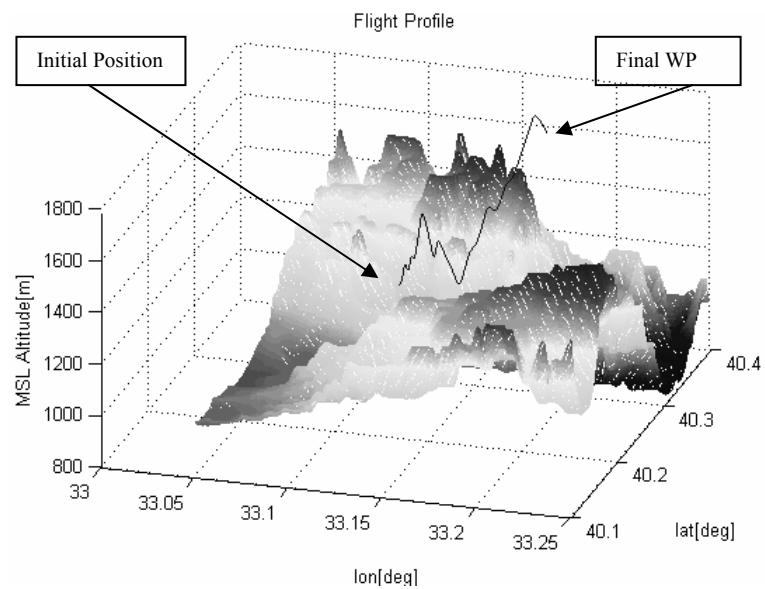


Figure 114. Flight Profile over the Terrain

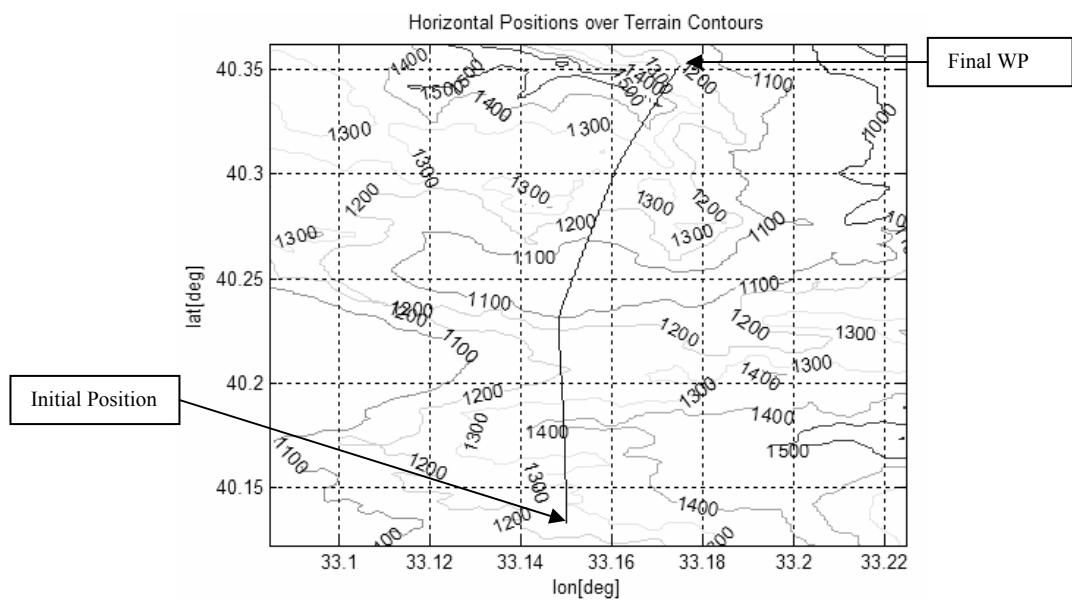


Figure 115. Latitude vs. Longitude over Terrain Contours

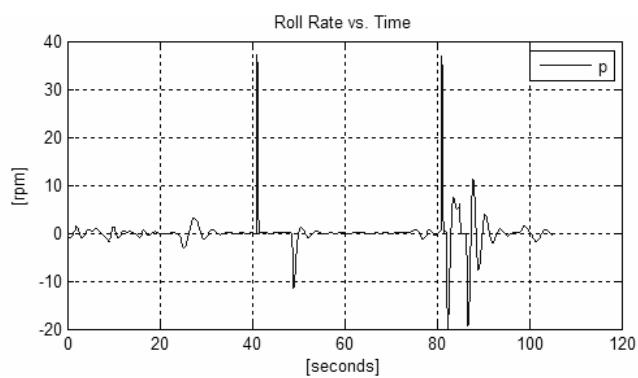


Figure 116. Roll Rate vs. Time

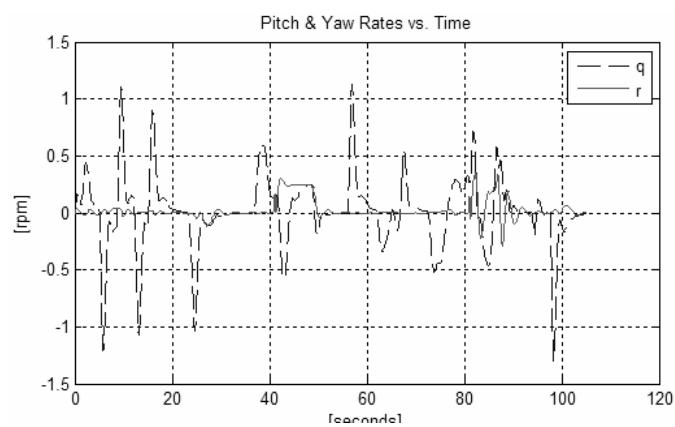


Figure 117. Pitch and Yaw Rates vs. Time

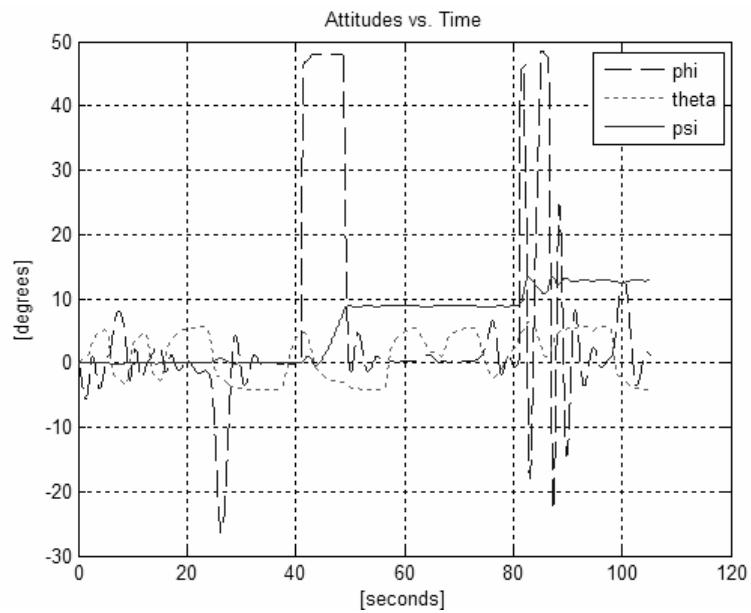


Figure 118. Attitudes vs. Time

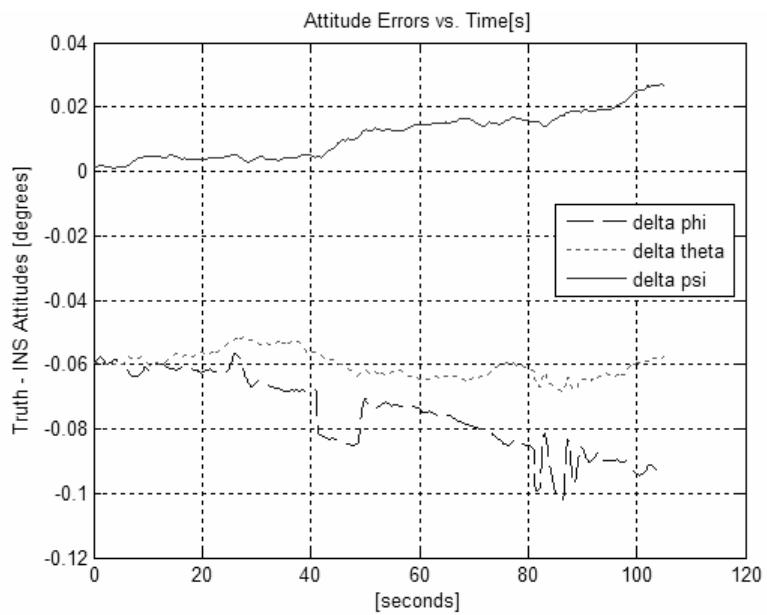


Figure 119. Attitude Errors vs. Time

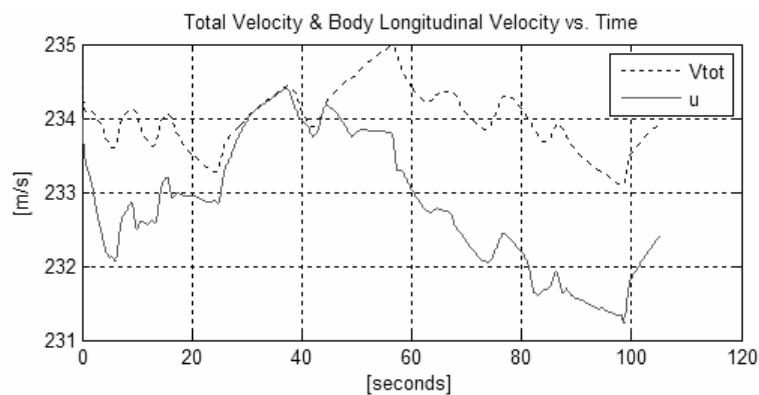


Figure 120. Total Velocity and Body Longitudinal Velocity vs. Time

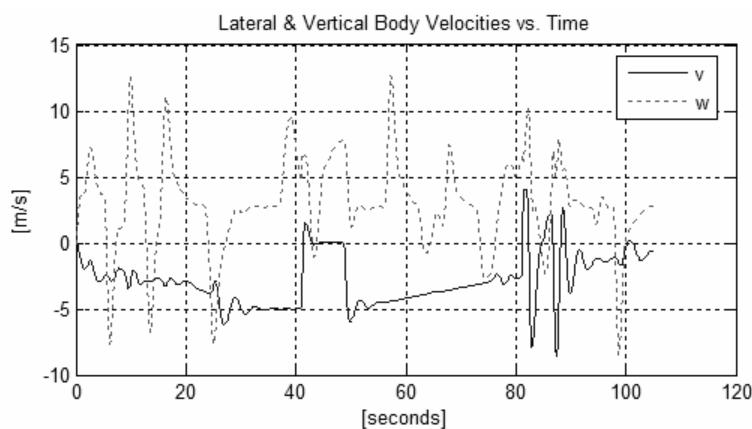


Figure 121. Lateral and Vertical Body Velocities vs. Time

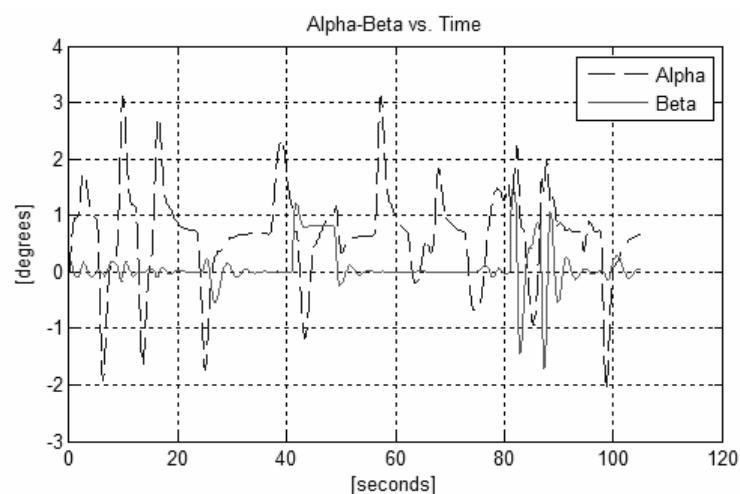


Figure 122. Angle of Attack and Side Slip Angle vs. Time

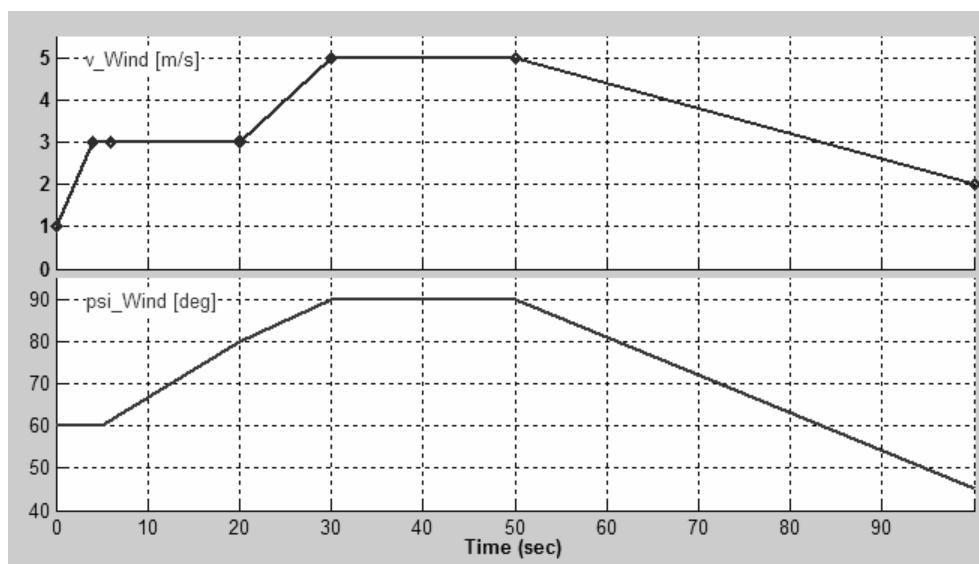


Figure 123. Wind Profile (Wind Velocity and Wind Heading) vs. Time

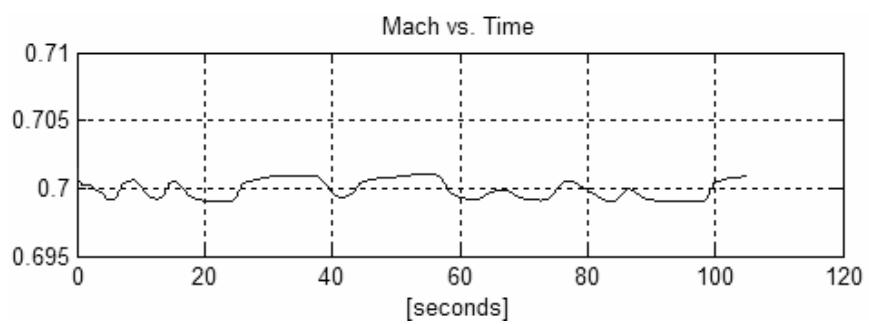


Figure 124. Mach vs. Time

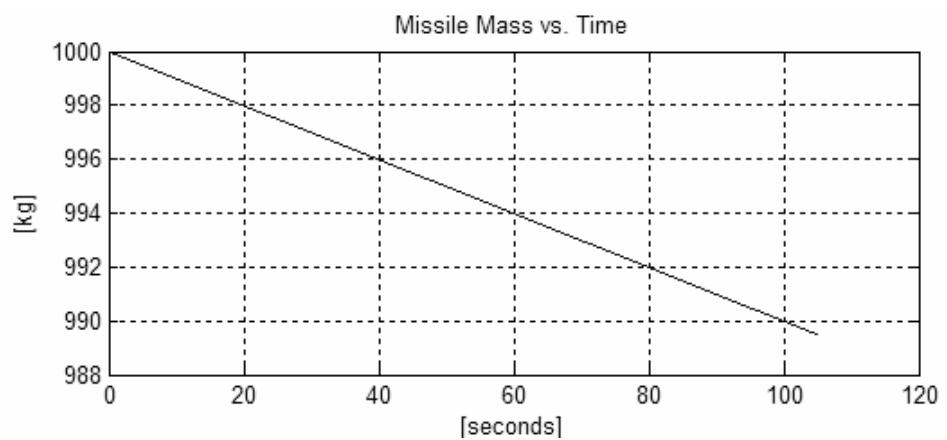


Figure 125. Missile Mass vs. Time

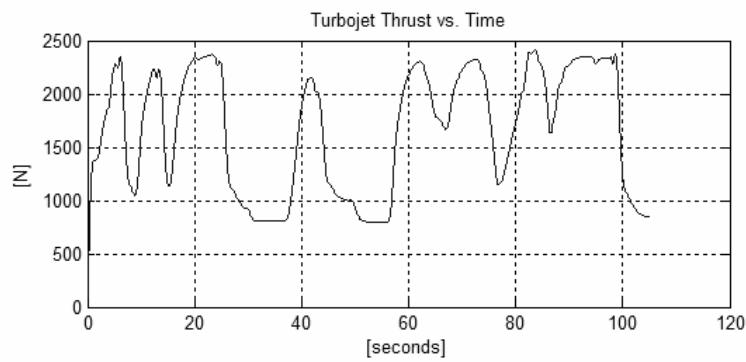


Figure 126. Turbojet Thrust vs. Time

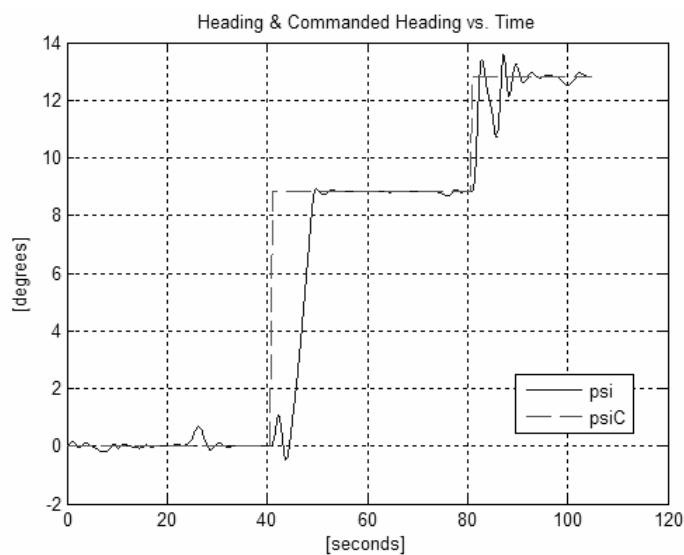


Figure 127. Missile Heading and Commanded Heading vs. Time

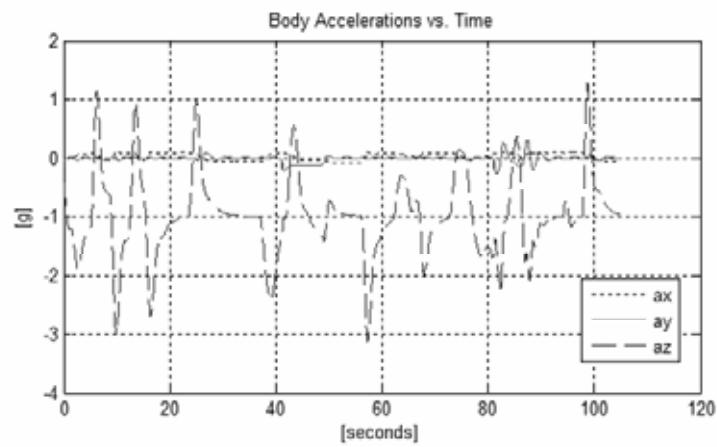


Figure 128. Body Accelerations vs. Time

From the simulation results of the system, controls applied for mid-course guidance phase can be seen easily. In Figure 113 to Figure 115 flight profile of the missile is given in detail. Terrain following guidance and heading corrections for the required waypoint can be seen from the results. Commanded height is given as 300 meters AGL and using acceleration controller, AGL height is kept between  $300 \pm 150$  meters which can be considered as an accepted range for the rough terrain considered.

Attitude rates are shown in Figure 116 and Figure 117. Heading commands are given at 41 and 81 seconds of the simulation. Since BTT control is applied, roll and yaw rates change rapidly at these instants as expected. On the other hand, rapid changes in pitch rate are due to terrain following controller where normal acceleration controller is used in order to follow the rapid changes in height channel. Attitudes can be explained in the same manner which is shown in Figure 118. Here, heading angle  $\varphi$  changes due to missile bank angle  $\phi$ , since BTT controller is used. Moreover, pitch angle  $\theta$  changes between  $\pm 10^\circ$  due to terrain following. Finally attitude errors are presented in Figure 119 which are consistent with the INS error model results. Attitude and attitude rates in the controllers are used from INS error model. Hence, the system worked as in the actual applications.

Missile total velocity and body velocity components are given in Figure 120 and Figure 121. Total velocity is kept constant within a range due to Mach control loop. Changes in lateral body velocity are due to wind and rapid changes in vertical body velocity are due to terrain following control. Angle of attack  $\alpha$  and side slip angle  $\beta$  are shown in Figure 122 and wind profile is given in Figure 123. Changes in  $\beta$  is due to wind profile where side wind components are considered for simulations. Finally Mach graph is presented in Figure 124 where Mach is controlled at 0.7 M within  $\pm 0.002$  M due to Mach control loop.

Changes in missile mass are given in Figure 125. Since Mach control is applied at constant altitude, constant mass flow rate, actually fuel flow rate is obtained as expected. Next, turbojet thrust is shown in Figure 126 where thrust changes between 750 – 2500 N in order to respond Mach control loop and terrain following controller. Then, missile heading versus commanded heading is given Figure 127. Here, missile response to heading commands can be seen well than the previous attitude plots. Heading corrections at 41 and 81 seconds of the simulation are applied considering the corrected missile position from TAN algorithms and waypoint position. It should be noted that INS updates from TAN algorithms are done at 40 seconds intervals. Hence, final position error decrease which will be discussed in the end of the chapter.

Finally, body accelerations are presented in Figure 128 where longitudinal and lateral accelerations are small as expected due to Mach and BTT controls. Vertical acceleration component changes up to 3 g's due to acceleration controller in order to achieve terrain following.

In the final section of the chapter, TAN results obtained from the simulations are presented. Here, original, updated INS and TAN algorithms' results are compared for both acquisition mode where initial position errors are huge for the first part of the simulation and for tracking mode. At the end of the chapter, final position errors are tabulated and discussed.

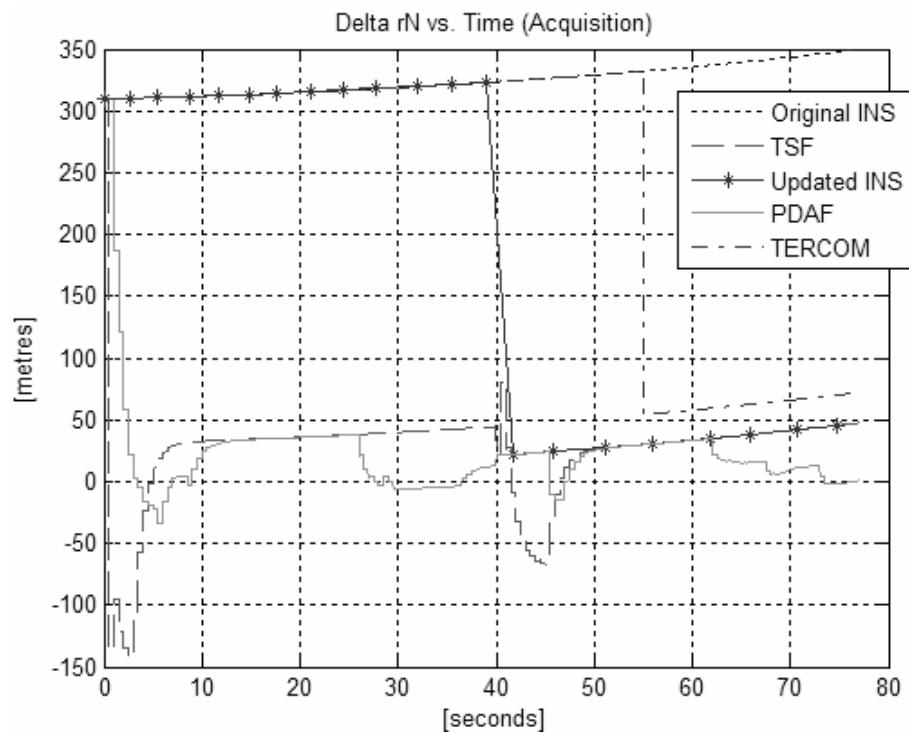


Figure 129. Northward Position Errors vs. Time (Acquisition Mode)

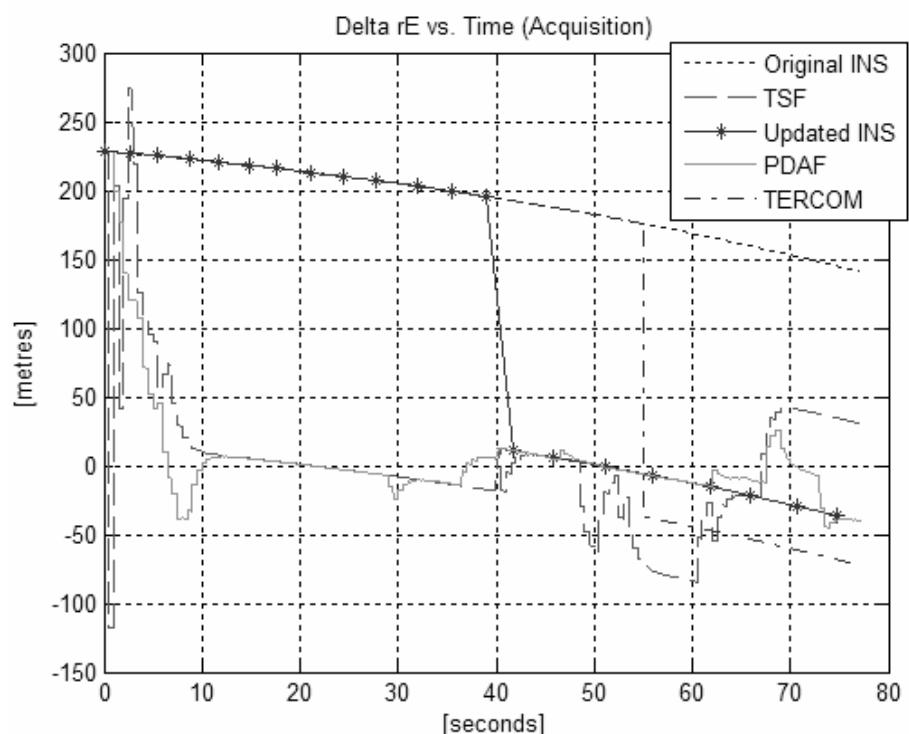


Figure 130. Eastward Position Errors vs. Time (Acquisition Mode)

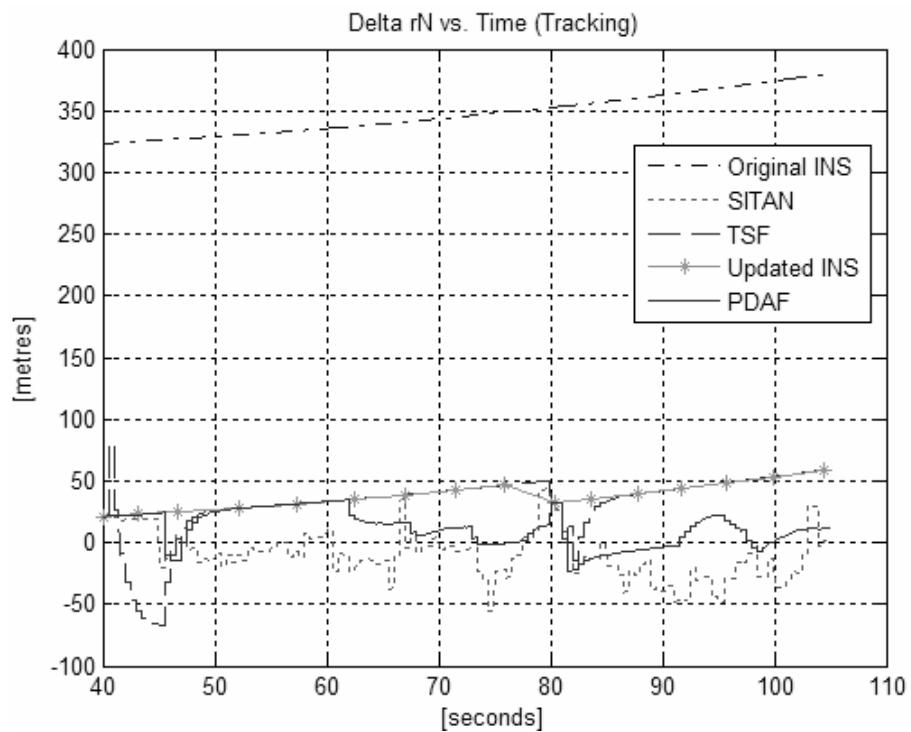


Figure 131. Northward Position Errors vs. Time (Tracking Mode)

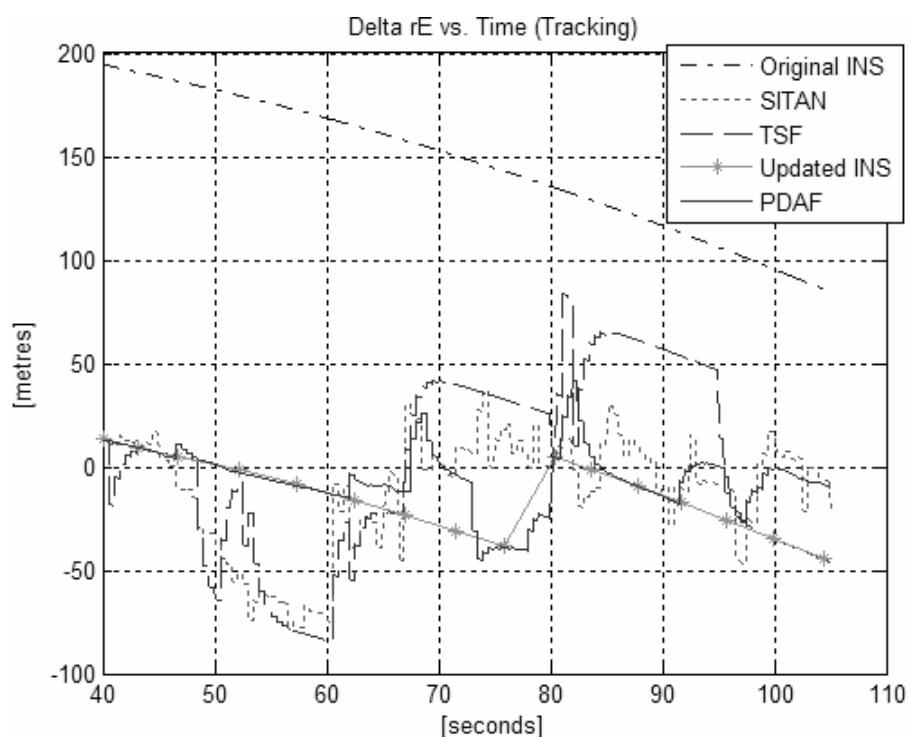


Figure 132. Eastward Position Errors vs. Time (Tracking Mode)

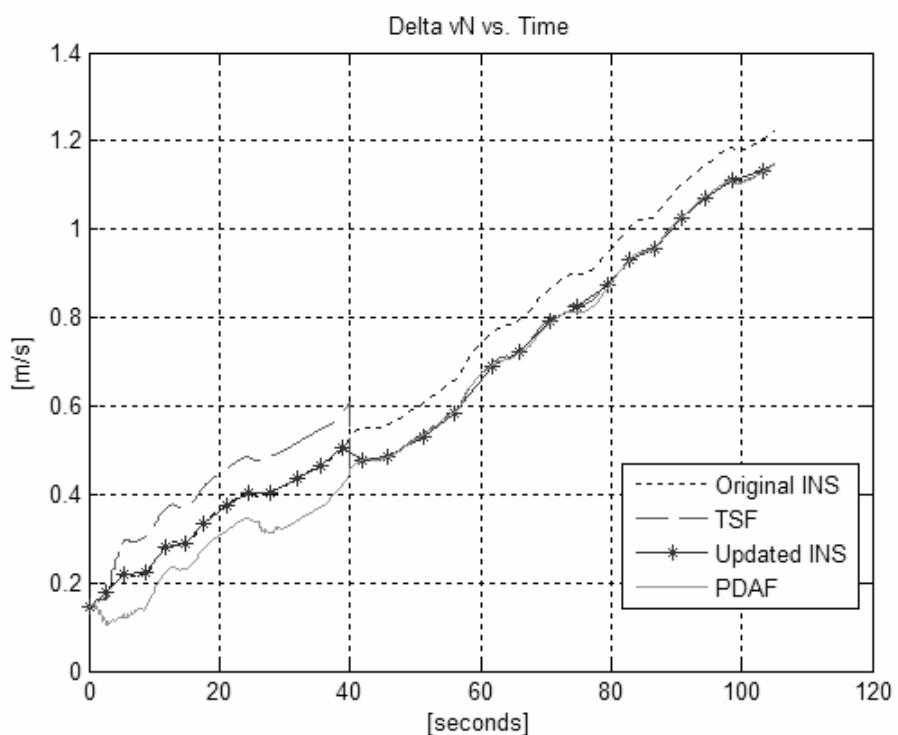


Figure 133. Northward Velocity Errors vs. Time

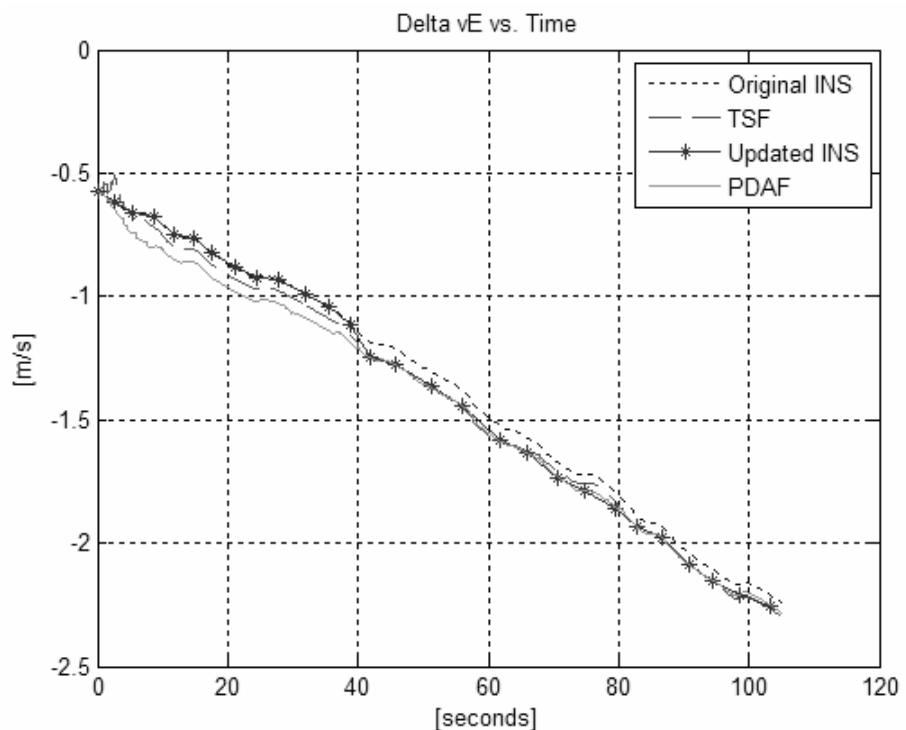


Figure 134. Eastward Velocity Errors vs. Time

TAN results are compared for both acquisition modes and tracking modes of operation. In the first 40 seconds of the operation, TAN algorithms are applied for acquisition mode with a larger DTED grid size of 11x11 given in Table 26. Here, large initial position errors are corrected using PDAF and TSF algorithms. Moreover, the results are compared with TERCOM results. In Figure 129, northward position errors and in Figure 130, eastward position errors are shown. At 40 seconds, PDAF and TSF corrections are done and INS is updated. On the other hand, TERCOM correction is done at 55 seconds. Both filters worked well in order to have errors less than 50 meters in both horizontal channels after a few seconds of operation. At the beginning of the TAN algorithms, past information is not used. Hence, filter results diverge for PDAF and TSF as it can be seen from the simulations.

Then, tracking mode position errors are shown in Figure 131 and Figure 132. Simulation results are given for 40 seconds to the end of the operation for tracking mode. In tracking mode, PDAF and TSF results are compared with SITAN results using a smaller DTED grid size of 5x5 given in Table 26. Here, INS is updated at 40 seconds intervals beginning from acquisition mode update at 40 seconds. As it can be seen from the results, real-time PDAF results are better than other algorithms. TSF results follow INS error model as expected due to the algorithm properties. In TSF, index correction for the navigation system is done as in TERCOM. Hence, TSF errors are sometimes within the neighborhood of the actual navigation solution. However, errors are limited within the grid size of DTED Level 1 (i.e. ~80 meters). Another interesting point is the divergence of TSF for a few seconds of operation at INS update times. This is due to lack of past measurement information. As older measurements come, TSF begins to follow INS errors. For PDAF, real-time corrections as in SITAN can be achieved since averages of the possible grid positions are considered throughout the operation.

In Table 27, RMS errors and final position errors from the required waypoint is tabulated for tracking mode simulations.

Table 27. Tracking Mode Position Errors

	RMS Error [m]	Final Position Error [m]
PDAF	24	16
TSF	60	75
SITAN	36	24
Updated INS	46	75
Original INS	379	390

As it can be seen from Table 27, PDAF results are better than other algorithms, especially than SITAN. Hence, it can be concluded that real-time PDAF improves navigation performance. On the other hand, if TSF divergence is avoided for the initial few seconds of update intervals, similar results with the updated INS can be achieved. However, it is shown from the results that TSF works better for acquisition mode than tracking mode. For the simulations, loosely coupled architecture is used where INS is updated at discrete time intervals. Actually, simulation results show that tightly coupled architecture where INS is updated continuously can be used for real-time PDAF and SITAN in order to have better updated INS results.

In the final part of the TAN simulations, horizontal position errors are presented in Figure 133 and Figure 134. TAN algorithms tend to correct northward velocity errors but not eastward velocity errors. Actually, with TAN algorithms velocity states can not be updated correctly, since measurements are only heights. However, if a detailed INS error model was used in TAN filters instead of simple error models derived in the previous chapter, better velocity errors could be obtained. In fact, TAN filter results can be improved by using better INS models in TAN models.

## **CHAPTER 5**

### **DISCUSSION AND CONCLUSION**

In the study, modern radar data association algorithms are implemented as new TAN algorithms which can be used with low-cost IMU's. After performing a thorough survey of the literature on mid-course navigation of cruise missiles, study on modern radar data association algorithms and their implementations to TAN are done. Finally, performances of the designed navigation systems with the implemented TAN algorithms are examined in detail with the help of the simulations performed.

In Chapter 1, theory about the study is given. Cruise missiles, cruise missile navigation performance and literature survey on TAN techniques are discussed. As it was discussed in detail in the chapter, the heart of TAN is the algorithms used which fall into two general algorithmic categories of batch and recursive algorithms. Therefore, main research of the study is concentrated on TAN algorithms.

From the literature survey, papers of Quintang, et al [40] and Dezert [43] gave inspiration for implementing modern data association algorithms to TAN in the Ph.D. study. Quintang, et al [40] propose a new TAN approach using PDAF to overcome irresolvable ambiguities in the correlation function used in TERCOM. The approach proposed is a batch algorithm which uses one of the modern radar

tracking algorithms. On the other hand, in the second paper, Dezert [43] proposes a new application of PDAF for improving the accuracy of autonomous strapdown INS. However, it is a real-time application of PDAF and relation with the former paper of Quintang, et al [40] can be obtained where batch implementation of PDAF is used. Therefore, it was thought whether PDAF could be used as a TAN algorithm for real-time applications.

TAN is a nonlinear estimation problem; since, terrain height information is used for navigation solution. Actually, TAN can be considered as a data association problem, especially for the acquisition operation mode where INS position errors are considerably large. From the literature survey as stated above, it has been thought that modern data association algorithms can be implemented for real-time TAN algorithms. Therefore, radar tracking, especially data association subject is investigated. At the end of Chapter 1, information about radar tracking techniques and possible implementations of radar data association algorithms to TAN is given.

In Chapter 2, major TAN methods are investigated. First, INS errors of the cruise missiles and need for TAN systems are discussed. Then, major TAN methods including TERCOM, SITAN and VATAN are presented in detail. Fundamentals of the major methods are discussed in this chapter with simulations in order to make comparisons for the implemented TAN algorithms in the Ph.D. study.

For the TERCOM process, several conclusions are achieved from the concept study and simulations performed. They are summarized as follows:

1. Correlation algorithm is simple but not smart. Many calculations should be performed in order to have a position fix and navigation solutions can be obtained for rough and unique terrains as expected.

2. It is thought that the algorithm was derived considering the capability of the computers of 1950's, performing only matrix calculations and simple mathematical operations.
3. Physical meaning of MAD and MSD processes is the minimization of the area difference between the measured and the reference areas along the route of the missile. Actually, TERCOM process is actually a Maximum Likelihood Estimator (MLE) which uses "Least Squares Estimation (LSE)" technique.
4. In the simulations, it was shown that MAD process shows better position fix than MSD process. For a terrain with small terrain height changes, MSD process neglects the small height difference terms and exaggerate the larger height difference terms. On the other hand, in MAD process absolute height difference terms are taken into account with same weights.
5. The critical parameter for best terrain correlation is sigma-Z value of the area concerned where standard deviation of the point-to-point changes in terrain elevation (i.e. the slope) are calculated instead of sigma-T value where standard deviation of height of the area is calculated. In other words, the slopes of the area concerned are more critical than the roughness of the area for correlation.
6. TERCOM process is independent of the target model where cruise missile is the target. Possible tracks for the missile are selected where tracks are the missile path formed by the terrain elevation file (DTED). Since, the target motion is not modeled; kinematical behavior of the system is not known.

For the SITAN process, conclusions achieved from the concept study and simulations performed is summarized as follows:

1. SITAN is a recursive TAN technique which uses EKF unlike TERCOM which is a batch process.
2. SITAN performance depends on the linearization of the terrain profiles since terrain slopes are required for the KF measurements. For large position errors, divergence can occur due to linearization errors in the EKF. In order to get rid of this, modified terrain linearization techniques and parallel KF structure are used.
3. SITAN improves position errors for rough and mountainous terrain types. However, due to slope determination process in SITAN, solutions have sometimes serious jumps for mountainous terrain type. This can be explained by the severe slope changes in the mountainous terrain modeling. Therefore, linearization of the terrain profiles is very critical especially for mountainous terrains in SITAN.
4. SITAN performance is better than both INS and terrain grids unlike TERCOM. In TERCOM, error can not be better than the terrain grid dimensions.
5. SITAN performs better for smaller position errors due to terrain linearization. Due to this fact, for large initial position errors TERCOM or SITAN with parallel KF structure must be used.

In the last part of Chapter 2, VATAN is investigated. VATAN uses VA which is a maximum a posteriori (MAP) estimator that estimates a sequence of system states from a sequence of observation values. VA is actually a dynamic programming technique for estimation which uses past information in data association problems. Therefore, it is thought that other data association algorithms can be used for TAN algorithms in the study.

In Chapter 3, implementation of target tracking algorithms to TAN is presented. First, general information about modern target tracking algorithms are given. Next, PDAF and TSF data association algorithms and their general implementations are investigated. Then, PDAF and TSF implementations to TAN are presented. At the end of the chapter, a simple simulation model is developed for the mid-course flight of the cruise missile. Finally, simulations are performed with the implemented TAN algorithms and the results are compared with the major TAN methods.

The advantages of the PDA and TS approach implemented for TAN solution can be summarized as follows:

1. Real-time TAN solution can be obtained with a single PDA filter or parallel TS filters.
2. PDA and TS filters can be used for both batch and recursive TAN solution. For batch solution, larger grid size is selected for navigation solution. For recursive solution, horizontal positions are calculated recursively in relatively small DTED grids.
3. Since past measurements are taken into account, smoothing of the measurements in the filter is achieved which decreases errors.

4. Since INS error model is used for navigation solution, application of the filters is simple and the filters are linear.
5. Batch size of the DTED area concerned can be changed. Both larger DTED areas for acquisition mode or smaller DTED areas for tracking modes can be selected using the same PDA filter.
6. TSF gives solutions for various tracks selected. Actually, all tracks converge to the same index of the DTED grid (i.e. solution grid). However, for smooth terrains, there exist more than one position solution index and the tracks can be investigated separately in order to give more than one but finite number of navigation solutions.

TSF approach is original when compared with other papers. However, PDA approach for TAN is found in the literature. The difference of the PDA algorithm developed from Qingtang, et al [40] is summarized as follows:

1. In the paper of Qingtang, et al [40], TAN using PDAF was investigated for the batch algorithm. The motion of the vehicle is not modeled.
2. In the paper of Qingtang, et al [40], performance of the TAN using PDA and TERCOM has been compared. It is stated that PDA was used in order to improve the performance of TAN compared to TERCOM.
3. In the Ph.D. study, real-time PDAF implementation is done. By using the error model of the INS used in the vehicle, system

dynamics is modeled. Using PDAF, error states of the system are estimated recursively.

4. In the Ph.D. study, PDAF equations are directly implemented for the TAN solution. Association probabilities obtained from height difference measurements for each element of the DTED grid concerned are used for position updates, considering the index of the DTED grid.

In the last part of Chapter 3, simulations are performed for both acquisition and tracking modes of operation considering a small period of time (i.e. 100 seconds for tracking mode) operation in order to visualize the performance of the implemented TAN algorithms. Simulations are performed for rough, smooth and mountainous terrain types. Moreover, effects of using different DTED types and DTED grid sizes are also investigated.

In tracking mode, it is seen that better results than SITAN are obtained for rough and mountainous terrain types. TSF and PDAF results are considerably good, since if navigation solution does not exist, the filters follow INS error model which is actually a desired feature. From the Monte Carlo simulations, position RMS errors of the TSF and PDAF algorithms become less than 50 meters for mountainous terrains; in other words, a decreased navigation error is obtained.

In acquisition mode, it is seen that similar results with TERCOM are obtained for rough terrain type. Critical point in the acquisition mode simulations is the percentage of false fix for acquisition mode. In TERCOM, since INS error model is not considered for correlation process, there is always a probability of false fix in the position solutions. In the Monte Carlo simulations performed, a few false

position fixes occurred for TERCOM for rough terrain type. On the other hand, with PDAF and TSF no false position fixes occurred.

Simulation results also show that TAN algorithms work with DTED Level 2 for acquisition and tracking modes. However, DTED Level 1 results seem to be better than DTED Level 2 results. Unfortunately, there were not sufficient DTED Level 2 maps for simulations in order to compare simulation results in detail. Actually, vehicle velocity directly influences TAN performance. For cruise missiles, DTED Level 1 maps are sufficient for mid-course flight navigation solution where INS position fixes less than 50 meters can be obtained. For faster vehicles like cruise missiles, rapid changes in the terrain profile as in DTED Level 2 decreases TAN performance. As a result of this, use of DTED Level 1 maps for TAN acquisition mode seems to perform better solutions.

At the end of Chapter 3, simulations are done with various DTED grid sizes for PDAF and TSF. Here same initial position errors are taken for simulations in tracking mode along rough terrain. It is seen that changing grid size for the same initial position errors for TSF slightly changes simulation results. Again errors are bounded and limited with the related grid solution. However, selecting larger grid sizes for PDAF solutions generally increase position errors. This is due to PDA procedure where weighted averages of the all grid points are taken into account for navigation solution. Therefore, it can be concluded that PDAF DTED grid size should be selected in accordance with the position errors.

In Chapter 4, case studies are performed for a cruise missile model with the help of the 6 DOF simulation tool developed. The simulation tool developed for the cruise missile is capable of performing full mid-course flight simulation of the cruise missile modeled. Actually, a generic simulation tool applicable to all air vehicles is considered except for guidance methods applied.

A variety of controllers are implemented for the mid-course flight of the cruise missile; Mach hold control, roll position control, heading angle control with bank-to-turn autopilot, yaw stability augmentation and altitude hold control with acceleration autopilot. All autopilots are derived by classical pole placement techniques summarized in Zipfel [79].

Then, simulations are performed with PDAF and TSF TAN models with actual flight conditions. Finally, simulation results are compared with major TAN algorithms considering other flight parameters of the cruise missile model. From the simulation results of the system, controls applied for mid-course guidance phase are clearly observed.

From the simulations performed in Chapter 4, better results are obtained for PDAF than other algorithms, especially than SITAN. Hence, it can be concluded that real-time PDAF improves navigation performance. However, simulation results also show that TSF works better for acquisition mode than tracking mode.

Several conclusions are achieved from the implemented PDAF and TSF algorithms from the simulations performed in Chapter 3 and Chapter 4. The advantages of the new algorithms proposed can be summarized as follows:

1. Real-time TAN solution can be obtained with a single PDA filter. Since past measurements are taken into account, by changing the buffer size of the measurements the filter, measurements are smoothed.
2. Real-time TAN solution can be obtained with a single TSF structure. However, TSF operations are more complex than SITAN. On the other hand, in TSF, more than one track is selected in order to determine navigation solution. Hence, probability of false fix decreases unlike in TERCOM.

3. Real-time TAN solution is obtained by considering horizontal position errors of DTED used in real-time PDA filter and TSF. Hence, horizontal position states are added to the Kalman filters used in PDAF and TSF.
4. Application of the filters is simple and the filters are linear, since INS error model is used.
5. Batch size of the DTED area concerned can be changed independent of the model used. Both larger DTED areas for acquisition mode or smaller DTED areas for tracking modes can be selected using the same filters.
6. Results of the filters are good for both recursive and batch algorithms. For tracking mode, position RMS error is less than 50 meters for DTED Level 1. Moreover, PDAF shows stable response. For smooth terrains where no navigation solution exists, PDAF follows the INS error model which is actually a desired feature.
7. TSF can be considered as a real-time TERCOM process for large position errors, i.e. large DTED batch size. Possibility of false position fixes decrease with TSF when compared with TERCOM. On the other hand, for small position errors, decreasing the weighting factor of the past measurements for TSF, better real-time solutions can be obtained. However, real-time results of TSF follows INS error model unlike PDAF where position errors decrease much.

### **Contributions of the Study:**

In order to declare the contributions of the study, disadvantages of the present TAN algorithms should be overviewed again. Several disadvantages of the TAN algorithms can be summarized as follows:

1. TAN requires terrain information for real-time navigation solution and the dynamics of the system is highly nonlinear which need considerable calculation work.
2. Real-time application for the TAN solution is generally impractical for high velocity vehicles like cruise missiles due nonlinear characteristics of the system.
3. In SITAN, terrain linearization and terrain slopes are required in order to apply extended Kalman filter equations which are actually critical stages for TAN solution.
4. TERCOM is a batch process and it is independent of the target model where cruise missile is the target. Since, the target motion is not modeled; kinematical behavior of the system is not known and possibility of false position fixes increase especially for terrains with similar height profiles.

Then, the contributions of the study can be summarized as follows:

1. Modern radar data association algorithms are implemented as new TAN algorithms which can be used with low-cost IMU's.

2. Acquisition mode performance of the TAN algorithms is improved when compared with TERCOM. Probability of false fix decreases with the implementation of PDAF and TSF for TAN.
3. Tracking mode performance of the TAN algorithms is improved when compared with SITAN especially with the implementation of PDAF.
4. Application of the filters is simple and the filters are linear, since INS error model is used for position updates.
5. No linearization for terrain is required for the implemented algorithms. DTED files can be used directly without any prior work for operation.
6. Implemented algorithms can be applied to existing systems with the use of the new micro-processors with relatively low costs.

### **Future Work:**

As the future work, implemented TAN algorithms can be used for tightly coupled integration architecture. Actually, simulation results show that tightly coupled architecture where INS is updated continuously can be used for real-time PDAF and SITAN in order to have better updated INS results.

Another future work can be the improvement of the implemented filters. Filter system and measurement models are constructed considering simple INS models with position and velocity errors only. If a detailed INS error model was used in the implemented filters, better results could be obtained than the simulation

results in Chapter 4. Finally, a hybrid filter algorithm can be implemented which uses both PDAF and TSF in order to have superior results than stand alone PDAF and TSF algorithms.

## REFERENCES

- [1]. Wikipedia, the free Encyclopedia, “*Cruise Missile*”,  
<http://www.wikipedia.org/w/wiki.phtml?search=cruise+missile&go=Go>,  
Last accessed in May 2003.
- [2]. Siouris, G., M., *Missile Guidance and Control Systems*, Springer-Verlag  
New York, Inc., pp. 521-522, 2004.
- [3]. Metzger, J., Wendel, J., Trommer, G., F., “*Hybrid Terrain Referenced  
Navigation System using a Bank of Kalman Filters and a Comparison  
Technique*”, AIAA, Guidance, Navigation, and Control Conference and  
Exhibit, Providence, Rhode Island, August 2004.
- [4]. Federation of American Scientists, “*Cruise Missiles*”,  
<http://www.fas.org/nuke/intro/cm>, Last accessed in May 2003.
- [5]. Federation of American Scientists, “*BGM-109 Tomahawk*”,  
<http://www.fas.org/man/dod-101/sys/smart/bgm-109.htm>, Last accessed in  
May 2003.

- [6]. Federation of American Scientists, “*AGM-86 Air-Launched Cruise Missile [ALCM]*”, [http://www.fas.org/nuke/guide/usa/bomber/\\_alcm.htm](http://www.fas.org/nuke/guide/usa/bomber/_alcm.htm), Last accessed in May 2003.
- [7]. CDISS, the Centre for Defence and International Security Studies, “*Cruise Missiles; Key Technologies: An Overview*”, <http://www.cdiiss.org/tabctechs.htm>, Last accessed in May 2003.
- [8]. DARPA Special Projects Office, “*Low Cost Cruise Missile Defense (LCCMD)*”, <http://www.darpa.mil/spo/Programs/lowcostcruisemissile-defense.htm>, Last accessed in May 2003.
- [9]. DARPA Special Projects Office, “*LCCMD Side Show*”, [http://www.darpa.mil/spo/Programs/LCCMD\\_slides/LCCMD\\_slide\\_show.pdf](http://www.darpa.mil/spo/Programs/LCCMD_slides/LCCMD_slide_show.pdf), Last accessed in May 2003.
- [10]. Aardvark, “*A DIY Cruise Missile*”, <http://www.interestingprojects.com/cruisemissile/>, Last accessed in May 2003.
- [11]. Johnson, N., Tang, W., Howell, G., “*Terrain Aided Navigation Using Maximum A Posteriori Estimation*”, IEEE, Position Location and Navigation Symposium, 1990.

- [12]. Hostetler, L., D., Andreas, R., D., “*Nonlinear Kalman Filtering Techniques for Terrain-Aided Navigation*”, IEEE Transactions on Automatic Control, Vol. AC-28, No. 3, March 1983.
- [13]. Boozer, D., D., “*Terrain Referenced Navigation*”, AGARD-AG-331, *Aerospace Navigation Systems*, pp. 152-157, June 1995.
- [14]. The MathWorks Inc., “*MATLAB Help, The Language of Technical Computing*”, Version 7.1.0.246 (R14SP3), 1984-2005.
- [15]. MIL-PRF-89020A, “*Performance Specification Digital Terrain Elevation Data (DTED)*”, US Department of Defense, 1996.
- [16]. “*Sayısal Harita ve Coğrafî Bilgi Sistemi Kurs Notları*”, T.C. M.S.B. Harita Genel Komutanlığı, Tasnif Dışı, TÜBİTAK-SAGE Kütüphanesi, CPP733, 2003, *in Turkish*.
- [17]. Henley, A., J., “*Terrain Aided Navigation – Current Status, Techniques for Flat Terrain and Reference Data Requirements*”, IEEE, Position Location and Navigation Symposium, 1990.
- [18]. Nielson, J., T., “*CALCM – The Untold Story of the Weapon Used to Start the Gulf War*”, IEEE Aerospace and Electronics Systems Magazine, July 1994.

- [19]. Hicks, S., “*Advance Cruise Missile Guidance System Description*”, Proceedings of the IEEE Aerospace and Electronics Conference, 1993.
- [20]. Bennett, P., J., “*The Use of Digital Map Data for Airborne Operations*”, IEE Colloquium on Serious Low Flying, Ref. No. 1998/223.
- [21]. Chen, Z., Yu, P., “*Model Study for Terrain Aided Navigation Systems*”, Proceedings of the IEEE International Symposium on Industrial Electronics, 1992.
- [22]. Wang, W., Chen, Z., “*Error Model Identification on Digital Map of TAN System Based on EKF*”, Proceedings of the IEEE International Conference on Industrial Technology, 1994.
- [23]. Yu, P., Chen, Z., Hung, J., C., “*Performance Evaluation of Six Terrain Stochastic Linearization Techniques for TAN*”, Proceedings of the IEEE National Aerospace and Electronics Conference, 1991.
- [24]. Paris, S., Le Cadre, J., “*Planification for Terrain-Aided Navigation*”, IEEE, Proceedings of the Fifth International Conference on Information Fusion, 2002.
- [25]. McFarland, M., B., Zachery, R., A., Taylor, B., K., “*Motion Planning for Reduced Observability of Autonomous Aerial Vehicles*”, Proceedings of the IEEE International Conference on Control Applications, 1999.

- [26]. Bar-Gill, A., Ben-Ezra, P., Bar-Itzhack, I., Y., “*Improvement of Terrain-Aided Navigation via Trajectory Optimization*”, IEEE, Transactions on Control Systems Technology, Vol. 2, No. 4, December 1994.
- [27]. Li, D., Zhou, D., Hu, Z., Hu, H., “*Optimal Preview Control Applied to Terrain Following Flight*”, Proceedings of the 40th IEEE Conference on Decision and Control, 2001.
- [28]. Baird, C., A., Snyder, F., B., “*Terrain-Aided Altitude Computations on the AFTI/F-16*”, IEEE, Position Location and Navigation Symposium, 1990.
- [29]. Hollowell, J., “*Heli/SITAN: A Terrain Referenced Navigation Algorithm for Helicopters*”, IEEE, Position Location and Navigation Symposium, 1990.
- [30]. Nordlund, P., Gustafsson, F., “*Recursive Estimation of Three-Dimensional Aircraft Position Using Terrain-Aided Positioning*”, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002.
- [31]. Bergman, N., Ljung, L., Gustafsson, F., “*Point-Mass Filter and Cramer-Rao Bound for Terrain-Aided Navigation*”, Proceedings of the 36th IEEE Conference on Decision and Control, 1997.
- [32]. Newman, P., Durrant-Whyte, H., “*Using Sonar in Terrain-Aided Underwater Navigation*”, IEEE International Conference on Robotics and Automation, 1998.

- [33]. Madhavan, R., Durrant-Whyte, H., Dissanayake, G., “*Natural Landmark-Based Autonomous Navigation Using Curvature Scale Space*”, IEEE International Conference on Robotics and Automation, 2002.
- [34]. Bruder, S., B., H., Wedeward, K., “*Terrain Aided INS Robot Navigation: A Deferred Decision Making Approach*”, IEEE, 42nd Midwest Symposium on Circuits and Systems, 1999.
- [35]. Morisue, F., Ikeda, K., “*Evaluation of Map-Matching Techniques*”, IEEE Vehicle Navigation and Information Systems Conference, 1989.
- [36]. McLellan, J., F., Schleppe, J., “*GPS/Barometry Height-Aided Positioning System*”, IEEE, Position Location and Navigation Symposium, 1994.
- [37]. Siouris, G., M., *Missile Guidance and Control Systems*, Springer-Verlag New York, Inc., pp. 551-576, 2004.
- [38]. Erhui, W., Guohua, G., Lincheng, S., Wensn, C., “*A Probability-Based Terrain-Aided Navigation Approach and Its Relative Terrain Navigability Analysis*”, Proceedings of the IEEE International Conference on Industrial Technology, 1996.
- [39]. Zhou, H., Zhang, C., “*Terrain Aided Navigation Based on Computer Vision*”, Proceedings of the 4th World Congress on Intelligent Control and Automation, 2002.

- [40]. Qingtang, F., Lincheng, S., Wenseng, C., “*Terrain Aided Navigation Using PDAF*”, Proceedings of the 2003 IEEE, International Conference on Robotics, Intelligent Systems and Signal Processing, China, 2003.
- [41]. Pei, Y., Chen, Z., Hung, J., C., “*BITAN-II: An Improved Terrain Aided Navigation Algorithm*”, Proceedings of the 1996 IEEE IECON 22 Industrial Electronics, Control, and Instrumentation, 1996.
- [42]. Enns, R., Morrell, D., “*Terrain-Aided Navigation Using the Viterbi Algorithm*”, AIAA, Journal of Guidance, Control, and Dynamics, Vol. 18, No. 6, November-December 1995.
- [43]. Dezert, J., “*Improvement of Strapdown Inertial Navigation using PDAF*”, IEEE Transactions on Aerospace and Electronic Systems, Vol. 35, No. 3, July 1999.
- [44]. Hongbo, X., Yan, T., JianZhong, S., Jinwen, T., Jian, L., “*Terrain Matching Based on Imaging Laser Radar*”, IEEE 6th International Conference on Signal Processing, 2002.
- [45]. Bevington, J., E., Marttila, C., A., “*Precision Aided Inertial Navigation Using SAR and Digital Map Data*”, IEEE, Position Location and Navigation Symposium, 1990.

- [46]. Chan, L., C., Snyder, F., B., “*System for Correlation and Recognition of Terrain Elevation*”, United States Patent, Patent No: 4,584,646, Date of Patent: Apr. 22, 1986.
- [47]. Baird, C., A., “*Map-Aided Navigation System Employing TERCOM-SITAN Signal Processing*”, United States Patent, Patent No: 4,829,304, Date of Patent: May 9, 1989.
- [48]. Lerche, H., “*Navigation of Aircraft by Correlation*”, United States Patent, Patent No: 4,910,674, Date of Patent: Mar. 20, 1990.
- [49]. Raymer, K., A., McGuffin, J., T., “*Terrain Referenced Navigation – Schuler Cycle Error Reduction Method and Apparatus*”, United States Patent, Patent No: 5,450,345, Date of Patent: Sep. 12, 1995.
- [50]. Goebel, R., H., Fogle, D., A., Torretta, D., C., Panagos, P., Heffern, P., A., “*Terrain Correlation System*”, United States Patent, Patent No: 6,218,980, Date of Patent: Apr. 17, 2001.
- [51]. Uhlmann, J., “*Introduction to the Algorithmics of Data Association in Multiple-Target Tracking*”, *Handbook of Multisensor Data Fusion*, CRC Press LLC, 2001.
- [52]. Kirubarajan, T., Bar-Shalom, Y., “*Target Tracking Using Probabilistic Data Association Based Techniques with Applications to Sonar, Radar, and*

*EO Sensors*”, *Handbook of Multisensor Data Fusion*, CRC Press LLC, 2001.

- [53]. Stieler, B., “*Inertial Navigation*”, AGARD-AG-331, *Aerospace Navigation Systems*, p. 45, June 1995.
- [54] Rogers, R., M., *Applied Mathematics in Integrated Navigation Systems*, AIAA Education Series, ISBN: 1-56347-397-6, Reston, VA, pp. 273-282, 2000.
- [55] Bar-Itzhack, I., Y., Berman, N., “*Control Theoretic Approach to Inertial Navigation Systems*”, AIAA Journal of Guidance, Vol. 13, pp. 237-245, May-June 1988.
- [56] Scherzinger, B., M., Reid, D., B., “*Modified Strapdown Inertial Navigator Error Models*”, Proceedings of the 1994 IEEE PLANS, Las Vegas NY, April 1994.
- [57]. Phillips, R., E., Schmidt, G., T., “*GPS/INS Integration*”, AGARD-LS-207, *System Implications and Innovative Applications of Satellite Navigation*, p. 9-12, June 1996.
- [58]. The MathWorks Inc., “*Simulink, Dynamic System Simulation for MATLAB*”, Version 6.3 (R14SP3), 1990-2005.

- [59] Kayton, M., Fried, W. R., ed., *Avionics Navigation Systems*, John Wiley & Sons, pp. 317-318, 1969.
- [60]. The MathWorks Inc., “*MATLAB, The Language of Technical Computing*”, Version 7.1.0.246 (R14SP3), 1984-2005.
- [61]. Newman, D., “*OziExplorer, GPS Mapping Software*”, Version 3.95.3.g2, D&L Software Pty Ltd., Australia, 2004.
- [62]. Newman, D., “*OziExplorer3D*”, Version 1.07b, D&L Software Pty Ltd., Australia, 2004.
- [63] Gelb, A., ed., *Applied Optimal Estimation*, The M.I.T. Press, pp. 182-189, Eleventh Printing, 1989.
- [64]. Chong, C., Garren, D., Grayson, T., P., “*Ground Target Tracking – A Historical Perspective*”, IEEE, Aerospace Conference Proceedings, 2000.
- [65]. Durrant-Whyte, H., “*Introduction to Estimation and Data Fusion*”, Lecture Notes, Australian Centre for Field Robotics, The University of Sydney, 2003.
- [66]. Chang, K., “*Syst680: Principles of C3I #4*”, Lecture Notes, Dept. of Systems Engineering, George Mason University, 2004.

- [67]. Maksarov, D., Durrant-Whyte, H., “*Mobile Vehicle Navigation in Unknown Environments: A Multiple Hypothesis Approach*”, IEE Proc.-Control Theory Appl., Vol. 142, No. 4, July 1995.
- [68]. Stone, L., D., “*A Bayesian Approach to Multiple-Target Tracking*”, *Handbook of Multisensor Data Fusion*, CRC Press LLC, 2001.
- [69] Blackman, S., Popoli, R., *Design and Analysis of Modern Tracking Systems*, Artech House, ISBN No: 1-58053-006-0, pp. 360-402, 1999.
- [70]. Reid, D., B., “*An Algorithm for Tracking Multiple Targets*”, IEEE Transactions on Automatic Control, Vol. AC-24, No. 6, Dec. 1979.
- [71]. Kurien, T., “*Issues in the Design of Practical Multitarget Tracking Algorithms*”, *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom (Ed.), Norwood, MA: Artech House, 1990, Chap. 3.
- [72]. Smith, P., B., Buechler, G., “*A Branching Algorithm for Discriminating and Tracking Multiple Objects*”, IEEE Transactions on Automatic Control, pp. 101-104, Feb. 1975.
- [73]. Stieler, B., “*Inertial Navigation*”, AGARD-AG-331, *Aerospace Navigation Systems*, p. 100, June 1995.

- [74]. Kumar, M., “*Coordinate Frames*”, AGARD-AG-331, *Aerospace Navigation Systems*, pp. 7-41, June 1995.
- [75]. The MathWorks Inc., “*MATLAB, Mapping Toolbox*”, Version 2.2 (R14SP3), 1984-2005.
- [76]. The MathWorks Inc., “*Simulink, Signal Processing Blockset*”, Version 6.2 (R14SP3), 1990-2005.
- [77]. “*Microdem/ TerraBase II Software*”, PETMAR Trilobite Breeding Ranch, Version 14.45.15.2, March 2003.
- [78]. Özgören, M. K., “*ME 502 Advanced Dynamics Lecture Notes*”, Middle East Technical University, 1997. (Unpublished)
- [79] Zipfel, P., H., *Modeling and Simulation of Aerospace Vehicle Dynamics*, AIAA Education Series, ISBN: 1-56347-456-5, Reston, VA, 2000.
- [80]. McDonnell Douglas Astronautics Company, “*The USAF Stability and Control Digital DATCOM, Volume I, Users Manual*”, Technical Report, AFFDL-TR-79-3032, Vol. I, St. Louis Division, St. Louis, Missouri, April 1979.
- [81]. “*International Gravity Formula*”, [http://geophysics.ou.edu/solid\\_earth/notes/potential/igf.htm](http://geophysics.ou.edu/solid_earth/notes/potential/igf.htm), Last accessed in March 2006.

- [82]. Lewantowicz, Z., H., Paschall, R., N., “*Deep Integration of GPS, INS, SAR, and Other Sensor Information*”, AGARD-AG-331, *Aerospace Navigation Systems*, p. 239, June 1995.
- [83]. Liang, D., F., “*An Overview of a Generic Multi-Sensor Integrated Navigation System Design*”, AGARD-AG-331, *Aerospace Navigation Systems*, pp. 219-220, June 1995.

## APPENDIX

### **PROBABILISTIC DATA ASSOCIATION EQUATIONS [52]**

The PDA algorithm calculates in real-time the probability that each validated measurement is attributable to the target of interest. This probabilistic (Bayesian) information is used in a tracking filter, the PDA filter (PDAF), which accounts for the measurement origin uncertainty.

#### **Past Measurement Information:**

The PDAF uses a decomposition of the estimation with respect to the origin of each element of the latest set of validated measurements, denoted as:

$$Z(k) = \{z_i(k)\}_{i=1}^{m(k)} \quad (\text{A.1})$$

where;

$z_i(k)$ :  $i$ 'th validated measurement,

$m(k)$ : Number of measurements in the validation region at time  $k$ .

The cumulative set (sequence) of measurements is:

$$Z^k = \{Z(j)\}_{j=1}^k \quad (\text{A.2})$$

### **Measurement Validation:**

From the Gaussian assumption, the validation region is the elliptical region:

$$V(k, \gamma) = \left\{ Z : [z - \hat{z}(k | k-1)]^T \cdot S(k)^{-1} \cdot [z - \hat{z}(k | k-1)] \leq \gamma \right\} \quad (\text{A.3})$$

where;

$\gamma$ : Gate threshold,

$$S(k) = H(k) \cdot P(k | k-1) \cdot H(k)^T + R(k) \quad (\text{A.4})$$

$S(k)$ : Covariance of the innovation corresponding to the true measurement.

The volume of the validation region given in equation (A.3) is:

$$V(k) = c_{n_z} \cdot |\gamma^{n_z} \cdot S(k)|^{1/2} = c_{n_z} \cdot \gamma^{n_z} \cdot |S(k)|^{1/2} \quad (\text{A.5})$$

where the coefficient  $c_{n_z}$  depends on the dimension of the measurement (it is the volume of the  $n_z$ -dimensional unit hyper sphere:  $c_1 = 2$ ,  $c_2 = \pi$ ,  $c_3 = 4\pi/3$ , etc.)

### The State Estimation:

In view of the assumptions listed, the association events;

$$\theta_i(k) = \begin{cases} \{z_i(k) \text{ is the target originated measurement}\} & i = 1, \dots, m(k) \\ \{\text{none of the measurements is target originated}\} & i = 0 \end{cases} \quad (\text{A.6})$$

are mutually exclusive and exhaustive for  $m(k) \geq 1$ .

Using the total probability theorem with regard to the above events, the conditional mean of the state at time  $k$  can be written as;

$$\hat{x}(k | k) = E[x(k) | Z^k]$$

$$\hat{x}(k | k) = \sum_{i=0}^{m(k)} E[x(k) | \theta_i(k), Z^k] \cdot P\{\theta_i(k) | Z^k\}$$

$$\hat{x}(k | k) = \sum_{i=0}^{m(k)} \hat{x}_i(k | k) \cdot \beta_i(k) \quad (\text{A.7})$$

where,  $\hat{x}_i(k | k)$  is the updated state conditioned on the event that the  $i$ 'th validated measurement is correct, and;

$$\beta_i(k) = P\{\theta_i(k) | Z^k\} \quad (\text{A.8})$$

is the conditional probability of this event; the association probability, obtained from the PDA procedure presented in the next subsection.

The estimate conditioned on measurement  $i$  being correct is;

$$\hat{x}_i(k | k) = \hat{x}(k | k-1) + K(k) \cdot v_i(k) \quad i = 1, \dots, m(k) \quad (\text{A.9})$$

where the corresponding innovation is;

$$v_i(k) = z_i(k) - \hat{z}(k | k-1) \quad (\text{A.10})$$

The gain  $K(k)$  is the same as in the standard Kalman filter;

$$K(k) = P(k | k-1) \cdot H(k)^T \cdot S(k)^{-1} \quad (\text{A.11})$$

since, conditioned on  $\theta_i(k)$ , there is no measurement origin uncertainty.

For  $i = 0$  (i.e. if none of the measurements is correct) or  $m(k) = 0$  (i.e. there is no validated measurement);

$$\hat{x}_0(k | k) = \hat{x}(k | k-1) \quad (\text{A.12})$$

### **The State and Covariance Update:**

Combining equations (A.9) and (A.12) into equation (A.7) yields the state update equation of the PDAF;

$$\hat{x}(k | k) = \hat{x}(k | k-1) + K(k) \cdot v(k) \quad (\text{A.13})$$

where the combined innovation is;

$$v(k) = \sum_{i=0}^{m(k)} \beta_i(k) \cdot v_i(k) \quad (\text{A.14})$$

The covariance associated with the updated state is;

$$P(k | k) = \beta_0(k) \cdot P(k | k-1) + [1 - \beta_0(k)] \cdot P^c(k | k) + \tilde{P}(k) \quad (\text{A.15})$$

where the covariance of the state updated with the correct measurement is;

$$P^c(k | k) = P(k | k-1) - K(k) \cdot S(k) \cdot K(k)^T \quad (\text{A.16})$$

and the spread of the innovations term (similar to the spread of the means term in a mixture) is;

$$\tilde{P}(k) = K(k) \cdot \left[ \sum_{i=0}^{m(k)} \beta_i(k) \cdot v_i(k) \cdot v_i(k)^T - v(k) \cdot v(k)^T \right] \cdot K(k)^T \quad (\text{A.17})$$

### **The Prediction Equations:**

The prediction of the state and measurement to  $k+1$  is done as in the standard filter, i.e.,

$$\hat{x}(k+1 | k) = \Phi(k) \cdot \hat{x}(k | k) \quad (\text{A.18})$$

$$\hat{z}(k+1 | k) = H(k+1) \cdot \hat{x}(k+1 | k) \quad (\text{A.19})$$

The covariance of the predicted state is, similarly;

$$P(k+1|k) = \Phi(k) \cdot P(k|k) \cdot \Phi(k)^T + Q(k) \quad (\text{A.20})$$

where  $P(k|k)$  is given by equation (A.15).

The innovation covariance (for the correct measurement) is, again, as in the standard filter;

$$S(k+1) = H(k+1) \cdot P(k+1|k) \cdot H(k+1)^T + R(k+1) \quad (\text{A.21})$$

### **The Probabilistic Data Association:**

To evaluate the association probabilities, the conditioning is broken down into the past data  $Z^{k-1}$  and the latest data  $Z(k)$ . A probabilistic inference can be made on both the number of measurements in the validation region (from the clutter density, if known) and on their location, expressed as;

$$\beta_i(k) = P\{\theta_i(k) | Z^k\} = P\{\theta_i(k) | Z(k), m(k), Z^{k-1}\} \quad (\text{A.22})$$

Using Bayes' formula, the above is rewritten as;

$$\begin{aligned} \beta_i(k) &= \frac{1}{c} \cdot p[Z(k) | \theta_i(k), m(k), Z^{k-1}] \cdot p\{\theta_i(k) | m(k), Z^{k-1}\} \\ &\quad i = 0, \dots, m(k) \end{aligned} \quad (\text{A.23})$$

The joint density of the validated measurements conditioned on  $\theta_i(k), i \neq 0$ , is the product of;

- The (assumed) Gaussian PDF of the correct (target-originated) measurements;
- The PDF of the incorrect measurements, which are assumed to be uniform in the validation region whose volume  $V(k)$  is given in equation (A.5).

The PDF of the correct measurement (with the  $P_G$  factor that accounts for restricting the normal density to the validation gate) is;

$$\begin{aligned} p[z_i(k) | \theta_i(k), m(k), Z^{k-1}] &= p_G^{-1} \cdot N[z_i(k) | z(k | k-1), S(k)] \\ &= p_G^{-1} \cdot N[v_i(k) | 0, S(k)] \end{aligned} \quad (\text{A.24})$$

The PDF from equation (A.23) is then;

$$p[Z(k) | \theta_i(k), m(k), Z^{k-1}] = \left\{ V(k)^{-m(k)+1} \cdot P_G \cdot N[\dots] \right\} \quad (\text{A.25})$$

The probabilities of the association events conditioned only on the number of validated measurements are;

$$p[Z(k) | \theta_i(k), m(k), Z^{k-1}] = \begin{cases} V(k)^{-m(k)+1} \cdot P_G \cdot N[v_i(k) | 0, S(k)] & i = 1, \dots, m(k) \\ V(k)^{-m(k)} & i = 0 \end{cases} \quad (\text{A.26})$$

where  $\mu_F(m)$  is the probability mass function (PMF) of the number of false measurements (false alarms or clutter) in the validation region.

Two models can be used for the PMF  $\mu_F(m)$  in a volume of interest  $V$ :

1. A Poisson model with a certain spatial density  $\lambda$ ;

$$\mu_F(m) = e^{-\lambda \cdot V} \cdot \frac{(\lambda \cdot V)^m}{m!} \quad (\text{A.27})$$

2. A diffuse prior model;

$$\mu_F(m) = \mu_F(m-1) = \delta \quad (\text{A.28})$$

where the constant  $\delta$  is irrelevant since it cancels out.

Using the (parametric) Poisson model in equation (A.26) yields;

$$\gamma_i[m(k)] = \begin{cases} P_D \cdot P_G \cdot [P_D \cdot P_G \cdot m(k) + (1 - P_D \cdot P_G) \cdot \lambda \cdot V(k)]^i & i = 1, \dots, m(k) \\ (1 - P_D \cdot P_G) \cdot \lambda \cdot V(k) \cdot [P_D \cdot P_G \cdot m(k) + (1 - P_D \cdot P_G) \cdot \lambda \cdot V(k)]^i & i = 0 \end{cases} \quad (\text{A.29})$$

The (nonparametric) diffuse prior equation A.28 yields;

$$\gamma_i[m(k)] = \begin{cases} \frac{1}{m(k)} \cdot P_D \cdot P_G & i = 1, \dots, m(k) \\ (1 - P_D \cdot P_G) & i = 0 \end{cases} \quad (\text{A.30})$$

The nonparametric model in equation (A.30) can be obtained from equation A.29 by setting;

$$\lambda = \frac{m(k)}{V(k)} \quad (\text{A.31})$$

i.e., replacing the Poisson parameter with the sample spatial density of the validated measurements. The volume  $V(k)$  of the elliptical (i.e., Gaussian-based) validation region is given in equation (A.5).

### **The Parametric PDA:**

Using equations (A.29) and (A.25) with the explicit expression of the Gaussian PDF in equation (A.23) yields, after some cancellations, the final equations of the parametric PDA with the Poisson clutter model;

$$\beta_i(k) = \begin{cases} \frac{e_i}{\frac{m(k)}{b + \sum_{j=1}^{m(k)} e_j}} & i = 1, \dots, m(k) \\ \frac{b}{b + \sum_{j=1}^{m(k)} e_j} & i = 0 \end{cases} \quad (\text{A.32})$$

where;

$$e_i = e^{-\frac{1}{2} v_i(k)^T \cdot S(k)^{-1} \cdot v_i(k)} \quad (\text{A.33})$$

$$b = \lambda \cdot |2\pi \cdot S(k)|^{1/2} \cdot \frac{1 - P_D \cdot P_G}{P_D} \quad (\text{A.34})$$

The last expression above can be rewritten as;

$$b = \left( \frac{2\pi}{\lambda} \right)^{\frac{n_z}{2}} \cdot \lambda \cdot V(k) \cdot c_{n_z}^{-1} \cdot \frac{1 - P_D \cdot P_G}{P_D} \quad (\text{A.35})$$

where;

$P_D$ : Probability of detection of a target originated measurement,

$P_G$ : Probability of measurements in the gate.

# **CURRICULUM VITAE**

## **PERSONAL INFORMATION:**

Surname, Name: EKÜTEKİN, Vedat

Nationality: Turkish (T.C.)

Date and Place of Birth: 15 May 1972, Eskişehir / TÜRKİYE

Marital Status: Married

Work Address: TÜBİTAK-SAGE  
(The Scientific & Technological Research Council of  
Türkiye –  
Defense Industries Research & Development Institute)  
P.K. 16, 06261 Mamak – ANKARA / TÜRKİYE  
Tel: +90 (312) 210 1374 / 228  
Fax: +90 (312) 210 1392  
E-mail: [veku@sage.tubitak.gov.tr](mailto:veku@sage.tubitak.gov.tr)

## **EDUCATION:**

<b>Degree</b>	<b>Institution</b>	<b>Year of Graduation</b>
MS	METU Mechanical Engineering	2000
BS	METU Mechanical Engineering	1994
High School	Ankara Fen Lisesi	1990

**WORK EXPERIENCE:**

<b>Year</b>	<b>Place</b>	<b>Enrollment</b>
2006 June – Present	TÜBİTAK-SAGE, Ankara Navigation Division	Chief Senior Researcher
1997 March – 2006 June	TÜBİTAK-SAGE, Ankara Flight Mechanics Division	Researcher Senior Researcher Chief Senior Researcher
1994 Sept. – 1995 August	ARÇELİK A.Ş., Eskişehir Buzdolabı İşletmesi	Product Development Engineer

**MILITARY SERVICE:**

<b>Year</b>	<b>Place</b>	<b>Enrollment</b>
1995 August –	T.S.K. K.K.K.	Reserve Officer
1996 December	33ncü Mknz. P. Tug. K. 3ncü Hd. Tb. Demirköy / Kırklareli	

**FOREIGN LANGUAGES:**

Advanced English.

## **PUBLICATIONS:**

1. Ekütekin, V., Özgören, K., “*Gerçek Zamanlı Olasılıklu Veri İlişkilendirme Filtresi Kullanılarak Arazi Destekli Seyriüsefer Çözümlerinin Gerçekleştirilmesi*”, SAVTEK 2006, Savunma Teknolojileri Kongresi, 29-30 Haziran 2006, Ankara.
2. Ataç, S., Platin, B., E., Balkan, T., Ekütekin, V., Özsoy, A., N., “*Bir İnsansız Hava Aracının Sayısal Arazi Yükseklik Verisi Kullanılarak Yükseklik Kontrolü*”, SAVTEK 2006, Savunma Teknolojileri Kongresi, 29-30 Haziran 2006, Ankara.
3. Ekütekin, V., “*Topçu Kargo Mühimmatı Bombacık Toplam Sürekleme Katsayılarının Deneysel Olarak Belirlenmesi*”, SAVTEK 2004, Savunma Teknolojileri Kongresi, 24-25 Haziran 2004, Ankara.
4. Ekütekin, V., “*Integration and Error Modeling of Global Positioning System (GPS) and Inertial Navigation System (INS)*”, METU, M.Sc. Thesis, May 2000, Ankara.
5. Mahmetyazıcıoğlu, G., Ekütekin, V., Korkmaz, Ş., Aytar, S., Özsoy, A., N., Bektaş, İ., Yazıcı, Z., T., Gözübük, S., “*TÜBİTAK-SAGE Uçuş Mekanığı Laboratuvarı*”, Mühendis ve Makina Dergisi, Sayı: 491, MMO, Aralık 2000, Ankara.