

The Evolution Of Highly Eccentric Orbits

by

Jack David Fischer

B.S. Astronautical Engineering
United States Air Force Academy, 1996

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 1998

©Jack D. Fischer. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly
paper and electronic copies of this thesis document in whole or in part.

Signature of Author: _____
Department of Aeronautics and Astronautics
May 8, 1997

Certified by: _____
Dr. Ronald J. Proulx
Thesis Supervisor, CSDL

Certified by: _____
Dr. Paul J. Cefola
Thesis Supervisor, CSDL
Lecturer, Department of Aeronautics and Astronautics

Accepted by: _____
Professor Jaime Peraire
Chair, Graduate Office

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUL 08 1998

LIBRARIES

1

ARCHIVES

[This page intentionally left blank.]

The Evolution of Highly Eccentric Orbits

by

Jack David Fischer

Submitted to the Department of Aeronautics and Astronautics
on May 8, 1997 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Astronautical Engineering

ABSTRACT

Highly eccentric orbits pose a unique challenge to the orbit determination process. These orbits may be perturbed by a broad spectrum of the natural forces which govern the motion of a space object and are difficult to model over long arcs. This work identifies the perturbation model improvements in the Draper R&D version of the Goddard Trajectory Determination System (GTDS) necessary to improve accuracy for a difficult orbit type known as the Molniya class orbit. Molniya orbits, with an eccentricity of approximately 0.7 and a period of slightly less than 12 hours, exaggerate the effects of eccentricity truncation in the J_2^2 and tesseral resonance models. Additionally, these orbits may produce a steep reentry path and magnify the effects of atmospheric drag, particularly when the decay occurs near the minimum in the approximate 11 year solar cycle. This work analyzes the evolution of these orbits, from the mission or operational phase to the final degradation of the orbit. For the mission phase, there is an investigation of the primary source of remaining un-modeled error for these orbits, the truncation of the J_2^2 term, and possible approaches for developing a working model of a closed form, second order, analytic solution for this term. For the primary focus of this work, the decay phase, a brief analysis of the modeling errors for various atmospheric effects is accomplished to create a functional representation of the process noise. Two variants of the error function are then used with a Kalman filter to develop an efficient algorithm for reducing observations for these orbits. The MSISE-90 atmospheric and a simplified lift model are also integrated into GTDS to more accurately model the decay phase. In addition, several utilities are created to efficiently analyze data characteristics of the observational data used for this work. By improving the necessary perturbation models, developing an efficient method for processing observations of these orbits, and identifying major sources of errors produced by the mission and decay environment, this work lays the groundwork for a refinement of previous work done in the mission phase, and serves as a first step in an effort to accurately model the evolution of highly eccentric orbits through the decay phase.

Thesis Supervisor: Dr. Ronald J. Proulx

Title: Technical Staff Engineer, The Charles Stark Draper Laboratory, Inc.

Thesis Supervisor: Dr. Paul J. Cefola

Title: Lecturer, Department of Aeronautics and Astronautics
Program Manager, The Charles Stark Draper Laboratory, Inc.

[This page intentionally left blank.]

Acknowledgments

Over the past two years I have received more blessings and learned more than I ever dreamt possible, and would like to take this opportunity to thank the multitude of people who are responsible.

First, I would like to thank my two thesis advisors, Ronald J. Proulx and Paul J. Cefola, who took a chance on me when no one else would. It has been a true honor to work under such distinguished professionals. To Dr. Proulx, thanks for your hours of help wading through GTDS and for being a continuous source of ideas and suggestions. To Dr. Cefola, for fostering my interest in management and leadership, as well as astrodynamics. In the face of hardship, you have continued to help and inspire me to become a better person, not just a better dynamicist.

I would like to thank the entire education office including, John Sweeney, Loretta Mitrano, and Arel Maguire for affording me the opportunity to participate in the Draper Fellow program and helping me along the way.

Draper Laboratory is full of other people worthy of thanks. To Wayne McClain for your help with, and intimate knowledge of, DSST; David Carter for your data processing code and atmospheric modeling knowledge; Joe Neelon and Brian Kantsiper for your friendship, conversations, and ideas; Jim Smith for sharing the office and the journey into fatherhood; and to Roger Medeiros, Rick Flanagan, Karen Koerber, Rick Metzinger, Warren Fitzgerald, and Jim Harrison for your support and constructive comments. I especially want to thank Naresh Shah for his friendship and continuing tutelage. Everything from lunchtime discussions that stretched my mind, to help with Battin homework, and the lucky 101, you've been a great friend.

The technical contributions for this paper also include non-Draper personnel. Thanks to Dick Hujsak (Logicon) for his countless ideas and advice on filtering; LTC David Vallado (Phillips Lab) for his comprehensive book and comments; Mike Picone (NRL) for his personal knowledge of the MSISE-90 model; Kenneth Schatten for his Ap prediction data; Mike Gaposchkin (Millstone Hill), Taft DeVere (SenCom Corporation), Dan Rypysc (Raytheon), and Dan Fonte (Phillips Lab) for their help in creating the SLAD; Nick Johnson (NASA JSC) and Steven Alderete (PL/WSA) for their expertise with Russian spacecraft; Bob Berman and the Macsymba Technical staff for their continuous support of this project; Gerald Karr (UAH) for his extensive help with analyzing the atmosphere and ideas on modeling lift; and Bill Kincaid for his help with the Agena comparative analysis. Special thanks to Bili Craig for numerous suggestions and guidance throughout the project, supplying the data, and conceiving the investigation topic.

As any author knows, support outside of work is just as important. Thanks go to Richard Battin for showing me the beauty of astrodynamics; to my mentors Gil Moore and Robert Giffen for their support and advice; to all the Air Force friends that have made Boston

fun; to Jerry Guida and Father DeAmato for helping me learn about my faith; Rebecca Martin for Saturday nights with Elizabeth and I; the Birches (especially Pauli) as a roommate, friend, and wedding pianist; to all the cadets and cadre of MIT ROTC Detachment 365 who let me help with the morning workout program; and to Eric Sundberg (the Magic Colonel) there's too much to mention, you've been like a father to me.

Special thanks to my family for always being the constant for me. The love and support you've given me have been the foundation of my life. To my new family, the Simonsons, thank you for welcoming me into the ranks.

To my greatest joy, my own family on Hunting Street, I thank you for bringing meaning to my life and a happiness I never thought possible. To Elizabeth my wife, best friend, confidant, and proofreader, you have made me whole, and this thesis legible. To my darling daughter Sariah, for melting my heart, making me a dad, and being the sweetest little girl in the world. To my Beagle Buddy, for teaching me patience.

Finally, and most importantly, I want to thank God for the many blessings he's given me through my life, and especially in the last two years. In that time, I've found my faith and been blessed with a new wife, a new daughter, a new dog, and a wonderful experience here in Cambridge.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., with support from Draper Laboratory's DFY98 IR&D.

Publication of this thesis does not constitute approval by The Charles Stark Draper Laboratory, Inc., or the Massachusetts Institute of Technology of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

Permission is hereby granted by the Author to the Massachusetts Institute of Technology to reproduce any or all of this thesis.

Jack Fischer, 2Lt, USAF

Contents

Chapter 1 Introduction.....	31
1.1 The Catastrophic Decay Problem	31
1.2 The Molniya Orbit	37
1.3 Previous Work	39
1.4 Mission and Decay Phase Stratification	41
1.5 Objects Analyzed	44
1.5.1 Molniya Spacecraft ..	45
1.5.2 Molniya Rocket Body	48
1.6 Thesis Overview	50
Chapter 2 Orbit Propagation and Impact Prediction Methods	53
2.1 Introduction	53
2.1.1 Orbit Generators	53
2.1.2 GTDS Integration Methods	57
2.1.3 Orbital Elements and Reference Frames	59
2.1.3.1 Keplerian Elements	61
2.1.3.2 Equinoctial Elements	64
2.2 Cowell Math Techniques	67
2.3 Variation of Parameters	69
2.3.1 Gaussian VOP Equations	72
2.3.2 Lagrange Planetary Equations	74
2.3.3 VOP Equations in Equinoctial Elements	77
2.4 Draper Semi-Analytic Satellite Theory	79
2.4.1 The Generalized Method of Averaging	80

2.4.2 Averaged VOP Equations of Motion	84
2.4.3 Elimination of Fast Variable Dependence	89
2.4.4 Determination of Short Period Functions	92
2.4.5 Computational Procedure	94
2.5 Impact Prediction Methods	96
Chapter 3 Statistical Orbit Determination Methods	103
3.1 Introduction	103
3.2 Least Squares Batch Methods	104
3.2.1 Linear Weighted Least Squares Formulation	105
3.2.2 Nonlinear Least Squares Formulation	109
3.3 Sequential Batch Least Squares Methods	111
3.4 Kalman Filters	113
3.4.1 Generalized Filter Equations	116
3.4.2 Linearized Kalman Filter	120
3.4.3 Extended Kalman Filter	122
3.4.5 Error Modeling	124
3.5 Performance Metrics	126
Chapter 4 The Research and Development Version of the Goddard Trajectory Determination System (R&D GTDS)	129
4.1 Introduction	129
4.2 Developmental History	134
4.3 Operation of R&D NT-GTDS	140

4.3.1	Compilation and Linking of NT-GTDS Code	140
4.3.2	NT-GTDS Data Assignment Files	143
4.3.3	NT-GTDS Input Card Data Files	150
4.3.4	NT-GTDS Batch Execution Files	153
4.3.5	Summary	153
Chapter 5	Mission Phase Analysis	157
5.1	Introduction	157
5.2	Environment Summary	158
5.3	J_2^2 Error Identification	165
5.4	Progress on the J_2^2 Closed-Form Solution	170
5.4.1	Development of the Gravitational Potential Function	174
5.4.2	Development of Necessary Partial Derivatives	176
5.4.2.1	Development of the Eccentric Longitude Partial Derivatives	176
5.4.2.2	Development of the (a/r) Partial Derivatives	177
5.4.2.3	Development of the True Longitude Partial Derivatives	183
5.4.2.4	Development of the Gravitational Potential Partial Derivatives	186
5.4.2.5	Summary of Necessary Partial Derivatives	187
5.4.3	Development of the F Functions	188
5.4.4	Macsyma Symbolic Algebra Utilities	191

Chapter 6	Decay Phase Analysis	197
6.1	Introduction	197
6.2	Atmospheric Drag and Density Modeling	198
6.2.1	MSISE-90 Atmospheric Model	201
6.2.2	GTDS Integration	202
6.2.3	Testing and Validation	210
6.3	Atmospheric Lift Modeling	213
6.3.1	Simplified Lift Model	215
6.3.1.1	Drag Model Partial Derivatives	217
6.3.1.2	Lift Model Velocity Partial Derivatives	219
6.3.1.3	Lift Model Position Partial Derivatives	221
6.3.1.4	Lift Model Dynamic Parameters Partial Derivatives ..	223
6.3.2	NT-GTDS Integration	223
6.3.3	Testing and Validation	225
6.4	Other Decay Phase Perturbations	227
6.4.1	Speed Ratio Error	227
6.4.2	Transition Region Error	228
6.4.3	Higher Order Drag and Oblateness Coupling	230
6.4.4	Short Periodics with Drag	234
6.4.5	Atmospheric Winds	235
6.4.6	Reentry Heating & Structural Failure	236
6.4.7	Hypersonic Flow Considerations	240
6.5	Error Modeling	242
6.5.1	Time Dependent Error Function	243
6.5.2	Altitude Dependent Error Function	250

6.5.3 GTDS Implementation	256
Chapter 7 Decay Phase Testing and Results	259
7.1 Introduction	259
7.2 Differential Correction Batch Runs	259
7.3 Kalman Filter Runs	264
7.3.1 Analysis Variables	265
7.3.2 Runs Summary	270
7.3.3 Impact Prediction	277
7.4 Recent Filter Modifications	279
7.4.1 Altitude Dependent Error Function (ADEF)	279
7.4.2 Simplified Lift Model (SLM)	282
Chapter 8 Conclusions and Future Work	287
8.1 Conclusions	287
8.1.1 Data Analysis Techniques	288
8.1.2 Mission Phase	292
8.1.3 Decay Phase	293
8.2 Future Work	295
8.2.1 Data Analysis Techniques	295
8.2.2 Mission Phase	298
8.2.3 Decay Phase	299

Appendix A Useful Program Utilities	303
A.1 NORAD Pre-Processor (NPP)	303
A.1.1 Problems Encountered	304
A.1.1.1 IBM 3090 Mainframe Compatibility	305
A.1.1.2 Comparison Data	307
A.1.2 Operation of the NORAD Pre-Processor (NPP)	310
A.1.3 NORAD B-3 Format	314
A.1.3 Station Location and Accuracy Database (SLAD)	317
A.2 Norad Pre-Processor Shell (NPP)	320
A.3 Single Orbit Observation Density Analysis Program (SOODAP)	324
A.4 Gap Analysis Program (GAP)	326
A.5 Generated Ephemeris Analysis Program (GEAP)	329
A.6 Filter Control Card Generator (FCCG)	331
A.7 Residual Data File Generator (RDFG)	332
A.8 Utility Coding Summary	334
Appendix B Data Analysis Plots	337
B.1 NSSC-13390	337
B.1.1 Element History Plots	337
B.1.2 Observation Density Analysis	341
B.1.3 Single Orbit Observation Density Analysis	342
B.1.4 Observation Gap Analysis	345
B.2 NSSC-12066	347
B.2.1 Observation Density Analysis	347
B.2.2 Observation Gap Analysis	348

B.3	NSSC-13112	349
	B.3.1 Observation Density Analysis	349
	B.3.2 Observation Gap Analysis	350
B.4	NSSC-13253	351
	B.4.1 Observation Density Analysis	351
	B.4.2 Observation Gap Analysis	352
B.5	NSSC-14199	353
	B.5.1 Observation Density Analysis	353
	B.5.2 Observation Gap Analysis	354
B.6	NSSC-14582	355
	B.6.1 Observation Density Analysis	355
	B.6.2 Observation Gap Analysis	356
B.7	NSSC-14830	357
	B.7.1 Observation Density Analysis	357
	B.7.2 Observation Gap Analysis	358
B.8	NSSC-15481	359
	B.8.1 Observation Density Analysis	359
	B.8.2 Observation Gap Analysis	360
B.9	NSSC-16805	361
	B.9.1 Observation Density Analysis	361
	B.9.2 Observation Gap Analysis	362
B.10	NSSC-16889	363
	B.10.1 Observation Density Analysis	363
	B.10.2 Observation Gap Analysis	364
B.11	Data Analysis Summary	365

Appendix C	GTDS Modifications and Additions	371
C.1	Control Card Modifications and Additions	371
C.2	Code Modifications and Additions	436
C.2.1	MSISE-90 Atmosphere Model Modifications	436
C.2.2	Filter Run-Time Modifications	440
C.2.3	Filter Input Processor Modifications	441
C.2.4	Simplified Lift Model Modifications	444
C.2.5	Error Function Modifications	451
C.2.6	PR4 Operational Modifications	453
C.2.7	Range-Gating Option Modifications	454
C.3	GTDS Modification Summary	455
Appendix D	Selected GTDS Control Cards	457
D.1	Differential Correction Batch and Ephemeris Generation Runs	457
D.1.1	J_2^2 Error Identification Runs	458
D.1.2	Real Data Runs	460
D.2	Kalman Filter Runs	463
D.3	Permanent File Report Runs	469
Appendix E	Macsyma Program Additions and Modifications	471
E.1	BPART2(I)	471
E.2	FFUNCT(I)	472
E.3	FFUNCT2(I)	473
E.4	FLOAD	475
E.5	F[L,M,P](I)	475

E.6	G[L,P,Q,MAXE](E)	476
E.7	LPART2(I)	477
E.8	POISSON2(I,J,MAXE)	478
E.9	UDERIV2(I)	481
E.10	UPART2(I)	481
E.11	UPARTIAL(I)	483
E.12	UZONCF[N](A,P,Q,B,L)	484
E.13	UZONCF2[N](A,P,Q,B,L)	485
E.14	ZEIS	486
References	489

[This page intentionally left blank.]

List of Figures

Figure 1.1	Satellite Orbits	35
Figure 1.2	Perigee Drift	38
Figure 1.3	Molniya Ground Track	38
Figure 1.4	Molniya Orbit Diagram	39
Figure 1.5	Radius of Perigee Element History	40
Figure 1.6	Radius of Apogee Element History	40
Figure 1.7	13390 Decay Profile	42
Figure 1.8	Top View of Orbital Decay	43
Figure 1.9	Molniya 1/S1 Technical Drawing	46
Figure 1.10	Molniya 1 Picture	46
Figure 1.11	Molniya 3 Solar Array Configuration	47
Figure 1.12	Molniya A6e-SL6, Block L Rocket Body Technical Drawing	48
Figure 1.13	Molniya A6e-SL6, Block L Rocket Body Picture	49
Figure 2.1	Keplarian Elements	62
Figure 2.2	Orbital Anomalies for Elliptic Motion	63
Figure 2.3	Direct Equinoctial Coordinate Frame	66
Figure 2.4	The Generalized Method of Averaging	80
Figure 2.5	GMA Harmonics Stratification	83
Figure 2.6	Molniya 1-55 Rocket Body Final and Mission Orbit Ground Tracks ..	100
Figure 3.1	Representation of a Linearized Kalman Filter	122
Figure 3.2	Representation of an Extended Kalman Filter	123
Figure 4.1	GTDS Initial Developmental History Diagram	134
Figure 4.2	Draper R&D GTDS Developmental History Diagram	137
Figure 4.3	Sample Lahey F77 Makefile	141

Figure 4.4	NT-GTDS Keyword Card Structure	150
Figure 5.1	Zonal Harmonics Diagram	162
Figure 5.2	Sectorial Harmonics Diagram	162
Figure 5.3	Tesseral Harmonics Diagram	163
Figure 5.4	Physical Representation of Nodal Regression	164
Figure 5.5	Motion of Orbital Plane on Celestial Sphere Due to Nodal Regression	165
Figure 5.6	Full Model Cross Track Position Prediction Error	167
Figure 5.7	Full Model Cross Track Velocity Prediction Error	167
Figure 5.8	Full Model Longitude of Ascending Node (Ω) Prediction Error	168
Figure 5.9	Cross Track Position Prediction Error for Minimized J_2^2	168
Figure 5.10	Cross Track Velocity Prediction Error for Minimized J_2^2	169
Figure 5.11	Longitude of Ascending Node (Ω) Prediction Error for Minimized J_2^2	169
Figure 5.12	Zeis Macsyms Blocks Flow Diagram	194
Figure 5.13	New Macsyms Blocks Flow Diagram	194
Figure 6.1	Karr Gas Surface Interaction Model	199
Figure 6.2	MSISE-90 Coding Structure	204
Figure 6.3	AERO Structure Diagram	204
Figure 6.4	MSISE-90 Binary Data File Flowchart	209
Figure 6.5	A_p Data Merging Technique	210
Figure 6.6	Atmospheric Model Semi-Major Axis Difference Comparison	211
Figure 6.7	Semi-Major Axis Element History	212
Figure 6.8	Effect of Satellite Orientation on Lift/Drag Coefficients	214
Figure 6.9	The SLM Coordinate System	217
Figure 6.10	SLM Routine Modifications	224
Figure 6.11	SLM Flow Diagram	225

Figure 6.12	Molniya Velocity Cylinder	229
Figure 6.13	Atmospheric Winds Error Magnitude	236
Figure 6.14	NSSC-13390 SMA/Eccentricity Decay History	238
Figure 6.15	NSSC-14830 SMA/Eccentricity Decay History	238
Figure 6.16	NSSC-13390 SMA Decay Profile Close Up	244
Figure 6.17	NSSC-13390 Perigee Decay Profile	245
Figure 6.18	Error Torus at Epoch State	248
Figure 6.19	Error Torus at Final Decay	248
Figure 6.20	Error Torus Graphical Representation	249
Figure 6.21	Cubics for the Four Boundary Conditions of the Hermite Cubic	253
Figure 6.22	Altitude Dependent Error Function (Default)	256
Figure 6.23	Error Function Implementation into NT-GTDS	257
Figure 7.1	Step Fit Span Diagram	263
Figure 7.2	February 1995 Data Gaps Plot	267
Figure 7.3	December-January 1995 – Data Gaps Plot	268
Figure 7.4	December-January 1995 – EGLQ SMA Residuals	268
Figure 7.5	27 March, 1995 – PRNQ SMA Residuals	268
Figure 7.6	Decay Mean Motion Acceleration	269
Figure 7.7	Decay Mean Motion Velocity	270
Figure 7.8	NSSC-13390 Decay Processing Diagram	274
Figure 7.9	Cowell Ephemeris Generations: January – March 1995	274
Figure 7.10	Cowell Ephemeris Generations: March 1995	275
Figure 7.11	January-March 1995, Hourly K_p Sums	276
Figure 7.12	January-March 1995, Daily A_p Values	276
Figure 7.13	January-March 1995, Daily $F_{10.7}$ Values	276

Figure 7.14	Impact Prediction Ground Tracks	278
Figure 7.15	Time Dependent Gaussian Approximation Error	280
Figure 7.16	Time Dependent Gaussian Approximation Error – Close Up	280
Figure 7.17	NSSC-13390 Drag Coefficient During Final Decay	283
Figure 8.1	The Interaction of the Atmosphere Variation Model with the SSS Algorithms	300
Figure A.1	Tree Structure Diagram of the NORAD Pre-Processor	310
Figure A.2	NORAD Pre-Processor Input Control Card	311
Figure A.3	Truncated NORAD Pre-Processor Station Names File	311
Figure A.4	NORAD Pre-Processor Print Messages File	313
Figure A.5	NT-GTDS FRN-15 OBSCARD File	313
Figure A.6	Station Location and Accuracy Database Description Block	318
Figure A.7	Truncated 16805 Station Matrix File	321
Figure A.8	Print Messages File Trailer Data	322
Figure A.9	Tree Structure Diagram of the NPP Shell Program	323
Figure A.10	Truncated Version of the Station Locations File	323
Figure A.11	Truncated Version of the Station Accuracy File	324
Figure A.12	Truncated Version of the Station Coordinate Frame File	324
Figure A.13	Tree Structure Diagram of the Single Orbit Observation Density Analysis Program	325
Figure A.14	Single Orbit Observation Density Analysis Program Control Card	326
Figure A.15	Single Orbit Observation Density Analysis Program Output File	326
Figure A.16	Tree Structure Diagram of the Gap Analysis Program	327
Figure A.17	Gap Analysis Program Control Card	328
Figure A.18	Gap Analysis Program Summary Output File	328
Figure A.19	Gap Analysis Program Plotting Data Ouput File	329

Figure A.20	NSSC-12066 November 1995 Gap Analysis Program Plot	329
Figure A.21	Tree Structure Diagram of the Generated Ephemeris Analysis Program	330
Figure A.22	Generated Ephemeris Analysis Program Control Card	331
Figure A.23	Generated Ephemeris Analysis Program Output File	331
Figure A.24	Filter Control Card Generator Program Control Card	331
Figure A.25	Tree Structure Diagram of the Filter Control Card Generator Program	332
Figure A.26	Filter Control Card Generator Program Output File	332
Figure A.27	Tree Structure Diagram of the Residual Data File Generator Program	333
Figure A.28	Residual Data File Generator Program Output File	333
Figure A.29	Residual Data File Generator Program Control Card	334
Figure B.1	NSSC-13390 Semi-Major Axis Lifetime History	338
Figure B.2	NSSC-13390 Semi-Major Axis Final Three Months History	338
Figure B.3	NSSC-13390 Eccentricity Lifetime History	338
Figure B.4	NSSC-13390 Eccentricity Final Three Months History	338
Figure B.5	NSSC-13390 Inclination Lifetime History	338
Figure B.6	NSSC-13390 Inclination Final Three Months History	338
Figure B.7	NSSC-13390 Right Ascension of Ascending Node Lifetime History ..	339
Figure B.8	NSSC-13390 Right Ascension of Ascending Node Final Three Months History	339
Figure B.9	NSSC-13390 Argument of Perigee Lifetime History	339
Figure B.10	NSSC-13390 Argument of Perigee Final Three Months History	339
Figure B.11	NSSC-13390 Mean Anomaly Lifetime History	339
Figure B.12	NSSC-13390 Mean Anomaly Final Three Months History	339
Figure B.13	NSSC-13390 Perigee Height Lifetime History	340
Figure B.14	NSSC-13390 Perigee Height Final Three Months History	340

Figure B.15	NSSC-13390 B* Lifetime History	340
Figure B.16	NSSC-13390 B* Final Three Months History	340
Figure B.17	NSSC-13390 $\dot{n}/6$ Final Three Months History	340
Figure B.18	NSSC-13390 $\dot{n}/2$ Final Three Months History	340
Figure B.19	NSSC-13390 Yearly Observation Density Profile	341
Figure B.20	NSSC-13390 Final Year Monthly Observation Density Profile	341
Figure B.21	NSSC-13390 Lifetime Single Orbit Observation Density Profile	342
Figure B.22	NSSC-13390 Single Orbit Observation Density Profile: Three Months Prior to Decay Month	343
Figure B.23	NSSC-13390 Lifetime Single Orbit Azimuth-Elevation Observation Density Profile	343
Figure B.24	NSSC-13390 Lifetime Single Orbit Range-Azimuth-Elevation Observation Density Profile	344
Figure B.25	NSSC-13390 Lifetime Single Orbit Right Ascension-Declination Observation Density Profile	344
Figure B.26	NSSC-13390 Lifetime Gap Analysis Plot	345
Figure B.27	NSSC-13390 Final Nine Months Gap Analysis Plot	345
Figure B.28	NSSC-13390 January 1995 Gap Analysis Plot	346
Figure B.29	NSSC-13390 February 1995 Gap Analysis Plot	346
Figure B.30	NSSC-13390 March 1995 Gap Analysis Plot	346
Figure B.31	NSSC-12066 Yearly Observation Density Profile	347
Figure B.32	NSSC-12066 Final Year Monthly Observation Density Profile	347
Figure B.33	NSSC-12066 Final Nine Months Gap Analysis Plot	348
Figure B.34	NSSC-12066 October 1995 Observation Gap Analysis Plot	348
Figure B.35	NSSC-12066 November 1995 Observation Gap Analysis Plot	348

Figure B.36	NSSC-13112 Yearly Observation Density Profile	349
Figure B.37	NSSC-13112 Final Year Monthly Observation Density Profile	349
Figure B.38	NSSC-13112 Final Nine Months Gap Analysis Plot	350
Figure B.39	NSSC-13112 October 1992 Observation Gap Analysis Plot	350
Figure B.40	NSSC-13112 November 1992 Observation Gap Analysis Plot	350
Figure B.41	NSSC-13253 Yearly Observation Density Profile	351
Figure B.42	NSSC-13253 Final Year Monthly Observation Density Profile	351
Figure B.43	NSSC-13253 Final Nine Months Gap Analysis Plot	352
Figure B.44	NSSC-13253 February 1993 Observation Gap Analysis Plot	352
Figure B.45	NSSC-13253 March – April 1993 Observation Gap Analysis Plot	352
Figure B.46	NSSC-14199 Yearly Observation Density Profile	353
Figure B.47	NSSC-14199 Final Year Monthly Observation Density Profile	353
Figure B.48	NSSC-14199 Final Nine Months Gap Analysis Plot	354
Figure B.49	NSSC-14199 January 1995 Observation Gap Analysis Plot	354
Figure B.50	NSSC-14199 February 1995 Observation Gap Analysis Plot	354
Figure B.51	NSSC-14582 Yearly Observation Density Profile	355
Figure B.52	NSSC-14582 Final Year Monthly Observation Density Profile	355
Figure B.53	NSSC-14582 Final Nine Months Gap Analysis Plot	356
Figure B.54	NSSC-14582 March 1994 Observation Gap Analysis Plot	356
Figure B.55	NSSC-14582 April 1994 Observation Gap Analysis Plot	356
Figure B.56	NSSC-14830 Yearly Observation Density Profile	357
Figure B.57	NSSC-14830 Final Year Monthly Observation Density Profile	357
Figure B.58	NSSC-14830 Final Nine Months Gap Analysis Plot	358
Figure B.59	NSSC-14830 August 1994 Observation Gap Analysis Plot	358
Figure B.60	NSSC-14830 September 1994 Observation Gap Analysis Plot	358

Figure B.61	NSSC-15481 Yearly Observation Density Profile	359
Figure B.62	NSSC-15481 Final Year Monthly Observation Density Profile	359
Figure B.63	NSSC-15481 Final Nine Months Gap Analysis Plot	360
Figure B.64	NSSC-15481 March 1995 Observation Gap Analysis Plot	360
Figure B.65	NSSC-15481 April 1995 Observation Gap Analysis Plot	360
Figure B.66	NSSC-16805 Yearly Observation Density Profile	361
Figure B.67	NSSC-16805 Final Year Monthly Observation Density Profile	361
Figure B.68	NSSC-16805 Final Nine Months Gap Analysis Plot	362
Figure B.69	NSSC-16805 November 1996 Observation Gap Analysis Plot	362
Figure B.70	NSSC-16805 December 1996 Observation Gap Analysis Plot	362
Figure B.71	NSSC-16889 Yearly Observation Density Profile	363
Figure B.72	NSSC-16889 Final Year Monthly Observation Density Profile	363
Figure B.73	NSSC-16889 Final Nine Months Gap Analysis Plot	364
Figure B.74	NSSC-16889 September 1996 Observation Gap Analysis Plot	364
Figure B.75	NSSC-16889 October 1996 Observation Gap Analysis Plot	364
Figure B.76	Yearly Observation Density Comparison Plot: 15 – 8 Years Before Decay	366
Figure B.77	Yearly Observation Density Comparison Plot: Final Seven Years of Decay	366
Figure B.78	Monthly Observation Density Comparison Plot: 12 – 7 Months Before Decay	367
Figure B.79	Monthly Observation Density Comparison Plot: Final Six Months Before Decay	367
Figure B.80	Monthly Observation Density Comparison Plot: Truncated Final Six Months Before Decay	367

Figure B.81	Truncated Contour Plot of Yearly Observation Density	368
Figure B.82	Truncated Contour Plot of Final Year Monthly Observation Density ..	368
Figure D.1	J22 Truth Model EPHEM-DC-EPHEM-COMPARE Control Card ...	460
Figure D.2	Minimized J2 Keyword Card	460
Figure D.3	Real Data DC-EPHEM Batch Fit Control Card – Cowell Propagator .	461
Figure D.4	Real Data DC-EPHEM Batch Fit Control Card – DSST Propagator ...	463
Figure D.5	Extended Kalman Filter Run Control Card – DSST Propagator	465
Figure D.6	Linearized Kalman Filter Run Control Card – Cowell Propagator	467
Figure D.7	Sample SPNOISE Constant Noise Matrix Definition	468
Figure D.8	Sample Altitude Dependent Error Function Definition	468
Figure D.9	Simplified Lift Model Ephemeris Generation Run Control Card	468
Figure D.10	Simplified Lift Model Solve-For Required Keyword Cards	469
Figure D.11	Permanent File Report Control Card – SLP File \$014	470
Figure D.12	Permanent File Report Control Card – SLP File \$078	470
Figure D.13	Permanent File Report Control Card – Jacchia-Roberts Atmosphere File S075	470
Figure D.14	Permanent File Report Control Card – MSISE-90 Atmosphere File S076	470

[This page intentionally left blank.]

List of Tables

Table 1.1	Tracking Data Objects	44
Table 1.2	Molniya 1 Parameters	45
Table 1.3	Molniya A6e-SL6, Upper Stage Block L Parameters	49
Table 2.1	Keplarian Orbital Elements	62
Table 4.1	NSSC-13390 Final Decay Database Files	144
Table 4.2	NT-GTDS Database Files	145
Table 4.3	Mandatory Keyword Cards	151
Table 4.4	Sub-deck Keyword Cards	152
Table 5.1	Disturbing Forces on a Satellite at Various Altitudes	159
Table 5.2	Summary Reference of Necessary Partial Derivatives Equation Numbers	188
Table 5.3	Zeis Macsyma Execution Blocks	192
Table 5.4	New Macsyma Execution Blocks	195
Table 6.1	MSISE-90 File Descriptions	203
Table 6.2	MSISE-90 Input Parameters	206
Table 6.3	MSISE-90 Temporary Data File Format	207
Table 6.4	MSISE-90 Binary Data File Format	208
Table 6.5	HODRAG Option Effects	231
Table 6.6	Frequency Effect Table	233
Table 6.7	Drag Short Periodic Options Effects	234
Table 6.8	Molniya Final Stage Breakups	240
Table 6.9	ADEF Atmospheric Region Error Estimates	252
Table 7.1	Molniya Optimized GTDS Input Card Options at Epoch	260
Table 7.2	Long Term Decay Predictions	261

Table 7.3	DSST – Extended Filter Run Descriptions	271
Table 7.4	Cowell – Linearized Filter Run Descriptions	271
Table 7.5	Cowell- Linearized Filter Run Variables	272
Table 7.6	Impact Prediction Comparisons	278
Table 7.7	ADEF Comparison Data	281
Table 7.8	Primary Effects of Coefficient Alterations	283
Table A.1	ASCII and EBDIC Conversion Value Table	306
Table A.2	GTDS Station Tracker Type Key	308
Table A.3	NT-GTDS Station Name Modifications and Additions	309
Table A.4	NORAD B-3 Observation Card Data Format	315
Table A.5	NORAD Observation Type Format	316
Table A.6	NORAD Coordinate System Type	316
Table A.7	Primary Station Data for Stations Used	318
Table A.8	Existing Code Modifications for the NORAD Pre-Processor	334
Table A.9	New Utilities Code	335
Table B.1	NSSC-13390 Object Suitability Comparison	365
Table B.2	Final Data Comparison	369
Table C.1	MSISE-90 Atmospheric Model NT-GTDS Existing Subroutine Modifications	436
Table C.2	MSISE-90 Atmospheric Model New NT-GTDS Subroutines	438
Table C.3	MSISE-90 Atmospheric Model New NT-GTDS Variables for the CSW Common Block	438
Table C.4	MSISE-90 Atmospheric Model New NT-GTDS Variables for the ELIPSD Common Block	439
Table C.5	MSISE-90 Atmospheric Model New NT-GTDS Variables for the F107AP Common Block	439

Table C.6	MSISE-90 Atmospheric Model New NT-GTDS Variables for the MSIS Common Block	439
Table C.7	Filter Run-Time NT-GTDS Existing Subroutine Modifications	440
Table C.8	Filter Input Processor NT-GTDS Existing Subroutine Modifications	441
Table C.9	Filter Input Processor New NT-GTDS Subroutines	442
Table C.10	Filter Input Processor New NT-GTDS Variables for the SKF Common Block	442
Table C.11	Filter Input Processor New NT-GTDS Variables for the ESKFCM Common Block	444
Table C.12	Simplified Lift Model NT-GTDS Existing Subroutine Modifications	444
Table C.13	Simplified Lift Model New NT-GTDS Subroutines	448
Table C.14	Simplified Lift Model New NT-GTDS Variables for the LFTPAR Common Block	448
Table C.15	Simplified Lift Model Future NT-GTDS Subroutine Modifications Listing	449
Table C.16	Simplified Lift Model New NT-GTDS LFTPAR/Drag Variable Comparison	450
Table C.17	Error Function NT-GTDS Existing Subroutine Modifications	451
Table C.18	Error Function New NT-GTDS Subroutines	452
Table C.19	Error Function New NT-GTDS Variables for the ERRFN Common Block	452
Table C.20	PR4 Operational NT-GTDS Existing Subroutine Modifications	453
Table C.21	Range-Gating Option NT-GTDS Existing Subroutine Modifications	454
Table C.22	Range-Gating Option New NT-GTDS Variables for the CONST Common Block	454

Table C.23	Listing of All NT-GTDS Subroutines Modified	455
Table C.24	Listing of All New NT-GTDS Subroutines	456

Chapter 1

Introduction

1.1 The Catastrophic Decay Problem

Since the launch of Sputnik on October 1, 1957, thousands of objects have been placed in orbit around the Earth including spacecraft, rocket bodies, and orbital debris. The North American Aerospace Defense Command (NORAD) currently tracks approximately 8,500 of these objects, ranging in size from the MIP (Mir) Space Station to objects as small as 10cm [1]. An even greater number of objects, some 15,300 [2], have fallen prey to the Earth's atmosphere and eventually reentered. The sheer number of objects in orbit makes the responsibility of tracking them an unenviable task, one which NORAD has endured for over 30 years.

NORAD accomplishes this feat with the aid of the Satellite Surveillance Network (SSN), a conglomerate of radars and high-resolution optical devices. These instruments obtain observations of the objects in various forms, and return them to NORAD with a corresponding time stamp. Measurements enable orbital analysts to determine an object's orbital path, or orbit, and the future location of the objects, a process known as orbit determination.

Orbit determination includes a broad family of statistical processes that estimate an object's location at a specified epoch using observations and a given set of initial

conditions. Carl Gauss described orbit determination as accumulating “the greatest number of the most perfect observations, and adjust[ing] the elements, not so as to satisfy this or that set of observations with absolute exactness, but so as to agree with all in the best possible manner.” [3]. The reader should note Gauss’s reference to “absolute exactness,” which identifies the inherent inaccuracy of orbit determination. This inaccuracy includes error in the mathematical modeling of the system, or process noise, error in the initial conditions, and errors in measurements. Math modeling errors can also include stochastic, random anomalies such as solar storms, atmospheric winds, or micrometeoroids, which can contribute even more uncertainty.

Unfortunately, there are certain orbit types which accentuate these errors to such a degree that they strongly challenge orbit determination processes. Catastrophic decay orbits, which are typically highly eccentric, are among the worst offenders. These orbits may be perturbed by a broad spectrum of the natural forces that govern the motion of a space object [4] :

- non-spherical gravitational effects
- lunar-solar point masses
- atmospheric drag
- atmospheric lift
- solar radiation pressure

In contrast, the typical Low Earth Orbit (LEO) is primarily perturbed by zonal harmonics and atmospheric drag. The small eccentricity simplifies the computation of the J_2^2 terms and tesseral resonance, if required. In addition, lunar-solar point masses and solar radiation pressure are usually not important for these orbits.

As an example of an eccentric orbit, consider the sun-synchronous, frozen line of apsides (SSFLA) orbit studied by Chris Sabol and Paul Cefola for the ELLIPSO Borealis

Sub-Constellation [5], [6]. An early configuration of this orbit had a semi-major axis of about 10,560 km, an eccentricity of 0.345, and an inclination of 116.5 degrees. This combination of semi-major axis, eccentricity, and inclination combine to produce an orbit which makes eight (8) orbital revolutions over a one (1) day repeat ground track interval. Due to the sharp commensurability between the orbital period and the Earth's rotation rate, tesseral resonance must be considered and, in fact, the tesseral harmonics of eighth (8) and sixteenth (16) orders make resonant contributions. Draper Semi-Analytic Satellite Theory (DSST), which allows mean element and short periodic motion to be separated, was the primary tool used in the SSFLA analysis.

SSFLA's particular combination of semi-major axis, eccentricity, and inclination values also leads to a nodal drift that is equivalent to the drift rate of the right ascension of the Sun; thus this orbit is in resonance with some of the terms in the solar point mass disturbing potential. In addition, this orbit is flown with an argument of perigee of approximately 270 degrees to stabilize the Earth coverage obtained. The combination of sun-synchronicity and fixed line of apsides makes this orbit particularly sensitive to direct solar radiation pressure perturbations [7]. The orbit has a perigee height of about 535 km which makes it subject to significant atmospheric drag perturbations as well. Finally, this orbit has an inclination near the retrograde critical inclination, thus making the orbit subject to the dynamics of critical inclination [8].

From the point of view of the semi-analytical theory, the SSFLA orbit requires comprehensive force models in the equations of motion :

- general zonal harmonic model
- J_2^2 terms including high eccentricity
- general tesseral resonance model with an appropriate Newcomb operator file including high eccentricity.

- lunar-solar point masses
- solar radiation pressure including eclipse
- atmospheric drag

Another class of catastrophic decay orbits which represents similar if not greater difficulty when compared to the SSFLA orbit, is the Молния (Molniya: mohl-nee-ya) orbit. Early in the Soviet space program, the Soviets had no launch vehicle large enough to place satellites in geosynchronous orbit from their high latitude launch sites. Considering their need for high latitude coverage, they needed an alternative and they found one in the Molniya orbit. This unique orbit designed to loiter over the Northern Hemisphere for 8 hours of its 12 hour orbital period [9], is ideal for covering the majority of the Commonwealth of Independent States. Much more common than the SSFLA orbit, Molniya orbits will require that:

- all of the force model terms noted for the SSFLA orbit are included
- J_2^2 terms must be considered with eccentricities up to 0.75
- the tesseral resonance calculations must admit eccentricities up to 0.75

In addition, the terminal (decay) phase of the Molniya orbit [10] experiences some of the highest atmospheric drag perturbing accelerations of any known orbiting vehicle due to steep reentry caused by lunar and solar gravity perturbations. These satellites also have perigees in southern latitudes, an area where few SSN assets exist, making it difficult to obtain observations during the most dynamic portion of the orbit [11]. When combined, these perturbations can lead to losing track of the satellite due to not knowing where to point the radar for observations. Obviously, without observations, it is hard to create an accurate orbit, which makes it hard to get future observations, and so on. For this and the aforementioned perturbation considerations, the Molniya orbit has been

chosen to serve as a test-bed for analyzing catastrophic decay orbits. Using this data, the objectives of this thesis are to:

- improve the necessary perturbation models and data analysis processes to more accurately track the decay of catastrophic decay orbits
- develop an efficient algorithm for processing observations of these orbits
- identify the major sources of error produced by both the mission and decay phase environments

In doing so, this work will serve as the first step in automating the orbit determination process for catastrophic decay orbits, and improving reentry prediction methods for high energy, highly eccentric orbits. The reader should note the importance of improving impact prediction methods, especially for highly eccentric orbits, which are among the most difficult to model. Figure 1.1 shows the distribution of tracked objects in various orbits, including Highly Eccentric Orbits (HEO), of which there are almost 900 objects. This high number of objects coupled with the threat of objects surviving reentry, make reentry prediction for these (and other) orbits very important.

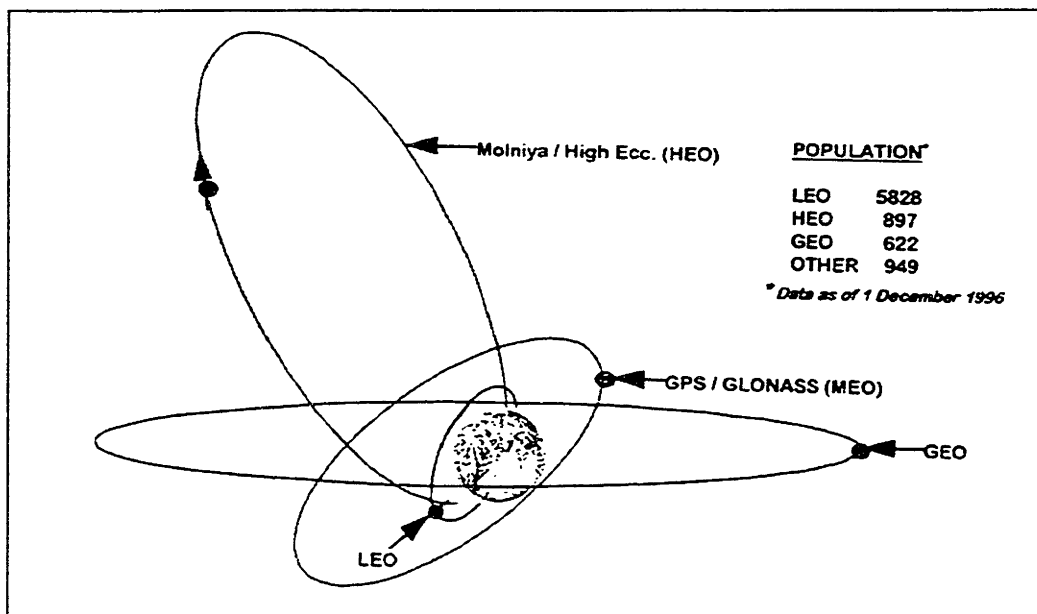


Figure 1.1 – Satellite Orbits [11]

On 16 November 1996, the Russian interplanetary probe, Mars-96, experienced an unknown failure at launch. Mars 96 is not the first satellite to fail at launch, but the 200 grams of radioactive plutonium in its on-board reactor made this satellite a special concern [12]. As the world turned its collective resources to tracking the satellite, technical and operational difficulties resulted in an inability to accurately predict the spacecraft's path. Fortunately, splashdown occurred off the coast of Chile, but the incident identified the need for a more robust tracking system. Other well-known reentries (which met with slightly more success than the highly eccentric Mars 96) include Skylab in 1979, Cosmos 954 in 1978 (which was also radioactive), and a Delta II second stage in 1997 [13].

Historically, estimates of the number of objects that will be destroyed during reentry have been high. In fact, recovered debris has shown that satellite components made of materials with high melting temperatures, such as stainless steel, titanium, and beryllium tend to survive reentry and impact the Earth's surface [13]. Rocket bodies, the primary focus of this study, are typically made of these type of materials and may therefore pose a threat, one which this thesis attempts to address.

In the past, significant improvements in orbit determination accuracy have resulted from the iterative processing of intensive, high quality tracking data and refinement of the physical models used in processing of this data. Examples include the LAGEOS and TOPEX systems which contributed to the enhancement of models for various non-conservative perturbations and geopotential forces [14]. By incorporating various perturbation upgrades into existing high precision orbit propagators (DSST and Cowell), the accuracy of position predictions can improve. These improved accuracy orbit predictions translate directly into improvements of the subsequent data processing for these re-entering space objects.

1.2 The Molniya Orbit

There are many highly eccentric orbit types including communications satellites, remote sensing satellites, and transfer orbits. None, however, are as widely employed as the Molniya orbit. By 1988, 126 Molniya spacecraft had been launched by the Soviet Union, with more launched every year [15]. Молния, the Russian word for lightning [16], is descriptive of how the object whips around “quick as lightning” [17] through perigee and spends the majority of its orbital period over the Northern Hemisphere (with an argument of perigee set at 270° , the apogee remains over the Northern Hemisphere). Another interesting characteristic of a Molniya orbit lies in its inclination.

The geopotential generates variations in all of the standard Keplerian orbital elements. The largest of these variations are secular and felt by the longitude of ascending node (Ω) and the argument of perigee (ω). The result is a drift in Ω and ω approximated by the following equations [9]:

$$\dot{\Omega}_{J_2} \cong -2.06474 \times 10^{14} \left(a^{-7} \right) (\cos i) (1 - e^2)^{-2} \quad (1.1)$$

$$\dot{\omega}_{J_2} \cong 1.03237 \times 10^{14} \left(a^{-7} \right) (4 - 5 \sin^2 i) (1 - e^2)^{-2} \quad (1.2)$$

In order for the Molniya orbit to maintain its coverage, the perigee cannot drift. To achieve this for any orbit, the inclination term in equation (1.2), $(4 - 5 \sin^2 i)$, must be set to zero. Figure 1.2 is a graph of equation (1.2) for the Molniya parameters ($a = 26,500$ km and $e = 0.74$) which shows the critical inclinations that satisfy this condition and freeze the argument of perigee. As the graph indicates, at approximately 63.4° and 116.6° , perigee drift is nullified.

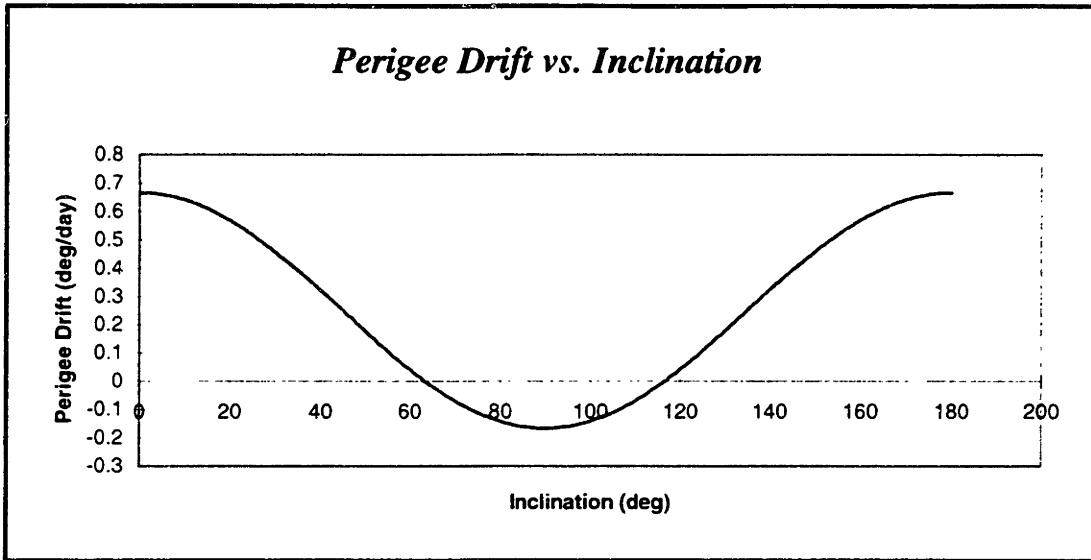


Figure 1.2 - Perigee Drift

Another attractive characteristic of the Molniya orbit is that its ground track repeats. With a 12 hour orbit, the satellites repeats its ground track configuration (of 2 revs/day), which is represented in Figure 1.3. This results in regularizing the coverage

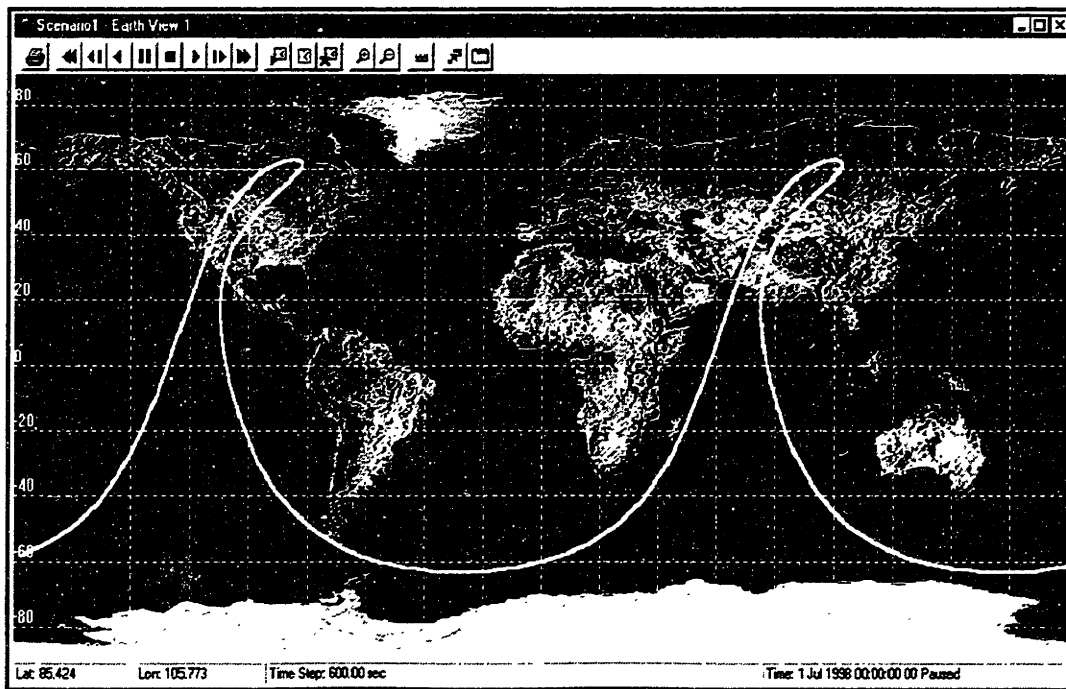


Figure 1.3 – Molniya Ground Track

and a simplification of the operations for the satellite. Figure 1.4 is another view of a typical Molniya orbit with a perspective drawing, please note the position of the perigee and apogee.

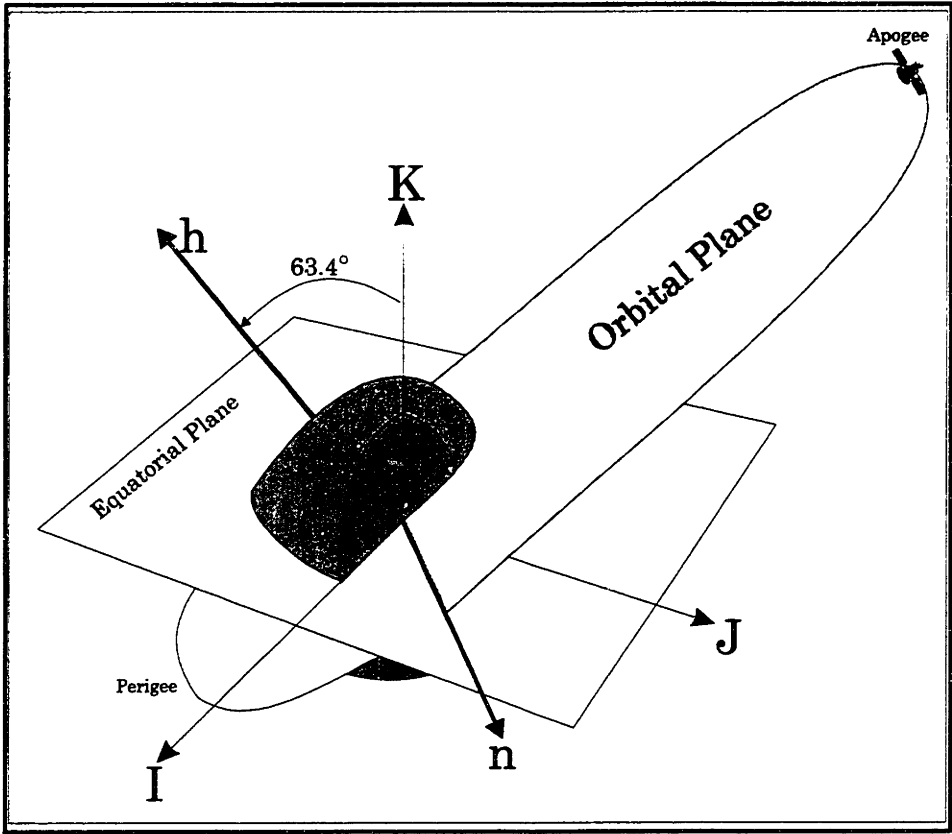


Figure 1.4 – Molniya Orbit Diagram

1.3 Previous Work

D.G. King-Hele, in his two papers “The Orbital Lifetimes of Molniya Satellites,” [18] and “Analysis of the Orbit of 1970-114F in its Last 20 Days” [10] has done extensive research into this area. “The Orbital Lifetimes of Molniya Satellites” identifies lunisolar perturbations as powerful disturbances for the Molniya orbit, shows that the satellites decay when the perigee is forced down to near 100km height, and demonstrates that the domain of permissible lifetimes is a function of the right ascension of node at

epoch. “Analysis of the Orbit of 1970-114F in its Last 20 Days” analyzes the decay of the final stage rocket of the Molniya 1S communications satellite which experienced an exceptionally rapid decay during its last 20 days. This rapid decay, and the numerous observations of the object allowed King-Hele to demonstrate three new techniques for the first time. He was able to use changes in inclination and right ascension of node to determine meridional winds near perigee, changes in argument of perigee to determine atmospheric oblateness (within 20%), and finally to determine the drag coefficient in transition flow. These two works served as the first of many investigations of the unique Molniya orbit, which include previous work at the Charles Stark Draper Laboratory (CSDL).

Martin Fieger, in his MIT thesis “An Evaluation of Semianalytical Satellite Theory Against Long Arcs of Real Data for Highly Eccentric Orbits,” tested the DSST

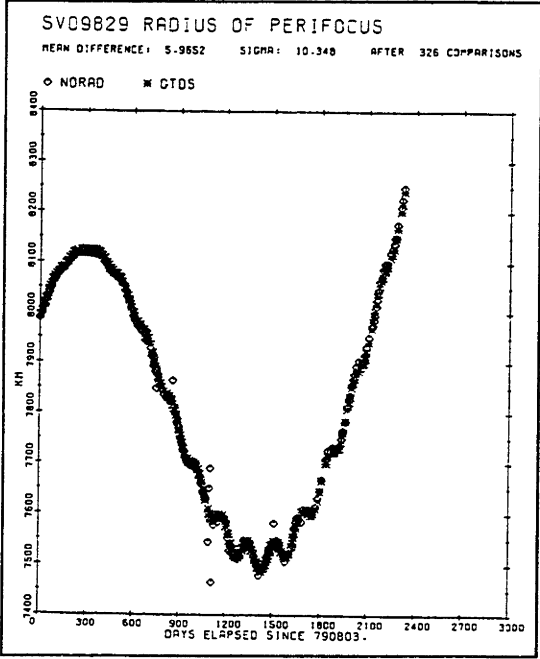


Figure 1.5: Radius of Perigee Element History

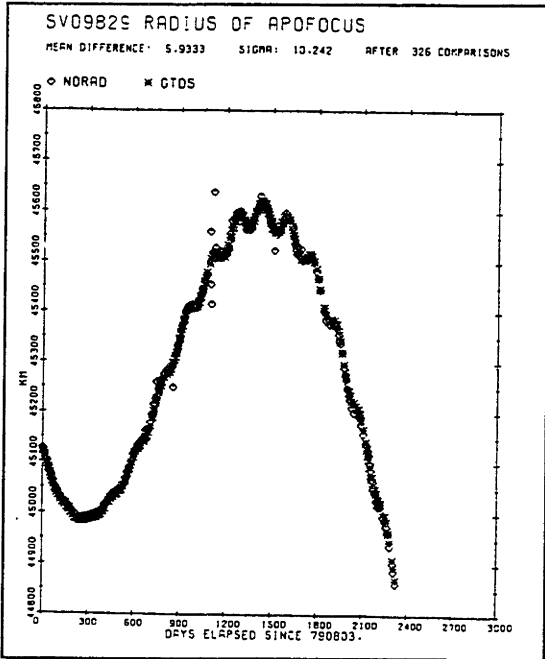


Figure 1.6: Radius of Apogee Element History

system against long arcs of real data from highly eccentric orbits [19]. Using data from NORAD, Fieger showed the ability of Semianalytical Theory to effectively predict highly

eccentric orbits over long intervals of time. He also showed that results obtained from using actual observational data to initialize semianalytical predicts, were superior compared to results obtained using NORAD element sets [20]. In fact the accuracy, demonstrated in Figures 1.5 and 1.6 with comparison plots of short-arc NORAD element sets and GTDS predictions, was sufficient to allow Fieger to solve for parameters such as the solar radiation coefficient, and further improve accuracy.

Unfortunately, Fieger's thesis was limited in that he chose spacecraft whose perigee remained outside the main portion of the atmosphere. This prevented him from testing DSST as the satellites transition into the atmosphere and decay. Inclusion of this portion of a satellite's lifetime, as well as improvements in all phases of the orbital lifetime, are necessary to make DSST, or any propagator, a highly accurate prediction tool for orbits of this type. The reader should note, however, that there has been significant work done with highly eccentric orbits using analytic and semi-analytic propagation methods by Brumberg [21], [22], Lidov and Solov'ev [23], Yurasov, Andrewshchenko, Batyr, Bratchikov, Dicky, and Veniaminov [24], [25]. In particular, the unique characteristics of mean elements and semi-analytic theory, have shown to be very successful in predicting highly eccentric reentry paths [24], [25].

1.4 Mission and Decay Phase Stratification

The analysis of catastrophic decay orbits has been stratified into two distinct phases of the orbital lifetime: the mission phase and the decay phase. The mission phase is defined by that portion of the orbital lifetime with the perigee sufficiently out of the Earth's atmosphere to ignore drag effects (above approx. 1000 km where the mean motion is less than 2.1 revolutions per day). Figure 1.7 demonstrates the reasoning for this stratification of the orbit by showing how drag makes a dramatic impact on the orbit

of a rocket body (launched in 1982) once the object plunges into the atmosphere. The majority of the data corresponds to a 12-hour orbit at two revolutions per day, and then rapidly evolves to more than 14 revs/day just before decay. The eccentricity remains at approximately 0.75 until the decay phase, at which point it rapidly approaches zero. The considerable increase in revs/day and decrease in eccentricity shown on the figure, indicate a circularizing and degradation of the orbit. Once this occurs, the primary modeling concerns shift, and therefore deserve independent analyses.

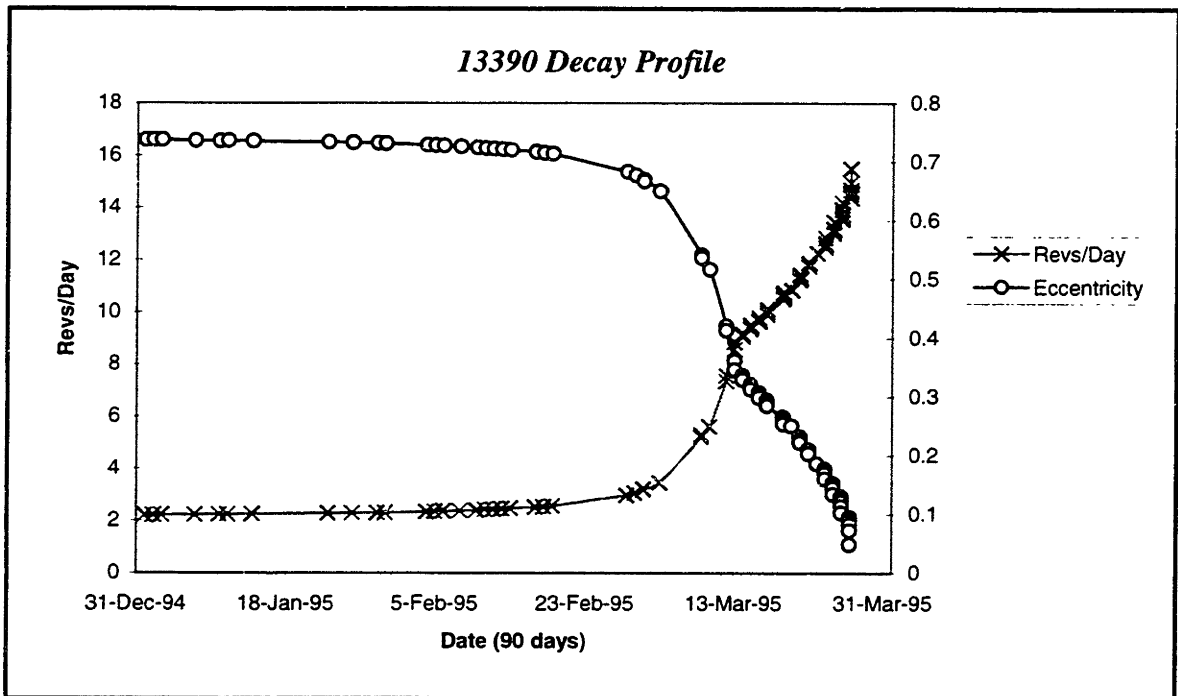


Figure 1.7: 13390 Decay Profile

The mission phase environment is dominated by the aforementioned perturbations of previous studies, most notably J_2^2 terms for high eccentricity and lunisolar perturbations. In general, the errors encountered from neglecting $J_2^2 e^2$ and higher order terms are relatively small, as was demonstrated by Fieger. As a result, the current study places more emphasis on the decay phase, which is comparatively much more difficult to model.

Once the objects reenter the atmosphere during perigee passage (refer to Figure 1.7), drag becomes the dominant force acting on the orbit. Every time the satellite passes through perigee, drag robs the orbit of energy, slowing the object down, and circularizing the orbit. Figure 1.8 depicts this circularizing for an eccentric orbit, as drag causes the orbit to decay. With every pass, more of the orbit falls prey to drag, drag's effect is amplified, and the object decays even more than during its previous orbit. This cycle continues until the apogee height is decreased sufficiently so that a considerable amount of the orbit is within the atmosphere and the satellite finally decays. Orbit 'A' in Figure 1.8 represents a satellite's operational orbit, still outside of the atmosphere. Orbits 'B' and 'C' show the progression of the orbit to final decay, indicated by orbit 'D'. Unfortunately, due to lunar and solar gravity perturbations, Molniya orbits do not undergo as drastic a circularization as is depicted in Figure 1.8 [10], and remain slightly elliptic through final decay. This ellipticity is responsible for creating much steeper reentry paths and amplifying uncertainties in drag, lift, and atmospheric modeling. The degree of circularization may also depend on solar activity (lower levels corresponding to less circularization), as is the case with NSSC-13390 when it decays during a minimum in the 11 year solar cycle.

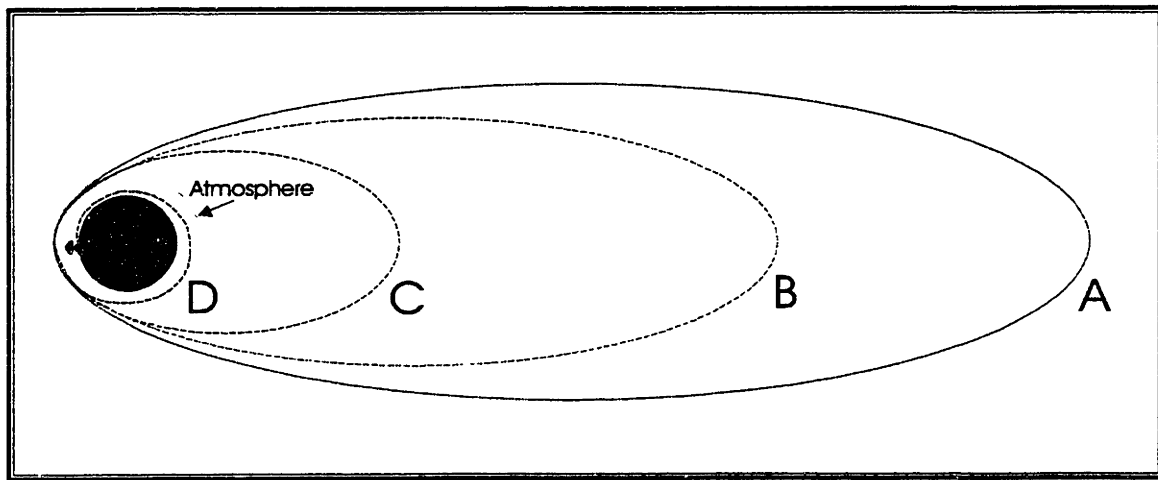


Figure 1.8: Top View of Orbit Decay

The decay phase of Molniya orbits plays host to deterministic and stochastic errors from multiple sources, including: density modeling errors, lift mis-modeling, reentry dynamics, structural failure, drag short periodics, higher order drag terms, drag-oblateness coupling, and more.

1.5 Objects Analyzed

The high number of decayed Molniya objects has provided a wealth of data from which the objects for this experiment were chosen. The ten objects used for this thesis were tracked from launch to reentry, a 10-15 year span of time. A majority of the tracking data was collected near the actual decay of the object, which is when the majority of this analysis takes place. Table 1.1 lists the NORAD identification, description, launch date, and decay date for the objects used in this analysis.

Table 1.1: Tracking Data Objects

<i>Satellite NORAD ID #</i>	<i>Satellite Description</i>	<i>Launch Date</i>	<i>Last Obs Date</i>	<i>Decay Date</i>
NSSC-12066	Molniya 1-48 Spacecraft	18 NOV 80	18 NOV 95	18 NOV 95
NSSC-13112	Molniya 3-18 Rocket Body	24 MAR 82	17 MAY 92	14 NOV 92
NSSC-13253	Molniya 1-54 Rocket Body	28 MAY 82	2 APR 93	2 APR 93
NSSC-13390	Molniya 1-55 Rocket Body	21 JUL 82	27 MAR 95	28 MAR 95
NSSC-14199	Molniya 1-58 Spacecraft	19 JUL 83	23 FEB 95	23 FEB 95
NSSC-14582	Molniya 3-22 Rocket Body	21 DEC 83	24 APR 94	24 APR 94
NSSC-14830	Molniya 1-60 Rocket Body	16 MAR 84	22 SEP 94	22 SEP 94
NSSC-15481	Molniya 3-23 Rocket Body	19 JUL 83	23 APR 95	24 APR 95
NSSC-16805	Molniya 3-29 Rocket Body	19 JUN 86	21 DEC 96	21 DEC 96
NSSC-16889	Molniya 1-67 Rocket Body	30 JUL 86	14 OCT 96	14 OCT 96

The rocket bodies were used as the primary test beds for this analysis to simplify modeling concerns for the decay phase. The only spacecraft, 12066 and 14199, were also

removed from consideration for poor data distribution (12066, 1-48) and a rare station keeping maneuver (14199, 1-58). When Molniya 1-58 was launched, its orbital plane with respect to the sun and moon caused perigee to decline, necessitating a rare station keeping maneuver to adjust the orbit [26]. The adjustment eventually brought perigee height to 430 km, a condition which made it a less than favorable candidate for studying catastrophic decay orbits.

1.5.1 Molniya Spacecraft

The first satellite to don the Molniya designation, Molniya 1-1, was launched from Tyuratam on April 23, 1965. Over the past 30 years of service, the Molniya spacecraft have undergone significant changes and improvements, but the basic satellites parameters have remained relatively constant. There are two types of Molniya satellites, including the older block 1 satellites and the newer Molniya 3 satellites. The block 1 models were the only ones studied in this analysis and the corresponding spacecraft information for these satellites is outlined in Table 1.2.

Table 1.2: Molniya 1 Parameters [15], [26], [27]

Mass	1600 kg
Length	3.4 m
Diameter	1.6 m
Transponders	1-3
Transmission Frequency	Nominal bandwidth of 15 MHz in the 800 – 1000 MHz range.
DC Power	BOL: 1000 W; Solar arrays; batteries unknown
Stabilization	Spin?
Attitude Control	Gyrostabilizer, cold gas jets
Orbit Configuration	Eight satellites in orbital planes, node spaced 45° apart
Mission	Real-time relay of Governmental and military communications
Design Life	46 months

Figure 1.9 is a technical drawing of the Molniya 1 with dimension approximations, and Figure 1.10 is an actual photograph of the spacecraft.

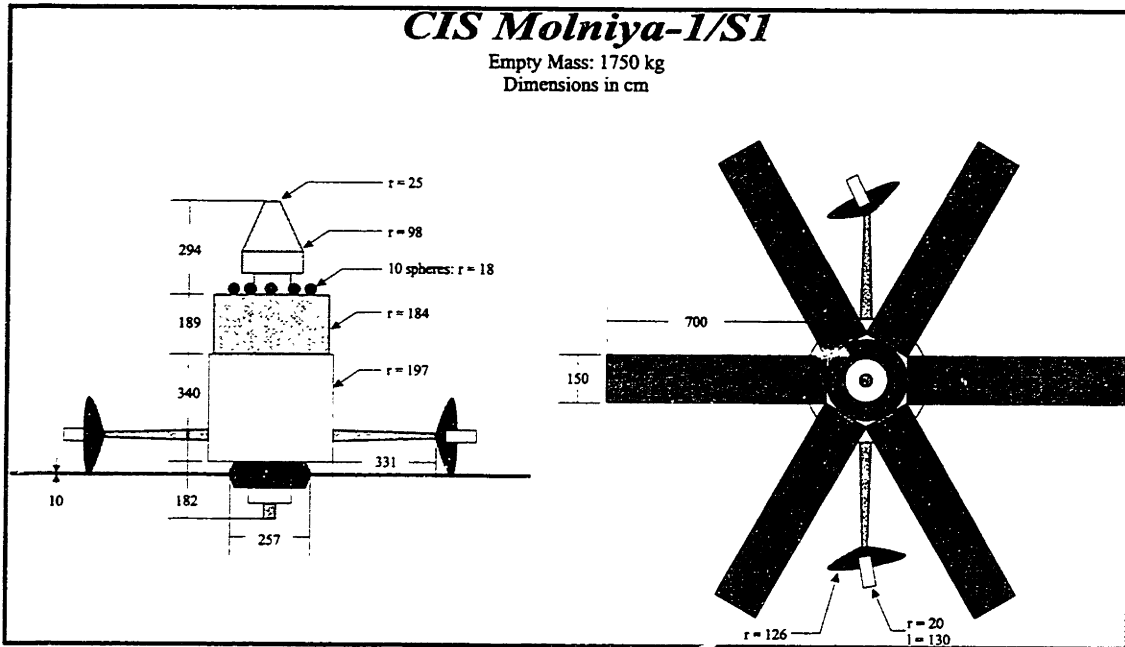


Figure 1.9: Molniya-1/S1 Technical Drawing [28]

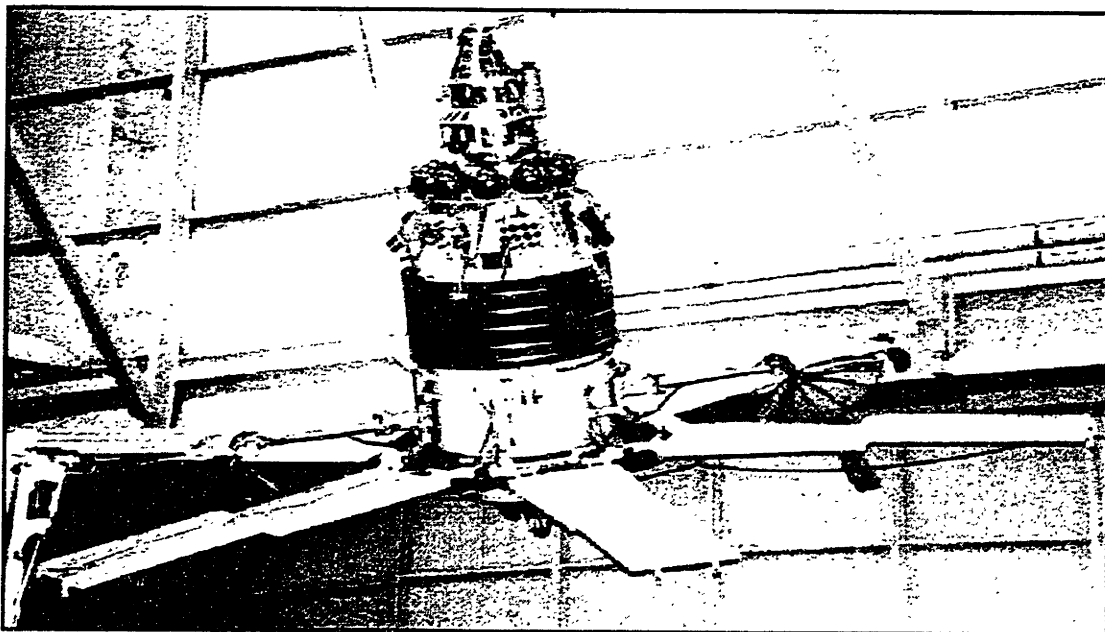


Figure 1.10: Molniya 1 Picture [29]

Another interesting characteristic of the Molniya orbital design is that each Molniya 1 is placed in its orbital plane to insure that every network satellite has identical ground track over the Earth each day at three hour intervals. As a result, the required tracking resources for the Орбита (Orbita) tracking stations are greatly reduced [26].

After a short-lived Molniya 2 program, the Molniya 3 class satellite made its debut in 1974 [15]. In contrast to the primarily governmental and military communications missions characteristic of the Molniya 1 satellites, Molniya 3 are typically used for television and communications. The most distinguishing feature between it and the Molniya 1 is a geometrically different and smaller solar configuration which can be seen in Figure 1.11. The Molniya 3's are located in four orbital planes spaced 90° apart which are coincident with four of the Molniya 1 orbital planes (to minimize tracking requirements) [15]. Due to the later inception of the Molniya 3 program, fewer have decayed, making it a less desirable test subject than even the Molniya 1, and much less desirable than the simple SL-6 rocket body.

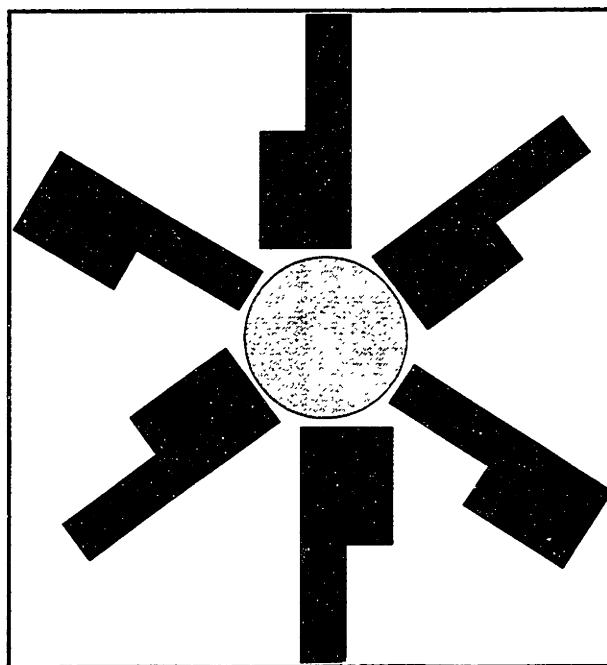


Figure 1.11: Molniya 3 Solar Array Configuration

1.5.2 Molniya Rocket Body

Derived from a Soviet ICBM, the Semiorka RD-7, the 4-stage Molniya launch vehicle known as the 8K78M (A6e-SL6) is capable of placing 1.6 tons into a Molniya orbit [27]. Since entering service in 1961, the Molniya launch vehicle has supported over 300 space missions and, despite early failures, has been extremely reliable during the past 25 years [30].

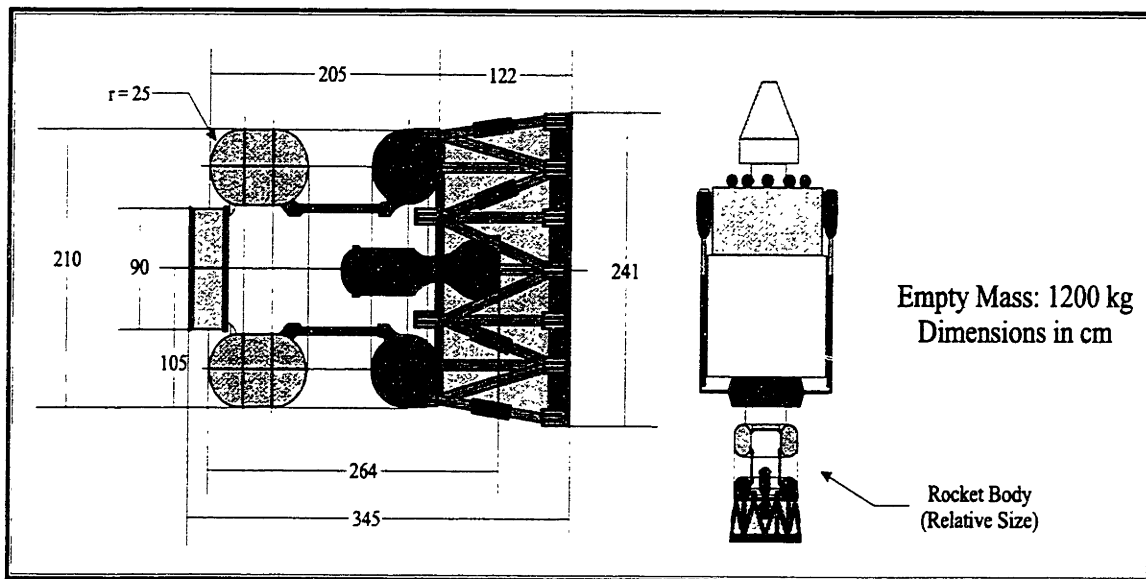


Figure 1.12: Molniya A6e-SL6, Block L Rocket Body Technical Drawing [30]

The final stage, known as the Block L third core stage, is of particular interest due to the fact that it remains in a Molniya orbit after delivering the satellite. In a similar orbit, the Block L allows for an analysis of the catastrophic decay orbit without the atmospheric interactions typical of the Molniya spacecraft's more complex structure. Figure 1.12 is a technical drawing which clearly show the simple, cylindrical structure. Figure 1.13 is a picture of the rocket body, and Table 1.3 is a listing of all known (to the author) characteristics of the Block L rocket body. Without the stochastically variable perturbations that may be caused by structures such as the solar arrays, the Block L is much simpler to model and therefore employed in the majority of this analysis.

Table 1.3: Molniya A6e-SL6, Upper Stage Block L Parameters [30]

Main Engine	1 main nozzle: 11D33
Length	2.41 m
Max Diameter	2.64 m
Empty Mass	1,200 kg (including 330 kg for a solid propellant BOZ launching provision block which is ignited prior to main engine start to settle propellants, then ejected)
Reentry Mass	870 kg (Empty – BOZ)
Fuel Mass	1,000 kg
Oxidizer Mass	3,000 kg
Fuel	Kerosene
Oxidizer	Liquid Oxygen
Propellant Mass Fraction	0.8327 (with BOZ)
Thrust	6,900 kg
Specific Impulse	346.6 sec (vacuum)
Typical Burn Time	150-200 sec

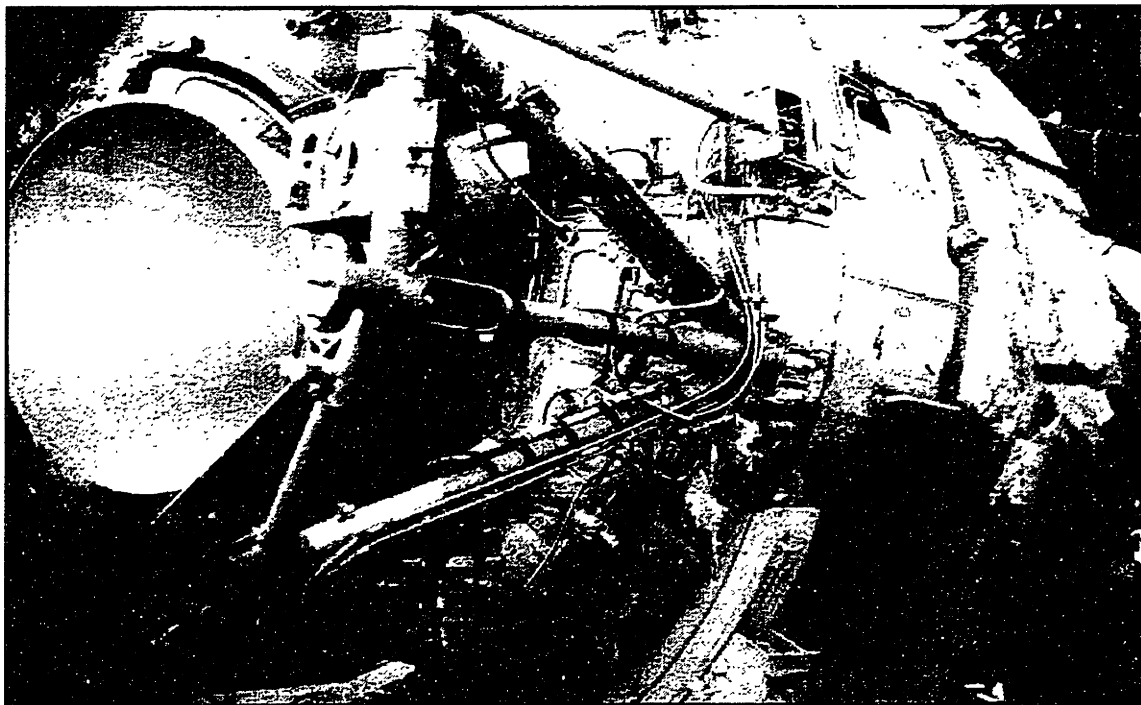


Figure 1.13: Molniya A6e-SL6, Block L Rocket Body Picture [31]

1.6 Thesis Overview

The remainder of this thesis covers the theory, process, testing, and results associated with incorporating various model improvements into R&D GTDS and with analyzing the decay environment in order to efficiently and accurately process data from catastrophic decay orbits and propagate position into the future to impact.

Chapter 2 is an overview of the various orbit propagation and prediction methods. Propagation, the process of moving the satellite ephemeris to a given epoch, is a pivotal part of this thesis due to modeling differences. As a result, both special and general perturbation techniques are covered, with special emphasis on the Draper Semi-Analytic Satellite Theory (DSST). Chapter 2 also includes a brief introduction to current impact prediction methods which will be utilized in Chapter 7 to estimate impact points of the reentering objects.

Chapter 3 covers the major classes of statistical orbit determination methods. The ability to overcome process noise, measurement errors, and other error sources is also a vital part of this thesis, thus necessitating an investigation of various orbit determination methods. Least squares, sequential batch least squares, and Kalman filtering are all covered, with an emphasis on the Kalman filter. This emphasis is due to the dynamic nature of the catastrophic decay orbits, for which the Kalman filter is ideally suited.

Chapter 4 introduces R&D GTDS, the orbital analysis tool used for this thesis. GTDS offers a multitude of orbital analysis tools in a flexible environment, a crucial attribute for the dynamic nature of the catastrophic decay problem. After a brief discussion of GTDS developmental history, there is a listing of available tools and an overview of program operation.

Chapter 5 is an initial analysis of the Mission Phase. Various perturbations identified by Fieger [19] are isolated individually to identify the main source of error - the

truncation of higher order J_2^2 terms. Due to time constraints, a closed form solution of the J_2^2 problem became improbable, but initial findings are presented in an attempt to trigger intellectual stimulation.

Chapter 6 is an in-depth analysis of the Decay Phase, which requires more model improvements than the Mission Phase. A detailed description of the inclusion of a new atmosphere model, the MSISE-90 model, is included as well as a description of various Decay Phase perturbations, and techniques for accounting for un-modeled perturbations with error modeling.

Chapter 7 delineates the testing and results compiled for the decay phase, including an initial attempt to process the data using a least squares batch filter, and Kalman filter runs. The Kalman filter results include runs with both the time dependent error function and the altitude dependent error function, as well as inclusion of a simplified lift model. Finally, a listing of impact predictions (time and locations) for the NSSC-13390 rocket body is presented.

Chapter 8 summarizes the conclusions resulting from this work, and outlines ideas and suggestions for further investigation.

Appendix A introduces the techniques used to process real data for analysis of these environments with several utilities developed during this investigation. Appendix A includes user guides for each utility and brief explanations of both their coding and input/output structure.

Appendix B is a compilation of the element histories, observation density, single orbit observation density, and observation gap analyses plots for each analysis object, primarily NSSC-13390.

Appendix C outlines the extensive GTDS modifications made for this study. Control card modifications and additions are listed in addition to specific code modifications pertaining to the MSISE-90 atmosphere model, filter input processor, filter run-time modifications, Simplified Lift Model modifications, error function modifications (for both time and altitude dependent models), and other miscellaneous modifications.

Appendix D presents select GTDS Control cards for sample runs of the batch, filter, and permanent file report runs with various perturbations, error functions, and lift options. Due to the large number of control cards required for this study, only representative cards are presented.

Appendix E is an overview of the Macsyma utilities used in the J_2^2 investigation. The new Macsyma execution blocks developed during this investigation are presented.

Chapter 2

Orbit Propagation and Impact Prediction Methods

2.1 Introduction

This thesis is designed to be a self-contained document, meaning that a reader with a sufficient mathematics background should be able to read through only what is contained within these pages to gain an understanding of what has been accomplished. In order to achieve this self-containment, it is necessary to provide the reader with some background material in the next few chapters on orbit propagation and determination, impact prediction methods, and the Research and Development version of the Goddard Trajectory and Determination System (R&D GTDS). This chapter will introduce the reader to Orbit propagation concepts, most notably an in-depth analysis of DSST, as well as a tertiary overview of current impact prediction methods.

2.1.1 Orbit Generators

Webster's Dictionary defines a perturbation as a "disturbance of motion: a disturbance of the regular and usual elliptic course of motion of a celestial body that is produced by some force additional to that which causes its regular motion." [33] Simply stated, things don't always go according to plan, and when dealing with orbital motion, things don't always go according to two body motion. The restricted two body equation

of motion, Equation (2.1), represents the motion of one body, with respect to another, taking into account only the gravitational attraction of the two bodies. The quantity μ represents the gravitational parameter of the two bodies and the vector \mathbf{r} is the vector between the two bodies.

$$\ddot{\mathbf{r}} + \mu \frac{\mathbf{r}}{r^3} = 0 \quad (2.1)$$

By neglecting perturbations caused by other bodies (for instance the moon or sun in the case of an artificial earth satellite), and other forces not considered in Keplerian motion (such as drag, nonspherical earth, etc.), real-world trajectory problems cannot be solved accurately with restricted two body theory [34]. Perturbations can cause deviations in the two-body orbital path that can sometimes be very significant. In fact, perturbations such as drag during reentry, may actually become larger than the force of gravitation and change the dynamics of the problem altogether.

Orbit generation, or perturbation, techniques are defined as the propagation of an orbit when faced with real-world perturbations. These techniques can be classified into two broad categories: special perturbations and general perturbations. Special perturbation techniques use numerical integration with the equations of motion including all necessary perturbation accelerations, while general perturbation techniques use analytic integration of series expansions of the perturbing accelerations [34]. A third method to be considered, semi-analytic techniques, offer a unique blend of special and general perturbation concepts.

Special perturbation orbit generators result from direct numerical integration of the equations of motion, represented as either a first or second order system. With these techniques, it is easy to add a new force model to the equations of motion by simply expressing the new perturbing acceleration in terms of time and the object's position and velocity. In addition, using a sufficiently small step size for integration, numerical

methods effectively model short periodics (periodic perturbations with cycle less than one orbital period). The biggest advantage of numerical methods is the ease of formulation and implementation (particularly for drag and solar radiation pressure), which allows an analyst to add a large number of perturbations and create a comprehensive model with relative ease.

Unfortunately, numerical (special perturbation) techniques also have several weaknesses. In the past, the primary problem with these techniques was that they were generally slower, and more demanding on computer resources than their analytic counterparts. Although current computing power and parallel processing techniques have made this disparity less of a concern, numerical techniques still suffer from errors that build up as a result of truncation errors (due to fixed computer wordlengths) which cause numeric solutions to degrade as the propagation interval increases [2]. The truncation error is countered by the error due to the length of the integration step size, which ideally is as small as possible to account for such things as short periodics. The smaller the step size, the better the integrator models the motion, at the cost of increasing truncation error. Several methods exist for determining the optimum step size to minimize these errors (refer to Vallado [2] for a comprehensive description), as well as methods for varying the step size during the orbit known as regularization which will be discussed in section 2.2.

General perturbation techniques (or analytical techniques), use analytical approximations to the equations of motion to allow for analytic integration while maintaining the essential character of the orbital motion, where truncated series expansions of the modeled perturbing accelerations form the basis of these analytical approximations. These theories are usually much faster than their numerical counterparts, but are typically more difficult to model, harder to program and decipher (perturbations are embedded in the solution), and are typically not as accurate. It should be noted, however, that the accuracy of analytic techniques is directly proportional to the

refinement of the given theory. For example, Andre Deprit and Arnold Rom, in their paper “The Main Problem of Satellite Theory for Small Eccentricities,” achieved impressive accuracy (12 significant figures for the integrals of energy and angular momentum) [35]. These theories have enjoyed wide use in the past due to limited computer resources, a condition which is rapidly changing. The one characteristic of these theories that should secure their future in a computer-driven future, is that they typically lend themselves to a better understanding of the perturbation source [2]. This is due to the fact that numerical techniques typically use position and velocity vectors for their integration, as opposed to analytic theories which usually express orbital motion in terms of orbital elements. Since most orbital elements undergo small variations (compared to position/velocity vectors), these elements make it simpler to notice changes in the orbit characteristics over time.

Since the mid 1960’s, NORAD has followed a two-pronged approach to orbit propagation by including both numerical and analytical techniques. Due to the increased computer resources required for numerical propagators, they have ordinarily been used only for decaying or other high priority objects. In contrast, the analytic propagators are used for the vast majority of the tracking done by NORAD. This two-pronged approach left NORAD with the choice of overtaxing their computers (with numeric propagators) or having low accuracy predictions (using analytic methods) for the preponderance of objects that they track.

In 1972 the question was posed, “What if analytic perturbation equations were integrated numerically?” By doing so, the advantages of numerical integrators and analytic theories are combined, and almost all disadvantages are overcome. The Draper Seminanalytic Satellite Theory (DSST), the result, is a highly accurate orbit propagator that efficiently uses computational resources, especially for long arcs. DSST achieves these results by separating long (cycle greater than one orbital period) and short periodics

which limit the integration step size for numerical methods. By removing the short periodic functions, step sizes as large as one day can be used with similar accuracy as numeric methods in a fraction of the time. DSST also represents a compromise with respect to implementation difficulty, when compared to numeric and analytic techniques. Despite the complexity involved in improving the DSST model, DSST's speed and efficiency made it the logical choice for modeling long arcs for the highly eccentric orbits being analyzed and as such, section 2.4 includes an in-depth investigation of the formulation of DSST.

2.1.2 GTDS Integration Methods

Once the optimal orbit generation technique for a specific problem has been chosen, an integration scheme must also be adopted. For most analytic techniques, the given theory dictates the integration method, but for numeric and semi-analytic techniques, a suitable numerical integration scheme must be selected. The two main types of numerical integration methods are the Single-step algorithms and the Multi-step or predictor-corrector algorithms.

Single-step methods update the satellite state vector using rates at several other times based on a single state value at a given epoch. The rates are readily obtained, and the integrator is self-starting, meaning that it doesn't require a sequence of back values to start the integrator [2]. Multi-step methods can improve on the accuracy and efficiency of single-step methods, but they do require back values (are not self-starting). Multi-step methods calculate an initial estimate, or predictor, using previous estimates of the function's rate of change. The method then completes a second series of calculations, known as the corrector, which uses the predictor calculations to further refine the result [2]. When comparing the two, multi-step methods are generally more accurate, efficient,

and complex than their single-step counterparts. Roy indicates that a typical multi-step algorithm (Gauss-Jackson) is nearly an order of magnitude more efficient than a comparable single-step algorithm (4th Order Runge-Kutta) [36]. For more difficult orbits, such as the highly eccentric, catastrophic decay orbits dealt with in this thesis, the disparity between the two methods is considerably smaller due to the complexity of multi-step methods [2].

Within the multi-step class of integration methods, a choice must be made among the following options outlined in reference [4]:

- **Type of formulation:** methods may be used which solve second order systems directly (such as Störmer's method), or which normalize the second order system into a higher dimensional first order system and use a first order formula (such as Adams-Bashforth).
- **Type of Algorithm:** several algorithms may be selected within the multistep predictor-corrector schemes including: PE (prediction only), P(EC)ⁿ, PE(CE)ⁿ, and PECE*, where P = predict, E = evaluate derivative, C = correct, and E* = pseudo-evaluate (correct or re-correct only part of the total derivative).
- **Order of Process:** various order formulas may be used in the algorithm. The higher order the formulas are, the more accurate and less stable they are.
- **Stepsize control:** since the orbit dynamics may undergo large variations during a revolution (such is the case with the highly-eccentric decay orbits), some algorithm must be adopted to allow stepsize variations. One method, that of time regularization, is discussed in Section 2.2.

GTDS uses primarily the multi-step Störmer-Cowell and Adams methods for numerical integration based on several efficiency studies ([37], [38]) which indicated they had the optimal blend of high accuracy and efficiency when compared to popular integration schemes of the time. A single-step 4th Order Runge-Kutta scheme is also used with sequential estimation and as a starter for certain multi-step processes. Tables 4.1 and 4.2 in Chapter 4 list the various GTDS orbit generators and the integration methods used with them. The reader should note that the Störmer-Cowell and Adams methods are the primary numerical integrators used. With Cowell's method, described in Section 2.2 for example, the three class II differential equations which are solved for the position vector use the Störmer-Cowell integration formulas, and the three class I equations which are solved for the velocity vector use the Adams integration formulas. For a detailed derivation of these formulas, and other numerical integration schemes in GTDS, please refer to reference [4].

2.1.3 Orbital Elements and Reference Frames

A fundamental step in any scientific analysis is to identify a set of parameters to describe the behavior of the object being analyzed. In the case of orbital propagation for an artificial earth satellite, the object being analyzed is the location and motion of the satellite in its orbital path, and the set of parameters which describe it are orbital elements. Six quantities are necessary to define the state or position of a satellite in a specific orbit with respect to the given coordinate system [39], and these quantities are known as orbital elements. Perhaps the most well known set of orbital elements were developed by Johannes Kepler and are referred to as Keplerian elements or Classical Orbital Elements (COE). These elements, which are discussed in Section 2.1.3.1, suffer

from singularities for certain orbit types. Another element set, the equinoctial elements discussed in Section 2.1.3.2, do not share these singularities and due to this characteristic, are used in the formulation of DSST in section 2.4.

In conjunction with the orbital elements, an inertial reference frame is a vital part of any orbital analysis. For artificial earth satellites, an earth-centered, or geocentric, reference frame is generally adopted. The two fundamental coordinate systems variants are the ecliptic and equatorial systems. The fundamental plane in the ecliptic system is the plane of the earth's orbit. For the equatorial system, the equatorial plane is the fundamental plane. For satellite motion about the earth, the geocentric-equatorial system is generally used, and is adopted for this analysis. The geocentric-equatorial frame has its origin at geocenter and a reference (\mathbf{i}_x) direction towards the vernal equinox, or a point of intersection of the two fundamental planes where the sun crosses the equator from south to north in its apparent annual motion along the ecliptic [39]. The \mathbf{i}_z axis is normal to the equatorial plane (positive in the northern direction), and the \mathbf{i}_y axis completes a right handed system.

Draper R&D GTDS supports a large number of geocentric-equatorial reference frames such as mean equator and equinox of fundamental epoch, true equator and equinox of reference time/date, and body fixed. For this analysis, however, only the Mean of 1950 reference frame was used to standardize results and promote interoperability of results among various versions of GTDS. Mean equator and equinox of fundamental epoch reference frames are inertial coordinate systems referenced to specific time; for the Mean of 1950 frame that specific time is January 1, 1950 (Julian Date 2433282.423 [40]). Please reference Carter's [40] thesis for a more detailed discussion of the reference frames and the inclusion of the J2000 frame into the VAX version of R&D GTDS. An unfortunate consequence of the multi-platform development of GTDS is that

functionality improvements are isolated to the development platform (any future initiative to consolidate GTDS models and improvements to one cross-platform version would be tremendously beneficial).

2.1.3.1 Keplerian Elements

All earth satellites have orbits which can be described with an ellipse. The size, shape, and orientation of that ellipse, as well as the position of the satellite in that ellipse, are described by six orbital elements. The Keplerian, or classical orbital elements are listed in Table 2.1, and represented graphically in Figure 2.1. The *semi-major axis*, a , describes the semi-major axis of the orbital ellipse and therefore the size of the orbit. *Eccentricity*, and therefore the shape of the ellipse is described by the variable e . The angle between the normal of the orbital plane (which is also the positive direction of the angular momentum vector - \mathbf{i}_h) and the positive direction of the z axis, is described by the angle of *inclination*, i . The line of intersection of the orbital ellipse and the equatorial plane is known as the *line of nodes*, and the ascending node is the point at which the satellite crosses the equatorial plane on the line of nodes (\mathbf{i}_n) with a positive component of velocity in the z direction [39]. The *longitude of ascending node*, Ω , is the angle from the \mathbf{i}_x direction (vernal equinox) to the ascending node. The line through the origin (occupied focus of the orbital ellipse, in this case the geocenter) and the perigee (point of closest approach) is known as the *line of apsides* (\mathbf{i}_p). It should also be noted that the point of furthest approach, the apogee, will occur at the intersection of the line of apsides with the orbit on the other side of the semi-major axis from the perigee. The *argument of perigee*, ω , is the angle measured from the line of nodes to the line of apsides. The sixth orbital

element is the *true anomaly*, v , which is the angle between the line of apsides and the satellite measured along the orbital path.

Table 2.1: Keplerian Orbital Elements

a	Semi-major axis
e	Eccentricity
i	Inclination
Ω	Longitude of ascending node
ω	Argument of perigee
v	True anomaly
M	Mean anomaly
ϖ	Longitude of perihelion
θ	Argument of latitude
L	True longitude
E	Eccentric anomaly
F	Eccentric longitude
λ	Mean orbital Longitude

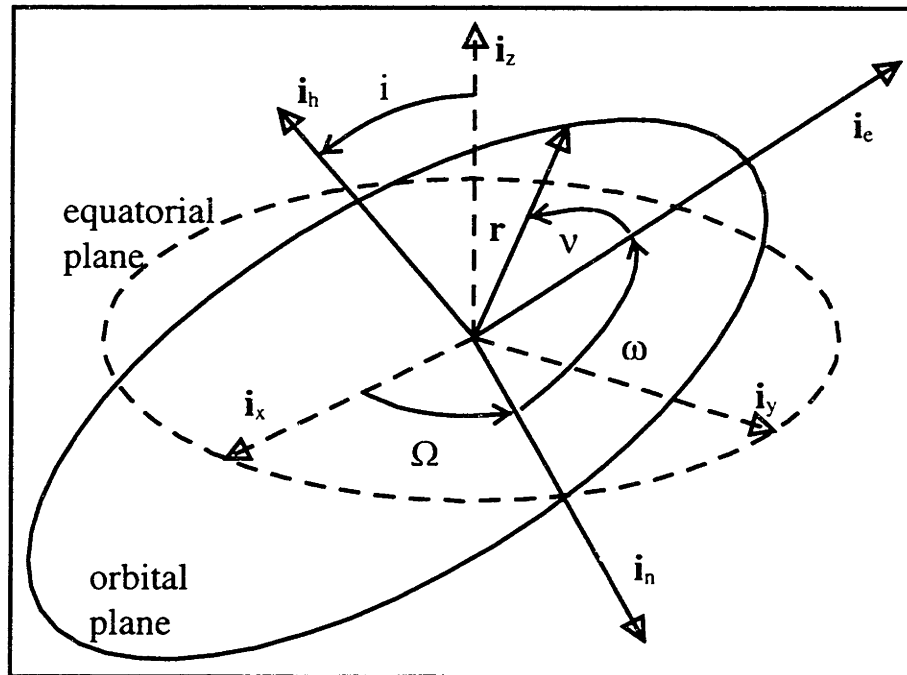


Figure 2.1: Keplerian Elements [41]

The additional elements listed in the table are also useful in describing the orbit. The *mean anomaly*, M , can be used to describe the location of the satellite in the orbit as

a function of time since perigee passage and is defined by Equation (2.2), where n is the mean motion (average angular rate of the object about the earth), and τ is the time of perigee passage [41].

$$M = \sqrt{\frac{\mu}{a^3}} (t - \tau) = n \Delta t \quad (2.2)$$

Figure 2.1 offers no geometric interpretation of the mean anomaly, but if the reader looks at Figure 2.2, where an auxiliary circle of radius 'a' has been drawn around the orbit, the mean anomaly corresponds to uniform angular motion on the auxiliary circle during the prescribed time period. Another angle of interest in Figure 2.2 is the *Eccentric anomaly*, E , was defined by Kepler as the angle $\angle QCA$, where Q is the point where the perpendicular of the major axis cuts the *auxiliary circle* as it passes through the satellite position. In Figure 2.2, C is the center of the orbital ellipse, and A is the end of the major axis at the perigee.

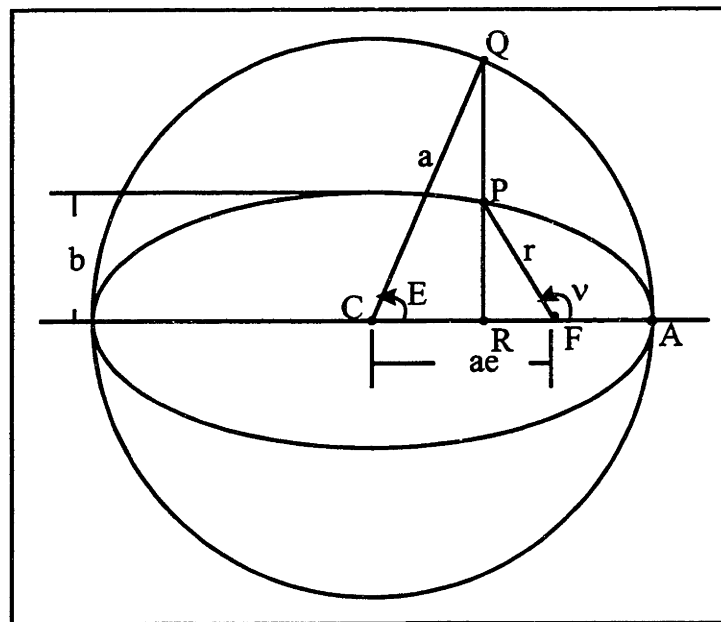


Figure 2.2: Orbital Anomalies for Elliptic Motion [39]

Sometimes called alternate or nonsingular elements, the remaining members of Table 2.1 represent angles that can substitute for the orbital elements in certain situations. The *longitude of perihelion* is denoted by the sum of the longitude of ascending node and the argument of perigee, $\varpi = \Omega + \omega$, and is not an ordinary longitude because it is the sum of angles in two different planes (unless inclination = 0° or 180°). The other alternate elements are also angles which represent sums of angles, including the: *argument of latitude*, $\theta = \omega + \nu$ (used when eccentricity = 0 and no perigee), the *true longitude*, $L = \varpi + \nu$ (used when $i = e = 0$ so there's no perigee or ascending node), and the *eccentric longitude*, $F = \varpi + E$ (defined unless inclination = 0). Mean orbital longitude, λ , will be described in Section 2.1.3.2.

As previously stated, the Keplerian element set experiences two significant singularities. One is caused by circular orbits, when the eccentricity is near zero, thus leaving the line of apsides undefined. The other singularity occurs with equatorial orbits, when the inclination is near zero, leaving the line of nodes undefined. It is because of these singularities that designers of DSST sought a more general set of elements. The set that was chosen as a singularity-free set of orbital parameters, was the equinoctial element set discussed next.

2.1.3.2 Equinoctial Elements

Originally developed by Lagrange, equinoctial elements do not fall prey to the singularities which plague the Keplerian elements. The following explanation of equinoctial elements is taken from McClain's *A Semianalytic Artificial Satellite Theory* [42] with slightly different notation to maintain commonality. Equations (2.2a-g)

describe the definition of the equinoctial elements in terms of the Keplerian elements, and Equations (2.3a-f) describe the reverse transformation of the two sets.

$$\begin{aligned}
 a &= a \\
 h &= e \sin(\omega + I\Omega) \\
 k &= e \cos(\omega + I\Omega) \\
 p &= \tan^{-1}\left(\frac{i}{2}\right) \sin \Omega \\
 q &= \tan^{-1}\left(\frac{i}{2}\right) \cos \Omega \\
 \lambda &= M + \omega + \Omega
 \end{aligned} \tag{2.2a-f}$$

where I is the retrograde factor shown in Equation (2.2g), which is positive for direct or prograde orbits, and negative for retrograde orbits ($90^\circ < i < 180^\circ$):

$$I = \begin{cases} 1, & \text{for } 0 \leq i < \pi \\ -1, & \text{for } \pi < i \leq 2\pi \end{cases} \tag{2.2g}$$

$$\begin{aligned}
 a &= a \\
 e &= \sqrt{h^2 + k^2} \\
 i &= \cos^{-1}\left[\frac{(1 - p^2 - q^2)I}{1 + p^2 + q^2}\right] \\
 \omega &= \tan^{-1}\left(\frac{h}{k}\right) - I \tan^{-1}\left(\frac{p}{q}\right) \\
 \Omega &= \tan^{-1}\left(\frac{p}{q}\right) \\
 M &= \lambda - \omega - I\Omega
 \end{aligned} \tag{2.3a-f}$$

The equinoctial elements are slightly more difficult to understand graphically without the use of the equinoctial reference frame shown in Figure 2.3 (for direct orbits). The orthogonal triad of $(\mathbf{i}_r, \mathbf{i}_g, \mathbf{i}_w)$ represent a right handed system with the satellite orbital plane as the fundamental plane. The vector \mathbf{i}_r is directed toward a point in the orbit that is

displaced by the angle $-\Omega$ (Ω for the retrograde system) from the ascending node. The vector \mathbf{i}_w points toward the north equinoctial pole and is also the unit angular momentum vector. The vector \mathbf{i}_g simply forms a right-handed system with the other two vectors.

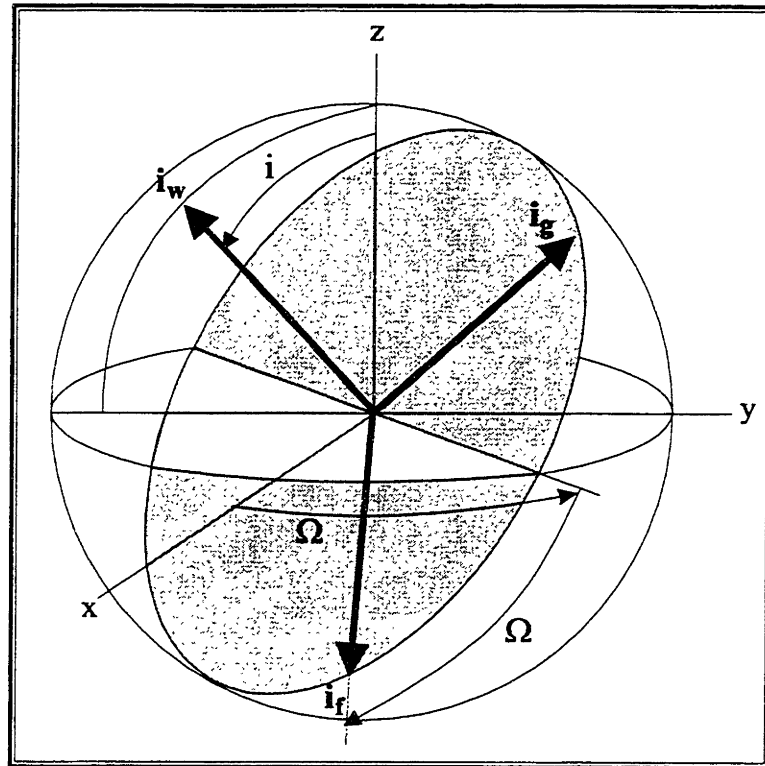


Figure 2.3: Direct Equinoctial Coordinate Frame [42]

Cefola and Broucke did significant work with the equinoctial element set to derive the partials of the position and velocity vectors with respect to the elements, state transition matrix, and the corresponding Lagrange brackets (discussed in section 2.3.1) [43]. These results were extended [44] and eventually used in the development of analytic equations necessary to build a propagator in the non-singular equinoctial elements. Others, like Yurasov, were able create propagators with similar equinoctial elements as well [24], making them useful to both special and general perturbation methods [41]. Of primary interest to this thesis, the equinoctial elements are used as a primary element set for the DSST propagator discussed in Section 2.4.

2.2 Cowell Math Techniques

Philip Herbert Cowell will always be remembered for the step-by-step numerical integration method of the planetary equations of motion in rectangular coordinates, that bears his name [39]. The integration formulas that Cowell used were first given by Gauss and adopted to the mathematical theory of Cowell's day. The method consists of integrating Equation (2.4), where \mathbf{Q} is the vector acceleration of all forces which prevent the orbit motion from being precisely described by a Keplerian orbit. Today, however, anytime Equation (2.4) is integrated numerically in rectangular coordinates, the method used is referred to as Cowell's Method.

$$\ddot{\mathbf{r}} + \mu \frac{\mathbf{r}}{r^3} = \mathbf{Q} \quad (2.4)$$

In GTDS, this set of three second order differential equations described by Equation (2.4) is solved directly for the position vector using the Störmer-Cowell numerical integration formulas discussed in Section 2.1.2. The three first order differential equations described by Equation (2.5) for the velocity vector are integrated using the Adams numerical integration formulas in the case of velocity dependent perturbations, such as atmospheric drag [4].

$$\frac{d\dot{\mathbf{r}}}{dt} + \mu \frac{\mathbf{r}}{r^3} = \mathbf{Q} \quad (2.5)$$

These equations, expressed in vector format, are given in Equations (2.6a) and (2.6b), where \mathbf{X} is the state vector [2].

$$\mathbf{x} = \begin{bmatrix} r_i \\ r_j \\ r_k \\ v_i \\ v_j \\ v_k \end{bmatrix} \quad \dot{\mathbf{x}} = \begin{bmatrix} v_i \\ v_j \\ v_k \\ -\frac{\mu r_i}{r^3} + a_{d_i} \\ -\frac{\mu r_j}{r^3} + a_{d_j} \\ -\frac{\mu r_k}{r^3} + a_{d_k} \end{bmatrix} \quad (2.6a-b)$$

The reader must also note the importance of the stepsize in the integration of the above equations. As discussed in Section 2.1.2, the determination of the stepsize is critical to the accuracy and efficiency of the integration process. This is due to the fact that the integration process is aided by the uniformization of the local error at each integration step. This uniformity is achieved for Equations (2.4) and (2.5) with respect to the dominant local error sources, generally the oblateness error [4]. Due to a relatively constant mean motion, circular orbits are inherently uniform by using a constant time step. For eccentric orbits, such as the focus of this analysis, the dominant local error sources are not evenly distributed about the orbit. For example, catastrophic decay orbits may have drag force perturbations around perigee that are nearly as strong as the gravitational potential. If a constant time step were used, the stepsize at apogee would be too small to achieve optimal efficiency, and the stepsize at perigee would be too large to accurately model the motion. One method of correcting for this irregularity in the dispersion of perturbation forces is to reformulate the Cowell equations in terms of a new variable s , defined by a form of the Sundman, or Regularization Transformation given in Equation (2.7) [4], [39], where n is the uniformization constant:

$$\frac{ds}{dt} = \frac{\sqrt{\mu}}{r^n} \quad (2.7)$$

GTDS uses a value of 3/2 for n, which uniformizes the local error with respect to oblateness effects, and results in the Time Regularized Cowell equation of motion given in equation (2.8a), where prime indicates differentiation with respect to the independent variable s:

$$\mathbf{r}'' = n \left(\frac{r' \mathbf{r}'}{r} \right) - r^{(2n-3)} \mathbf{r} + \frac{r^{2n}}{\mu} \mathbf{Q} \left(t, \mathbf{r}, \frac{\sqrt{\mu}}{r^n} \mathbf{r}' \right) \quad (2.8a)$$

As with the standard Cowell formulation in GTDS, the position vector is found by integrating Equation (2.8a) with the Störmer-Cowell formulas and the velocity vector is found by integrating the equation with the Adams formulas. However, since the velocity appears explicitly in (2.8a), the velocity equation must be integrated even in the case of velocity free perturbations. In addition, the second order differential Equation (2.8b) must be integrated to properly adjust the time [4]. By doing so, the stepsize will vary to more accurately achieve uniformity of the perturbations around elliptic orbits. Time Regularized Cowell is used extensively in Chapter 8 for propagating final decay of the Molniya orbits due to this analytic stepsize regulation, and the improved performance it provides.

$$\dot{t}'' = \frac{nr^{2n-1} \dot{r}}{\mu} \quad (2.8b)$$

2.3 Variation of Parameters

Developed by Euler in 1748, the Variation of Parameters (VOP) method of perturbations differs from Cowell's method in a fundamental way. Whereas Cowell's method is concerned with finding the position and velocity, or coordinates, VOP calculates the orbital elements (or any other set of parameters which sufficiently describe the satellite's position in a specific orbit) [34]. The fundamental principle of the VOP

method is to find the time variations due to the perturbations of a given set of parameters. This is accomplished by finding analytic expressions for the time rate of change of the parameters in terms of the perturbations and integrating them numerically (special perturbations) or analytically (general perturbations), to find their values at a later time [34].

In order to fully understand the VOP method, in particular how it applies to the development of the DSST, the following derivation is presented. This derivation is taken from reference [42] and personal discussions with Paul Cefola, Ron Proulx, and Wayne McClain. Any number of other references are available for the VOP method ([2], [34], [36], [39], etc.), but the following has been adopted to maintain commonality in notation throughout this Section and Section 2.4.

Starting with Equation (2.4), we first need to admit a simple solution of the form (2.9a) for position where a_k represents the time dependent parameters (such as the equinoctial elements). These elements represent an orbital ellipse at a given point of time. Since this ellipse “kisses” the actual satellite trajectory represented by a space curve in 3-D space (consisting of the locus of points describing the position as a function of time), they are called *osculating elements*. To illustrate, if the perturbations vanished altogether, the satellite would follow the 2-body equation of motion of the ellipse defined by the osculating elements at that instant. The analog of Equation (2.9a) for velocity is found through use of the chain rule for differentiation.

$$\mathbf{r} = \mathbf{f}(a_1, a_2, \dots, a_6, t) \quad (2.9a)$$

$$\dot{\mathbf{r}} = \frac{\partial \mathbf{r}}{\partial t} + \sum_{j=1}^6 \frac{\partial \mathbf{r}}{\partial a_j} \dot{a}_j \quad (2.9b)$$

Even though the time dependent parameters, a_k , are not constants (which is why the parameter rates vanished for the unperturbed case), one can still constrain the sum (2.9c) of Equation (2.9b) to zero to simplify the equation. This simplification will again be taken into consideration in Section 2.3.2 for the development of the Gaussian VOP equations. Using this constraint, and differentiating once more with the chain rule, yields Equation (2.9d) for the acceleration.

$$\sum_{j=1}^6 \frac{\partial \mathbf{r}}{\partial a_j} \dot{a}_j \equiv 0 \quad (2.9c)$$

$$\sum_{k=1}^6 \frac{\partial \dot{\mathbf{r}}}{\partial a_k} \dot{a}_k = \mathbf{Q}(\mathbf{r}, \dot{\mathbf{r}}, t) \quad (2.9d)$$

By plugging (2.9c) back into (2.4) and comparing it with the 2-body equation of motion (2.1), we can solve for the perturbation vector (which is a function of position, velocity, and time).

$$\ddot{\mathbf{r}} = \frac{\partial^2 \mathbf{r}}{\partial t^2} + \sum_{k=1}^6 \frac{\partial \dot{\mathbf{r}}}{\partial a_k} \dot{a}_k \quad (2.10)$$

Therefore, the perturbed equations of motion differ only by the terms which contain the time dependent parameters. Since \mathbf{Q} is now defined as a linear combination of the element rates, VOP equations are sought in the following form where the functions G_i are selected to depend either on the state or elements a_k .

$$\frac{da_i}{dt} = G_i \quad (2.11)$$

Two important forms of these differential equations are the Lagrange Planetary Equations (Section 2.3.1) and the Gaussian VOP Equations (Section 2.3.2). The difference between the two is in the types of forces to which they apply. The Lagrange Planetary Equations are applicable to conservative perturbing forces (forces which are

independent of path), while the Gaussian equations are applicable to both conservative and nonconservative forces. *Conservative forces* have a constant total energy, whereas *nonconservative forces* may lose or gain energy through interactions with external sources [2]. An example of a conservative force might be gravitational effects, and an example of a nonconservative force might be thrust, drag, lift, or solar radiation pressure.

2.3.1 Gaussian VOP Equations

The first step in the formulation of the Gaussian VOP equations is a ‘weighted’ sum the partials with respect to velocity of Equation (2.9d) with the partials with respect to position of Equation (2.9c) to arrive at Equation (2.12).

$$\sum_{k=1}^6 \left(\frac{\partial a_j}{\partial \dot{\mathbf{r}}} \cdot \frac{\partial \dot{\mathbf{r}}}{\partial a_k} + \frac{\partial a_j}{\partial \mathbf{r}} \cdot \frac{\partial \mathbf{r}}{\partial a_k} \right) \dot{a}_k = \frac{\partial a_j}{\partial \dot{\mathbf{r}}} \cdot \mathbf{Q} \quad (2.12)$$

Since the elements are mutually independent, and are functions of only position and velocity, a Jacobian and identity map can be utilized to relate their partials to the Kronecker delta function $\delta_{j,k}$ (1 if $j = k$, 0 otherwise). By defining an identity function and taking the Jacobian (first derivative matrix of a given number n of functions with respect to n variables \mathbf{x}) of the identity function, the result will be the identity matrix, as shown in Equation (2.13a).

$$\begin{aligned} \mathbf{f}(\mathbf{x}) &= \mathbf{x}, \quad \mathbf{x} = (x_1, x_2) \\ J(\mathbf{f}, \mathbf{x}) &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{I} \end{aligned} \quad (2.13a)$$

Now define the vector \mathbf{a} as the set of elements, and a function of position and velocity, it can be mapped directly to a specific set of \mathbf{a} in an identity map given in (2.13b).

$$\mathbf{a} = \mathbf{a}(\dot{\mathbf{r}}(\mathbf{a}), \mathbf{r}(\mathbf{a})) \quad (2.13b)$$

When the Jacobian of the identity map given in (2.13b) is taken, the result is an identity matrix as in (2.13a). Using the chain rule to expand this expression in terms of position and velocity, and recognizing that the identity matrix can be represented with the Kronecker delta function, Equation (2.13c) is obtained.

$$\frac{\partial a_j}{\partial \dot{\mathbf{r}}} \cdot \frac{\partial \dot{\mathbf{r}}}{\partial a_k} + \frac{\partial a_j}{\partial \mathbf{r}} \cdot \frac{\partial \mathbf{r}}{\partial a_k} = \delta_{j,k}, \quad (j = 1, 2, \dots, 6) \quad (2.13c)$$

Substitution of (2.13c) back into (2.12) yields (2.14a), or alternately, (2.14b):

$$\sum_{k=1}^6 \delta_{j,k} \dot{a}_k = \frac{\partial a_j}{\partial \dot{\mathbf{r}}} \cdot \mathbf{Q} \quad (2.14a)$$

$$\dot{a}_j = \frac{\partial a_j}{\partial \dot{\mathbf{r}}} \cdot \mathbf{Q}, \quad (j = 1, 2, \dots, 6) \quad (2.14b)$$

The Gaussian VOP equations offer two important advantages over the Lagrange equations. Primarily, these equations are open to both conservative and non-conservative forces. They also allow for closed form expressions of the osculating element rates due to a formulation of the G functions in terms of the components of acceleration directly (Lagrange modeled conservative perturbation forces as the gradient of a potential function, which explains the restriction to only conservative forces). However, the Gaussian form also suffers weaknesses that may become important for this problem. Namely, time consuming conversions from the elements to position and velocity must be applied at every integration step. In addition, periodic phenomena are not readily isolated

from the acceleration model because most perturbing accelerations are best modeled in terms of position or position and velocity rather than a Fourier series expansion, necessitating a sometimes inefficient numerical procedure to isolate specific frequencies in the motion.

2.3.2 Lagrange Planetary Equations

The Lagrange planetary equations overcome the repeated conversions between elements and position/velocity at the expense of not having closed-form expressions for the equations of motion. As previously stated, the Lagrange equations assume a disturbing potential due to a conservative force as the gradient of a potential function (\mathcal{R}).

$$\mathbf{Q}(\mathbf{r}) = \frac{\partial \mathcal{R}(\mathbf{r})}{\partial \mathbf{r}} \quad (2.15)$$

Next take the dot product of $\frac{\partial \mathbf{r}}{\partial a_j}$ with Equation (2.9d) and $\frac{\partial \dot{\mathbf{r}}}{\partial a_j}$ with (2.9c).

Using the relation described in Equation (2.16), known as the Lagrange Brackets, (2.15) becomes (2.17)

$$[a_j, a_k] = \left(\frac{\partial \mathbf{r}}{\partial a_j} \cdot \frac{\partial \dot{\mathbf{r}}}{\partial a_k} - \frac{\partial \dot{\mathbf{r}}}{\partial a_j} \cdot \frac{\partial \mathbf{r}}{\partial a_k} \right) \quad (2.16)$$

$$\sum_{k=1}^6 [a_j, a_k] \dot{a}_k = \frac{\partial \mathcal{R}}{\partial a_j} \quad (2.17)$$

With i and j each having six components, there should be 36 Lagrange Brackets, but because of the following two relations (2.18a) and (2.18b), there are only 15.

$$[a_j, a_j] = 0 \quad (2.18a)$$

$$[a_j, a_k] = -[a_k, a_j] \quad (2.18b)$$

\mathbf{L} is then defined as the matrix of Lagrange Brackets (2.19) and permits the expression of (2.17) in matrix notation (2.20).

$$\mathbf{L} = \begin{bmatrix} [a_1, a_1] & \dots & [a_1, a_6] \\ \dots & \dots & \dots \\ [a_6, a_1] & \dots & [a_6, a_6] \end{bmatrix} \quad (2.19)$$

$$\mathbf{L}\dot{\mathbf{a}} = \frac{\partial \mathcal{R}}{\partial \mathbf{a}} \quad (2.20)$$

The Lagrange Brackets depend only on the functional relationship between the elements and position/velocity vectors of the 2-body problem. As a result of this dependence, Lagrange Brackets are independent of time (2.21). The significance of this independence is that the brackets can be evaluated at any time. The brackets can also be expressed explicitly in the elements and the system of equations inverted to yield expressions for each element rate, a_k .

An alternate derivation of the Lagrange planetary equations can be obtained from the Equation (2.21) given by Broucke [45], and the relation given in (2.22) known as the Poisson Bracket. The Poisson Brackets share the same properties as the Lagrange Brackets, described in Equations (2.18a) and (2.18b).

$$\frac{\partial a_k}{\partial \dot{\mathbf{r}}} = -\sum_{j=1}^6 (a_k, a_j) \frac{\partial \mathbf{r}}{\partial a_j} \quad (2.21)$$

$$(a_k, a_j) = \left(\frac{\partial a_k}{\partial \mathbf{r}} \cdot \frac{\partial a_j}{\partial \dot{\mathbf{r}}} - \frac{\partial a_k}{\partial \dot{\mathbf{r}}} \cdot \frac{\partial a_j}{\partial \mathbf{r}} \right) \quad (2.22)$$

Expressing the Gaussian form of the VOP equations (2.1.4b) in terms of the disturbing function results in Equation (2.23). Substitution of Equation (2.21) and cancellation due to R 's dependence only on position yields (2.24).

$$\dot{a}_k = \frac{\partial a_k}{\partial \dot{\mathbf{r}}} \cdot \frac{\partial \mathcal{R}}{\partial \mathbf{r}} \quad (2.23)$$

$$\dot{a}_k = -\sum_{j=1}^6 (a_k, a_j) \frac{\partial \mathcal{R}}{\partial a_j} \quad (2.24)$$

Definition of a matrix \mathbf{P} , analogous to the Langrange matrix \mathbf{L} for Poisson Brackets, then results in the matrix form of Equation (2.24), given in Equation (2.25). The reader should note the relationship of Poisson and Langrange Brackets given in Equation (2.26). This result is derived from substituting (2.24) back into (2.17) and simplifying.

$$\dot{\mathbf{a}} = -\mathbf{P} \frac{\partial \mathcal{R}}{\partial \mathbf{a}} \quad (2.25)$$

$$\mathbf{L}\mathbf{P}^T = \mathbf{I} \quad (2.26)$$

The final result of the Poisson Bracket derivation, is a modified version of the Langrange Planetary Equations which will be used for the development of DSST in Section 2.4. For the first 5 elements ($I = 1, 2, \dots, 5$), Equation (2.27a) is used. a_6 , or ℓ , is the fast variable (rapidly rotating phase angle). It is a linear combination of the time with an element as shown in (2.28a), where n is the mean motion, and measures the angular distance of the satellite from some departure point in the orbit (this avoids the unnecessary appearance of mixed secular terms in the VOP equations). For this variable, Equation (2.28b) is used to avoid differencing two large secular terms in the formulation of the VOP equations, a situation which would result in large errors.

$$\frac{da_i}{dt} = -\sum_{j=1}^6 (a_i, a_j) \frac{\partial \mathcal{R}}{\partial a_j} \rightarrow i = 1, 2, \dots, 5 \quad (2.27a)$$

$$\ell = nt + a_6 \quad (2.28a)$$

$$\frac{d\ell}{dt} = n - \sum_{j=1}^6 (\ell, a_j) \frac{\partial \mathcal{R}}{\partial a_j} \quad (2.28b)$$

2.3.3 VOP Equations in Equinoctial Elements

Equations (2.27a) and (2.28b) are the form of the Lagrange planetary equations used for the development of DSST in Section 2.4. As mentioned previously, DSST is also developed using the equinoctial element set to avoid singularities. Therefore Equations (2.27a) and (2.28b) will be developed here in the equinoctial elements for use in Section 2.4.

The first step is to define variables and relations that will be used to simplify the derivation. Equations (2.29a-c) are specific relations of the equinoctial elements used to simplify the expressions. The direction cosines with respect to the equinoctial reference frame of either the equatorial z axis or the third body position vector are then calculated. Expressing the disturbing potential in terms of these direction cosines, rather than in terms of p and q, make it easier to conceptualize the results. Finally, Equation (2.30) is defined to simplify notation.

$$A = na^2 \quad (2.28a)$$

$$B = \sqrt{1 - h^2 - k^2} \quad (2.28b)$$

$$C = 1 + p^2 + q^2 \quad (2.28c)$$

$$R_{x,y} = x \frac{\partial \mathcal{R}}{\partial y} - y \frac{\partial \mathcal{R}}{\partial x} \quad (2.30)$$

The partial of the disturbing potential with respect to the direction cosines is taken using the chain rule to accommodate the particular form of the disturbing functions necessary. These partials, as well as the partials of the direction cosines with respect to p and q , can be found in Volume II of reference [42], but are not reprinted here. Instead, the result of substituting the direction cosine partials and (2.28) equations into the disturbing potential partials results in Equations (2.31a) and (2.31b).

$$\frac{\partial \mathcal{R}}{\partial p} = \frac{2}{C} \left[\alpha \frac{\partial \mathcal{R}}{\partial \gamma} - \gamma \frac{\partial \mathcal{R}}{\partial \alpha} + qI \left(\alpha \frac{\partial \mathcal{R}}{\partial \beta} - \beta \frac{\partial \mathcal{R}}{\partial \alpha} \right) \right] \quad (2.31a)$$

$$\frac{\partial \mathcal{R}}{\partial q} = -\frac{2I}{C} \left[\beta \frac{\partial \mathcal{R}}{\partial \gamma} - \gamma \frac{\partial \mathcal{R}}{\partial \beta} + p \left(\alpha \frac{\partial \mathcal{R}}{\partial \beta} - \beta \frac{\partial \mathcal{R}}{\partial \alpha} \right) \right] \quad (2.31b)$$

Using one final simplification, given in Equation (2.32) which occurs for the nonresonant 3rd body and zonal harmonic perturbations, the final form of the VOP equations of motion can be obtained. Again the reader is directed to Volume II of reference [42] for a more detailed derivation of Equations (2.33a) – (2.33f).

$$R_{h,k} - R_{\alpha,\beta} \equiv 0 \quad (2.32)$$

VOP Equations of Motion Expressed in Equinoctial Elements

$$\dot{a} = \frac{2a}{A} \frac{\partial \mathcal{R}}{\partial \lambda} \quad (2.33a)$$

$$\dot{h} = \frac{B}{A} \frac{\partial \mathcal{R}}{\partial k} + \frac{k}{AB} (pR_{\alpha,\gamma} - IqR_{\beta,\gamma}) - \frac{hB}{A(1+B)} \frac{\partial \mathcal{R}}{\partial \lambda} \quad (2.33b)$$

$$\dot{k} = - \left[\frac{B}{A} \frac{\partial \mathcal{R}}{\partial h} + \frac{h}{AB} (pR_{\alpha,\gamma} - IqR_{\beta,\gamma}) - \frac{kB}{A(1+B)} \frac{\partial \mathcal{R}}{\partial \lambda} \right] \quad (2.33c)$$

$$\dot{p} = \frac{C}{2AB} \left[p \left(R_{h,k} - R_{\alpha,\beta} - \frac{\partial \mathcal{R}}{\partial \lambda} \right) - R_{\beta,\gamma} \right] \quad (2.33d)$$

$$\dot{q} = \frac{C}{2AB} \left[q \left(R_{h,k} - R_{\alpha,\beta} - \frac{\partial \mathcal{R}}{\partial \lambda} \right) - R_{\alpha,\gamma} \right] \quad (2.33e)$$

$$\dot{\lambda} = n - \frac{2a}{A} \frac{\partial \mathcal{R}}{\partial \lambda} + \frac{B}{A(1+B)} \left(h \frac{\partial \mathcal{R}}{\partial h} - k \frac{\partial \mathcal{R}}{\partial k} \right) + \frac{1}{AB} (pR_{\alpha,\gamma} - IqR_{\beta,\gamma}) \quad (2.33f)$$

2.4 Draper Semi-Analytic Satellite Theory (DSST)

There is some indication that the unique characteristics of mean elements used with semi-analytic theories are more successful with highly eccentric orbits than typical osculating elements and standard propagation methods [19], [20], [24], [25]. It is for this reason that DSST has been chosen as a primary propagation method for this analysis for both the mission and decay phases. Primarily during the mission phase, a fundamental source of yet un-modeled error is the truncation of the J_2^2 terms, and to understand this problem, it is necessary to have a comprehensive understanding of DSST, which is presented below. As with the derivation of the VOP equations, this section is taken from McClain's *A Recursively Formulated First-Order Semianalytic Artificial Satellite Theory Based on the Generalized Method of Averaging* [42], to maintain commonality of notation. The reader is encouraged to reference other sources on semi-analytic theories such as Vallado [2] and Danielson [46].

2.4.1 The Generalized Method of Averaging

Lagrange's investigation of planetary motion was based on the Lagrange planetary equations with the disturbing function expanded as a literal Fourier series. The long period and secular terms were then isolated by inspection, which yielded excellent results with small perturbations. Higher order solutions are necessary, however, when perturbations grow larger or the validity of the solution must be expanded over greater time intervals. It is therefore necessary to develop an efficient, systematic method for finding complete higher order approximations to the satellite's motion. The first step in that method is to isolate long period and secular contributions to the motion from the short periodics, as demonstrated in Figure 2.4.

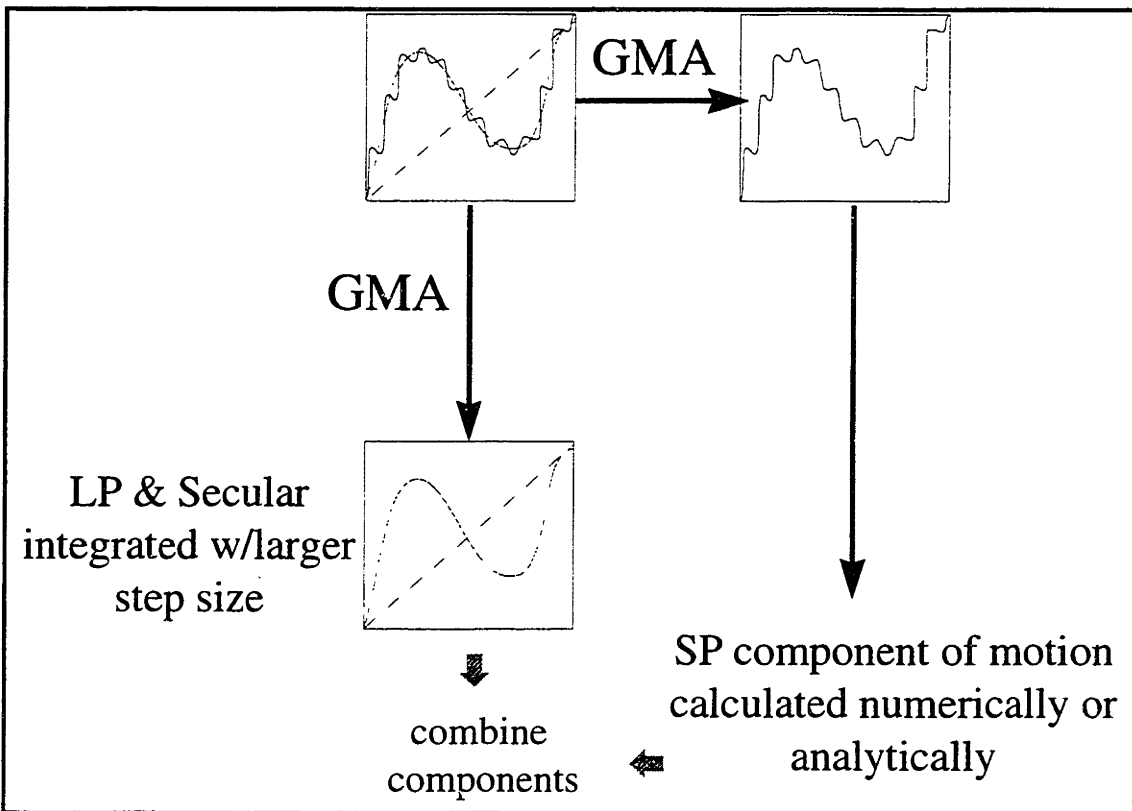


Figure 2.4: The Generalized Method of Averaging

This process is known as the Generalized Method of Averaging (GMA), and it results in a system of averaged equations with a parameter set solution known as the *mean elements*. These elements represent the solution to the averaged equations of motion over the averaging interval of the osculating elements, and describe only the long period and secular deviations from two-body motion. In the absence of resonance, the averaging interval for mean elements is usually the satellite period, which ensures elimination of all satellite-dependent short period terms. The reader should also note that mean elements are similar to Brouwer's single primed elements which reflect both the long-period and secular motion of the satellite, averaged over an interval equal to the period of the satellite.

The GMA theory is based on Poincaré's theory of asymptotic expansions [47], the concept of a near-identity transformation by Krylov and Bogoliubov [48], and substantial work by Mitropol'ski [49]. Several other authors have contributed to current GMA theory, all of which are listed in reference [42]. The current theory can be applied to either the Gaussian or Lagrangian formulation of the VOP equations, and as such, the generalized expressions (2.34a) and (2.34b) are used for the remaining derivation of DSST.

$$\frac{da_i}{dt} = \varepsilon F_i(\mathbf{a}, \ell) \rightarrow (i = 1, 2, \dots, 5) \quad (2.34a)$$

$$\frac{d\ell}{dt} = n(a) + \varepsilon F_6(\mathbf{a}, \ell) \quad (2.34b)$$

For GMA theory to be applied to the above generalized VOP equations and to determine the equations for the long period and secular motion, short period terms must first be selected. When integrating the equations of motion, the perturbation with the shortest period within acceptable accuracy tolerances is effectively the constraint on the

integration step size that can be used. Removing these terms allows the user to increase the stepsize, and with it, the speed of the integration process. To maximize efficiency, the stepsize should be as large as possible while still retaining the fundamental character of the motion over a given time interval. In order to determine these short-periodic perturbations, it is useful to examine the frequencies of the artificial satellite problem.

The first step is to determine the short period satellite dependent frequencies and, generally, all terms with periods of the same order of magnitude (and smaller) as the satellite period are considered short period terms. Another source of short periodics are the third body effects on motion, which will contribute terms with a fundamental period of approximately 28 days for the moon and one year for the sun. Whereas these sun and moon effects can be considered long period effects for the vast majority of artificial earth satellites, any number of the multiples of the third body fast variable may contribute periodicities P_n^* to the motion of the satellite. These periodicities are described by equation (2.35), where P^* is the fundamental period produced by the fast variable of the disturbing body.

$$P_n^* = \frac{P^*}{n} \rightarrow (n = 1, 2, \dots) \quad (2.35)$$

With very high harmonics in the third body model, n increases, introducing short period terms. However, in the absence of resonance, these terms are very small and in the case of strong resonance, they are significantly larger than the period of the satellite.

The nonsphericity of the Earth's gravitational field may also contribute short period effects. The rotation of the Earth (represented as the Greenwich Hour Angle θ , GHA) contributes terms with a fundamental period of 24 hours and can be considered long period only for close-earth satellites with orbits periods less than a few hours. These effects should therefore be grouped with medium period contributions. High degree multiples of the GHA ($m = 11, 12, 13, \dots$; $P < \text{few hours}$) appear in the spherical harmonic

expansion representing the Earth's gravitational model and cannot be considered medium period, they are θ -dependent short period terms. Tesseral harmonics also contribute terms which depend on linear combinations of the satellite fast variable and GHA. Contributions are generally larger than short period, and smaller than medium period except for the short period contribution of J_2 , which is larger than either short or medium period contributions.

Overall, the fundamental factor in determining short period and long period terms is the orbital period of the satellite. Knowing that all periods introduced by the fast variable and satellite dependent (or θ -dependent) terms are short period, all terms smaller than or equal to the satellite's orbital period are also short period. Figure 2.5 is a graphic representation of the relationship of various contributing terms. The figure demonstrates an approximate stratification for short and long period terms.

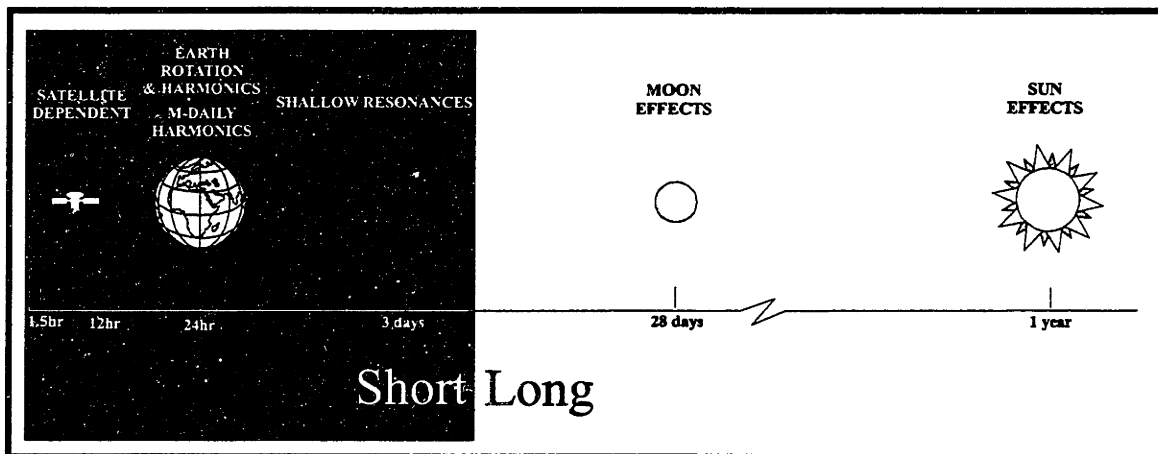


Figure 2.5: Generalized Method of Averaging Harmonics Stratification

What this implies for the GMA is that cases with quite distinct groups or families of frequencies are ideally suited for removal of these terms. If two frequencies overlap, removal of one family may not remove similar terms of different families. Similarly, if two are resonant, a single averaging pass will remove all frequencies contributed by both sources up to a cut-off frequency specified in the averaging operation. To avoid a

complex second averaging procedure, it is also possible to restrict the tesseral harmonic terms in the nonspherical gravitational model to eliminate the θ -dependent short period terms they introduce.

With DSST, a single application of GMA is used and θ -dependent terms are eliminated by appropriate restriction of the potential model (θ -dependent terms are most commonly included in the short periodics model). A medium period theory is developed but not applied, and a third body theory is developed with restricted use for nonresonant cases and satellites with periods significantly shorter than the third body orbital period.

2.4.2 Averaged VOP Equations of Motion

After developing guidelines for the stratification of short period and long period contributions, it is necessary to develop the averaged equations of motion from the generalized VOP equations (2.34). The near identity transformation from (\mathbf{a}, ℓ) to $(\bar{\mathbf{a}}, \bar{\ell})$ is assumed to take the form given in Equations (2.36a) and (2.36b) where the bar symbol indicates mean elements, $\eta_{i,j}$ functions are 2π periodic in ℓ , and ε is assumed to be a small parameter. Chapter 5 will focus on ε as J_2^2 , but it is defined here as a coefficient in one of the terms of the spherical harmonic expansion of the geopotential model, or the ratio of the semi-major axes of the satellite and third-body orbits in the series expansion of the third body disturbing function

$$a_i = \bar{a}_i + \sum_{j=1}^N \varepsilon^j \eta_{i,j}(\bar{\mathbf{a}}, \bar{\ell}) + O(\varepsilon^{N+1}) \rightarrow (i = 1, 2, \dots, 5) \quad (2.36a)$$

$$\ell = \bar{\ell} + \sum_{j=1}^N \varepsilon^j \eta_{6,j}(\bar{\mathbf{a}}, \bar{\ell}) + O(\varepsilon^{N+1}) \quad (2.36b)$$

In GMA, the equations of motion for the mean elements take the form given in Equations (2.37a) and (2.37b) where the rate of change of the mean elements depends only on the slowly varying mean elements.

$$\frac{d\bar{a}_i}{dt} = \sum_{j=1}^N \varepsilon^j A_{i,j}(\bar{\mathbf{a}}) + O(\varepsilon^{N+1}) \rightarrow (i = 1, 2, \dots, 5) \quad (2.37a)$$

$$\frac{d\bar{\ell}}{dt} = n(\bar{a}_1) + \sum_{j=1}^N \varepsilon^j A_{6,j}(\bar{\mathbf{a}}) + O(\varepsilon^{N+1}) \quad (2.37b)$$

These equations represent Equations (2.34), expressed in mean elements. The function F on the right hand side of (2.34) is expanded in a Taylor series about the mean elements and rearranged as a power series in the small parameters ε . It is then averaged such that all dependence on the mean fast variable is eliminated. The result yields order-by-order expressions for the mean element rates, $A_{i,j}(\bar{\mathbf{a}})$, in terms of suitably averaged functions of F and its partial derivatives (where i represents the variable and j represents the order of perturbation). The derivation also gives a formulation of short periodic motion.

It is now possible to formulate the generalized VOP equations (2.34) in mean elements. The first step is to differentiate (2.36) to obtain expressions for the osculating element rates which depend only on the corresponding mean elements and their rates, such as is given in Equations (2.38a) and (2.38b) (where a_k is understood to represent the fast variable ℓ).

$$\frac{da_i}{dt} = \frac{d\bar{a}_i}{dt} + \sum_{j=1}^N \varepsilon^j \sum_{k=1}^6 \frac{\partial \eta_{i,j}}{\partial \bar{a}_k} \frac{d\bar{a}_k}{dt} + O(\varepsilon^{N+1}) \rightarrow (i = 1, 2, \dots, 5) \quad (2.38a)$$

$$\frac{d\ell}{dt} = \frac{d\bar{\ell}}{dt} + \sum_{j=1}^N \varepsilon^j \sum_{k=1}^6 \frac{\partial \eta_{6,j}}{\partial \bar{a}_k} \frac{d\bar{a}_k}{dt} + O(\varepsilon^{N+1}) \quad (2.38b)$$

Substituting Equations (2.37) into (2.38) to introduce the mean element rates expressions into the osculating rates, results in Equation (2.39a) and (2.39b).

$$\frac{da_i}{dt} = \sum_{j=1}^N \varepsilon^j \left[A_{i,j}(\bar{\mathbf{a}}) + n(\bar{\mathbf{a}}_1) \frac{\partial \eta_{i,j}}{\partial \ell} + \sum_{m=1}^{N-j} \varepsilon^m \sum_{k=1}^6 A_{k,m} \frac{\partial \eta_{i,j}}{\partial \bar{a}_k} \right] + O(\varepsilon^{N+1}) \rightarrow (i = 1, 2, \dots, 5) \quad (2.39a)$$

$$\frac{d\ell}{dt} = n(\bar{\mathbf{a}}_1) + \sum_{j=1}^N \varepsilon^j \left[A_{6,j}(\bar{\mathbf{a}}) + n(\bar{\mathbf{a}}_1) \frac{\partial \eta_{6,j}}{\partial \ell} + \sum_{m=1}^{N-j} \varepsilon^m \sum_{k=1}^6 A_{k,m} \frac{\partial \eta_{6,j}}{\partial \bar{a}_k} \right] + O(\varepsilon^{N+1}) \quad (2.39b)$$

These equations can then be reordered to allow for future substitutions into the form given in Equations (2.40a) and (2.40b).

$$\frac{da_i}{dt} = \sum_{j=1}^N \varepsilon^j \left[A_{i,j}(\bar{\mathbf{a}}) + n(\bar{\mathbf{a}}_1) \frac{\partial \eta_{i,j}}{\partial \ell} + \sum_{k=1}^6 \sum_{p=1}^{j-1} A_{k,p}(\bar{\mathbf{a}}_1) \frac{\partial \eta_{i,j-p}}{\partial \bar{a}_k} \right] + O(\varepsilon^{N+1}) \rightarrow (i = 1, 2, \dots, 5) \quad (2.40a)$$

$$\frac{d\ell}{dt} = n(\bar{\mathbf{a}}_1) + \sum_{j=1}^N \varepsilon^j \left[A_{6,j}(\bar{\mathbf{a}}) + n(\bar{\mathbf{a}}_1) \frac{\partial \eta_{6,j}}{\partial \ell} + \sum_{k=1}^6 \sum_{p=1}^{j-1} A_{k,p}(\bar{\mathbf{a}}_1) \frac{\partial \eta_{6,j-p}}{\partial \bar{a}_k} \right] + O(\varepsilon^{N+1}) \quad (2.40b)$$

The reader should note that the summation over p is not performed for $j = 1$ and therefore does not contribute to first order terms. The function F on the right side of (2.34) is then expanded via a Taylor series about the mean elements as shown in Equation (2.42), where $\Delta a_k = a_k - \bar{a}_k$, and Equation (2.41) reminds the reader of the form of the Taylor series [50].

$$f(z) = \sum_{m=0}^{\infty} \frac{f^{(m)}(z_0)}{m!} (z - z_0)^m \quad (2.41)$$

$$F_i(\mathbf{a}, \ell) = \sum_{n=0}^{\infty} \frac{1}{n!} \left(\sum_{k=1}^6 \Delta a_k \frac{\partial}{\partial a_k} \right)^n F_i(\mathbf{a}, \ell) \Big|_{a=\bar{a}, \ell=\bar{\ell}} \rightarrow i = (1, 2, \dots, 6) \quad (2.42)$$

After defining $\frac{\partial}{\partial \bar{a}_i} = \frac{\partial}{\partial a_i} \Big|_{a=\bar{a}}$, (2.42) can be rearranged as a power series in ε by substituting Δa values from (2.36) to yield Equation (2.43), where f is defined by Equations (2.44a) through (2.44d).

$$F_i(\mathbf{a}, \ell) = \sum_{j=0}^N \varepsilon^j f_{i,j}(\bar{\mathbf{a}}, \bar{\ell}) + O(\varepsilon^{N+1}) \rightarrow i = (1, 2, \dots, 6) \quad (2.43)$$

$$f_{i,0}(\bar{\mathbf{a}}, \bar{\ell}) = F_i(\bar{\mathbf{a}}, \bar{\ell}) \quad (2.44a)$$

$$f_{i,1}(\bar{\mathbf{a}}, \bar{\ell}) = \sum_{k=1}^6 \eta_{k,1} \frac{\partial F_i}{\partial \bar{a}_k} \quad (2.44b)$$

$$f_{i,2}(\bar{\mathbf{a}}, \bar{\ell}) = \sum_{k=1}^6 \left[\eta_{k,2} \frac{\partial F_i}{\partial \bar{a}_k} + \frac{1}{2} \sum_{m=1}^6 \eta_{k,1} \eta_{m,1} \frac{\partial^2 F_i}{\partial \bar{a}_k \partial \bar{a}_m} \right] \quad (2.44c)$$

$$f_{i,3}(\bar{\mathbf{a}}, \bar{\ell}) = \sum_{k=1}^6 \left\{ \eta_{k,3} \frac{\partial F_i}{\partial \bar{a}_k} + \frac{1}{2} \sum_{m=1}^6 \left[(\eta_{k,2} \eta_{m,1} + \eta_{k,1} \eta_{m,2}) \frac{\partial^2 F_i}{\partial \bar{a}_k \partial \bar{a}_m} + \frac{1}{3} \sum_{j=1}^6 \eta_{k,1} \eta_{m,1} \eta_{j,1} \frac{\partial^3 F_i}{\partial \bar{a}_k \partial \bar{a}_m \partial \bar{a}_j} \right] \right\} \quad (2.44d)$$

The mean motion $n(a_1)$ is also expanded in a Taylor series about \bar{a}_1 , as shown in Equation (2.45). Then, using the same method as for F , expanded as a power series in ε to yield Equation (2.46) where the N function is defined in Equations (2.47a) through (2.47d).

$$n(a_1) = \sum_{k=0}^{\infty} \frac{(\Delta a_1)^k}{k!} \frac{d^k n}{d\bar{a}_1^k} \quad (2.45)$$

$$n(a_1) = \sum_{k=0}^N \varepsilon^k N_k(\bar{\mathbf{a}}, \bar{\ell}) \quad (2.46)$$

$$N_0(\bar{\mathbf{a}}, \bar{\ell}) = n(\bar{a}_1) = \bar{n} \quad (2.47a)$$

$$N_1(\bar{\mathbf{a}}, \bar{\ell}) = -\frac{3}{2} \frac{\bar{n}}{\bar{a}_1} \eta_{1,1} \quad (2.47b)$$

$$N_2(\bar{\mathbf{a}}, \bar{\ell}) = \frac{15}{4} \frac{\bar{n}}{\bar{a}_1^2} \eta_{1,1}^2 - \frac{3}{2} \frac{\bar{n}}{\bar{a}_1} \eta_{1,2} \quad (2.47c)$$

$$N_3(\bar{\mathbf{a}}, \bar{\ell}) = -\frac{35}{16} \frac{\bar{n}}{\bar{a}_1^3} \eta_{1,1}^3 + \frac{15}{4} \frac{\bar{n}}{\bar{a}_1^2} \eta_{1,1}^2 - \frac{3}{2} \frac{\bar{n}}{\bar{a}_1} \eta_{1,2} \quad (2.47d)$$

Finally, Equations (2.37), (2.43), (2.44) and (2.47) are substituted into (2.34). Equating terms with like powers of ε yields the expressions for the j^{th} – order contribution to the osculating element rates given in Equations (2.48a) and (2.48b).

$$A_{i,j}(\bar{\mathbf{a}}) + \bar{n} \frac{\partial \eta_{i,j}}{\partial \ell} + \sum_{k=1}^6 \sum_{p=1}^{j-1} A_{k,p}(\bar{\mathbf{a}}) \frac{\partial \eta_{i,j-p}}{\partial \bar{a}_k} = f_{i,j-1}(\bar{\mathbf{a}}, \bar{\ell}) \quad (2.48a)$$

$$A_{6,j}(\bar{\mathbf{a}}) + \bar{n} \frac{\partial \eta_{6,j}}{\partial \ell} + \sum_{k=1}^6 \sum_{p=1}^{j-1} A_{k,p}(\bar{\mathbf{a}}) \frac{\partial \eta_{6,j-p}}{\partial \bar{a}_k} = f_{6,j-1}(\bar{\mathbf{a}}, \bar{\ell}) + N_j \quad (2.48b)$$

2.4.3 Elimination of Fast Variable Dependence

Equations (2.48) are really an intermediate step in the determination of both the averaged equations of motion and the short periodic functions. The next step involves the $A_{i,j}(\bar{\mathbf{a}})$ functions, which depend only on the slowly varying mean elements, and must now be related to the perturbing function (or a power series representation of it). Since the functions $\eta_{i,j}$ are underdetermined (except their periodic constraint discussed above), (2.48) still includes the short periodic functions. Fortunately, the periodicity of the $\eta_{i,j}$ functions permits the elimination of the mean fast variable dependence, which is accomplished by integrating both sides of (2.48) over the mean fast variable. This process, known as the averaging operation and defined in Equation (2.49), integrates over

the interval $[0, 2\pi]$ to eliminate the $\frac{\partial \eta_{i,j}}{\partial \bar{\ell}}$ function. Properties of the averaging function are listed in Equations (2.50a) through (2.50e), where X and Y are two functions (continuous and differentiable) which are 2π periodic in $\bar{\ell}$, and ρ is any function independent of the mean fast variable.

$$\langle H(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\ell}} = \frac{1}{2\pi} \int_0^{2\pi} H(\bar{\mathbf{a}}, \bar{\ell}) d\bar{\ell} \quad (2.49)$$

$$\langle X(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\ell}} = C(\bar{\mathbf{a}}) \quad (2.50a)$$

$$\langle X(\bar{\mathbf{a}}, \bar{\ell}) Y(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\ell}} \neq \langle X(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\ell}} \langle Y(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\ell}} \quad (2.50b)$$

$$\langle X(\bar{\mathbf{a}}, \bar{\ell}) + Y(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\ell}} = \langle X(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\ell}} + \langle Y(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\ell}} \quad (2.50c)$$

$$\langle \rho X(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\ell}} = \rho \langle X(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\ell}} \quad (2.50d)$$

$$\frac{\partial}{\partial a_k} \langle X(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{i}} = \left\langle \frac{\partial}{\partial a_k} X(\bar{\mathbf{a}}, \bar{\ell}) \right\rangle_{\bar{i}} \quad (2.50e)$$

It follows that, since the η functions are 2π periodic in ℓ , they will average to zero, as shown in (2.51a).

$$\left\langle \frac{\partial \eta_{i,j}}{\partial \ell} \right\rangle_{\bar{i}} = 0 \quad (2.51a)$$

The reader should also note what happens to the summations of Equation (2.48) once averaged, as shown in (2.51b), where the second summation now neglects the fast variable. As a result of this relation, Equations (2.34a) and (2.34b) then become Equation (2.52a) and (2.52b).

$$\left\langle \sum_{p=1}^{j-1} \sum_{k=1}^6 A_{k,p}(\bar{\mathbf{a}}) \frac{\partial \eta_{i,j-p}}{\partial a_k} \right\rangle_{\bar{i}} = \sum_{p=1}^{j-1} \sum_{k=1}^5 A_{k,p}(\bar{\mathbf{a}}) \left\langle \frac{\partial \eta_{i,j-p}}{\partial a_k} \right\rangle_{\bar{i}} \quad (2.51b)$$

$$A_{i,j}(\bar{\mathbf{a}}) = \left\langle f_{i,j-1}(\bar{\mathbf{a}}, \bar{\ell}) \right\rangle_{\bar{i}} - \sum_{p=1}^{j-1} \sum_{k=1}^5 A_{k,p}(\bar{\mathbf{a}}) \left\langle \frac{\partial \eta_{i,j-p}}{\partial a_k} \right\rangle_{\bar{i}} \quad (2.52a)$$

$$A_{6,j}(\bar{\mathbf{a}}) = \left\langle f_{6,j-1}(\bar{\mathbf{a}}, \bar{\ell}) + N_j \right\rangle_{\bar{i}} - \sum_{p=1}^{j-1} \sum_{k=1}^5 A_{6,p}(\bar{\mathbf{a}}) \left\langle \frac{\partial \eta_{6,j-p}}{\partial a_k} \right\rangle_{\bar{i}} \quad (2.52b)$$

These equations can be further simplified with the application of a series of constraints, the first of which is listed in Equations (2.53a). This constraint can also be expressed as an equivalence of mean and osculating elements (no short periodics), as shown in Equation (2.53b).

$$\left\langle \frac{\partial \eta_{i,j}}{\partial a_k} \right\rangle_{\bar{i}} \equiv 0 \quad \begin{array}{l} (i = 1, 2, \dots, 6) \\ (k = 1, 2, \dots, 6) \end{array} \quad (2.53a)$$

$$\left\langle \frac{da_i}{dt} \right\rangle_{\bar{\tau}} = \frac{d\bar{a}_i}{dt} \quad \left\langle \frac{d\ell}{dt} \right\rangle_{\bar{\tau}} = \frac{d\bar{\ell}}{dt} \quad (2.53b)$$

Next a constraint is placed upon the mean elements to represent the long-period and secular contributions to the osculating elements to within a constant C , as shown in Equation (2.54) (which follows from (2.51a) and (2.53a)).

$$\left\langle \eta_{i,j}(\bar{\mathbf{a}}, \bar{\ell}) \right\rangle_{\bar{\tau}} = C_{i,j} \quad (2.54)$$

A logical extension to the constraint given in (2.53) is to require that the constant $C = 0$, such that the averaged osculating elements are identical to the mean elements. With these constraints, the $\eta_{i,j}$ functions are restricted to containing only short period and mixed short period (product of a short period term and a long period term) terms only. In doing so, (2.52) can be reduced to Equations (2.55a) and (2.55b), which are now completely specified in terms of the expansion of the perturbing function and, for the mean fast variable, the expansion of the osculating mean motion. Therefore, by substituting (2.55) into (2.37) yields the averaged equations of motion given in Equations (2.56a) and (2.56b).

$$A_{i,j}(\bar{\mathbf{a}}) = \left\langle f_{i,j-1}(\bar{\mathbf{a}}, \bar{\ell}) \right\rangle_{\bar{\tau}} \quad (i = 1, 2, \dots, 5) \quad (2.55a)$$

$$A_{6,j}(\bar{\mathbf{a}}) = \left\langle f_{6,j-1}(\bar{\mathbf{a}}, \bar{\ell}) + N_j \right\rangle_{\bar{\tau}} \quad (2.55b)$$

$$\frac{d\bar{a}_i}{dt} = \sum_{j=1}^N \varepsilon^j \left\langle f_{i,j-1}(\bar{\mathbf{a}}, \bar{\ell}) \right\rangle_{\bar{\tau}} + O(\varepsilon^{N+1}) \rightarrow (i = 1, 2, \dots, 5) \quad (2.56a)$$

$$\frac{d\bar{\ell}}{dt} = n(\bar{a}_1) + \sum_{j=1}^N \varepsilon^j \left\langle f_{6,j-1}(\bar{\mathbf{a}}, \bar{\ell}) + N_j \right\rangle_{\bar{\tau}} + O(\varepsilon^{N+1}) \quad (2.56b)$$

The reader should note that the averaging operation does not free (2.56) of all contributions from the short period terms due to the fact that the product of two short

period functions can yield a long period term. Inspection of (2.44) and (2.47) shows that partials of the osculating force function (F) and $\eta_{i,j}$ functions (as well as product of two or more $\eta_{i,j}$ functions) exist, and can produce long period terms. These terms will always be of second (or higher) order in the small parameter, a characteristic that will become important in Chapter 5.

2.4.4 Determination of Short Period Functions

To obtain the $\eta_{i,j}$ functions, without the constraints given in (2.53), and (2.54), first subtract (2.52) from (2.48) to get Equations (2.57a) and (2.57b).

$$\bar{n} \frac{\partial \eta_{i,j}}{\partial \ell} + \sum_{p=1}^{j-1} \sum_{k=1}^6 A_{k,p} \left(\frac{\partial \eta_{i,j-p}}{\partial \bar{a}_k} - \left\langle \frac{\partial \eta_{i,j-p}}{\partial \bar{a}_k} \right\rangle_{\bar{\tau}} \right) = f_{i,j-1} - \langle f_{i,j-1} \rangle_{\bar{\tau}} \rightarrow (i = 1, 2, \dots, 5) \quad (2.57a)$$

$$\bar{n} \frac{\partial \eta_{6,j}}{\partial \ell} + \sum_{p=1}^{j-1} \sum_{k=1}^6 A_{k,p} \left(\frac{\partial \eta_{6,j-p}}{\partial \bar{a}_k} - \left\langle \frac{\partial \eta_{6,j-p}}{\partial \bar{a}_k} \right\rangle_{\bar{\tau}} \right) = f_{6,j-1} + N_j - \langle f_{6,j-1} + N_j \rangle_{\bar{\tau}} \quad (2.57b)$$

In order to simplify, the superscript s identifier is defined in Equation (2.58) to reduce (2.57) to Equations (2.59a) and (2.59b):

$$f^s_{i,j-1} = f_{i,j-1} - \langle f_{i,j-1} \rangle_{\bar{\tau}} \quad (2.58)$$

$$\bar{n} \frac{\partial \eta_{i,j}}{\partial \ell} = f^s_{i,j-1} - \sum_{p=1}^{j-1} \sum_{k=1}^6 A_{k,p} \frac{\partial \eta^s_{i,j-p}}{\partial \bar{a}_k} \rightarrow (i = 1, 2, \dots, 5) \quad (2.59a)$$

$$\bar{n} \frac{\partial \eta_{6,j}}{\partial \ell} = f^s_{6,j-1} + N_j^s - \sum_{p=1}^{j-1} \sum_{k=1}^6 A_{k,p} \frac{\partial \eta^s_{6,j-p}}{\partial \bar{a}_k} \quad (2.59b)$$

When looking at $\eta_{i,j}$, i represents the variable (a, h, k, ...), and j represents the order. Therefore, inspection of (2.59) shows that $\eta_{i,j}$ depends only on quantities of lower order. The one exception is the case of variable $\bar{\ell}$, because $\eta_{6,j}$ also depends on the j^{th} order function $\eta_{1,j}$ introduced through the term N_j , which necessitates $\eta_{1,j}$ being determined before $\eta_{6,j}$. The $\eta_{i,j}$ functions are calculated (within an arbitrary function of slow variables) by expanding the right hand side of (2.59) into multiple Fourier series and integrating one term at a time. Multiplying (2.59) by $1/\bar{n}$ and integrate with respect to the mean fast variable will yield (2.60), where α is defined in Equations (2.61a) and (2.61b) which are 2π periodic in ℓ (and therefore average to 0), and C is an arbitrary function of integration which depends only on the slowly varying mean elements and is constrained to 0 after averaging.

$$\eta_{i,j}(\bar{\mathbf{a}}, \bar{\ell}) = \alpha_{i,j}(\bar{\mathbf{a}}, \bar{\ell}) + C_{i,j}(\bar{\mathbf{a}}) \quad (2.60)$$

$$\alpha_{i,j} = \frac{1}{\bar{n}} \int \left[f^{s_{i,j-1}}(\bar{\mathbf{a}}, \bar{\ell}) - \sum_{p=1}^{j-1} \sum_{k=1}^6 A_{k,p} \frac{\partial \eta_{i,j-p}}{\partial \bar{a}_k} \right] d\bar{\ell} \quad (2.61a)$$

$$\alpha_{6,j} = \frac{1}{\bar{n}} \int \left[\{ f^{s_{6,j-1}}(\bar{\mathbf{a}}, \bar{\ell}) + N_j^s \} - \sum_{p=1}^{j-1} \sum_{k=1}^6 A_{k,p} \frac{\partial \eta_{6,j-p}^s}{\partial \bar{a}_k} \right] d\bar{\ell} \quad (2.61b)$$

These constraints are similar to those given in the previous section, with the same result, the $\eta_{i,j}$ functions are constrained to be purely short periodic and/or mixed periodic, such that: $\eta_{i,j} = \eta_{i,j}^s$. Therefore, a set of functions containing only short periodic terms can be obtained, and the near identity transformation given in (2.36) are completely specified by Equations (2.62a) and (2.62b).

$$a_i = \bar{a}_i + \frac{1}{\bar{n}} \sum_{j=1}^N \varepsilon^j \int \left[f^{s_{i,j-1}}(\bar{\mathbf{a}}, \bar{\ell}) - \sum_{k=1}^6 \sum_{p=1}^{j-1} A_{k,p} \frac{\partial \eta^{s_{i,j-p}}}{\partial \bar{a}_k} \right] d\bar{\ell} + O(\varepsilon^{N+1}) \quad (2.62a)$$

$$\ell = \bar{\ell} + \frac{1}{\bar{n}} \sum_{j=1}^N \varepsilon^j \int \left[\{f^{s_{6,j-1}}(\bar{\mathbf{a}}, \bar{\ell}) + N_j^s\} - \sum_{k=1}^6 \sum_{p=1}^{j-1} A_{k,p} \frac{\partial \eta^{s_{6,j-p}}}{\partial \bar{a}_k} \right] d\bar{\ell} + O(\varepsilon^{N+1}) \quad (2.62b)$$

2.4.5 Computational Procedure

After formulating the averaged equations of motion, and the tools used with those equations, it is necessary to develop an algorithm for calculating the various components of the equations. This process must proceed serially, on an order-by-order basis to account for the interdependency of the determination of the j^{th} – order contribution to the mean element rates and the $\eta_{i,j}$ functions. To illustrate this procedure, expand Equations (2.56) to second order to yield Equations (2.63a) and (2.63b).

$$\frac{d\bar{a}_i}{dt} = \varepsilon \langle f_{i,0}(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\tau}} + \varepsilon^2 \langle f_{i,1}(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\tau}} + O(\varepsilon^3) \quad (2.63a)$$

$$\frac{d\bar{\ell}}{dt} = n(\bar{a}_1) + \varepsilon \langle f_{6,0}(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\tau}} + \varepsilon^2 \langle f_{6,1}(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\tau}} + O(\varepsilon^3) \quad (2.63b)$$

Next, using the constraints from Section 2.4.3 for C and the averaged $\eta_{i,j}$ functions, as well as the N function definitions (2.47), the averaging function yields the following simplifying equations.

$$\langle f_{i,j-1}(\bar{\mathbf{a}}, \bar{\ell}) + N_j(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\tau}} = \langle f_{i,j-1}(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\tau}} + \langle N_j(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\tau}} \quad (2.64a)$$

$$\langle N_1(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\tau}} = \left\langle -\frac{3\bar{n}}{2\bar{a}_1} \eta_{1,1}(\bar{\mathbf{a}}, \bar{\ell}) \right\rangle_{\bar{\tau}} = 0 \quad (2.64b)$$

$$\langle N_2(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\tau}} = \left\langle \frac{15\bar{n}}{4\bar{a}_1^2} \eta_{1,1}^2 - \frac{3\bar{n}}{2\bar{a}_1} \eta_{1,2} \right\rangle_{\bar{\tau}} = \left\langle \frac{15\bar{n}}{4\bar{a}_1^2} \eta_{1,1}^2 \right\rangle_{\bar{\tau}} \quad (2.64c)$$

These simplifications can then be used with (2.44) to reduce (2.63) to Equations (2.65a) and (2.65b)

$$\frac{d\bar{a}_i}{dt} = \varepsilon \langle F_i(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\tau}} + \varepsilon^2 \left\langle \sum_{k=1}^6 \eta_{k,1}(\bar{\mathbf{a}}, \bar{\ell}) \frac{\partial F_i(\bar{\mathbf{a}}, \bar{\ell})}{\partial \bar{a}_k} \right\rangle_{\bar{\tau}} + O(\varepsilon^3) \rightarrow (i = 1, 2, \dots, 5) \quad (2.65a)$$

$$\frac{d\bar{\ell}}{dt} = \bar{n} + \varepsilon \langle F_6(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\tau}} + \varepsilon^2 \left\langle \sum_{k=1}^6 \eta_{k,1}(\bar{\mathbf{a}}, \bar{\ell}) \frac{\partial F_6(\bar{\mathbf{a}}, \bar{\ell})}{\partial \bar{a}_k} + \frac{15\bar{n}}{8\bar{a}_1^2} \eta_{1,1}^2(\bar{\mathbf{a}}, \bar{\ell}) \right\rangle_{\bar{\tau}} + O(\varepsilon^3) \quad (2.65b)$$

Inspection of (2.65) shows that first order contributions to the mean element rates are independent of the $\eta_{i,j}$ functions, but second order contributions are not. Therefore, computation of the mean element rates and $\eta_{i,j}$ functions for the averaged equations of motion for a single perturbing function, must proceed in the following manner, given by Equations (2.66a) through (2.66e).

$$A_{i,1} = \langle F_i(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\tau}} \rightarrow (i = 1, 2, \dots, 6) \quad (2.66a)$$

$$\eta_{i,1} = \frac{1}{\bar{n}} \int F_i^s(\bar{\mathbf{a}}, \bar{\ell}) d\bar{\ell} \rightarrow (i = 1, 2, \dots, 5) \quad (2.66b)$$

$$\eta_{6,1} = \frac{1}{\bar{n}} \int \left[F_6^s(\bar{\mathbf{a}}, \bar{\ell}) - \frac{3\bar{n}}{2\bar{a}_1} \eta_{1,1} \right] d\bar{\ell} \quad (2.66c)$$

$$A_{i,2} = \left\langle \sum_{k=1}^6 \eta_{k,1} \frac{\partial F_i}{\partial \bar{a}_k} \right\rangle_{\bar{r}} \quad (2.66d)$$

$$A_{6,2} = \left\langle \sum_{k=1}^6 \eta_{k,1} \frac{\partial F_6}{\partial \bar{a}_k} + \frac{15}{4} \frac{\bar{n}}{\bar{a}_1^2} \eta_{1,1}^2 \right\rangle_{\bar{r}} \quad (2.66e)$$

These Equations ((2.66a) through (2.66e)) represent the basis of the J_2^2 investigation in Chapter 5.

2.5 Impact Prediction Methods

The science of reentry modeling is a complex and comprehensive art, one that will certainly not be addressed here in any appreciable detail. The serious reader is encouraged to reference any number of documents and books which treat the topic much more rigorously [51], [52], and [53]. This section, rather, is meant only to introduce the reader to some of the main programs and tools (and their fundamental technical foundation), currently employed for impact prediction of reentering objects. Special attention will be paid to the Lifetime prediction model as implemented in the Satellite Toolkit, and used in Chapter 8 to predict impact points of the various objects in this study.

Domestic methods for impact prediction include the NORAD multi-step process, and several specialized prediction programs developed by NASA, JPL, DoD, and industry. NORAD uses their Tracking and Impact Prediction (TIP) Special Perturbations software for impact predictions of reentering objects. Between fifteen and twenty days prior to the estimated decay date, a special perturbations propagator and sliding differential correction batch fits are utilized in conjunction with increased radar tasking to step the satellite orbit predictions down to final decay. The models used with these tools

are the Jacchia-65 atmosphere model and the World Geodetic System Earth gravity model of 1972 (WGS-72) truncated to sixth order and degree [54]. Cowell's method is used with the Gauss-Jackson eighth order predictor-corrector formulas to integrate the equations of motion, with the partials computed analytically to increase speed [54].

At specified intervals, NORAD issues warning to various agencies of possible reentry locations, while continually updating the estimate with new data and fits. A more modern technique, introduced by Barker, Eller, and Herder reduces variability in decay estimates with the parameterization of the ballistic coefficient in the form of a simple linear function with time, thus making it a new model parameter. This method has shown, with a sample base of some 156 cases, to improve on the NORAD prediction time by as much as 61% [54]. The reader should note that Professor Andrey Nazarenko has created a highly accurate and efficient atmosphere model (GOST) which takes into account the variation of density calculations through a process known as adaptive density estimation. This model, which will be summarized in Chapter 6, has achieved impressive accuracy for density estimation, particularly in the 500-600 km region [55].

Another domestic contributor to the art of reentry prediction is the Aerospace Corporation, who has improved or developed several major reentry prediction tools including the Generalized Trajectory Simulation (GTS), Trajectory Optimization Program (TOP), Reentry Trajectory Reconstruction Program (RENTTRAN), and Lifetime Prediction Program. GTS and TOP are general purpose trajectory simulation and optimization programs, which have evolved over a 30 year period. These programs have powered flight capabilities in addition to the following reentry trajectory analysis capabilities [56]:

- Deorbit planning
- Debris footprint generation

- Crossrange & Downrange aerodynamic reentry
- Sensor placement for planned reentry to maximize visibility
- Trajectory reconstruction based on sensor observations
- Tumbling reentry analysis
- Monte Carlo dispersion analysis
- Reentry heating calculations

RENTTRAN reconstructs the trajectory state and drag profile based on multiple sensor observations, and estimates the breakup altitude. Notable features include the same MSISE-90 atmosphere model used for this thesis, a transitional drag model, and state of the art numerical techniques [56]. RENTTRAN has been used for the following applications:

- Shuttle External Tank (ET) breakup analysis
- Random reentry trajectory analysis
- Planned reentry trajectory analysis
- Real time Launch vehicle anomaly analysis

Another Aerospace tool is the Lifetime analysis program which is one of the most popular reentry/impact prediction tools on the market today. Lifetime can be used for various applications, including [56]:

- Highly accurate and efficient orbit decay prediction
- Orbit sustenance fuel requirement estimates
- Impact point prediction using NORAD data
- Differential correction of the ballistic coefficient

- High resolution plots of decay and impact

In Chapter 8, Satellite Toolkit 4.0.3 and its version of the Lifetime prediction tool are used to predict impact points for the objects listed in Chapter 1. Based originally on algorithms from NASA's Langley Research Center and enhanced by the Aerospace Corporation, Lifetime utilizes the Jacchia 1970 model (90 – 2500 km) and U.S. Standard Atmosphere 1976 (USSA-76) model (below 90 km) to model atmospheric density, and a simplified Earth gravitational model (which considers oblateness effects through J_5) for efficiency. Lifetime also includes models for the effects of luni-solar point mass effects and solar radiation pressure [57]. The STK Lifetime tool was chosen for this analysis due to its extensive graphics capability, ease of use, and availability. In addition, the U.S. Standard Atmosphere model helps to improve predictions for the last few orbits where no observations exist to correct for density mis-modeling.

The STK Lifetime tool requires the user to input several of the satellite's physical characteristics, including:

- Mass
- Area
- Coefficient of Drag
- Coefficient of Reflection
- Solar flux data file (can use STK default file)
- Planetary geomagnetic index data file (can use STK default file)
- Orbit count limit (maximum number of orbits processed before Lifetime stops calculations)
- Number of orbits per Lifetime calculation
- Number of Gaussian quadratures used in drag integration routine
- 2nd order oblateness correction flag (on/off – for highly eccentric orbits, should be on)

- Rotating atmosphere model flag (on/off)

Lifetime, as employed in Satellite Toolkit, calculates the date of satellite decay, total satellite lifetime, and number of orbits within the lifetime. Orbital elements, ground tracks, coordinates, and other information are also available in several graph and report formats. Lifetime also gives the user a graphical representation of the final orbit ground track, as seen in Figure 2.6 with an impact label. The figure includes ground tracks for both the epoch orbit of a typical Molniya satellite, and the final orbit down to impact.

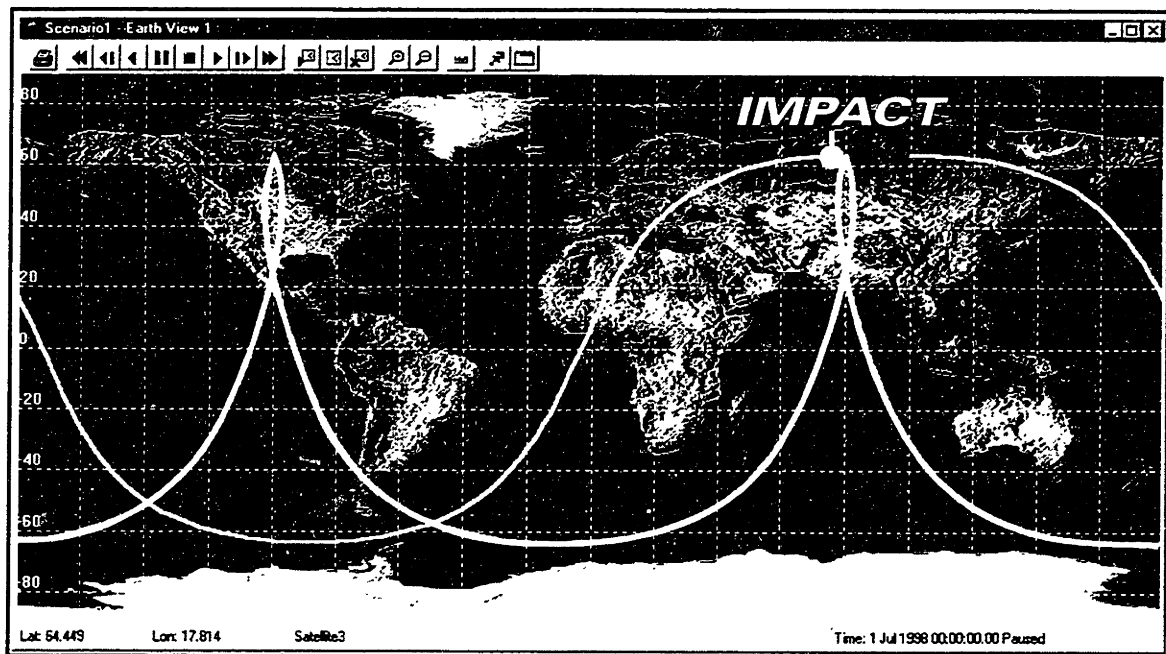


Figure 2.6: Molniya 1-55 Rocket Body Final and Mission Orbit Ground Tracks

There are also a multitude of reentry prediction tools available on the world market, including programs from the Russian Space Agency, CNUCE, an institute of the Italian National Research Council (CNR), and the European Space Operations Center (ESOC). The Russian Space Agency program boasts impressive accuracy using Nazarenko's GOST adaptive density estimation model, but are still in need of further improvement, especially in the below 150 km region, where the GOST model is as yet not completely refined [25]. CNUCE is currently developing an automated reentry

determination system as a result of a study [58] which compared various propagators, density models, and automation techniques. The study boasted 10% prediction errors and nearly 20% for a tether satellite due to simplified models (for example, a constant value of the ballistic coefficient) [58], but only represent a first step in the evolution of the specific reentry determination capability.

At ESOC, there are two methods of reentry prediction currently employed [53]. One method is a more completely modeled and computerized version of the historic King-Hele technique which uses observational data to determine various parameters and update state estimates [10], [18], [59]. In addition, ESOC uses a reentry program known as FOCUS.

The FOCUS program's most notable feature is that it limits error due to analytically derived drag perturbations by stopping the semi-analytic state propagation of mean elements after passing through a pre-defined altitude, then propagating the osculating state to impact [60], [53]. The FOCUS program uses the GEM 10B geopotential model (J2 – J7), MSIS-86 (> 120 km), USSA-76 (< 90 km), and a bridging function (between 90 and 120 km) to model the atmosphere, a variable drag coefficient model, rotating Earth atmosphere, luni-solar point mass effects, and solar radiation pressure. To integrate the equations of motion, FOCUS uses Cowell's method with a Runge-Kutta/Shanks 7/8 starter and Adams-Bashforth/Adams-Moulton multi-step numerical integrator [53]. The major limiting feature of FOCUS is that the trajectory prediction is truncated at 30 km because "the governing laws of perturbed Keplerian motion become invalid below this threshold altitude [53]." Basically, the vehicle is considered to be in a nearly vertical fall for the remainder of the reentry trajectory, and as such, the Center Of Impact Window (COIW) is defined at the ground track coordinates through which the satellites passes at an altitude of 30 km. Depending on the satellite configuration, this assumption of neglecting the hypersonic flow regime during reentry,

can lead to significant errors, but due to the highly stochastic nature of these errors, the assumption can be considered valid for a majority of reentering objects.

The aforementioned methods have all been deterministic in nature. Unfortunately, since modeling reentry depends on several parameters that are difficult or impossible to predict, deterministic models may be limited in the accuracy they can achieve. Another option lies in the development of a probabilistic reentry model, several of which currently exist. References [61] and [62] present two possible methods for incorporating probabilistic behavior. Barker [61] uses a Monte Carlo simulation to analyze various factors such as the addition of pseudo-noise compensation for unmodeled dynamics, adaptive estimation methods, and state covariance deweighting techniques. From these analyses he was able to develop a fading memory differential correction model compensation method which improved impact prediction accuracy [53]. Lefebvre [62] used multiple linear regression models to improve TIP decay procedures with success. Another model using Monte Carlo techniques and complex probability trees, recently developed for the Cassini mission reentry analysis, modeled probabilities of survival of critical components and a corresponding impact footprint for surviving components [13].

Whereas there may be a multitude of tools available for impact prediction, no one tool is accurate enough to close the door on this complex problem. Probabilistic methods seem to have more promise in future applications, but with an environment as dynamic as that created by an object reentering the Earth's atmosphere at orbital speeds, all models need improvements to create a comprehensive and reliable reentry modeling program.

Chapter 3

Statistical Orbit Determination Methods

3.1 Introduction

The fundamental challenge of orbit determination is to estimate a set of orbital parameters which minimize the error between computed and real world orbital trajectories. The set of parameters used in the process can include orbital elements, position and velocity vectors, observation station locations and biases [4], and coefficients of various perturbative effects such as the coefficients of drag, solar radiation pressure and atmospheric lift.

Unfortunately, measurements obtained from the observation stations have an inherent inaccuracy due to biases and noise created by the sensing equipment. In addition, the mathematical constructs used to model the orbital motion can contribute error. As a result, no trajectory can perfectly match the observations, and the user must employ a method which finds the “best fit” estimate of the actual trajectory. Several methods exist for determining this estimate, but this chapter will only cover the broad categories used for this thesis, including the classical least squares method and a sequential estimation method known as the Kalman Filter.

An important definition, necessary for this discussion of orbit determination techniques, is differential correction. *Differential Correction (DC)* is simply the process of processing the methods of solution to an orbit determination process which require

iteration or incremental updates to the state vector [2]. This process can be used with linear and nonlinear least squares methods, as well as sequential and filter methods, and typically involves groups of data known as *batches*. The DC process takes a specific batch of data, processes the observations within the batch to determine an incremental update, updates the states with this correction and, if a specified residual tolerance has not been met, repeats the above steps.

The following derivations are meant to briefly introduce the reader who is unfamiliar with the fundamentals of the orbit determination process. An introduction to the least squares problem (as well as the weighted linear and nonlinear variants of the solution), is followed by a brief overview of both the extended and linearized versions of the Kalman Filter. This discussion, primarily of the Kalman Filter, is a necessary tool for developing the functional representation of process noise used for this thesis, which is described in Chapter 6. For a more detailed derivation of either the least squares methods, or the Kalman Filter, literally hundreds of sources are available, including those sources used in the development of Chapter 3 (references [2], [4], [34], [39], [50], and [63] through [73]).

3.2 Least Squares Batch Methods

Carl Gauss first developed and used the Method of Least Squares in 1795, but allowed nearly 14 years to pass before mentioning it in his book *Theoria Motus*. Battin [39] describes the ensuing conflict that arose when Adrien-Marie Legendre independently developed the method and published the results in 1806. Gauss acknowledged Legendre's work in *Theoria Motus* by stating:

“Our principle, which we have made use of since the year 1795, has lately been published by Legendre in the work *Nouvelles méthodes pour la détermination des orbites des comètes*, Paris, 1806, where several other properties of this principle have been explained, which, for the sake of brevity, we here omit.”

The angered Legendre wrote to Gauss “You, who are already so rich in discoveries, might have had the decency not to appropriate the Method of Least Squares.” Despite Gauss’ gracious apology to Legendre, the evidence clearly substantiated Gauss’ priority, and the method is now remembered as Gauss’ Method of Least Squares [39].

In its simplest form, Gauss’ Method of Least Squares is simply the solution, or more exactly, a close approximation of the solution for a set of linear algebraic equations of the form given in Section 3.2.1, which have more equations than unknowns (overdetermined).

3.2.1 Linear Weighted Least Squares Formulation

The aforementioned goal of orbit determination, and the Method of Least Squares is to find a mathematical model which will allow us to predict the motion and location of an object, given a set of observations. In its simplest form, consider a set of data given in two dimensional coordinates (x,y) , and it is desired to create a mathematical model, or function of x of the form (3.1), to predict y [50]. The reader should note that the following derivation is a synthesis of references [2], [4], [39], and [50], but primarily from Battin [4]. An interesting linear algebra approach to the least squares problem can also be found in Strang [63], but is not presented here. Effort has been made to minimize duplication of variables used previously in Chapter 2, and to standardize new variables.

$$f(x_1) \approx y_1, f(x_2) \approx y_2, \dots, f(x_n) \approx y_n \quad (3.1)$$

The goal is to determine a function f , which fits the given observations as closely as possible, but for this simplified example, we will assume a solution in the form of a straight line (3.2) which “best fits” the data.

$$f(x) = mx + b \quad (3.2)$$

The overdetermined system of equations given in (3.3), describe the data points that our function should match. Unless the data points lie on the same line, however, (3.3) has no unique solution and necessitates an approximation method such as least squares. Rewriting Equation (3.3) in matrix form yields the basic set of equations given in (3.4), where state vector \mathbf{X} must be determined so that the left side (\mathbf{MX}) is as close as possible to the right (\mathbf{Y}).

$$\begin{aligned} y_1 &= mx_1 + b \\ y_2 &= mx_2 + b \\ &\vdots \\ y_n &= mx_n + b \end{aligned} \quad (3.3)$$

$$\mathbf{MX} = \mathbf{Y}, \text{ where: } \mathbf{M} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}, \mathbf{X} = \begin{bmatrix} m \\ b \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (3.4)$$

In order to determine the best \mathbf{X} solution, the residual e is used as a metric. Simply put, the residual is the difference (or error) between the computed and actual value at a given point. For the straight line example, the residuals can be defined as the vertical distances between the data points and the line defined by \mathbf{X} , as shown in Equation (3.5).

$$e_i = |y_i - f(x_i)|, \quad i = 1, 2, \dots, n; \quad (3.5)$$

$$\mathbf{E}_x = \mathbf{MX} - \mathbf{Y}$$

The best fit line will minimize the sum of all the errors. To avoid sign complications, the sum of the squares of all the errors, given by the function J defined in Equation (3.6). Equation (3.6) offers both the standard and matrix form of the equation but the reader should note, that the remainder of this derivation is done in matrix form to simplify calculations.

$$J = e_1^2 + e_2^2 + \dots + e_n^2 = \sum_{i=1}^n [y_i - mx_i - b]^2 \quad (3.6)$$

$$\mathbf{J} = \mathbf{E}_x^T \mathbf{E}_x$$

Before minimizing \mathbf{J} , consider the case, where the observations are from different sources, each with a given accuracy. As such, the observations should be weighted according to the accuracy of each station to obtain the best solution. Using a weighting matrix \mathbf{W} , given in Equation (3.7), the updated version of \mathbf{J} is given in Equation (3.8).

$$\mathbf{W} = \begin{bmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_n \end{bmatrix} \quad (3.7)$$

$$\begin{aligned} \mathbf{J} &= \mathbf{E}_x^T \mathbf{W} \mathbf{E}_x = (\mathbf{X}^T \mathbf{M}^T - \mathbf{Y}^T) \mathbf{W} (\mathbf{M} \mathbf{X} - \mathbf{Y}) \\ &= \mathbf{X}^T \mathbf{M}^T \mathbf{W} \mathbf{M} \mathbf{X} - \mathbf{Y}^T \mathbf{W} \mathbf{M} \mathbf{X} - \mathbf{X}^T \mathbf{M}^T \mathbf{W} \mathbf{Y} + \mathbf{Y}^T \mathbf{W} \mathbf{Y} \end{aligned} \quad (3.8)$$

The next step is to minimize \mathbf{J} by taking the partial with respect to the solve-for vector \mathbf{X} and setting it to 0, as shown in (3.9). Simple matrix manipulation yields Equation (3.10), the weighted least squares solution of the overdetermined system in Equation (3.3). The reader should note Battin's conclusion that "solving a set of

overdetermined linear algebraic equations using the pseudo-inverse method is formally equivalent to Gauss' Method of Least Squares [39]."

$$\begin{aligned}\frac{\partial \mathbf{J}}{\partial \mathbf{X}} &= 2\mathbf{X}^T \mathbf{M}^T \mathbf{W} \mathbf{M} - 2\mathbf{Y}^T \mathbf{W} \mathbf{M} = \mathbf{0}^T \\ \mathbf{M}^T \mathbf{W} \mathbf{M} \mathbf{X} &= \mathbf{M}^T \mathbf{W} \mathbf{Y}\end{aligned}\tag{3.9}$$

$$\mathbf{X} = (\mathbf{M}^T \mathbf{W} \mathbf{M})^{-1} \mathbf{M}^T \mathbf{W} \mathbf{Y}\tag{3.10}$$

The matrix \mathbf{P} , defined as $(\mathbf{M}^T \mathbf{W} \mathbf{M})^{-1}$, is known as the *covariance matrix*. To better understand the elements of the covariance matrix, it is first necessary to define the *Probability Mass Function* (PMF) for the discrete case, or equivalently for the continuous case, the *Probability Density Function* (PDF). A PMF/PDF is simply a measure of the probability that the experimental value of a random variable, x , obtained on a performance of a specific experiment is equal to x_0 [64]. The *variances*, which are the diagonal elements of the covariance matrix, are then defined as the measure of the second power of the spread of the PMF (or PDF) for random variable x about its expected value. The *standard deviation*, which is the square root of variance, characterizes the spread of the PMF/PDF around the expectation of x [64]. In other words, the variance is the expected square of the error between the actual and computed trajectory, and the standard deviation characterizes the quality of the fit of the computed trajectory to the observations. The covariance terms, or correlation coefficients, are the off-diagonal terms in the covariance matrix which describe the degree of independence among the elements of the state vector (0.0 indicates ideal independence, 1.0 total dependence, and negative terms indicate an inverse relationship) [2].

3.2.2 Nonlinear Least Squares Formulation

Unfortunately, Gauss' Method of Least Squares was designed for linear systems, and real-world orbit determination problems, especially catastrophic decay cases, are simply nonlinear. To correct for this, the problem can be linearized and the resulting approximate solution can be iterated to refine the answer in a DC process. The problem can be seen by defining a function \mathbf{f} of a set of parameters, like Equation (3.2), only using a nonlinear function such as $y = m \cdot \sin(x+b)$. This function cannot be separated into two variables (m and b in the linear case), and the function is therefore nonlinear in b . To obtain an estimate of the new parameters, an alternate method must be employed.

As described in reference [4] (the primary source for this section), for the nonlinear problem, \mathbf{Y} denotes a set of m observations, \mathbf{f} is a known vector function of a set of p parameters (denoted by the p -dimensional \mathbf{X} vector), and \mathbf{Q} represents additive random noise. The result is the nonlinear regression equation, given in Equation (3.11), and the nonlinear orbit determination problem is to estimate \mathbf{X} , given \mathbf{Y} , the functional form of \mathbf{f} , and the statistical properties of \mathbf{Q} [4].

$$\mathbf{Y} = \mathbf{f}(\mathbf{X}) + \mathbf{Q} \quad (3.11)$$

Similar to the linear method, the estimation process involves determining an estimate for \mathbf{X} that minimizes the weighted sum of the residual squares, \mathbf{J} . An analog to Equation (3.9), in nonlinear form, is:

$$\frac{\partial \mathbf{J}}{\partial \mathbf{X}} = -2[\mathbf{Y} - \mathbf{f}(\mathbf{X})]^T \mathbf{W} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{X}} \right) = \mathbf{0}^T \quad (3.12)$$

To solve this nonlinear minimization, (3.12) is linearized by expanding $\mathbf{f}(\mathbf{X})$ in a truncated Taylor series about the a priori estimate \mathbf{X}_0 as shown in Equation (3.13), where $\Delta \mathbf{X}$ and \mathbf{F} are defined in Equations (3.14) and (3.15). (3.15b) is an example of the \mathbf{F}

matrix for observation vector triplets of the form $(\rho_i, \beta_i, \gamma_i)$, and (3.15c) is the approximation used as the result of linearization.

$$\mathbf{f}(\mathbf{X}) = \mathbf{f}(\mathbf{X}_0) + \mathbf{F}(\Delta\mathbf{X}) \quad (3.13)$$

$$\Delta\mathbf{X} = \mathbf{X} - \mathbf{X}_0 \quad (3.14)$$

$$\mathbf{F} = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{X}} \right)_{\mathbf{X}=\mathbf{X}_0} \quad (3.15a)$$

$$\mathbf{F} = \begin{bmatrix} \frac{\partial \rho_1}{\partial r_{I_0}} & \frac{\partial \rho_1}{\partial r_{J_0}} & \frac{\partial \rho_1}{\partial r_{K_0}} & \frac{\partial \rho_1}{\partial v_{I_0}} & \frac{\partial \rho_1}{\partial v_{J_0}} & \frac{\partial \rho_1}{\partial v_{K_0}} \\ \frac{\partial \beta_1}{\partial r_{I_0}} & \frac{\partial \beta_1}{\partial r_{J_0}} & \frac{\partial \beta_1}{\partial r_{K_0}} & \frac{\partial \beta_1}{\partial v_{I_0}} & \frac{\partial \beta_1}{\partial v_{J_0}} & \frac{\partial \beta_1}{\partial v_{K_0}} \\ \frac{\partial \gamma_1}{\partial r_{I_0}} & \frac{\partial \gamma_1}{\partial r_{J_0}} & \frac{\partial \gamma_1}{\partial r_{K_0}} & \frac{\partial \gamma_1}{\partial v_{I_0}} & \frac{\partial \gamma_1}{\partial v_{J_0}} & \frac{\partial \gamma_1}{\partial v_{K_0}} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \gamma_n}{\partial r_{I_0}} & \dots & \dots & \dots & \dots & \frac{\partial \gamma_n}{\partial v_{K_0}} \\ \frac{\partial r_{I_0}}{\partial r_{I_0}} & & & & & \frac{\partial v_{K_0}}{\partial v_{K_0}} \end{bmatrix} \quad (3.15b)$$

$$\mathbf{F} = \begin{bmatrix} \frac{\Delta \rho_1}{\Delta r_{I_0}} & \dots & \frac{\Delta \rho_1}{\Delta v_{K_0}} \\ \frac{\Delta \beta_1}{\Delta r_{I_0}} & \dots & \frac{\Delta \beta_1}{\Delta v_{K_0}} \\ \frac{\Delta \gamma_1}{\Delta r_{I_0}} & \dots & \frac{\Delta \gamma_1}{\Delta v_{K_0}} \\ \vdots & \ddots & \vdots \\ \frac{\Delta \gamma_n}{\Delta r_{I_0}} & \dots & \frac{\Delta \gamma_n}{\Delta v_{K_0}} \\ \frac{\Delta r_{I_0}}{\Delta r_{I_0}} & & \frac{\Delta v_{K_0}}{\Delta v_{K_0}} \end{bmatrix}_{(3*n) \times 6} \quad (3.15c)$$

The linearized observation vector becomes:

$$\Delta\mathbf{Y} = \mathbf{F}(\Delta\mathbf{X}) + \mathbf{Q} \quad (3.16)$$

Substituting (3.13) and (3.16) into (3.12), the linearized partial derivative of the loss function in (3.12) can then be described by Equation (3.17).

$$-2(\Delta\mathbf{Y} - \mathbf{F}(\Delta\mathbf{X}))^T \mathbf{W}\mathbf{F} = \mathbf{0} \quad (3.17)$$

(3.17) can then be solved for $\Delta\mathbf{X}$ to yield the classical equation for the best estimate given in Equation (3.18), where the updated value of the estimate, \mathbf{X}_u , is defined in Equation (3.19).

$$\Delta\mathbf{X}_u = (\mathbf{F}^T \mathbf{W}\mathbf{F})^{-1} \mathbf{F}^T \mathbf{W}\Delta\mathbf{Y} \quad (3.18)$$

$$\mathbf{X}_u = \mathbf{X}_0 + \Delta\mathbf{X}_u \quad (3.19)$$

To maintain linearity after the linearization, $\Delta\mathbf{X}_u$ must be small. Therefore, the apriori estimate \mathbf{X}_0 , should be relatively close to the true solution of the least squares problem. A differential correction process (GTDS uses a Newton-Raphson procedure) is then used to iteratively improve the solution estimate, each time using the last best estimate \mathbf{X}_u as a reference for the linearization. The iterations continue until the update vector ($\Delta\mathbf{X}_u$) approaches zero within a given tolerance level (or another predetermined metric).

3.3 Sequential Batch Least Squares Methods

The Method of Least Squares offers an adequate method for processing a batch of observations, but for a real-time orbit determination problem, it suffers from a fundamental flaw. If a Least Squares fit is applied to a specific batch of n observations, and ten new observations then become available, the user is forced to run the fit again for $(n+10)$ observations. Given the computationally expensive process involved with the Method Least Squares, this option of reprocessing observations is unacceptable. An alternative to classical Least Squares methods, are Sequential Batch Least Squares, and

Kalman Filter Techniques. These techniques process observations recursively and avoid the pitfall of redundant processing. They include the Sequential Batch Estimator (which simply steps the epoch along with successive batch fit runs), Kalman Filter (all filter processes update the epoch to the current time), and Extended Kalman Filter methods. Kalman Filters enjoy the following advantages over their Least Squares counterparts [4], [65]:

- The estimation procedure can be interrupted at any time and the best estimates of the state parameters (and their uncertainties) based on all accumulated data to that time are known.
- At each step the calculations are fixed in size and format (they don't grow with the number of observations, or change form).
- The need for storing previous data points is eliminated by the use of recursion.
- They make immediate use of new observations.
- Sequential processes are better suited for processing objects in highly dynamic environments, such as the catastrophic decay cases. For cases where highly varying parameters (such as the coefficient of drag) are held constant in the batch filter and cause increasing error due to their rapidly changing values. Sequential methods estimate a new value for these parameters with each new observation. In doing so, sequential processes improve state estimation and future processing in these dynamic environments.

These techniques begin by making a best guess initial estimate of the state vector from a minimum set of observations. Each new observation is then combined with the previous parameter estimates by appropriately weighting the observation to give an improved estimate of the state parameters. This process is repeated for each new observation to improve the parameter estimates. A filtering algorithm must be chosen to

specify exactly how to determine a new estimate after the receipt of each new observation, and for R&D GTDS, the sequential filtering algorithms used include the Linearized and Extended Kalman Filters, discussed in Section 3.4.

3.4 Kalman Filters

Perhaps the most widely known sequential type filter is the Kalman Filter developed in 1959 by Kalman and Bucy [66], [67]. Although a similar concept was developed almost simultaneously by Battin for a guidance algorithm used in the Apollo space program [39], the greater generality offered by Kalman's method quickly propelled the Kalman Filter to a wide use in estimation theory. Lefferts, Markley, and Shuster describe the Kalman Filter as a filter which "uses a dynamical model for the time development of the system and a model of the sensor measurements to obtain the most accurate estimate possible of the system state using a linear estimator based on present and past measurements [68]." Vallado describes two very important differences from the least squares method, noting that the Kalman filter continuously updates the epoch time and carries all the information about past measurements in its current "maximum-likelihood" state and covariance estimates [2]. This characteristic, described in Section 3.3, alleviates the need to reprocess the old data after each observation and makes the filter much more accurate and efficient. When compared with Least Squares methods, the Kalman Filter enjoys the following advantages [2], [69]:

- It estimates the state and covariance, including all new observations, at the current state. This not only allows the filter to process through a maneuver (such as the perigee passage of catastrophic decay cases), but also effectively creates the capability for real-time on-board error analysis.

- It can make use of human judgement. The reader may consider that a mixed blessing, but in most circumstances, the high degree of flexibility inherent in the filter allows the user to tailor it to meet the specific needs of a given problem.
- It allows the user to solve for observable measurement biases simultaneously in the state space. With enough data present, a filter can more accurately adapt to the time-varying nature of sensor biases.
- It allows the user to incorporate new values without reprocessing all data to date, as with a traditional Least Squares method.
- It models time-varying coefficients, (such as drag, solar radiation pressure, or aerodynamic lift) much more effectively than batch methods. Batch methods hold these coefficients constant over the observation span which, for highly dynamic variables (like the coefficient of drag in catastrophic decay cases), can lead to significant error. The filter updates parameters at each step and therefore avoids these errors.

Unfortunately, the Kalman Filter suffers from some weaknesses as well. These include:

- The covariance matrix can become too small, and cause the filter to become *smug*. A smug filter will then trust itself more than new data, and diverge. As the filter gain (K) decreases, new observations exert less influence on the solutions. Several methods exist to overcome this weakness, some of which will be discussed in Section 3.4.4.
- Depending on the formulation, filters can become computationally expensive. The primary source of this burden is computation of the covariance and partial derivative matrices (which are usually integrated numerically). Sub-optimal

filters have been used in the past for high data rate problems, but current computational power has nearly negated their need.

- It isn't self-starting and requires an initial estimate of the state, covariance, process noise, and measurement error. Depending on the problem, acquiring these estimates can be difficult.
- It's very susceptible to proper modeling of system dynamics. There is obviously a tradeoff between complexity of the model used and computer efficiency, but a worse problem is when the exact model of the system isn't well known as with catastrophic decay cases. Accounting for improperly modeled dynamics can be accomplished with a well-developed noise model.

As mentioned above, one of the main weaknesses of filtering is filter divergence which can be caused by several factors, including [4]:

- Linearization errors (such as from measurement linearization)
- Computational errors (resulting from things such as the loss of semi-definiteness by the covariance matrix)
- Unknown noise statistics
- Modeling errors

Linearization errors are typically minimized by iterating the solution. Computational procedures can be minimized with new algorithms (such as Potter's square root formulation as presented by Battin which avoids the loss of semi-definiteness [39]), and software coding techniques [70]. Noise statistics and modeling errors can be minimized by several different techniques, and for the catastrophic decay problem, are essential to the successful implementation of the filter. These techniques will be addressed in Section 3.4.4. In addition, the specific method used in this thesis to model error for the catastrophic decay problem, is outlined in Chapter 6.

First, however, Section 3.4.1 follows Vallado's [2] derivation of the generalized Kalman Filter equations. Sections 3.4.2 and 3.4.3 identifies the specific equations of the Linearized and Extended Filter equations, respectively. Simply stated, the Linearized Filter uses a reference trajectory and calculates the differences to that trajectory to provide the necessary state and covariance information for propagation. In contrast, the Extended Filter uses a current state estimate for each propagation instead of a single reference trajectory. This contrariety lends itself to a dichotomy in the decay profile that will be discussed in Chapter 7.

3.4.1 Generalized Filter Equations

To avoid confusion in terminology from the Least Squares Section, the generalized filter equations will be derived from scratch using, whenever possible, like notation. The following derivation is taken primarily from Vallado [2], as well as reference [4] and [71]. The first step is to linearize the equations of motion for two trajectories whose states are represented by \mathbf{X} and \mathbf{Y} and, defined by Equations (3.20) and (3.21), where f is a function of the state.

$$\mathbf{X}(t_0) = \mathbf{X}_0 \quad \dot{\mathbf{X}} = \bar{f}(\mathbf{X}) \quad (3.20)$$

$$\mathbf{Y}(t_0) = \mathbf{Y}_0 \quad \dot{\mathbf{Y}} = \bar{f}(\mathbf{Y}) \quad (3.21)$$

A simplified approximation, as a result of the linearization, of the difference between \mathbf{X} and \mathbf{Y} is given in Equation (3.22).

$$\mathbf{Y} = \mathbf{X} + \delta\mathbf{x} \quad (3.22)$$

The time derivative of \mathbf{Y} , $f(\mathbf{X} + \delta\mathbf{x})$, is nonlinear so, as was done in Section 3.2.2, we expand the function in a Taylor's series about \mathbf{X} , to yield (3.23).

$$\dot{\mathbf{Y}} = \bar{f}(\mathbf{X}) + \frac{\partial \bar{f}(\mathbf{X})}{\partial \mathbf{X}} \delta \mathbf{x} + \frac{\partial^2 \bar{f}(\mathbf{X})}{2! \partial \mathbf{X}^2} \delta \mathbf{x}^2 + \dots \quad (3.23)$$

Substitution of the time derivative of (3.22) for the left hand side, the time-dependent \mathbf{F} matrix (3.15 - which represents the partials of $f(x)$), and introduction of the \mathbf{u} vector (which represents neglected terms of second order and higher resulting from the truncation of the Taylor's series in (3.23)), yields Equation (3.24).

$$\delta \dot{\mathbf{x}} = \frac{\partial \bar{f}(\mathbf{X})}{\partial \mathbf{X}} \delta \mathbf{x} + \mathbf{u} = \mathbf{F}(t) \delta \mathbf{x} + \mathbf{u} \quad (3.24)$$

The solution of (3.24) represents the time-varying difference between \mathbf{X} and \mathbf{Y} , and is assumed to be of the form shown in Equation (3.25), where Φ is defined as the *State Transition Matrix*.

$$\delta \mathbf{x} = \Phi(t, t_0) \delta \mathbf{x}_0 \quad (3.25)$$

The State Transition Matrix propagates the state differences from t_0 to t . Some important properties of Φ are given in (3.26). Note that Φ must be invertible.

$$\begin{aligned} \Phi(t, t_0) &= \Phi^{-1}(t_0, t) \\ \Phi(t_1, t_0) &= \Phi(t_1, t) \Phi(t, t_0) \\ \Phi(t_1, t_0) &= \frac{\delta \mathbf{x}_1}{\delta \mathbf{x}_0} \end{aligned} \quad (3.26)$$

Temporarily neglecting the \mathbf{u} vector, and substituting a time differentiated (3.24) into (3.25) yields Equation (3.27a), with an initial condition given in (3.27b).

$$\dot{\Phi}(t, t_0) = \mathbf{F}(t) \Phi(t, t_0) \quad (3.27a)$$

$$\Phi(t_0, t_0) = \mathbf{I} \quad (3.27b)$$

(3.27a) represents a first order differential equation which can be numerically integrated with the equations of motion to determine Φ , while \mathbf{F} is determined by

calculating the partial derivative of the acceleration vector. The vector \mathbf{u} is determined with a simplifying assumption that \mathbf{u} is constant over small time intervals. Applying Euler's method yields (3.28a), and with Φ , allows the user to map error to any time in the orbit. The total effect on the state can then be determined by summing these contributions over the specified time interval as shown in Equation (3.28b), where \mathbf{v} represents the *process noise*, or uncertainty in the dynamics modeling used in the propagation process.

$$\delta\mathbf{x} = \mathbf{u}\delta t \quad (3.28a)$$

$$\mathbf{v} = \int_{t_0}^t \Phi(t, t_0) \mathbf{u} dt \quad (3.28b)$$

The vector \mathbf{v} represents the noise due to linearization and improper modeling, and it has a 0 mean. Its introduction results in a linear approximation to the propagated state update of the form given in Equation (3.29).

$$\delta\bar{\mathbf{x}}_{k+1} = \Phi\delta\hat{\mathbf{x}}_k + \mathbf{v} \quad (3.29)$$

Remembering that covariance is the expected value of the square of the errors in the state, and its predicted value is based on the predicted state update, (3.29) can be used to determine the covariance update equation given in (3.30), where $E\{ \}$ represents the expectation operator, bars represent predicted values, and hats represent best estimates.

$$\bar{\mathbf{P}}_{k+1} = E\left\{ (\Phi\delta\hat{\mathbf{x}}_k + \mathbf{v})(\Phi\delta\hat{\mathbf{x}}_k + \mathbf{v})^T \right\} \quad (3.30)$$

Assuming that the state update and process noise are uncorrelated yields (3.31), where \mathbf{Q} represents the second moment of the process noise (accounts for the error resulting from the \mathbf{v} model in propagating the covariance matrix through time).

$$\bar{\mathbf{P}}_{k+1} = \Phi\hat{\mathbf{P}}_k\Phi^T + \mathbf{Q} \quad (3.31)$$

Use of equations (3.29) and (3.31) produce predictions of the state, state update, and covariance at the current time. The next step is to develop expressions, and an algorithm for using the expressions, to refine predictions and ultimately update the state. The state correction estimated from all observations through the current time, using the notation from Section 3.2.2, is given in Equation (3.18). Equation (3.18) can then be reduced to (3.32a) using (3.27), (3.29), and (3.15). The \mathbf{H} matrix is the analog of \mathbf{F} at the current state (instead of the epoch state). $\tilde{\mathbf{E}}_x$ is the residual matrix defined in (3.5) and updated with current observations. In light of these substitutions, Equation (3.32a) represents the state correction estimate.

$$\delta\hat{\mathbf{x}}_{k+1} = \left(\mathbf{H}^T \mathbf{W} \mathbf{H} + \bar{\mathbf{P}}_{k+1}^{-1} \right)^{-1} \left(\mathbf{H}^T \mathbf{W} \tilde{\mathbf{E}}_x + \bar{\mathbf{P}}_{k+1}^{-1} \delta\bar{\mathbf{x}}_{k+1} \right) \quad (3.32a)$$

In the pursuit of efficiency, a mathematical relation known as the Matrix Inversion Lemma was developed to eliminate the need for inverting the 6x6 matrix of the covariance update. Generally attributed to Georg Ferdinand Frobenius [39], the general Lemma expression (using \mathbf{A}_{nn} , \mathbf{B}_{nm} , and \mathbf{C}_{mm} as dummy matrices) is shown in (3.32a) [39]. The reader should note that while the dimension of the inverted matrix on the left is $n \times n$, the right side it is a $m \times m$ matrix. For the covariance equation, that results in calculating the inverse of a 3x3 matrix instead of a 6x6.

$$\left(\mathbf{A}^{-1} + \mathbf{B} \mathbf{C}^{-1} \mathbf{B}^T \right)^{-1} = \mathbf{A} - \mathbf{A} \mathbf{B} \left(\mathbf{C} + \mathbf{B}^T \mathbf{A} \mathbf{B} \right)^{-1} \mathbf{B}^T \mathbf{A} \quad (3.32b)$$

By defining the inverse of the weighting matrix as \mathbf{R} , the measurement noise matrix (the equivalent of the \mathbf{C} matrix in (3.32b)), and noting that \mathbf{A} is the equivalent of the covariance matrix, and \mathbf{B} is the transpose of the \mathbf{H} matrix, Equation (3.33) follows easily.

$$\hat{\mathbf{P}} = \bar{\mathbf{P}} - \bar{\mathbf{P}} \mathbf{H}^T \left(\mathbf{W}^{-1} + \mathbf{H} \bar{\mathbf{P}} \mathbf{H}^T \right)^{-1} \mathbf{H} \bar{\mathbf{P}} \quad (3.33)$$

Next, define the Kalman Gain Matrix (\mathbf{K}) as shown in (3.34) to produce the final version of the error covariance estimate equation in (3.35).

$$\mathbf{K} \equiv \hat{\mathbf{P}}\mathbf{H}^T\mathbf{W} = \bar{\mathbf{P}}\mathbf{H}^T(\mathbf{W}^{-1} + \mathbf{H}\bar{\mathbf{P}}\mathbf{H}^T)^{-1} \quad (3.34)$$

$$\hat{\mathbf{P}} = \bar{\mathbf{P}} - \mathbf{K}\mathbf{H}\bar{\mathbf{P}} = [\mathbf{I} - \mathbf{K}\mathbf{H}]\bar{\mathbf{P}} \quad (3.35)$$

The original complex inversion of a nxn matrix is now reduced to the inversion of a scalar quantity for a single observation [71]. Final substitution of (3.35) into the estimated state correction expression, expansion of the Kalman Gain Matrix, identification of an identity matrix relation, and reordering of terms results in the final form of the state update estimate equation, given in (3.36).

$$\delta\hat{\mathbf{x}} = \delta\bar{\mathbf{x}} + \mathbf{K}(\tilde{\mathbf{E}}_x - \mathbf{H}\delta\bar{\mathbf{x}}) \quad (3.36)$$

3.4.2 Linearized Kalman Filter

The Linearized Kalman Filter (LKF) is similar to Encke's Method in that it generates a reference trajectory from the apriori initial state and calculates the state error as the difference of this reference trajectory and the predicted trajectory. The LKF updates the state vector and covariance matrix, but not the reference orbit, to maximize computational speed. As a result, any significant perturbations or modeling errors which cause the orbit to change, can quickly cause the LKF to diverge. Rectification points can be inserted in the algorithm, but at a high computational cost.

The following algorithm, presented by Vallado [2] and cross checked with the GTDS Filter Math Specifications [71], represents the computational procedure for the

LKF using the initial state, covariance, \mathbf{Q} matrix, measurement noise matrix \mathbf{R} , and observations as inputs.

1. **Calculate** the necessary partials matrices (\mathbf{F} and \mathbf{H}), and the State Transition Matrix Φ . The reader should note that Vallado presents Equation (3.37) as a more accurate technique for determining Φ , which can be obtained through the numerical integration of (3.37) or (3.27).

$$\Phi(t_{k+1}, t_k) = \mathbf{I} + \int_{t_k}^{t_{k+1}} \mathbf{F}(t_{k+1}) \Phi(t_{k+1}, t_k) dt \quad (3.37)$$

2. **Predict** the state (3.38), state update (3.39), and error covariance (3.40). The reader should note that (3.38) is only an approximation, and a propagation technique should be used to advance the epoch state to the current time.

$$\bar{\mathbf{X}}_{k+1} \cong \Phi \hat{\mathbf{X}}_0 \quad (3.38)$$

$$\delta \bar{\mathbf{x}}_{k+1} = \Phi \delta \hat{\mathbf{x}}_k \quad (3.39)$$

$$\bar{\mathbf{P}}_{k+1} = \Phi \hat{\mathbf{P}}_k \Phi^T + \mathbf{Q} \quad (3.40)$$

3. **Update** the Kalman Gain (3.41), state update estimate (3.42), error covariance update (3.43), and state estimate (3.44), to complete the process.

$$\mathbf{K}_{k+1} = \bar{\mathbf{P}}_{k+1} \mathbf{H}_{k+1}^T (\mathbf{R}_k + \mathbf{H}_{k+1} \bar{\mathbf{P}}_{k+1} \mathbf{H}_{k+1}^T)^{-1} \quad (3.41)$$

$$\delta \hat{\mathbf{x}}_{k+1} = \delta \bar{\mathbf{x}}_{k+1} + \mathbf{K}_{k+1} (\tilde{\mathbf{E}}_x|_{k+1} - \mathbf{H}_{k+1} \delta \bar{\mathbf{x}}_{k+1}) \quad (3.42)$$

$$\hat{\mathbf{P}}_{k+1} = \bar{\mathbf{P}}_{k+1} - \mathbf{K}_{k+1} \mathbf{H}_{k+1} \bar{\mathbf{P}}_{k+1} \quad (3.43)$$

$$\hat{\mathbf{X}}_{k+1} = \bar{\mathbf{X}}_{k+1} + \delta \hat{\mathbf{x}}_{k+1} \quad (3.44)$$

3.4.3 Extended Kalman Filter

As mentioned above, the LKF is not typically the ideal filter for truly nonlinear systems such as real world orbit determination problems. The Extended Kalman Filter (EKF) continually updates the reference trajectory, as opposed to maintaining one constant reference trajectory. The trajectory update is based on the latest state estimate so the update computations must be done in real time. As a result, the state update prediction is effectively zero because there is no state update to propagate (the current state is the current best guess). Restarting the integrator at each observation time (to recompute the new reference trajectory), places a higher demand on computer resources. Fortunately, today's computer resources are fast, cheap, and available, making the application of the EKF to real world problems increasingly more feasible. Figures 3.1 and 3.2 graphically demonstrate the update process for the LKF and EKF respectively. The reader should note how the LKF references each update to the original reference trajectory, while the EKF actually uses the update to continually improve the reference orbit.

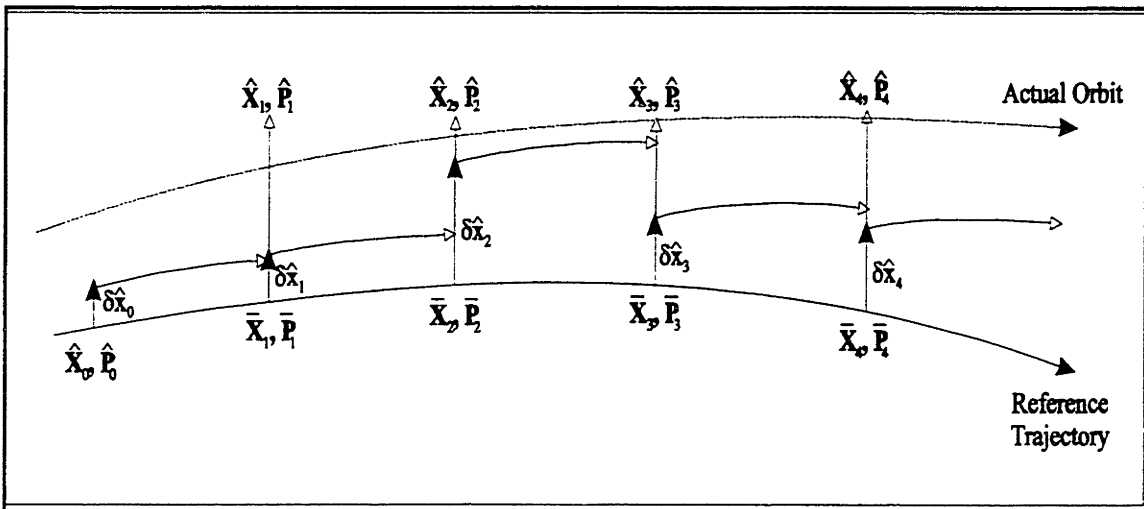


Figure 3.1 Representation of a Linearized Kalman Filter [2]

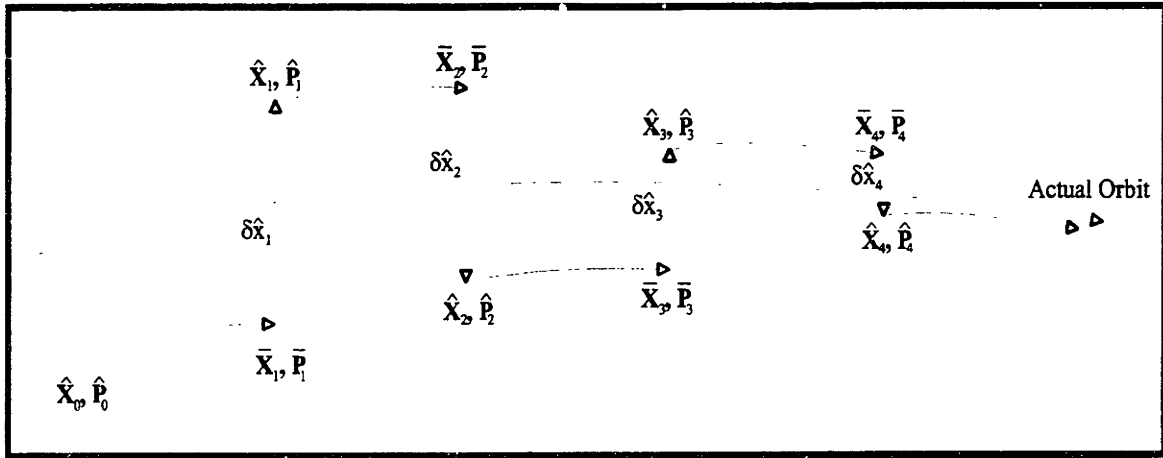


Figure 3.2 Representation of an Extended Kalman Filter [2]

The following algorithm, also presented by Vallado [2] and cross checked with the GTDS Filter Math Specifications [71], represents the computational procedure for the EKF using the initial state, covariance, \mathbf{Q} matrix, measurement noise matrix \mathbf{R} , and observations as inputs.

1. **Calculate** the necessary partials matrices (\mathbf{F} and \mathbf{H}), and the State Transition Matrix Φ . There are no differences in the calculation of these matrices for the EKF compared to the LKF, other than the fact that the \mathbf{F} matrix must be recalculated with each new step due to an essentially changing epoch.
2. **Predict** the state, state update, and error covariance as in the LKF algorithm, with the exception of the state update equation in (3.45). Care should also be taken to choose a propagation technique which has a simple formulation and computationally inexpensive initializations, because it's at each observation time.

$$\delta \bar{\mathbf{x}}_{k+1} = 0 \quad (3.45)$$

3. *Update* the Kalman Gain, error covariance update, and state estimate, using the same equations as shown in the LKF algorithm ((3.41), (3.43), and (3.44) respectively). The state update estimate is calculated using Equation (3.46).

$$\delta \hat{\mathbf{x}}_{k+1} = \mathbf{K}_{k+1} \tilde{\mathbf{E}}_{\mathbf{x}} \Big|_{k+1} \quad (3.46)$$

3.4.4 Error Modeling

In order to successfully use a filter, the squared errors should be comparable to the computed variances. If the covariance is too small, the filter becomes ‘smug,’ rejects new data, and doesn’t make the required corrections to approximate the orbit. If the covariance is too large, the filter over-compensates for errors, begins unstable oscillations, and eventually diverges. Process noise is used to overcome the dynamic modeling errors which cause the covariance to stray from acceptable bounds. There are two main methods for addressing modeling errors, non-adaptive and adaptive methods. *Non-adaptive* methods modify the filter structure to artificially inflate process noise and maintain the Kalman Gain at a suitable level to continue filter operation. *Adaptive* methods can be further distinguished as structural and statistical techniques. Structural (dynamic model compensation) methods are designed to adaptively estimate the true unmodeled acceleration along with the state, whereas statistical correct the basic filter to accommodate the combined effect of all known errors inherent in the orbit determination problem, such as neglected nonlinearities, unknown noise statistics, model errors, and computational error effects [4]. Examples of each technique, and sources the reader may investigate for more detail, are listed below.

- ***Non-Adaptive Methods:***
 - Dowd's white noise model, using constant strength rate which is set by trial and error or physical considerations [72].
 - The Modified Extended Kalman Filter (MEKF) by Torroglosa, which also uses constant strength rates and fading memory to compensate [73].
- ***Adaptive Structural Methods:***
 - Wright analytic development of modeling errors based on geodetic error analysis [74], and Hujsak's extension of this work [75].
 - Myer's modeling the process noise as a first order Markov process, with the initial conditions and dynamics selected for optimum performance [76]. Tapley, Ingram, and Schutz have utilized similar methods [77], [78], [79].
- ***Adaptive Statistical Methods:***
 - The Jazwinski Filter [80] is an example of a statistical filter which determines the mean and covariance of both the state and observation noise to gain better estimates of the state [4].

GTDS primarily uses the first method because most orbits can take advantage of assuming a near-linearity and slowly-varying elements. Currently, GTDS uses two variants of non-adaptive techniques which have been very successful with most orbit types; fading memory and a constant process noise rate. There are certain situations (Molniya orbits for example) where the orbit is so dynamic that it becomes desirable to weight the current observations more than previous ones. One method to achieve this effect is to multiply the entire covariance matrix by a memory factor as shown in Equations (3.47a) and (3.47b), where Δt is the time step between observations and τ is the memory time constant, chosen so that s is greater than or equal to one [71].

$$s = e^{\frac{\Delta t}{\tau}} \quad (3.47a)$$

$$\bar{\mathbf{P}}_{k+1} = s\Phi\hat{\mathbf{P}}_k\Phi^T \quad (3.47b)$$

Fading memory amplifies the entire covariance by the multiplicative factor s . In contrast, GTDS's constant process noise matrix rate method provides a minimum value for the covariance components corresponding to the nonzero noise matrix components. A user-defined process noise rate matrix $\dot{\mathbf{Q}}$ is multiplied by the time between observations and added to the covariance estimation equation, (3.48a) and (3.48b). Taylor [65], outlines a detailed algorithm for obtaining estimates of the process noise rate defined by the Q-dot matrix. Taylor's estimate of the $\dot{\mathbf{Q}}$ matrix from an orbit type similar to catastrophic decay cases, is used for NSSC-13390 during the highly dynamic month of February, which will be discussed in Chapter 8.

$$\mathbf{Q} = \Delta t \dot{\mathbf{Q}} \quad (3.48a)$$

$$\bar{\mathbf{P}}_{k+1} = \Phi\hat{\mathbf{P}}_k\Phi^T + \mathbf{Q} \quad (3.48b)$$

While these methods seem to function adequately for most orbit types, they are ineffective for catastrophic decay orbits. As a result, two improved non-adaptive noise models have been created to improve noise estimates by isolating process noise to certain regions of the orbit. A derivation of these models is included in Section 6.4.

3.5 Performance Metrics

An important part of any scientific analysis is to have metrics to measure the performance of experiment variables. These metrics can characterize the system with a minimal number of statistics, as well as giving a check for iteration control. There is no

limit to the number and types of metrics that an investigator may use to present data, but some of the most common (used in GTDS after each DC run – [4]) are presented here.

- **Root Mean Square (RMS):** the square root of the mean squared expected value of the observations residuals, as shown in Equation (3.49), where E is the expectation operator, and n is the number of observations.

$$RMS \equiv \sqrt{E(x^2)} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n x_i^2} \quad (3.49)$$

- **Mean:** represents the expected value of a function f(x) and is simply the weighted sum of all possible experimental values of f(x), each weighted by its probability of occurrence on a performance of the experiment [64]. Stated another way, the mean represents the first moment, or center of mass of the given PDF or PMF discussed in Section 3.2.1. Using p_x as the PDF/PMF, and summing over all experimental values x_0 yields Equation (3.50).

$$\bar{x} = E(x) = \sum_{x_0} x_0 p_x(x_0) \quad (3.50)$$

- **Variance:** As described in Section 3.2.1, the second central moment of the expectation measures the second power of the spread of the PDF or PMF around the expected value. Using E as the expectation operator, the variance is given in Equation (3.51).

$$\sigma^2 = E\{[x - E(x)]^2\} \quad (3.51)$$

- **Standard Deviation:** The positive square root of the variance and characterizes the spread of a PDF or PMF around the expected value, or mean. The *sample standard deviation* of each residual group is a measure of the dispersion of the observation data and is calculated by taking the square root of the variance divided by the number of residuals in the group, n_s .

- **Sum of Squares About Mean:** the sum of the squares of the residuals about the mean of each residual group. A group can be described as the total number of residuals for a station and/or data type.
- **Group Mean:** the mean value of each residual group is a measure of the bias in the observation and is calculated by dividing the sum of the residuals for a group by n_s , as shown in Equation (3.52). It's desirable for the group mean to be zero, which indicates that the measurement noise has zero mean.

$$\bar{m} = \frac{1}{n_s} \sum_{j=1}^{n_s} (E_x)_j \quad (3.52)$$

- **Confidence Interval for Group Mean:** the confidence, measured in probability, that the experimental variable will fall within a specified interval. If the observation residual group is normally distributed with a zero mean, the variable t described in Equation (3.53) has a student's t -distribution with $(n_s - 1)$ degrees of freedom. The confidence intervals for the mean can then be constructed from tables of the t -distribution [4]. In accordance with the Central Limit Theorem [64], as n_s becomes large, the t -distribution approaches the normal distribution, and confidence intervals can then be constructed using the normal distribution.

$$t = \frac{\bar{m}}{\sqrt{\frac{\sigma^2}{n_s}}} \quad (3.53)$$

- **Observation Residual Groups:** At the end of each DC run in GTDS, the following statistics are presented for each residual group:
 - Total number of observations, n_s .
 - The number of rejected and accepted observations.
 - The histograms of observations by true anomaly.

Chapter 4

The Research and Development Version of the Goddard Trajectory Determination System (R&D GTDS)

4.1 Introduction

In order to fully understand the results presented in this thesis, and to build upon this work in the future, it is necessary to explain the development, capabilities, and operation of the primary tool used in this investigation, the Research & Development version of the Goddard Trajectory Determination System (R&D GTDS). GTDS is a multipurpose computer system designed:

“To provide operational support for individual earth, lunar, and planetary space missions and for the research and development requirements of the various projects of the NASA/Goddard Space Flight Center scientific community [81].”

Since its development in the early 1970's, GTDS has undergone significant changes and technical improvements, but the primary functionality has remained the same. To accomplish the wide variety of its system objectives, GTDS is partitioned into the following nine programs (program keywords – abbreviations). For a more complete description of each program and its use, please refer to the source for this section, the *GTDS User's Guide* [82], [129].

- *Ephemeris Generation Program (EPHEM – EP)*
- *Ephemeris Generation Compare Program (COMPARE – CM)*
- *Differential Correction Program (DC – DC)*
- *Data Management Program (DATAMGT – DM)*
- *Data Simulation Program (DATASIM – DS)*
- *Error Analysis Program (ANALYSIS – EA)*
- *Early Orbit Determination Program (EARLYORB – EO)*
- *Permanent File Report Generation Program (FILERPT – FR)*
- *Filter Program (FILTER – FL)*

The *Ephemeris Generation Program (EPHEM)* propagates the satellite state and state partial derivatives from epoch over a given time interval. To accommodate a wide range of accuracy and efficiency requirements, the program can use any of the nineteen propagation techniques, discussed below. In addition, the program can generate the State Transition Matrix (STM) by analytic approximation or numerical integration, as discussed in Chapter 3. The Ephemeris program can create a variety of output files, including EPHEM data output, ORB1, and ORBIT files, which will be discussed in Section 4.3.3. These files can be used in conjunction with the other programs, or for orbital analysis by themselves.

The *Ephemeris Generation Compare Program (COMPARE)* compares two ephemerides generated by the EPHEM program in any of the three output file formats. Comparisons can be restricted to overlap intervals, or specific user-defined intervals to isolate observable effects. Differences in radial, cross-track, and along track measurements are presented in tabular format to a data file as well as corresponding plots. Plots of state variable differences can also be produced.

The *Differential Correction Program (DC)* estimates the values of specified solve-for parameters at a given epoch using a Least Squares fitting technique in conjunction with orbital observations. The parameters are constrained in the fitting process to satisfy a priori estimates to within a specified uncertainty. Solve-for parameters can include any combination of up to 20 variables such as state parameters, station locations and biases, and dynamic coefficients (drag, lift, solar radiation pressure, etc.). Output includes both first and second order statistics, like the mean and covariance matrix, as determined by the estimated parameters. The program also allows the user to tailor output options to include tabular residuals, various statistics, and residual plots after specified iterations.

The *Data Management (DATAMGT)* is designed to retrieve data from the R&D GTDS database to create temporary working files to be used by other programs in R&D GTDS. In this capacity, the program is used in conjunction with the program that will actually use the working files being created. **DATAMGT** can also be used as a standalone program to create working files for future program execution.

The *Data Simulation Program (DATASIM)* computes simulated observations at a specified frequency. The simulated observations can be for any specified tracking station or interval, and can include random and bias errors, atmospheric refraction errors, antenna mount errors, transponder delays, and light time delays. These simulated observations can be used as input to other R&D GTDS programs, and in determining tracking station schedules.

The *Error Analysis Program (ANALYSIS)* allows the user to determine satellite state uncertainties about a reference orbit. These uncertainties are given as a function of uncertainties in the epoch state, observation data, and system parameters for a given

station dependent tracking schedule. The program is primarily used with DC observation types, but its flexible design allows for future expansion to other types.

The *Early Orbit Determination Program (EARLYORB)* produces an initial estimate of an orbit when there is insufficient data to start the DC program. The **EARLYORB** program uses as few as six observations to quickly determine an initial estimate of the spacecraft state. This estimate can then be used as an apriori ephemeris with the DC program.

The *Permanent File Report Generation Program (FILERPT)* generates descriptive reports of the data and models in the R&D GTDS database and Solar/Lunar/Planetary (SLP) Ephemeris Files. The user has control of several report output options from condensed summary reports to highly descriptive complete reports. Observations, station positions, astrodynamical constants, potential fields, integration coefficients, and other data files can be analyzed with this program.

The *Filter Program (FILTER)* uses the four sequential estimation algorithms currently available in R&D GTDS to process observations and recursively determine the satellite state vector. The four filters used in R&D GTDS include a Linearized Kalman Filter (LKF), Extended Kalman Filter (EKF), Semianalytical Kalman Filter (SKF), and Extended Semianalytical Kalman Filter (ESKF), the latter two developed by Taylor [65]. A filter's unique ability to determine state and solve-for parameters in highly dynamic environments make it a logical choice for a majority of the estimation necessary for this study, and is therefore the primary focus of this thesis.

As a powerful orbit analysis system, R&D GTDS includes nineteen different orbit propagation theories to allow the user to choose the best propagator for the given R&D GTDS program, and the analysis problem. The available orbit theories include numerical integration, analytic theories based on general perturbations, and semianalytic theories

[40]. A mathematical derivation and analysis of the following theories can be found in reference [4]. In addition to the Draper Semi-analytic Satellite Theory (DSST), R&D GTDS has multiple mechanizations of numerical propagators, including:

- Cowell
- Time-regularized Cowell

and the following analytic theories:

- Brouwer
- Brouwer-Lyddane
- Brouwer-Gordon
- Vinti
- Naval Space Command PPT2
- Draper PPT2 / M-Daily [128]
- NORAD theories
 - Simplified General Perturbations (SGP)
 - GP4
 - DP4
 - HANDE
 - SALT

The remainder of this Chapter is to serve as a reference to familiarize the reader with the Windows NT version of R&D GTDS. First, the reader will be introduced to the developmental history of GTDS and functionality which accompanied that development. Next, the propagators used in R&D GTDS are presented to show the broad functionality of R&D GTDS. Finally, a comprehensive overview of the operation of NT-GTDS will be presented to simplify the execution of this complex program to a step by step procedure.

4.2 Developmental History

As an evolutionary system, GTDS represents a mathematical springboard from which many versions, and corresponding functionality, have been developed in the past twenty years. The continuous stream of new functionality is part of a continuing effort for the evolution of this tool to include the state of the art in current space mission

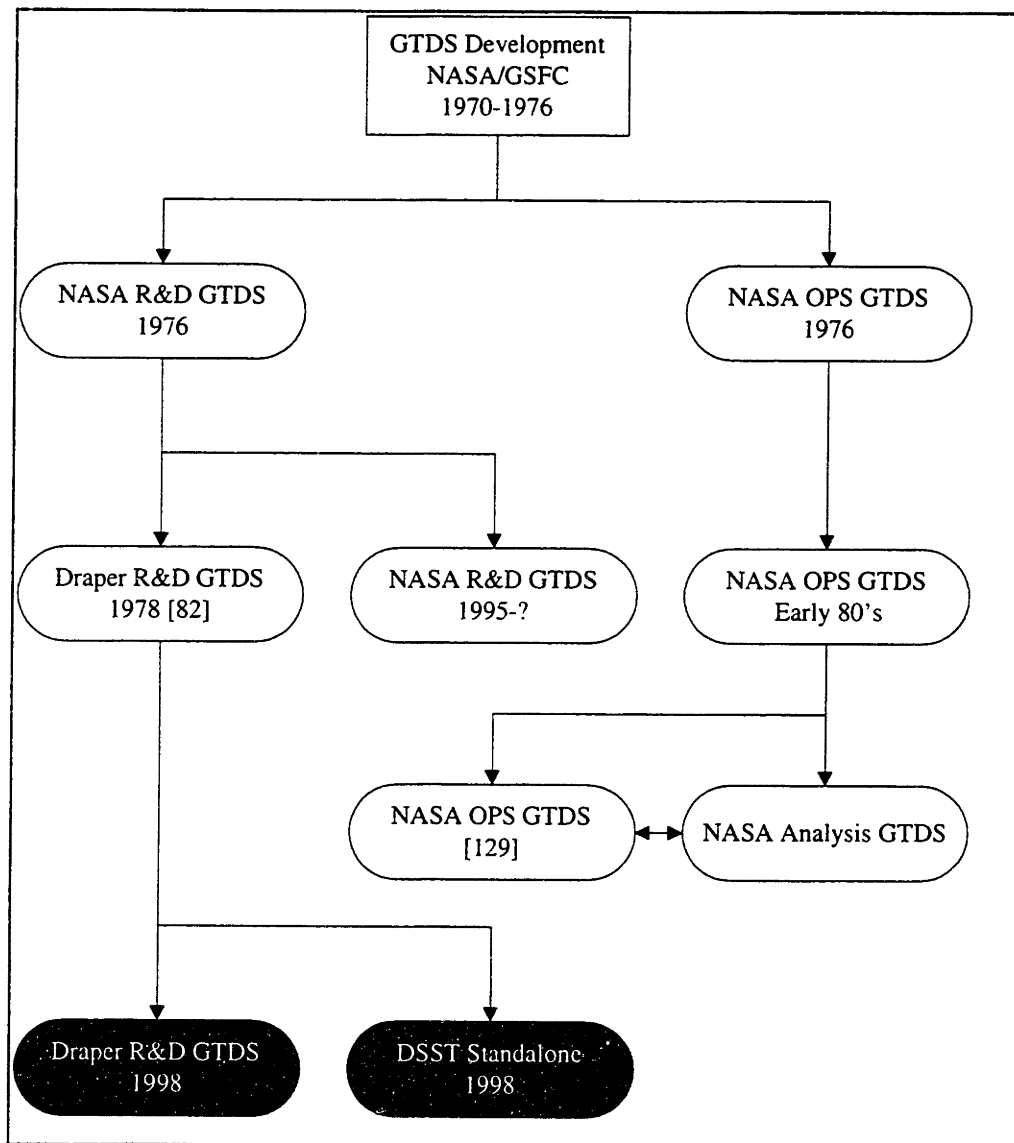


Figure 4.1: GTDS Initial Developmental History Diagram

analysis capabilities. After initial GTDS development in the early 1970's, this evolution occurred along two primary paths beginning in 1976 when GTDS was split into two versions: the operational and R&D versions [83]. Figure 4.1 provides a flow diagram of the initial development of GTDS, including the initial version of R&D GTDS acquired by the Charles Stark Draper Laboratory in 1978 to support a project for the U.S. Air Force Space Division.

As shown in Figure 4.1, the 1978 version of Draper R&D GTDS has evolved into two primary products, the current Draper R&D GTDS and the Draper Semianalytic Satellite Theory (DSST) Standalone. These products incorporate the extensive Draper upgrades to GTDS, including [83],[84]:

- *Algorithm improvements:*
 - High Precision Mean Element (PME) orbit generation including short periodics and efficient interpolators
 - Orbit determination using PME: Weighted Least Squares and Kalman Filter
 - Development of the SKF and ESKF Semianalytical Filters [65]
 - New observation models and coordinate systems for use with the Precision Conversion of Element observation data (PCE) and onboard MANS functionalities
 - Maintenance and improvement of numerous database files (geopotential, atmospheric model, SLP, and timing coefficient files)
 - Standalone tracking data pre-processors
 - Inclusion of NORAD GP Theories and the Naval Space Command (NAVSPASUR) second generation Position, Partial, and Time (PPT2) Theory

- Refinements of semianalytical theory, as demonstrated in DSST
- *Software improvements:*
 - User interface for analysts (IBM MVS)
 - Cross-platform development on Mainframe, VAX, Sun Workstation, Silicon Graphics (SGI), Macintosh, and PC platforms
 - Software maintenance:
 - Replacement of the GTDS assembly language routines with FORTRAN equivalents
 - Conversion of GTDS source to FORTRAN 77 with improved coding style, standardization of common block usage, etc.

To take advantage of platform specific advantages, and tailor R&D GTDS and the DSST Standalone to various user needs, Draper developed a variety of versions on multiple platforms such as the IBM Mainframe, VAX, Sun Workstation, SGI, Macintosh (DOS on Mac usage), and PC environments. Although significant effort is necessary to consolidate the versions and capabilities into a single cross-platform version, extensive testing done by Metzinger [85] has shown that the various versions yield nearly identical results for the basic R&D GTDS functions.

In addition to these improvements and version testing, a long line of MIT graduate students, participating in the Draper Fellow program, have made significant contributions to the development of R&D GTDS. Figure 4.2 outlines the development of R&D GTDS at Draper, highlighting the improvements made by some of the most recent Draper Fellows (shaded) and the versions developed for use in this thesis (white on black).

Fonte [86] extended the gravity field model of VAX R&D GTDS to include a 50x50 gravity field model in the Cowell and SST orbit generators. Fonte also analyzed

the impact of necessary calculations for the 50x50 model on computational accuracy and overflow/underflow constraints for several platforms (PC, VAX, Sun, and SGI).

Wallace [87] coupled DSST with the Parallel Virtual Machine (PVM) software package to demonstrate the power of the networked computing paradigm. Using genetic algorithm software and the PVM package, Wallace was able to automate the constellation design optimization process.

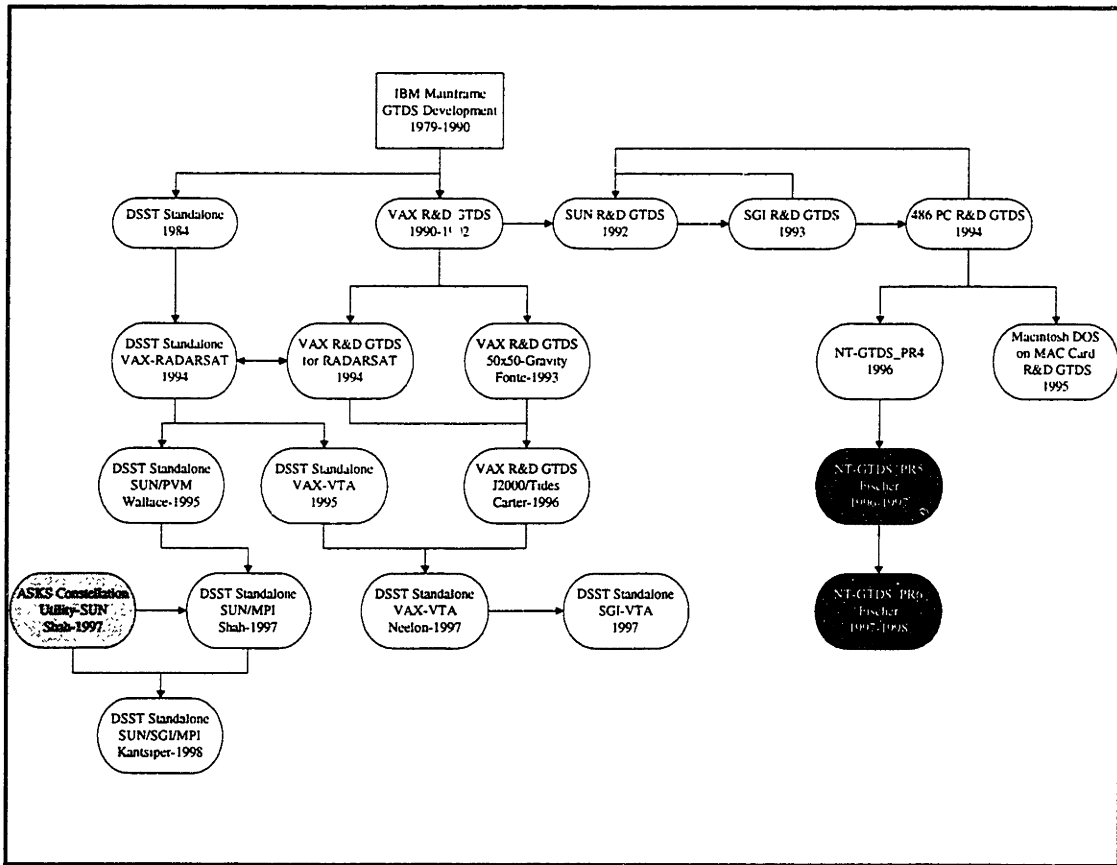


Figure 4.2: Draper R&D GTDS Developmental History Diagram

Carter [40] made several contributions to VAX R&D GTDS in his investigation of using GPS navigation solutions as an observation source in a Least Squares fitting process. As a result of this investigation, Carter included coordinate systems compatible with the Fifth Fundamental Catalogue (FK5) and an inertial true equator and equinox of date frame (J2000). He also introduced solid earth tide modeling for the numerical

integration techniques and refined the inclusion of these perturbations in the semianalytic methods. The new GTDS models are good at the one meter level for the 15 cm accurate TOPEX reference orbit as a result of Carter's modifications.

Neelon [83] synthesized functionality of various versions to improve the accuracy and maintainability of the DSST Standalone. The specific model improvements include Carter's J2000 coordinate system reference and solid earth tides, Fonte's 50x50 geopotential model, and modifications to the short periodic model.

Shah [41] develops and tests a new methodology for the automated station-keeping of satellite constellations by introducing Message Passing Interface (MPI) code to the DSST standalone. Using both conventional station keeping strategies, and developing Lyapunov non-linear control strategies, Shah analyzes the control and coverage characteristics of the Ellipso communications constellation. In addition, Shah developed the Automated Station-Keeping Simulator (ASKS) which combines an optimal n-impulse rendezvous variable and user interface with DSST to refine the MPI constellation optimization process.

Kantsiper [88] developed a systematic approach to determining station-keeping requirements for individual members of constellations based on the coverage requirements imposed on that constellation. To accomplish this, he developed an analytical approach to calculating constellation coverage, and an algorithm for determining optimal orbital element limits on individual satellites to meet a specified coverage requirement. Kantsiper also used the ASKS utility to validate his work, and adapted this utility and the DSST Standalone to the SGI environment.

When the current investigation began in 1996, the PC was chosen as the testing and development platform due to availability of the DSST input processor. In addition, the VAX version model enhancements (Carter [40] and Fonte [86]) improvements such as

the 50x50 gravity field model), were deemed unnecessary for use with highly eccentric orbits. A wide variety of propagators and programs were necessary and, as such, a version of R&D GTDS was chosen over the DSST Standalone. There was also a desire to extend the PC version into a 32-bit operating environment, namely Windows NT 4.0 and Windows 95. Accordingly, the PC variant of R&D GTDS was selected in August of 1996. The current PC version at the time was the PR3 version developed by Paul Cefola and Dan Fonte at Phillips Lab in 1995. The routines were then transferred to an NT machine and recompiled, creating NT-GTDS PR4. Using Metzinger's [85] test cases, PR4 was tested against all versions of R&D GTDS to verify code and, with only a few conditional loops to correct for double precision errors, was validated as an operational version of R&D GTDS.

In order to stratify major changes in the code due to the current investigation within specific versions, NT-GTDS PR5 was then developed during 1996 and 1997 by:

- Incorporating the MSISE-90 atmospheric model
- Adding a filter input processor
- Enhancing the atmospheric file database
- Adding a time dependent functional process noise model
- Correcting several small filter bugs

Finally, during March and April of 1998, the current version of NT-GTDS (PR6) was created by adding a simplified lift model, and an altitude dependent functional process noise model. A detailed account of the operation of this version, and NT-GTDS PR5, will be given in the following section.

4.3 Operation of R&D NT-GTDS

To encourage future use of NT-GTDS, and to create a concise manual for the reader interested in taking this work a step further, a comprehensive description of the construction and operation of NT-GTDS becomes necessary. The following includes the primary files essential to make a NT-GTDS run:

- Fortran MAKEFILE
- NT-GTDS Data Assignment Files
- NT-GTDS Output Files
- NT-GTDS Batch Execution Files
- NT-GTDS Input Card Data Files

While there are some examples of each file type given, all files used in this investigation, as well as all source code for each version of NT-GTDS, can be found at the Charles Stark Draper Laboratory in an E53 computer titled "Bob." Future reference to this computer will include path statements for the exact location of files, and the phrase \$E53-BOB to avoid cumbersome mention of the computer's location.

4.3.1 Compilation and Linking of NT-GTDS Code

The first step in making an operational version of NT-GTDS on a PC is to compile and link the nearly 1200 subroutines, common blocks, and include statements which comprise the program. Conventional methods of compiling and linking would make this task formidable, but Lahey Fortran 77 includes a make utility which simplifies this process to a single command, and makes incorporation of new routines straightforward. The make utility allows the user to define the relationships between the

modules, then make utility will update them automatically when a change occurs in the files or relationships (as opposed to re-compiling everything) [89]. Figure 4.3 lists a sample NT-GTDS makefile (with the majority of the routines truncated) to demonstrate the basic structure of these files.

Initialization	<pre> # # MKMF template makefile for protected mode executables. # FC = f7713 LINKER LINKER = 386link PROGRAM = GTDS.EXE DEST = . VPATH = ..\PPT2_SOU;..\NORAD_SO;..\MSIS_SOU EXTHDRS = FFLAGS = HDRS = LDFLAGS = -SYMBOLS -stub runb LDMAP = GTDS.MAP LIBS = MAKEFILE = Makefile </pre>
Objects Definition	<pre> OBJS = adconsbd.obj \ anavdpbd.obj \ . . . odsexec.obj \ . . GNEF4.OBJ \ SETESKF.OBJ </pre>
Source Files Definition	<pre> SRCS = adconsbd.for \ . . . SETESKF.FOR </pre>
Create Executable	<pre> \$(PROGRAM): \$(OBJS) \$(LIBS) \$(LINKER) \$(OBJS) -EXE \$@ -MAP \$(LDMAP) -LIB \$(LIBS) \$(LDFLAGS) clean:: @del -f \$(OBJS) depend:: @mkmf -f \$(MAKEFILE) PROGRAM=\$(PROGRAM) DEST=\$(DEST) install: \$(PROGRAM) @echo Installing \$(PROGRAM) in \$(DEST) @if not \$(DEST)x==.x copy \$(PROGRAM) \$(DEST) </pre>
'Include' Dependencies	<pre> *** OPCS MKMF: Do not remove this line! Automatic dependencies follow. LFTPARBD.OBJ: lftpar.cmn . . . ancast.obj: aanc01.cmn aanc02.cmn aancer.cmn aancin.cmn aanca.cmn \ aancpe.cmn aancrs.cmn amathc.cmn mcodes.cmn . . </pre>

Figure 4.3: Sample Lahey F77 Makefile

The first section is the initialization, where the user defines the name of the created executable, linker used, directory locations of subroutines, common blocks, and includes compilation flags, map file definition, and other options defined in reference [89]. The compilation flags used in Figure 4.3 are recommended for use when compiling NT-GTDS. The initialization section is followed by a listing of the object and source code files for all routines and common blocks used in the program. The current convention is to list common block objects first, then subroutines, where the line is extended with a \ to allow for more routines. To add a subroutine to NT-GTDS, simply add the object and source code files at the end of the OBJ/FOR sections, taking care to add a \ at the end of the preceding line. If the new file is a new common block, add it at the end of the common block listings (before *.FOR files).

The create executable section gives the actual commands for compiling and linking the routines together. These lines tell the compiler, using the information from the above sections, how to create the executable and where to install it. Finally, the 'INCLUDE' statement dependencies are defined, where if more than one routine depend on an include statement, they are separate by spaces as shown in Figure 4.3.

Using the existing makefiles makes the process of compiling and linking a very simple task, and allows the user a great deal of flexibility when adding code. Once the makefile is completed, the user simply types MAKEL (L is for large programs), and Fortran will re-compile any routines that have been modified, link all the program files, and create a new executable. Please refer to \$E53-BOB for sample NT-GTDS makefiles in the following directories:

- BOB\D:\NTGTDS\PR4\GTDS_SOU\MAKEFILE
- BOB\D:\NTGTDS\PR5\GTDS_SOU\MAKEFILE
- BOB\D:\NTGTDS\PR6\UPDATE\GTDS_SOU\MAKEFILE

4.3.2 NT-GTDS Data Assignment Files

Essential to any execution of NT-GTDS is to properly link the database files into the NT-GTDS equivalents for use during the execution. These files define geopotential fields, atmospheric density model data, timing coefficients, error messages, SLP data, modified Newcomb operator files, and more. Without them, NT-GTDS would not be able to function, and it is therefore necessary to define an efficient way of assigning these files at the start of every NT-GTDS execution. To do so, the user simply makes two batch files to assign the data files at the start of a run (GTD_SET) and to move the files back at the end of a run (GTD_KIL). Reproduction of these files here is unnecessary because they are comprised of the following simple line formats, and are then saved with a *.bat extension under the names GTD_SET.BAT and GTD_KIL.BAT.

- **GTD_SET:** move e:\ntgtds\PR6\update\data\sfdir.dat GTDS\$001
- **GTD_KIL:** move GTDS\$001 c:\ntgtds\pr6\update\data\sfdir.dat

Obviously the files used are run-specific and depend on the models, options, and programs chosen. As an example, Table 4.1 identifies the files used with GTD_SET/GTD_KIL in the majority of this analysis, concerning the final decay of NSSC-13390 in 1994 and 1995 with the MSISE-90 atmospheric density model. Explanation of the files uses are from reference [129]. Table 4.2 then identifies all database files available in NT-GTDS and pertinent information about each file. The user must be careful to use consistent files with a run (e.g. the JN94_S50 file with the JN94_TMC file).

The construction of the MSISE-90 atmospheric data files is discussed in Section 6.1. The construction of the ERRMSG2 error file involved the use of the READERR and WRITERR utilities which can be found on the \$E53-BOB system at

D:\NTGTDS\PR6DATA_ARCHIVE\ERROR. New errors are simply typed into the text error data file (produced by READERR), then the data is translated back into a useable NT-GTDS format by using WRITERR.

Table 4.1: NSSC-13390 Final Decay Database Files

GTDS Logical Name	File Name (*.DAT)	File Function
GTDS\$001	SFDIR	<i>Small Files Directory</i> : contains file header information for various NT-GTDS files. Used internally.
GTDS\$002	ATMOSDEN	<i>Atmospheric Density Models File</i> : contains various sets or models of atmospheric density data used by NT-GTDS in drag computations.
GTDS\$008	RSAT_FLD	<i>Earth Potential Fields File</i> : contains data relating to the Earth's potential field. The data set is accessed when Earth potential field data other than that in COMMON storage is required.
GTDS\$013	ERRMSG2	<i>Error Messages File</i> : contains all the numbered NT-GTDS error messages in EBCDIC format. It is accessed by NT-GTDS whenever an error occurs.
GTDS\$014	JN94_S50	<i>Solar/Lunar/Planetary (SLP) Ephemeris File</i> : contains planetary ephemeride data in a coordinate system referenced to 1950.0 and is used throughout NT-GTDS.
GTDS\$023	HI_NUK	<i>Modified Newcomb Operator File</i> : contains data for Newcomb operators which are dependent on the orbit and potential model.
GTDS\$038	JN94_TMC	<i>Time Conversion Coefficients File</i> : contains timing conversion and polar motion data which may be accessed throughout NT-GTDS runs.
GTDS\$075	Not Used	<i>Jacchia Atmosphere File</i> : contains solar activity data and geomagnetic indices for use with the Jacchia-Roberts Atmospheric density model.
GTDS\$076	MS90_HOT	<i>MSISE-90 Atmosphere File</i> : analog of file 75 for use with the MSISE-90 Atmospheric Density Model.
GTDS\$078	JN94_STD	<i>SLP Ephemeris File (True of date coordinates)</i> : contains planetary ephemeris data and is accessed throughout NT-GTDS computation when the user has specified True of date coordinates for the integration.

Table 4.2: NT-GTDS Database Files

GTDS Logical Name	File Name (*.DAT)	Function	Effective Time Span
GTDS\$001	SFDIR	stub associated with small files directory	
GTDS\$002	ATMOSDEN	Harris-Priester atmosphere density tables	
GTDS\$008	RSAT_FLD	<p>Earth Geopotential Field (21 x 21 models) updated for use in the Radarsat FD Program</p> <p>1 = GEM T3</p> <p>2 = GEM 10B</p> <p>3 = WGS 84 (21 x 21 created from 12 x 12)</p> <p>4 = JGM 2</p> <p>5 = JGM 2 Clone</p> <p>6 = WGS 72 (12 x 12)</p> <p>7 = WGS 72 (12 x 12)</p>	
GTDS\$008	GRAVITY	<p>The baseline 21 x 21 gravity models file</p> <p>1 = SAO 1969 Standard Earth Model</p> <p>2 = Earth Potential for Manned Flight Computations (EPMFC)</p> <p>3 = GSFC Earth Model (GEM 1)</p> <p>4 = GSFC Earth Model (GEM 7)</p> <p>5 = GSFC Earth Model (GEM 9)</p> <p>6 = GSFC Earth Model (GEM 10B truncated)</p> <p>7 = WGS 72 (12 x 12)</p> <p>8 = GSFC Earth Model (GEM L2 truncated)</p> <p>9 = WGS 84 (12 x 12)</p>	

GTDS\$008	GRAVITY2	A modified 21 x 21 gravity models file developed at Phillips Lab/VTA 1 = Unused 2 = Joint Gravity Model 2 (JGM 2) 3-9 = Unused	
GTDS\$013	ERRORMSG	NT-GTDS Error Messages	
GTDS\$013	ERRMSG2	NT-GTDS Error Messages for PR5 and PR6	
GTDS\$014	JN94_S50	SLP Mean of 1950 file supplied by GSFC in June 94	24 DEC 1987 to 18 DEC 2007
GTDS\$014	OLD_S50	SLP Mean of 1950 file supplied by GSFC for use with the 1982 epoch benchmark test cases	1 JAN 1971 to 23 NOV 1981
GTDS\$014	ORIG_S50	SLP Mean of 1950 file supplied by GSFC for use with the 1982 epoch benchmark test cases	1 JAN 1974 to 17 JAN 1986
GTDS\$014	NEW_S50	SLP Mean of 1950 file obtained from the NASA GSFC in December 96	24 DEC 1987 to 18 DEC 2007
GTDS\$023	HI_NUK	Modified Newcomb Operator File. Designed for use with Molniya class orbits and gravity fields bounded by the 8 x 8 Geopotential	
GTDS\$023	LO_NUK	Modified Newcomb Operator File. Designed for use with low eccentricity orbits and gravity fields bounded by the 21 x 21 Geopotential	
GTDS\$023	ME_NUK	Modified Newcomb Operator File. Designed for use with medium eccentricity orbits and gravity fields bounded by the 21 x 21 Geopotential; used in the analyses of the ELLIPSO™ Borealis orbit (e = 0.3).	

GTDS\$038	JN94_TMC	Timing Coefficient File supplied by GSFC in June 94	
GTDS\$038	OLD_TMC	Timing Coefficient File supplied by GSFC; this file should be used with the 1982 epoch GTDS benchmark test cases.	
GTDS\$038	ORIG_TMC	Timing Coefficient File supplied by GSFC; this file should be used with the 1982 epoch GTDS benchmark test cases.	
GTDS\$038	NEW_TMC	Timing Coefficient File obtained from the NASA GSFC in December 96	
GTDS\$047 GTDS\$048	GRAVIT50	A modified 50 x 50 gravity models file developed at Phillips Lab/VTA 10-12 = Not Used 13 = Goddard Earth Model T3 (GEMT3) (tbc) 14-18 = Not Used 19 = Joint Gravity Model 2 (JGM 2) (tbc)	
GTDS\$075	JRS_COLD	Jacchia-Roberts Atmosphere Density Model data based on Ken Schatten's "low" solar activity and geomagnetic index predictions; valid for time interval 1994 thru 2007	14 JAN 1994 to 26 AUG 2007
GTDS\$075	JRS_HOT	Jacchia-Roberts Atmosphere Density Model data based on Ken Schatten's "high" solar activity and geomagnetic index predictions; valid for time interval 1994 thru 2007	14 JAN 1994 to 26 AUG 2007
GTDS\$075	JRS_NOM	Jacchia-Roberts Atmosphere Density Model data based on Ken Schatten's "nominal" solar activity and geomagnetic index predictions; valid for time interval 1994 thru 2007	14 JAN 1994 to 26 AUG 2007
GTDS\$075	JAC_SCHT	Jacchia-Roberts Atmosphere Density Model data based on Ken Schatten's "nominal" solar activity and geomagnetic index predictions; valid for time interval 1994 thru 2007 [believed to be equivalent to JRS_NOM.DAT]	14 JAN 1994 to 26 AUG 2007

GTDS\$075	JRMS_FIS	Jacchia-Roberts Atmosphere Density Model data based on real NOAA data. File created with Fischer utilities and accounts for Kp hourly adjustment for MSISE-90 comparison cases.	10 FEB 1980 to 11 NOV 1996
GTDS\$075	JAC_ORIG	Jacchia-Roberts Atmosphere Density Model data; this file should be used with the 1982 epoch GTDS benchmark test cases.	28 FEB 1965 to 17 FEB 1986
GTDS\$075	JAC_REAL	Jacchia-Roberts Atmosphere Density Model data; this file contains actual data from the 1995 epoch. This file should be used with the PL/VTA SCN Training Case.	16 FEB 1994 to 9 AUG 1995
GTDS\$076	MS90_CLD	MSISE-90 Atmosphere Density Model data based on real NOAA data through February 1997, and on Ken Schatten's "low" solar activity and geomagnetic index predictions through September 2008. The file is for use with "nominal" cycle timing.	10 FEB 1980 to 30 SEP 2008
GTDS\$076	MS90_HOT	MSISE-90 Atmosphere Density Model data based on real NOAA data through February 1997, and on Ken Schatten's "high" solar activity and geomagnetic index predictions through September 2008. The file is for use with "nominal" cycle timing.	10 FEB 1980 to 30 SEP 2008
GTDS\$076	MS90_NOM	MSISE-90 Atmosphere Density Model data based on real NOAA data through February 1997, and on Ken Schatten's "nominal" solar activity and geomagnetic index predictions through September 2008. The file is for use with "nominal" cycle timing.	10 FEB 1980 to 30 SEP 2008
GTDS\$076	MS9E_CLD	MSISE-90 Atmosphere Density Model data shifted by one based on real NOAA data through February 1997, and on Ken Schatten's "low" solar activity and geomagnetic index predictions through September 2008. The file is for use with "early" cycle timing.	10 FEB 1980 to 30 SEP 2008
GTDS\$076	MS9E_HOT	MSISE-90 Atmosphere Density Model data based on real NOAA data through February 1997, and on Ken Schatten's "high" solar activity and geomagnetic index predictions through September 2008. The file is for use with "early" cycle timing.	10 FEB 1980 to 30 SEP 2008

GTDS\$076	MS9E_NOM	MSISE-90 Atmosphere Density Model data based on real NOAA data through February 1997, and on Ken Schatten's "nominal" solar activity and geomagnetic index predictions through September 2008. The file is for use with "early" cycle timing.	10 FEB 1980 to 30 SEP 2008
GTDS\$076	MS9L_CLD	MSISE-90 Atmosphere Density Model data shifted by one based on real NOAA data through February 1997, and on Ken Schatten's "low" solar activity and geomagnetic index predictions through September 2008. The file is for use with "late" cycle timing.	10 FEB 1980 to 30 SEP 2008
GTDS\$076	MS9L_HOT	MSISE-90 Atmosphere Density Model data based on real NOAA data through February 1997, and on Ken Schatten's "high" solar activity and geomagnetic index predictions through September 2008. The file is for use with "late" cycle timing.	10 FEB 1980 to 30 SEP 2008
GTDS\$076	MS9L_NOM	MSISE-90 Atmosphere Density Model data based on real NOAA data through February 1997, and on Ken Schatten's "nominal" solar activity and geomagnetic index predictions through September 2008. The file is for use with "late" cycle timing.	10 FEB 1980 to 30 SEP 2008
GTDS\$078	JN94_STD	SLP True of Date file supplied by GSFC in June 94	24 DEC 1987 to 18 DEC 2007
GTDS\$078	NEW_STD	SLP True of Date file obtained from the NASA GSFC in December 96	24 DEC 1987 to 18 DEC 2007
GTDS\$078	OLD_STD	SLP True of Date file supplied by GSFC for use with the 1982 epoch benchmark test cases	1 JAN 1974 to 17 JAN 1986
GTDS\$078	ORIG_STD	SLP True of Date file supplied by GSFC for use with the 1982 epoch benchmark test cases	1 JAN 1971 to 23 NOV 1981

4.3.3 NT-GTDS Input Card Data Files

Input Card Data Files are the control cards of the NT-GTDS run. These cards define the programs, databases, models, options, solve-for parameters, time parameters, and much more, by specific NT-GTDS Keyword Cards. Keyword cards consist of seven input values including the following, which are also shown in Figure 4.4:

- *Keyword name* (columns 1-8): Eight character alphanumeric variables which represents some program option or data quantity.
- *Three integer fields* (9-11, 12-14, and 15-17): Integer input options.
- *Three real fields* (18-38, 39-59, and 60-80): Real number input options.

Keyword Name	Integer Fields	Real Fields
GAUSSERR	82 7 22	0.0 1000.0 50.0

Figure 4.4: NT-GTDS Keyword Card Structure

The specific structure of an input card will vary dramatically depending on the program(s) being used in the run. As an example of constructing an input card, the FILTER program input card structure will be investigated. The following keyword cards are required for a FILTER run, and are described in Table 4.3: CONTROL, ELEMENT1, ELEMENT2, EPOCH, OBSINPUT, ORBTYPE, and FIN.

The first card in a FILTER input card must be the CONTROL card to initiate the run. This card is then followed by the ELEMENT1, ELEMENT2, EPOCH, OBSINPUT, and ORBTYPE cards. The above cards are known as mandatory keyword cards and, like the above cards and the filter program, must be used in the specified programs. All mandatory keyword cards used in NT-GTDS are listed in Table 4.3. There are also

hundreds of optional keyword cards, which can be found in references [82], and [129], and these cards must be used with sub-deck keyword cards, which are listed in Table 4.4. If any data management functions are necessary, the mandatory cards must be followed by the DMOPT. Other sub-deck keywords such as DCOPT, OGOPT, and FILOPT can also be used in this control card with the proper optional keywords and END delimiters (at the end of each sub-deck)

Table 4.3: Mandatory Keyword Cards

<i>Keyword Card</i>	<i>Description</i>	<i>Program Use</i>
CONTROL	Initiates input processor in a run, discussed above.	CM, DC, DM, DS, EA, EO, EP, FL, FR
ELEMENT1	Sets coordinate system, reference central body, and first three orbital elements of the orbital state.	DC, EP, FL
ELEMENT2	Sets the second three orbital elements of the orbital state.	DC, EP, FL
EPOCH	Specifies the epoch of the initial state and changes the output report epoch.	DC, EO, EP, FL
FIN	Indicates the end of the FILTER program card deck.	CM, DC, DM, DS, EA, EO, EP, FL, FR
OBSINPUT	Specifies the observation input sources and the start/end times of the data and the time of the output report.	DC, EO, FL
OBSNUME	Selects observation numbers for the Early Orbit Determination Program.	EO
ORBTYPE	Sets orbit generator type and related parameters.	DC, EP, FL
OUTPUT	Specifies orbit generator printer output	EP
TYPE	Selects method for Early Orbit Determination Program computation.	EO

Table 4.4: Sub-deck Keyword Cards

Keyword Card	Description	Program Use
COMPOPT	Initiates the Ephemeris Comparison Program sub-deck processing.	CM
DCOPT	Initiates the Differential Correction sub-deck processing.	DC, DS, EA, EO, FL
DMOPT	Initiates the Data Management sub-deck processing.	DC, DM, DS, EA, EO, EP, FL
EAOPT	Initiates the Error Analysis sub-deck processing.	EA
END	Identifies the end of a sub-deck	CM, DC, DS, DS, EA, EO, EP, FL, FR
FILOPT	Initiates the Filter sub-deck processing.	FL
OGOPT	Initiates Orbit Generator sub-deck processing.	DC, DS, EA, EO, EP, FL
PFR OPT	Initiates the Permanent File Report sub-deck processing.	FR

Examples of completed input cards are included in Appendix D for the FILTER, DC, EPHEM, and FILERPT programs. The reader is encourage to reference these completed cards and the *GTDS User's Guide*, Section I [82], [129] to gain a better understanding of the options and models available in NT-GTDS. The interested reader should also reference Fonte and Sabol [90] for a listing of optimal DSST input card decks for various orbit types. The User's Guide Section II and Appendix B contain Keyword card descriptions that are used in the input cards, and should also be referenced.

4.3.4 NT-GTDS Batch Execution Files

The final file necessary to make a NT-GTDS run, is a batch execution file. The batch execution file is a DOS batch file that:

- Calls the GTD_SET batch file to move the necessary database files to the proper NT-GTDS locations
- Copies the input card data file to NT-GTDS file GTDS\$005
- Copies the input observations (if used, for example – in a DC or FILTER run) data file to NT-GTDS file GTDS\$015
- Calls the NT-GTDS executable
- Copies the output file, GTDS\$006 to a specified permanent data file
- Copies any other desired output files (such as an ORB1 file, GTDS\$024 or GTDS\$081, lift acceleration output file, GTDS\$050, etc.) to permanent data files
- Call the GTD_KIL batch file to move the database files back to their previous locations in the NT-GTDS data archive and erase any unused temporary NT-GTDS files

4.3.5 Summary

In summary, to execute a NT-GTDS run, the user must accomplish the following steps:

- Make any necessary modifications to the NT-GTDS source code
- Modify the NT-GTDS makefile to reflect those changes
- Execute the makefile to compile and link a current version of the NT-GTDS executable

- Select the necessary database files to be used with the run by referencing Tables 4.1 and 4.2, and making the necessary changes to GTD_SET and GTD_KIL files to reflect those choices
- Create an input data card file using reference [82] with Tables 4.3 and 4.4, to represent the programs, models, and options desired for the run
- Create a batch execution file which incorporates the above files and saves any desired output files to permanent locations
- Run the batch execution file

The reader should note that NT-GTDS has the capability to interpret multiple input card decks (which specify a given program) within a single input card data file. This affords the user a great amount of flexibility by allowing several NT-GTDS programs to run in succession and build upon results. For example, the user may wish to fit elements to simulated data, and can do so by running an EPHEM run, followed by a DC fit over the same interval and using the OUTPUT option for the CONTROL card.

The above example brings up another interesting note for the reader, that of the output files available in NT-GTDS. The first, and most obvious output file, is the equivalent to the printer output file in Mainframe GTDS, which prints out all specified information concerning a run, including error messages, residuals, statistics, plots, and input options. The ORBIT file serves two functions in NT-GTDS. They can first be used in the retrieval of a set of interpolator coefficients, representative of an object's ephemeris and partial derivatives, which can be used to develop grid points for interpolation schemes used in the integration process. These files can also be used to store position and velocity information for simulated data over a given time period, that can also be used by other NT-GTDS programs [40]. The EPHEM output files, which are primarily used as inputs to the EPHEM COMPARE program, can also be used for tracking data

acquisition or maneuver planning [82]. Utilities developed for this investigation can also take these files and create plots of any variable in the EPHEM output listing. These plots can be useful when predicting satellite behavior and trends. Another useful output file, created using the OUTOPT keyword card, is the ORB1 file. This file can be used for several things such as ephemeris comparisons, input files to DC and/or FILTER runs, output files to post-processing utilities [40].

There are a number of other output files that can be used with NT-GTDS such as filter de-bugging files (GTDS\$043 and GTDS\$044), aerodynamic acceleration files (GTDS\$050), and many more, all of which are defined in a subroutine named SETDAF.FOR. Any output the user desires can be obtained by creating a new file in SETDAF (or using an old one), and outputting the necessary data to that file.

One crucial ingredient to the above process that has yet to be mentioned, is the use of a well-developed text editor. During this investigation, the Codewright for Windows version 3.1 editor was used. Without the code chromacoding, file GREP, search, and other capabilities within the Codewright editor, working with NT-GTDS would have been a very arduous task. Any editor can be used within the NT environment, but the reader should take care to choose a well developed one.

The flexibility apparent in the output options of NT-GTDS is primary to the very design of GTDS. GTDS is a living tool, growing in functionality and complexity with each new version, each new technical advance, and each new process that is added to it. As this investigation will show, NT-GTDS, as only one small part of the GTDS family, represents a ready platform for the integration of new functionality to meet the ever changing demands of the orbital dynamics community.

[This page intentionally left blank.]

Chapter 5

Mission Phase Analysis

5.1 Introduction

This primary focus of this thesis is the decay phase of the highly eccentric orbit's evolution. As such, the analysis presented in Chapter 5 is meant to serve as a source of ideas and intellectual stimulation rather than a complete report on the improvement of tracking these objects during the mission phase. Reasoning for this decision comes from Fieger, who in his 1987 study of the effectiveness of DSST with long terms predictions for highly eccentric orbits [19], showed the that the perturbative environment facing these objects, while complex, is fairly well modeled.

The remainder of this Chapter will delineate Fieger's, and other's, investigation of the significant perturbations affecting highly eccentric orbits. This analysis will lend itself to a discussion of the geopotential function, and identification of higher order oblateness terms as the primary source of error in the dynamic modeling of these orbits. Building upon the results and utilities of Zeis [32], Cefola [91], [92], McClain [42], [92], [93], Slutsky [93], [94], Green [95], Kaniecki [96], Metris [97]-[100], and Salma [101], an incomplete attempt was made to solve the higher order oblateness error by developing closed-form, second order, equations of motion in J_2 for use with DSST. These results, and the corresponding utilities, are presented in the hopes that they will present a new

form of an old problem, and inspire the reader to eventually develop a useable J_2^2 closed-form solution for use with DSST.

5.2 Environment Summary

As described in Chapter 1, the mission phase describes the portion of the orbital lifetime where the perigee is sufficiently out of the Earth's atmosphere, such that drag effects can be neglected. For the highly eccentric Molniya orbit, the arbitrary stratification of approximately 1000 km perigee altitude (mean motion less than 2.1 revolutions per day) has been chosen for this analysis. Fieger identifies several perturbations which will effect highly eccentric orbits in this region, the largest of which include [19]:

- *Earth oblateness effects*: discussed below.
- *Atmospheric Drag*: although not a major perturbation for the decay phase, drag does result in a decrease in orbital energy and corresponding secular reduction in semi-major axis.
- *Lunar-Solar activity*: perigee variations due to effects of the sun and moon result in four effects which demonstrate periods of twelve hours (short), fourteen days (intermediate), 183 days (long), and several years (very long). Long and very long perturbations become important when determining orbital lifetimes for these satellites, but like their shorter counterparts, these perturbations are well modeled in R&D GTDS.

The reader is encouraged to reference Fieger for sources on other perturbations such as resonance resulting from longitude dependence on the geopotential, and resonant

equilibrium longitudes for Molniya orbits (as a function of inclination and argument of perigee).

For an object in a Molniya orbit, until it reaches the decay phase and succumbs to drag, Earth oblateness effects represent the greatest source of perturbations to its motion. As a comparison, consider the moderately eccentric Vanguard 1 ($a = 8640$ km, $e = 0.186$, $A/m = 0.0212$ m²/kg) and Echo 1 ($a = 7940$ km, $e = 0.15$, $A/m = 10.2$ m²/kg) satellites and the disturbing force of the following perturbations upon them, presented by Blitzer [102]:

Table 5.1: Disturbing Forces on a Satellite at Varying Altitudes [152], [41]

(m/sec ²)	150 km Altitude	750 km Altitude	1500 km Altitude	Geosynchronous
Central Gravity	9.35	7.85	6.42	0.22
Geopotential				
<u>Zonals</u>				
J_2	30×10^{-3}	20×10^{-3}	14×10^{-3}	16×10^{-6}
J_3	0.09×10^{-3}	0.06×10^{-3}	0.04×10^{-3}	8×10^{-9}
J_4	0.07×10^{-3}	0.04×10^{-3}	0.02×10^{-3}	1×10^{-9}
<u>Tesserals</u>				
$J_{2,2}$	0.09×10^{-3}	0.07×10^{-3}	0.04×10^{-3}	0.05×10^{-6}
Atmospheric Drag				
Vanguard 1	3×10^{-3}	10×10^{-7}	negligible	Negligible
Echo 1	1.3	500×10^{-7}		
Solar/Lunar Third Body	1×10^{-6}	1×10^{-6}	1×10^{-6}	70×10^{-7}
Solar Radiation Pressure				
Vanguard 1	10×10^{-7}	10×10^{-7}	10×10^{-7}	10×10^{-7}
Echo 1	500×10^{-7}	500×10^{-7}	500×10^{-7}	500×10^{-7}

The table clearly shows the dominance of the geopotential in comparison with the other major perturbations in an eccentric environment, especially for the J_2 zonal term. In fact, Kozai [103] points out that the main problem in the theory of artificial satellite

movement lies in the determination of the effects due to the oblateness J_2 of the Earth, to the order 2 in this parameter. The reason Kozai identifies J_2^2 as the main problem in analytic and semianalytic propagation methods is that the extremely complex mathematical formulation of closed form, second order equations of motion, leaves J_2^2 as the largest un-modeled perturbation for a majority of satellites during their mission phase.

To more completely understand the problem associated with J_2^2 , it is first necessary to introduce the reader to the concept of spherical harmonics. The first step is to define U as the disturbing potential \mathcal{R} of the Earth's gravitational field. Next, as presented by Zeis [32], a basic expansion of the gravitational potential can be obtained by solving the Laplace equation, $\nabla^2 U = 0$, in spherical coordinates relative to an Earth-fixed coordinate system, as shown in Equation (5.1).

$$U = \frac{\mu}{r} \sum_{n=2}^{\infty} \sum_{m=0}^n \left(\frac{R_e}{r} \right)^n P_{nm} \sin(\phi) [C_{nm} \cos(m\lambda) + S_{nm} \sin(m\lambda)] \quad (5.1)$$

where:

μ = Earth's gravitational constant

r = distance from Earth's center of mass to satellite

R_e = Earth's mean equatorial radius

ϕ = geocentric latitude

λ = geographic longitude

$P_{nm} \sin(\phi)$ = associated Legendre functions of first kind of degree n and order m

C_{nm}, S_{nm} = harmonic coefficients of degree n and order m

To better understand the origin of the 'J' coefficient terminology, an alternate form of Equation (5.1) is given by Blitzer in Equation (5.2), and shows the new J_{nm}

constants which, like the C_{nm} , S_{nm} , and λ_{nm} constants, depend on the Earth's mass distribution.

$$U = -\frac{\mu}{r} \left[1 + \sum_{n=2}^{\infty} \sum_{m=0}^n J_{nm} \left(\frac{R_e}{r} \right)^n P_{nm}(\sin \phi) \cos(m(\lambda - \lambda_{nm})) \right] \quad (5.2)$$

The functions $\cos(m\lambda)P_{nm}\sin(\phi)$ and $\sin(m\lambda)P_{nm}\sin(\phi)$ (or, alternately, J_{nm} coefficients) are called the *spherical harmonics*, due to their periodicity on the surface of a unit sphere. Specifically, the indices n and m determine lines on the unit sphere along which the harmonic functions will vanish [102]. There exist three primary classes of spherical harmonics which are differentiated by the values of n and m , including:

- **Zonal Harmonics ($m = 0$):** The $m = 0$ condition negates the dependence of U on longitude, making the gravitational field symmetric about the polar axis. These harmonics are simply thought of as bands of latitude. For any $P_k\sin(\phi)$ there exists k circles of latitude along which P_k is zero. This defines $(k+1)$ zones of alternating sign, whose boundaries represent a root of the corresponding Legendre polynomial [2]. Figure 5.1 provides a graphic illustration of the location of these harmonics on the Earth.
- **Sectorial Harmonics ($n = m$):** When $n = m$, the lines along which $\cos(n\lambda)P_{nn}\sin(\phi)$ and $\sin(n\lambda)P_{nn}\sin(\phi)$ vanish represent meridians of longitude, which divide the Earth into $2n$ “orange slice” sectors with alternating signs, as shown in Figure 5.2 [102].

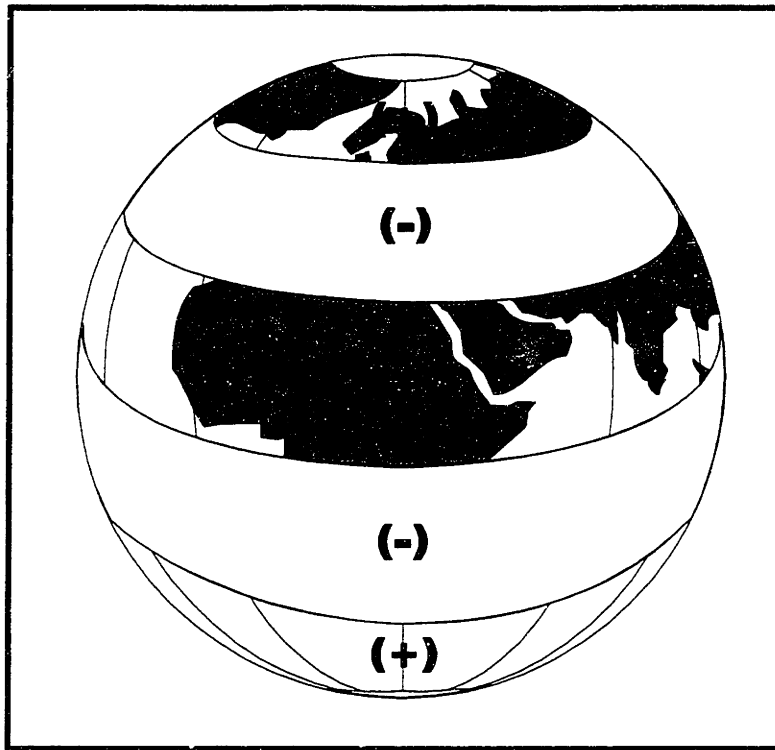


Figure 5.1: Zonal Harmonics Diagram

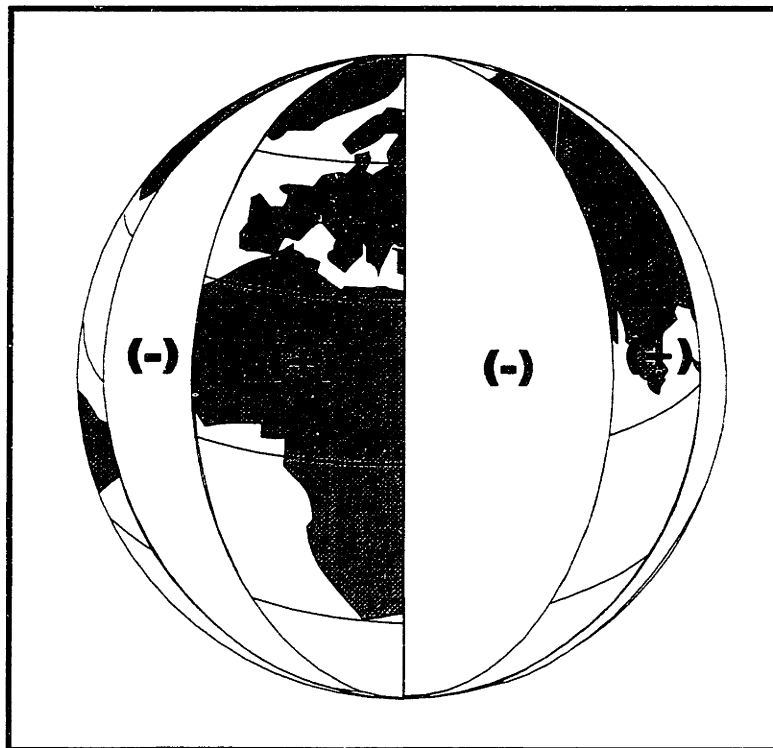


Figure 5.2: Sectorial Harmonics Diagram

- **Tesseral Harmonics ($n \neq m$):** If $n \neq m$, the functions $\cos(m\lambda)P_{nm}\sin(\phi)$ and $\sin(m\lambda)P_{nm}\sin(\phi)$ are referred to as tesseral, or square, harmonics. This name comes from the checkerboard array of alternating signs (shown in Figure 5.3) which the Earth is divided into. There are $(n-m)$ circles of latitude which correspond to a zero value for the $P_{nm}\sin(\phi)$ function and, alternately, $2m$ meridians of longitude along which the $\sin(m\lambda)$, $\cos(m\lambda)$ equal zero.

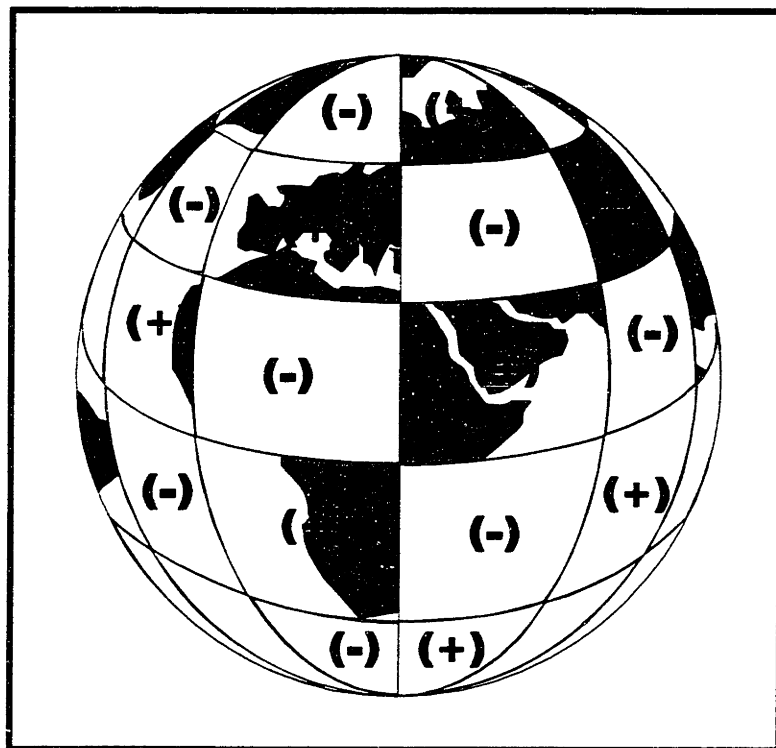


Figure 5.3: Tesseral Harmonics Diagram

Of particular interest is the J_2 zonal coefficient, which represents the equatorial obliquity of the Earth, and is by far the strongest gravitational potential effect (nearly 1000 times greater than the next largest coefficient J_3 [2]). The principal long term effects of oblateness are secular motions of the longitude of ascending node (Ω), and the

argument of perigee (ω) [102]. Motion in the node results from the added gravitational attraction of the equatorial bulge, which introduces a perturbative force toward the equator. This acceleration causes the satellite to reach the ascending node short of the crossing point for a spherical earth, as shown by Blitzer in Figure (5.4). This early crossing of the node results in the regression of the node and a corresponding rotation of the orbit plane around the polar axis, as shown in Figure (5.5).

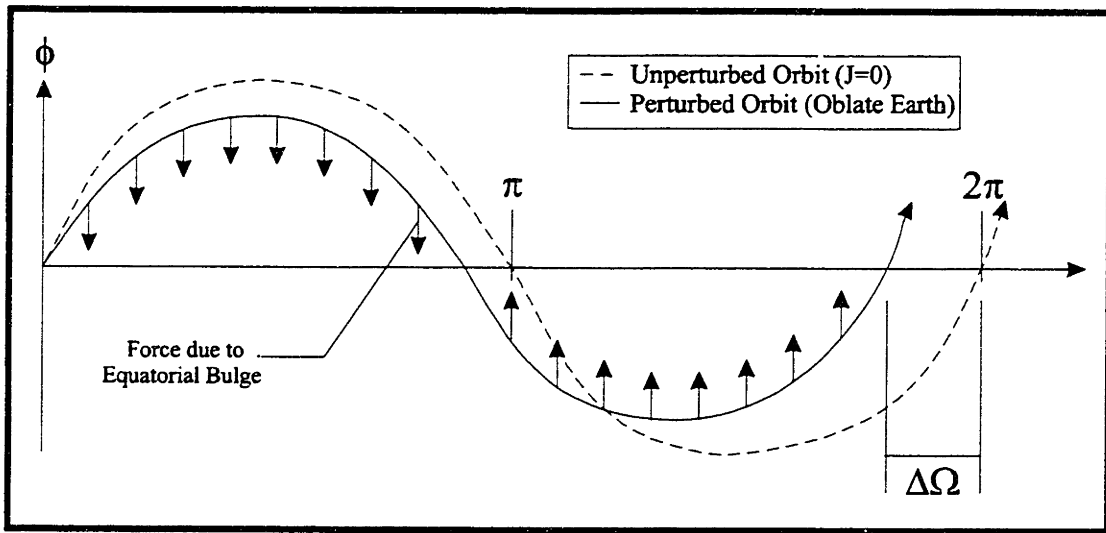


Figure 5.4: Physical Representation of Nodal Regression [102]

The secular error growth in the line of apsides is not as simple to explain physically. Blitzer summarizes this effect by saying that with the force no longer being constrained as the inverse square, the orbit is no longer a closed ellipse. Since the motion is nearly Keplerian, however, the orbit can be represented by a precessing osculating ellipse [102].

In summary, there are a significant number of perturbations which affect highly eccentric orbits during their mission phase. Fortunately, the majority of these perturbations are well modeled (as demonstrated by Fieger in Figures 1.5 and 1.6), and accurate long term predictions are attainable. The major remaining source of error is

caused by the truncation of the equations of motion and corresponding failure to account for higher order potential terms, most notably J_2^2 . Section 5.3 describes an experiment done for a sample Molniya object (NSSC-12066) which identifies J_2^2 as the primary source of error for this investigation, and Section 5.4 describes the progress made towards correcting for this error source.

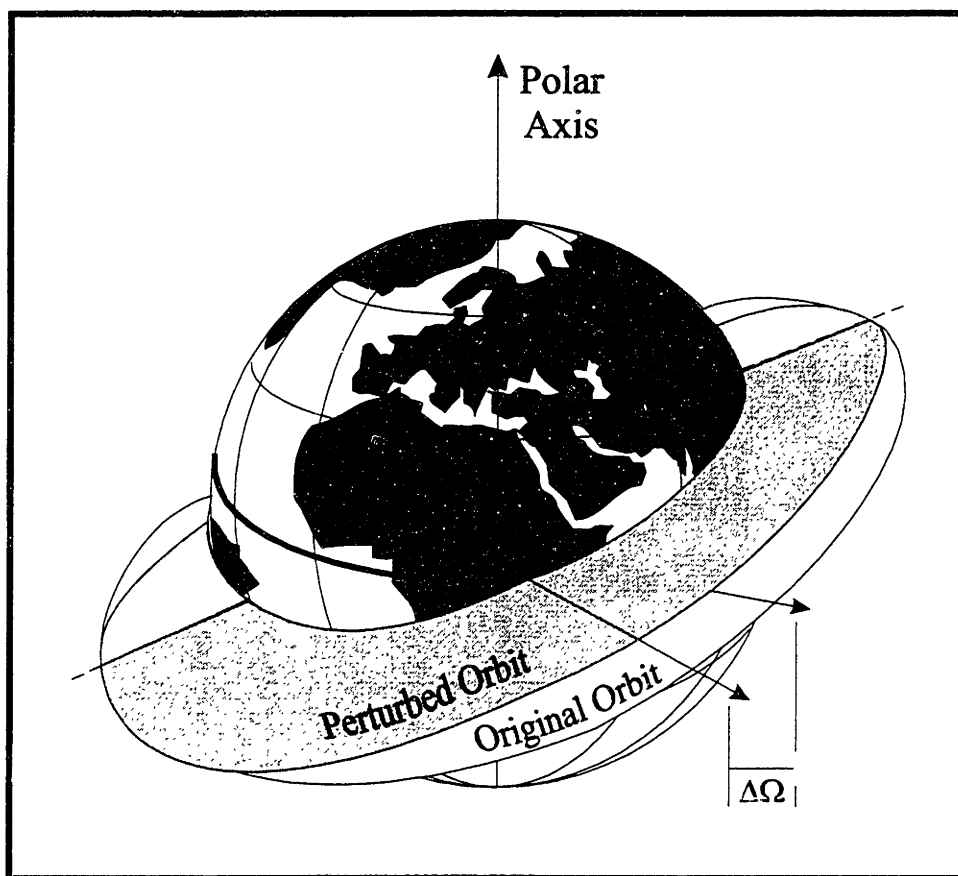


Figure 5.5: Motion of Orbital Plane on Celestial Sphere Due to Nodal Regression

5.3 J_2^2 Error Identification

The first step in finding the incorrectly modeled perturbations of the mission phase was to attempt to process real data (NSSC 12066 – which had the best mission

phase observation density) with the current propagation tools. The result was a very large RMS and slow convergence for each Differential Correction (DC) approximation of the orbit. In order to reduce error and speed convergence, it was necessary to isolate perturbations one at a time and compare the output to “truth” simulated data.

The truth simulation data was produced by the EPHEM program with a highly accurate numerical Cowell generator over a 60 day time span, using all available special perturbations and no drag. The DC program then used this output to do a 30 day fit in the middle of the EPHEM data to complete a PCE Run. Finally, the elements of the DC fit were fed into another EPHEM run which propagated the satellite over a 30 day time span, thus creating the truth comparison file. One by one, perturbations were then isolated with the DSST propagator, using the above process, and the DSST output was compared to the truth data in the hopes of finding an explainable reason for prediction errors. The first error source investigated was reducing the gravity field model from a complete 8x8 model to a 2x0 model. The error signature remained the same, and other perturbations were investigated. Figures 5.6, 5.7, and 5.8, show the cross-track and Ω prediction errors respectively between DSST and Cowell, using a 2x0 gravitational field and all available perturbations except drag. The characteristic secular growth in Ω prediction error is clearly evident in Figure 5.8.

Figures 5.9, 5.10, and 5.11 show the same errors using a value for J_2 three orders of magnitude smaller than the physical value ($J_{2\text{-new}} = 1.08263 \times 10^{-5}$) to isolate the J_2^2 signature. The reader should note that the errors are several orders of magnitude smaller in Figures 5.9-5.11, than in 5.6-5.8, indicating the isolation of a significant error source. The initial result of this experiment has therefore been the identification of the J_2^2 term (higher order oblateness terms) as a primary factor in the DSST’s prediction error.

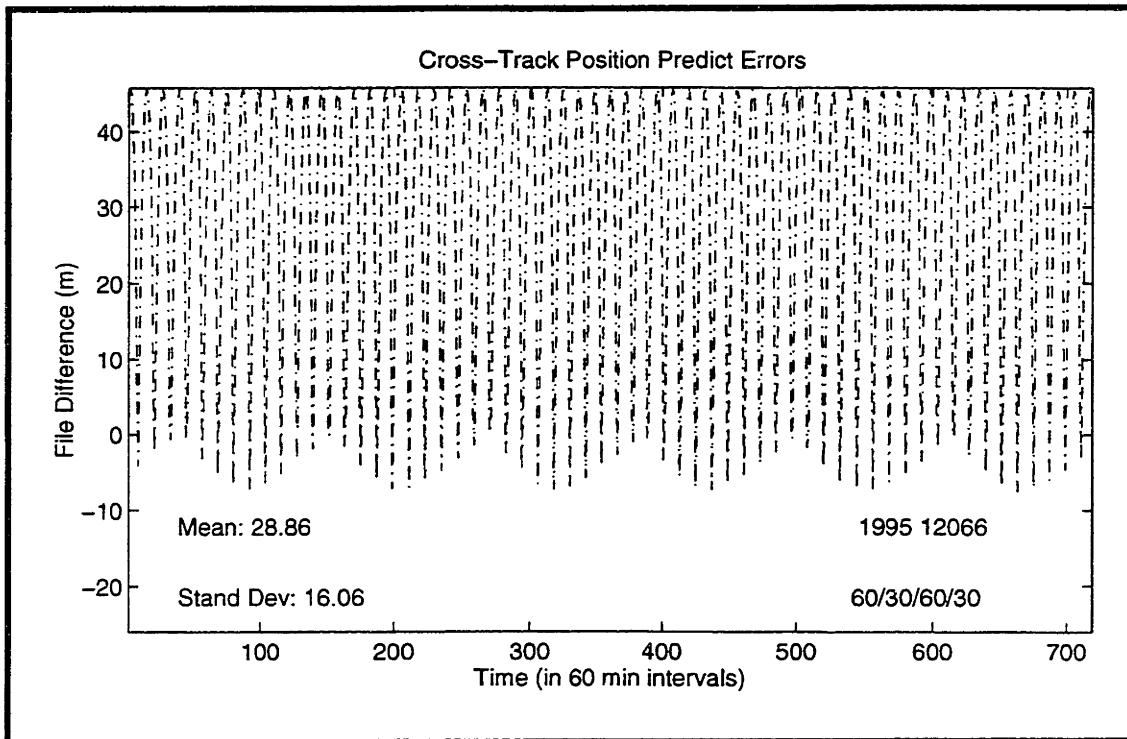


Figure 5.6: Full Model Cross Track Position Prediction Error

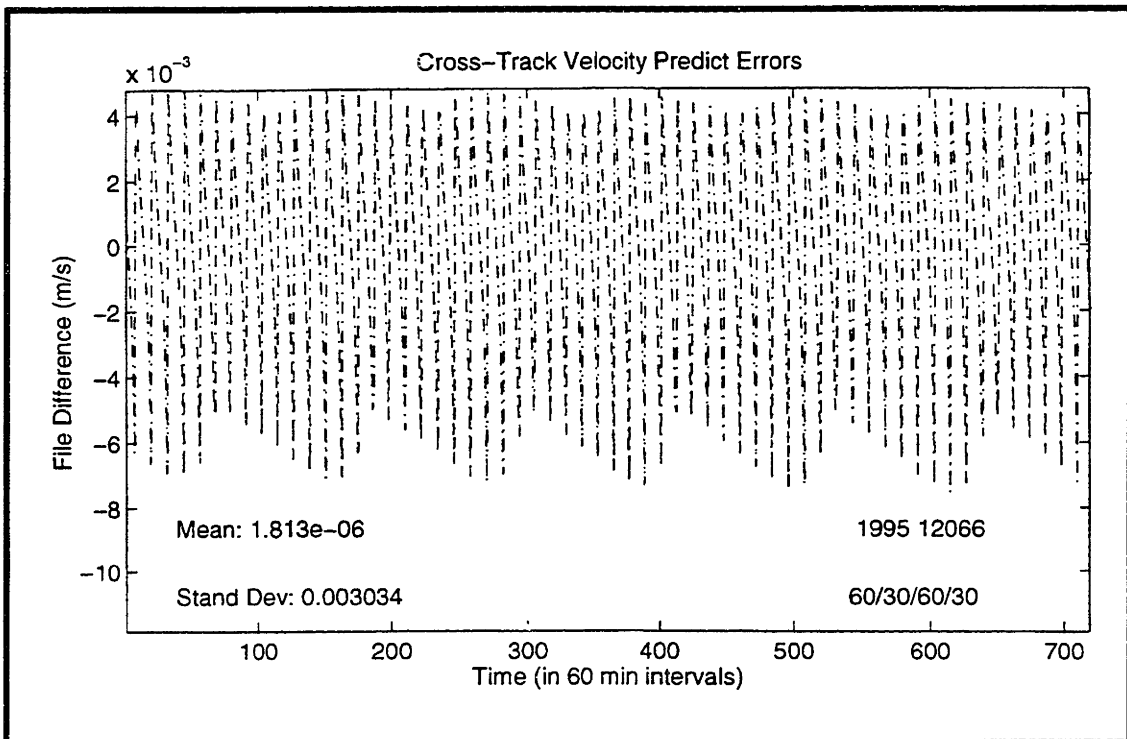


Figure 5.7: Full Model Cross Track Velocity Prediction Error

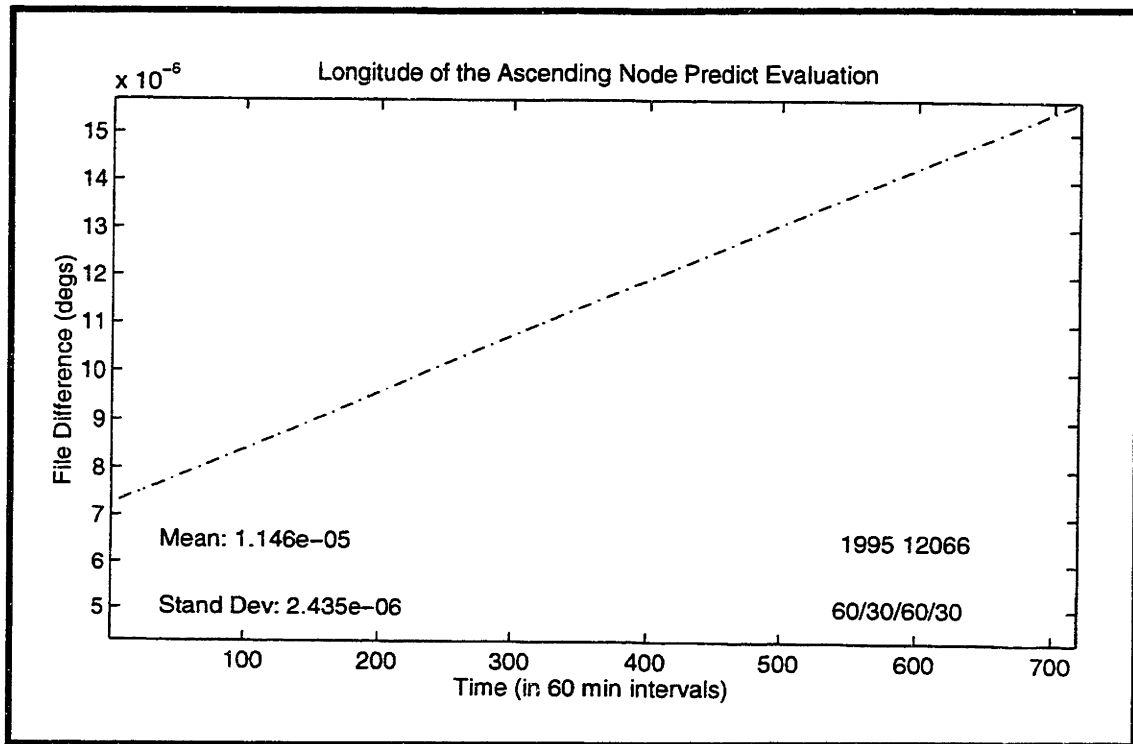


Figure 5.8: Full Model Longitude of Ascending Node (Ω) Prediction Error

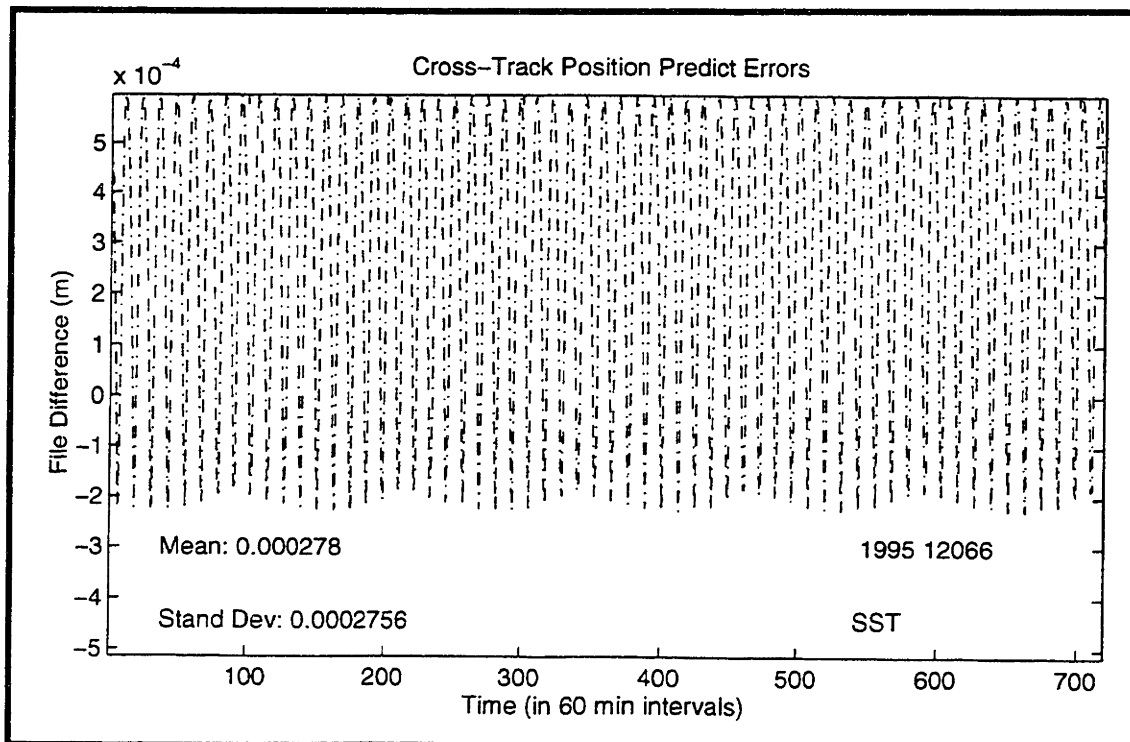


Figure 5.9: Cross Track Position Prediction Error for Minimized J_2^2

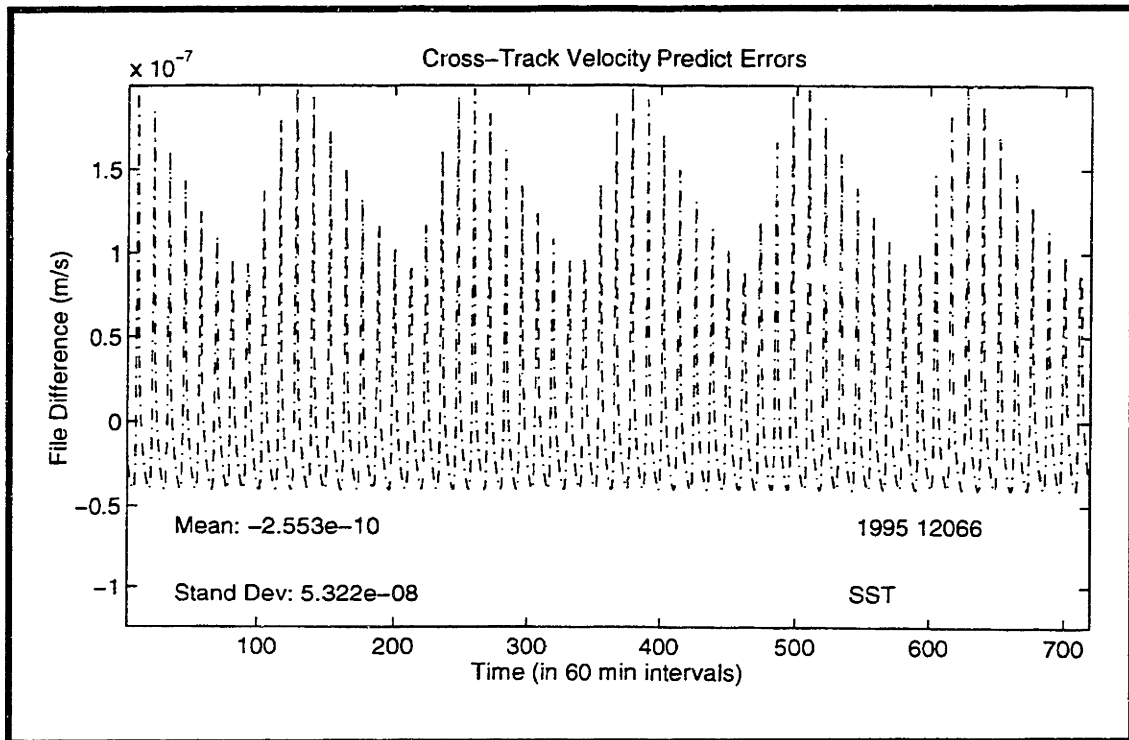


Figure 5.10: Cross Track Velocity Prediction Error for Minimized J_2^2

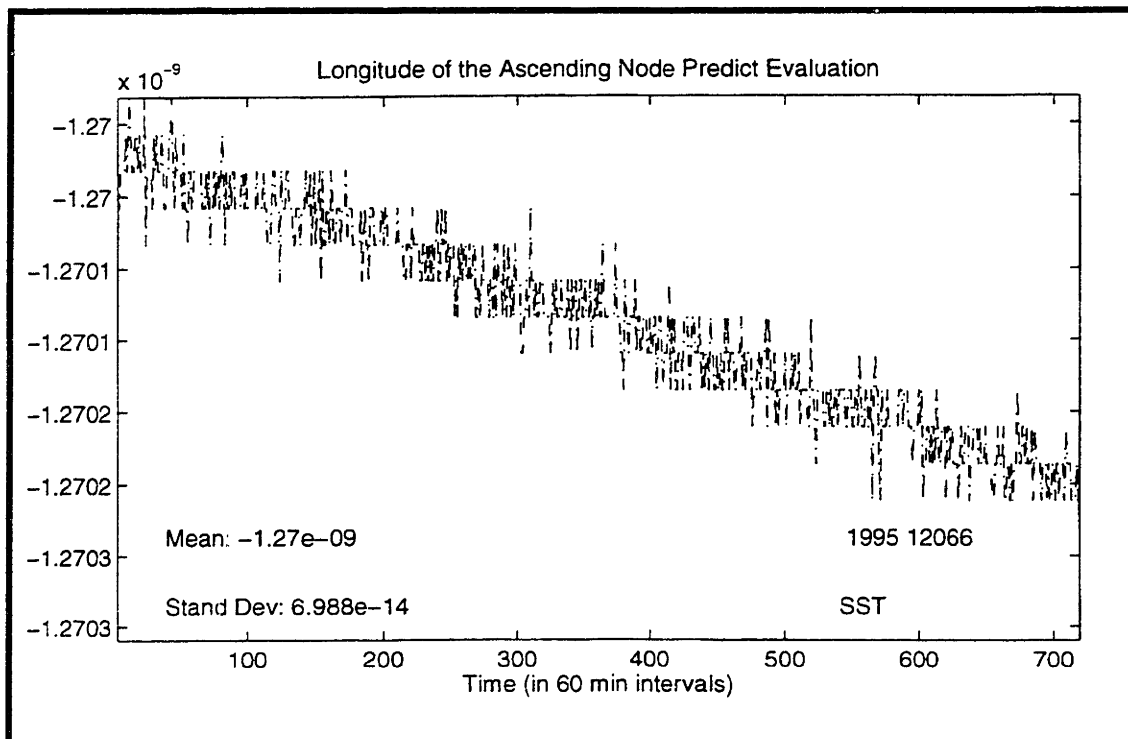


Figure 5.11: Longitude of Ascending Node (Ω) Prediction Error for Minimized J_2^2

The above plots were created with utilities developed by Scott Carter [40] for the VAX environment. In order to transfer data, in binary form, to the VAX system and back to the PC, several utilities were used. These utilities are self explanatory, so the interested reader should simply reference them at:

- \$E53-BOB\D:\MOLYDATA\DATA-TRANSFER

Carter's MATLAB utilities can be found at

- \$E53-BOB\D:\EVO-HEO\MATLAB.

5.4 Progress on the J_2^2 Closed-Form Solution

In order to successfully solve the J_2^2 problem, a computational procedure must be developed for Equations (2.66a) through (2.66e) (DSST algorithm) through second order. To do so, the J^{th} – order contributions must be determined explicitly.

$$A_{i,1} = \langle F_i(\bar{\mathbf{a}}, \bar{\ell}) \rangle_{\bar{\tau}} \rightarrow (i = 1, 2, \dots, 6) \quad (2.66a)$$

$$\eta_{i,1} = \frac{1}{\bar{n}} \int F_i^s(\bar{\mathbf{a}}, \bar{\ell}) d\bar{\ell} \rightarrow (i = 1, 2, \dots, 5) \quad (2.66b)$$

$$\eta_{6,1} = \frac{1}{\bar{n}} \int \left[F_6^s(\bar{\mathbf{a}}, \bar{\ell}) - \frac{3}{2} \frac{\bar{n}}{\bar{a}_1} \eta_{1,1} \right] d\bar{\ell} \quad (2.66c)$$

$$A_{i,2} = \left\langle \sum_{k=1}^6 \eta_{k,1} \frac{\partial F_i}{\partial \bar{a}_k} \right\rangle_{\bar{\tau}} \quad (2.66d)$$

$$A_{6,2} = \left\langle \sum_{k=1}^6 \eta_{k,1} \frac{\partial F_6}{\partial \bar{a}_k} + \frac{15}{4} \frac{\bar{n}}{\bar{a}_1^2} \eta_{1,1}^2 \right\rangle_{\bar{\tau}} \quad (2.66e)$$

Looking at the general equations for the osculating element rates, given in Equations (2.27) and (2.28), the first order contributions can be determined by averaging these equations over the fast variable and expanding to first order in a small parameter (such as eccentricity). These derivation of these contributions is shown in Equation (5.3).

$$\frac{da_i}{dt} = -\sum_{j=1}^6 (a_i, a_j) \frac{\partial \mathbf{R}}{\partial a_j} \rightarrow (i = 1, 2, \dots, 5) \quad (2.27)$$

$$\frac{d\ell}{dt} = n - \sum_{j=1}^6 (\ell, a_j) \frac{\partial \mathbf{R}}{\partial a_j} \quad (2.28)$$

$$\varepsilon A_{i,1} = -\sum_{j=1}^6 (\bar{a}_i, \bar{a}_j) \frac{1}{2\pi} \int_0^{2\pi} \frac{\partial \mathbf{R}}{\partial a_j} d\ell \quad (5.3a)$$

$$\varepsilon A_{i,1} = -\sum_{j=1}^6 (\bar{a}_i, \bar{a}_j) \frac{\partial}{\partial a_j} \left[\frac{1}{2\pi} \int_0^{2\pi} \frac{\partial \mathbf{R}}{\partial a_j} d\ell \right] \quad (5.3b)$$

$$\varepsilon A_{i,1} = -\sum_{j=1}^6 (\bar{a}_i, \bar{a}_j) \frac{\partial}{\partial a_j} \bar{\mathbf{R}} \quad (5.3c)$$

Alternately, Cefola [92] defines writes (5.3c) in the form given in Equation (5.4) where U represents the gravitational potential.

$$\varepsilon F_i = -\sum_{j=1}^6 (a_i, a_j) \frac{\partial U_{nmst}}{\partial a_j} \quad (5.4)$$

To develop closed-form, second order, equations of motion in J_2 for use with DSST, the following steps must be taken:

- The explicit development of the closed form disturbance potential due to J_2 , $U_{2,0}$.

- The explicit development of the right hand side of the osculating, closed-form, VOP equations of motion, F_i .
- Determining the partials of F , $\frac{\partial F_i}{\partial \bar{a}_k}$.
- Identify the specific form for J_2 of the general form of the first order zonal short periodics, as described by Slutsky [94] and given in Equation (5.5).

$$\eta_{k,1} = C_{i,0} + S_{i,0}(L - \lambda) + \sum_{k=1}^5 C_{i,k} \cos(KL) + S_{i,k} \sin(KL) \quad (5.5)$$

- Determine the C and S coefficients using Slutsky's SPZONAL within GTDS.
- Determine the EOM relationships (2.66d) and (2.66e), by developing the given summations through determination of the product of (5.5) with the partials of F .
- Average the EOM relationships over the fast variable.

The current investigation has shown that F functions are extremely complex, and accordingly, the product of the partials of F and the short periodic functions are very difficult to average. One method of solution may be to average these equations as they are using some form of numerical quadrature. Another option explored in this investigation, is to expand (2.66d) and (2.66e) into a different form, given in Equation (5.6).

$$A_{i,2} = - \sum_{k=1}^6 \sum_{j=1}^6 \left\{ \left[\frac{\partial}{\partial \bar{a}_k} (a_i, a_j) \right] \left\langle \eta_{k,1} \frac{\partial U_{2,0}}{\partial \bar{a}_j} \right\rangle_{\bar{\tau}} \right\} - \sum_{k=1}^6 \sum_{j=1}^6 \left[(a_i, a_j) \left\langle \eta_{k,1} \frac{\partial^2 U_{2,0}}{\partial \bar{a}_k \partial \bar{a}_j} \right\rangle_{\bar{\tau}} \right] \quad (5.6)$$

This form of the second order contributions allows the user to avoid computation of the F function, and by putting the gravitational potential in a Fourier series like that shown in Equation (5.7), some of the difficulty of the averaging process can be alleviated.

$$U_{2,0} = \alpha_0 + \sum_{k=1}^5 [\alpha_k \cos(KL) + \beta_k \sin(KL)] \quad (5.7)$$

Unfortunately, the general form of the derivative of (5.6), given in Equation (5.8) becomes extremely complicated, especially for h, k, and λ because True Longitude (L) is a function of them.

$$\frac{\partial U_{2,0}}{\partial \bar{a}_i} = \frac{\partial \alpha_0}{\partial \bar{a}_i} + \sum_{k=1}^5 \left[\frac{\partial \alpha_k}{\partial \bar{a}_i} \cos(KL) + \frac{\partial \beta_k}{\partial \bar{a}_i} \sin(KL) \right] + \sum_{k=1}^5 \left\{ k \frac{\partial L}{\partial \bar{a}_i} [\beta_k \cos(KL) - \alpha_k \sin(KL)] \right\} \quad (5.8)$$

Adding insult to injury is that the second derivative of (5.7) must also be taken to determine the second portion of (5.6). By determining a similar Fourier series representation of the partials of U, determination of the second derivatives would be fairly simple and the averaging process would also be simplified. Unfortunately reorganizing the terms of explicitly writing out (5.8) involves thousands of terms.

Using the symbolic algebra program Macsyma as a test bed, and building upon utilities developed by Zeis, several steps in the above process have been accomplished. Unfortunately time and computer constraints have limited progress and, although others (such as Metris [97]) have attained closed form solutions with different element sets, that was not the case for this investigation (which utilized equinoctial elements for use with DSST). For ideas of future work in this area, please reference Chapter 8.

5.4.1 Development of the Gravitational Potential Function

To develop the proper equations of motion for the J_2^2 problem, it is necessary to derive the gravitational potential due only to the second zonal harmonic, $U_{2,0}$. The complex notation of Cefola [91] (which simplifies Equation (5.1)), is employed to yield Equations (5.9), (5.10), and (5.11), where θ is Greenwich Sidereal Time (GST), and α is the inertial right ascension.

$$U_{nm} = \text{Real}\{U_{nm}^*\} \quad (5.9)$$

$$U_{nm}^* = \frac{\mu}{r} \left(\frac{R_e}{r} \right)^n C_{nm}^* e^{-jm\theta} P_{nm} \sin(\phi) e^{jm\alpha} \quad (5.10)$$

$$C_{n,m}^* = C_{n,m} - jS_{n,m} \quad (5.11)$$

Cefola then performs an expansion of the spherical harmonics to obtain a closed form expression for P (using equinoctial elements), given in Equation (5.12). Equations (5.13a) – (5.13c) define the S functions for $m \geq 0$, and (5.13d) defines γ .

$$P_{nm} \sin(\phi) e^{jm\alpha} = \sum_{k=-n}^n \frac{(n-k)!}{(n-m)!} P_{n,k} (0) S_{2n}^{(m,k)}(p,q) e^{jkl} \quad (5.12)$$

$$S_{2n}^{(m,k)}(p,q) = (1+p^2+q^2)^k (p-jq)^{m-k} P_{n+k}^{(m-k,-m-k)}(\gamma) \rightarrow k \leq -m \quad (5.13a)$$

$$S_{2n}^{(m,k)}(p,q) = \frac{(n+m)!(n-m)!}{(n+k)!(n-k)!} (1+p^2+q^2)^{-m} (p-jq)^{m-k} P_{n-m}^{(m-k,k+m)}(\gamma) \rightarrow -m \leq k \leq +m \quad (5.13b)$$

$$S_{2n}^{(m,k)}(p,q) = (-1)^{m-k} (1+p^2+q^2)^{-k} (p+jq)^{k-m} P_{n-k}^{(k-m,k+m)}(\gamma) \rightarrow k \geq m \quad (5.13c)$$

$$\gamma = \frac{1-p^2-q^2}{1+p^2+q^2} \quad (5.13d)$$

Substitution of (5.12) into (5.10) yields Equation (5.14).

$$U_{nm}^* = \frac{\mu}{r} \left(\frac{R_e}{r} \right)^n C_{nm}^* e^{-jm\theta} \sum_{k=-n}^n \frac{(n-k)!}{(n-m)!} P_{n,k}(0) S_{2n}^{(m,k)}(p,q) e^{jkl} \quad (5.14)$$

This equation is simplified by setting $m = 0$ for the zonal harmonics, and the introduction of the term (a/r) , creating Equation (5.15) for U , and Equation (5.16) for C^* .

$$U_{n,0}^* = \frac{\mu}{a} \left(\frac{R_e}{a} \right)^n \left(\frac{a}{r} \right)^{n+1} C_{n,0}^* \sum_{k=-n}^n \frac{(n-k)!}{n!} P_{n,k}(0) S_{2n}^{(0,k)}(p,q) e^{jkl} \quad (5.15)$$

$$C_{n,0}^* = C_{n,0} - jS_{n,0} = C_{n,0} = -J_n \quad (5.16)$$

For further simplification, the V function defined in Equations (5.17a) and (5.17b) are substituted into (5.15) to yield Equation (5.18).

$$V_{n,k}^m = \left(\frac{(n-k)!}{(n-m)!} \right) P_{n,k}(0) \rightarrow k > 0 \quad (5.17a)$$

$$V_{n,k}^m = (-1)^k \left(\frac{(n-k)!}{(n-m)!} \right) P_{n,k}(0) \rightarrow k < 0 \quad (5.17b)$$

$$U_{n,0} = \text{Real} \left\{ -\frac{\mu}{a} \left(\frac{R_e}{a} \right)^n \left(\frac{a}{r} \right)^{n+1} J_n \sum_{k=-n}^n V_{n,k}^0 S_{2n}^{(0,k)}(p,q) e^{jkl} \right\} \quad (5.18)$$

Therefore, for order 2, Equation (5.19) represents the gravitational potential due only to the second zonal harmonic.

$$U_{2,0} = \text{Real} \left\{ -\frac{\mu}{a} \left(\frac{R_e}{a} \right)^2 \left(\frac{a}{r} \right)^3 J_2 \sum_{k=-2}^2 V_{2,k}^0 S_4^{(0,k)}(p,q) e^{jkl} \right\} \quad (5.19)$$

Using Macsyma block utilities developed for this thesis, and others developed by Zeis (all of which will be outlined in Section 5.4.4), (5.19) was further simplified in equinoctial

elements. Using LEGZ1 to find $V_{2,k}^0$, SFUNCT1 to find $S_4^{(0,k)}(p,q)$, and substituting $B \equiv \left(\frac{a}{r}\right) \rightarrow c \equiv \tan^2 \frac{i}{2}$, (5.19) is simplified to its final form given in (5.20).

$$U_{2,0} = - \frac{J_2 R_e^2 \mu B^3 \left\{ \frac{3[(p^2 - q^2)\cos(2L) - 2pq\sin(2L)]}{(c+1)^2} - \frac{c^2 - 4c + 1}{2(c+1)^2} \right\}}{a^3} \quad (5.20)$$

5.4.2 Development of Necessary Partial Derivatives

As discussed above, several partial derivatives are needed for this derivation, which are presented below.

5.4.2.1 Development of the Eccentric Longitude Partial Derivatives

Since Eccentric Longitude (F), defined in Equation (5.21), is not dependent on a , p , or q , partial derivatives with respect to these elements are therefore equal to zero.

$$F = E + \omega + I\Omega \quad (5.21)$$

Using Green's [95] Equation (2-12) for the mean longitude in equinoctial elements, given in (5.22), the partial of F with respect to λ is given in Equation (5.23).

$$\lambda = F - k \sin F + h \cos F \quad (5.22)$$

$$1 = \frac{\partial F}{\partial \lambda} - k \cos F \frac{\partial F}{\partial \lambda} - h \sin F \frac{\partial F}{\partial \lambda} \quad (5.23)$$

Substituting Green's (2-6) for r , given in (5.24), yields the partial of F with respect to λ , shown in (5.25).

$$r = a(1 - k \cos F - h \sin F) \quad (5.24)$$

$$\frac{\partial F}{\partial \lambda} = \frac{a}{r} \quad (5.25)$$

Next, the partial of (5.22) is taken with respect to h , resulting in (5.26). Substitution of (5.24) into (5.26) results in the derivative of F with respect to h , given in Equation (5.27)

$$0 = \frac{\partial F}{\partial h} - k \cos F \frac{\partial F}{\partial h} + \cos F - h \sin F \frac{\partial F}{\partial h} \rightarrow -\cos F = \frac{\partial F}{\partial h} (1 - k \cos F - h \sin F) \quad (5.26)$$

$$\frac{\partial F}{\partial h} = -\frac{a}{r} \cos F \quad (5.27)$$

Finally, the partial of (5.22) is taken with respect to k , which results in (5.28). A similar substitution of (5.24) yields the partial derivative of F with respect to k , given in Equation (5.29)

$$0 = \frac{\partial F}{\partial k} - k \cos F \frac{\partial F}{\partial k} - \sin F - h \sin F \frac{\partial F}{\partial k} \rightarrow \sin F = \frac{\partial F}{\partial k} (1 - k \cos F - h \sin F) \quad (5.28)$$

$$\frac{\partial F}{\partial k} = \frac{a}{r} \sin F \quad (5.29)$$

5.4.2.2 Development of the (a/r) Partial Derivatives

The ratio of the semi-major axis and position vector magnitude, (a/r) , has been shown by Green [95] and Cefola [91], [92] to be beneficial in the development of closed form solutions. As such, the derivation of these partials begins with Green's (2-5), shown

in (5.30), which defines r in equinoctial elements and true longitude. Accordingly, (a/r) can be defined as in Equation (5.31).

$$r = \frac{a(1-h^2-k^2)}{1+k \cos L + h \sin L} \quad (5.30)$$

$$\frac{a}{r} = \frac{1+k \cos L + h \sin L}{(1-h^2-k^2)} \quad (5.31)$$

Remember that L is a function of h , k , and λ . As a result, the partials with respect to a , p , and q are equal to zero. For λ , however, the first step in deriving the partials comes from Green's (2-10), given in (5.32a) and (5.32b), which relates true longitude and equinoctial elements, where β is defined in (5.33).

$$\cos L = \frac{a}{r} \left[(1-h^2\beta) \cos F + hk\beta \sin F - k \right] \quad (5.32a)$$

$$\sin L = \frac{a}{r} \left[(1-k^2\beta) \sin F + hk\beta \cos F - h \right] \quad (5.32b)$$

$$\beta = \frac{1}{1 + \sqrt{1-h^2-k^2}} \quad (5.33)$$

Next, the partials of (5.32a) and (5.32b) are taken with respect to λ , and (5.32a) and (5.32b) are then substituted back in to simplify partials to Equations (5.34a) and (5.34b).

$$-\sin L \frac{\partial L}{\partial \lambda} = \frac{\partial \left(\frac{a}{r} \right)}{\partial \lambda} \left[\frac{\cos L}{\frac{a}{r}} \right] + \frac{a}{r} \left[-(1-h^2\beta) \sin F \frac{\partial F}{\partial \lambda} + hk\beta \cos F \frac{\partial F}{\partial \lambda} \right] \quad (5.34a)$$

$$\cos L \frac{\partial L}{\partial \lambda} = \frac{\partial \left(\frac{a}{r} \right)}{\partial \lambda} \left[\frac{\sin L}{\frac{a}{r}} \right] + \frac{a}{r} \left[(1-k^2\beta) \cos F \frac{\partial F}{\partial \lambda} - hk\beta \sin F \frac{\partial F}{\partial \lambda} \right] \quad (5.34b)$$

Multiplying (5.34a) by $\cos(L)$ and (5.34b) by $\sin(L)$, and substitution of the partial of F from above, results in Equations (5.35a) and (5.35b).

$$-\sin L \cos L \frac{\partial L}{\partial \lambda} = \frac{\partial \left(\frac{a}{r} \right)}{\partial \lambda} \left[\frac{\cos^2 L}{\frac{a}{r}} \right] + \left(\frac{a}{r} \right)^2 \cos L [hk\beta \cos F - (1 - h^2\beta) \sin F] \quad (5.35a)$$

$$\cos^2 L \frac{\partial L}{\partial \lambda} = \frac{\partial \left(\frac{a}{r} \right)}{\partial \lambda} \left[\frac{\cos L \sin L}{\frac{a}{r}} \right] + \left(\frac{a}{r} \right)^2 \cos L \left[\frac{\dot{Y}_1}{na^2} \right] \quad (5.35b)$$

A simple substitution, using the following velocity coordinate relations from Green (2-11c) and (2.11d), given in (5.36a) and (5.36b), followed by an addition of (5.35a) and (5.35b), results in Equation (5.37).

$$\dot{X}_1 = \frac{na^2}{r} [hk\beta \cos F - (1 - h^2\beta) \sin F] \quad (5.36a)$$

$$\dot{Y}_1 = \frac{na^2}{r} [(1 - k^2\beta) \cos F - hk\beta \sin F] \quad (5.36b)$$

$$0 = \frac{\partial \left(\frac{a}{r} \right)}{\partial \lambda} \left[\frac{1}{\frac{a}{r}} \right] + \left[\left(\frac{a}{r} \right)^2 \cos L \frac{\dot{X}_1}{\frac{na^2}{r}} \right] + \left[\left(\frac{a}{r} \right)^2 \sin L \frac{\dot{Y}_1}{\frac{na^2}{r}} \right] \quad (5.37)$$

This equation then simplifies to (5.38).

$$\frac{\partial \left(\frac{a}{r} \right)}{\partial \lambda} = - \left(\frac{a}{r} \right)^2 \left(\frac{1}{na} \right) [\dot{X}_1 \cos L + \dot{Y}_1 \sin L] \quad (5.38)$$

Next, by substituting Green's (2-9d) and (2-9e), given in (5.39a) and (5.39b), Equation (5.40) is produced.

$$\dot{X}_1 = \frac{-na(h + \sin L)}{\sqrt{1-h^2-k^2}} \quad (5.39a)$$

$$\dot{Y}_1 = \frac{-na(k + \cos L)}{\sqrt{1-h^2-k^2}} \quad (5.39b)$$

$$\frac{\partial L}{\partial \lambda} = \left(\frac{a}{r}\right)(1-h^2-k^2)^{-\frac{1}{2}} \left(\frac{a}{r}\right)(1-h^2-k^2) \quad (5.40)$$

Further simplification yields the final form of the partial of (a/r) with respect to λ , given in Equation (5.41).

$$\frac{\partial L}{\partial \lambda} = \left(\frac{a}{r}\right)^2 \sqrt{1-h^2-k^2} \quad (5.41)$$

The first step in determining the partial with respect to h, is to define the partials of β with respect to h and k, as shown in (5.42a) and (5.42b).

$$\frac{\partial \beta}{\partial h} = \frac{h}{\sqrt{1-h^2-k^2} \left(1 + \sqrt{1-h^2-k^2}\right)^2} \quad (5.42a)$$

$$\frac{\partial \beta}{\partial k} = \frac{k}{\sqrt{1-h^2-k^2} \left(1 + \sqrt{1-h^2-k^2}\right)^2} \quad (5.42b)$$

Next the partials of (5.32a) and (5.32b) are taken with respect to h, as shown in (5.43a) and (5.43b).

$$-\sin L \frac{\partial L}{\partial h} = \frac{\partial \left(\frac{a}{r}\right)}{\partial h} \left[\frac{\cos L}{\frac{a}{r}} \right] + \frac{a}{r} \left[-(1-h^2\beta) \sin F \frac{\partial F}{\partial h} + hk\beta \cos F \frac{\partial F}{\partial h} + \frac{\partial \beta}{\partial h} hk \sin F + k\beta \sin F + \left(-\frac{\partial \beta}{\partial h} h^2 - 2h\beta\right) \cos F \right] \quad (5.43a)$$

$$\cos L \frac{\partial L}{\partial h} = \frac{\partial \left(\frac{a}{r}\right)}{\partial h} \left[\frac{\sin L}{\frac{a}{r}} \right] + \frac{a}{r} \left[-hk\beta \sin F \frac{\partial F}{\partial h} + (1-\beta k^2) \cos F \frac{\partial F}{\partial h} - \frac{\partial \beta}{\partial h} k^2 \sin F + \frac{\partial \beta}{\partial h} hk \cos F + k\beta \cos F - 1 \right] \quad (5.43b)$$

(5.43a) is multiplied by $\cos(L)$ and (5.43b) by $\sin(L)$, and the two are added to remove the L partial. Several substitutions are then made, including:

- the partials of F , which are substituted in Equations (5.36) and (5.39) to produce (5.44a) and (5.44b)
- the definition of χ , given in (5.45), and the corresponding partials of β , given in (5.46)

$$\left[hk\beta \cos F - (1 - h^2\beta) \sin F \right] \frac{\partial F}{\partial h} = \chi(h + \sin L) \cos F \quad (5.44a)$$

$$\left[(1 - k^2\beta) \cos F - hk\beta \sin F \right] \frac{\partial F}{\partial h} = -\chi(k + \cos L) \cos F \quad (5.44b)$$

$$\chi = \frac{1}{\sqrt{1 - h^2 - k^2}} \quad (5.45)$$

$$\frac{\partial \beta}{\partial h} = \chi h \beta^2 \quad (5.46)$$

The result of substituting all of these equations into the summation of (5.43), results in Equation (5.47).

$$\frac{\partial \left(\frac{a}{r} \right)}{\partial h} = - \left(\frac{a}{r} \right)^2 \left[\begin{array}{l} -\beta^2 \chi h k^2 \sin F \sin L + \beta^2 \chi h^2 k \cos F \sin L - \chi k \cos F \sin L + \beta k \cos F \sin L - \sin L \\ +\beta^2 \chi h^2 k \sin F \cos L + \beta k \sin F \cos L - \beta^2 \chi h^3 \cos F \cos L + \chi h \cos F \cos L - 2\beta h \cos F \cos L \end{array} \right] \quad (5.47)$$

Grouping like terms results in (5.48).

$$\frac{\partial \left(\frac{a}{r} \right)}{\partial h} = - \left(\frac{a}{r} \right)^2 \left[\begin{array}{l} -\beta^2 \chi h k \sin F (k \sin L - h \cos L) + \beta^2 \chi h^2 \cos F (k \sin L - h \cos L) + \beta \cos F (k \sin L - h \cos L) \\ -\chi \cos F (k \sin L - h \cos L) + \beta \cos L (k \sin F - h \cos F) - \sin L \end{array} \right] \quad (5.48)$$

Using the expression for $\beta-\chi$, given in (5.49), (5.48) then simplifies to the final form of the partial of (a/r) with respect to h , as shown in (5.50).

$$\beta-\chi = \frac{\sqrt{1-h^2-k^2} - (1+\sqrt{1-h^2-k^2})}{\sqrt{1-h^2-k^2}(1+\sqrt{1-h^2-k^2})} = \frac{-1}{\sqrt{1-h^2-k^2}(1+\sqrt{1-h^2-k^2})} = -\chi\beta \quad (5.49)$$

$$\frac{\partial\left(\frac{a}{r}\right)}{\partial h} = -\left(\frac{a}{r}\right)^2 \left[[\cos F(-\chi\beta) - \beta^2\chi h(F-\lambda)](k \sin L - h \cos L) + \beta \cos L(F-\lambda) - \sin L \right] \quad (5.50)$$

Using the same method for k , the reader can develop the k equivalent of (5.47), given in Equation (5.51).

$$\frac{\partial\left(\frac{a}{r}\right)}{\partial k} = -\left(\frac{a}{r}\right)^2 \left[\begin{aligned} &-\beta^2\chi k^3 \sin F \sin L - 2\beta k \sin F \sin L + \beta^2\chi h k^2 \cos F \sin L - \chi k \cos F \sin L + \beta h \cos F \sin L \\ &+ \beta^2\chi h k^2 \sin F \cos L + \beta h \sin F \cos L - \beta^2\chi h^2 k \cos F \cos L + \chi h \cos F \cos L - \cos L \end{aligned} \right] \quad (5.51)$$

Grouping the terms results in (5.52),

$$\frac{\partial\left(\frac{a}{r}\right)}{\partial k} = -\left(\frac{a}{r}\right)^2 \left[\begin{aligned} &\beta^2\chi h k \cos F(k \sin L - h \cos L) - \beta^2\chi k^2 \sin F(k \sin L - h \cos L) - \beta \sin F(k \sin L - h \cos L) \\ &- \chi \cos F(k \sin L - h \cos L) - \beta \sin L(k \sin F - h \cos F) - \cos L \end{aligned} \right] \quad (5.52)$$

Substitution of (5.49) results in the final form of the partial derivative of (a/r) with respect to k , given in (5.53).

$$\frac{\partial\left(\frac{a}{r}\right)}{\partial k} = -\left(\frac{a}{r}\right)^2 \left[[-\chi \cos F - \beta \sin F - \beta^2\chi k(F-\lambda)](k \sin L - h \cos L) - \beta \sin L(F-\lambda) - \cos L \right] \quad (5.53)$$

5.4.2.3 Development of the True Longitude Partial Derivatives

As with eccentric longitude, true longitude is not dependent on a , p , or q , partial derivatives with respect to these elements are therefore equal to zero. For the partial with respect to λ , the same method used for (a/r) may be employed, where (5.34a) is multiplied by $\sin(L)$ and (5.34b) is multiplied by $\cos(L)$. Substituting (5.25) and (5.36) into these products yields (5.54a) and (5.54b).

$$-\sin^2 L \frac{\partial L}{\partial \lambda} = \frac{\partial \left(\frac{a}{r} \right)}{\partial \lambda} \left[\frac{\cos L \sin L}{\frac{a}{r}} \right] + \left(\frac{a}{r} \right)^2 \sin L \left[\frac{\dot{X}_1}{na^2} \right] \left[\frac{1}{r} \right] \quad (5.54a)$$

$$\cos^2 L \frac{\partial L}{\partial \lambda} = \frac{\partial \left(\frac{a}{r} \right)}{\partial \lambda} \left[\frac{\cos L \sin L}{\frac{a}{r}} \right] + \left(\frac{a}{r} \right)^2 \cos L \left[\frac{\dot{Y}_1}{na^2} \right] \left[\frac{1}{r} \right] \quad (5.54b)$$

Subtracting these two equations results in (5.55).

$$\frac{\partial L}{\partial \lambda} = \left(\frac{a}{r} \right)^2 \left[\frac{\dot{Y}_1 \cos L - \dot{X}_1 \sin L}{\frac{na^2}{r}} \right] \quad (5.55)$$

Substitution of (5.39) then simplifies the expression as shown in Equation (5.56).

$$\frac{\partial L}{\partial \lambda} = \left(\frac{a}{r} \right) (1 - h^2 - k^2)^{-\frac{1}{2}} \left(\frac{a}{r} \right) (1 - h^2 - k^2) \quad (5.56)$$

Using the definition of (a/r) from Equation (5.31), results in the final form of the derivative of L with respect to λ , as shown in Equation (5.57).

$$\frac{\partial L}{\partial \lambda} = \left(\frac{a}{r} \right)^2 \sqrt{1 - h^2 - k^2} \quad (5.57)$$

For the partial with respect to h, a similar approach to that used for (a/r) is utilized. (5.43a) is multiplied by sin(L), and (5.43b) by cos(L) to remove the (a/r) partial, resulting in Equation (5.58).

$$\begin{aligned} \frac{\partial L}{\partial h} = & \frac{a}{r} \cos L \left[-hk\beta \sin F \frac{\partial F}{\partial h} + (1-\beta k^2) \cos F \frac{\partial F}{\partial h} - \frac{\partial \beta}{\partial h} k^2 \sin F + \frac{\partial \beta}{\partial h} hk \cos F + k\beta \cos F - 1 \right] \\ & - \frac{a}{r} \sin L \left[-(1-h^2\beta) \sin F \frac{\partial F}{\partial h} + hk\beta \cos F \frac{\partial F}{\partial h} + \frac{\partial \beta}{\partial h} hk \sin F + k\beta \sin F + \left(-\frac{\partial \beta}{\partial h} h^2 - 2h\beta \right) \cos F \right] \end{aligned} \quad (5.58)$$

Substitution of (5.44), then results in Equation (5.59).

$$\begin{aligned} \frac{\partial L}{\partial h} = & \frac{a}{r} \cos L \left[-\chi(k + \cos L) \cos F - \frac{\partial \beta}{\partial h} k^2 \sin F + \left(\frac{\partial \beta}{\partial h} hk + k\beta \right) \cos F - 1 \right] \\ & - \frac{a}{r} \sin L \left[\chi(h + \sin L) \cos F + \left(\frac{\partial \beta}{\partial h} hk + k\beta \right) \sin F + \left(-\frac{\partial \beta}{\partial h} h^2 - 2h\beta \right) \cos F \right] \end{aligned} \quad (5.59)$$

Next, (5.45) and (5.46) are substituted and (5.59) expanded to yield (5.60).

$$\begin{aligned} \frac{\partial L}{\partial h} = & \frac{a}{r} \left[-\chi k \cos L \cos F - \chi \cos^2 L \cos F - \chi h \beta^2 k^2 \cos L \sin F + (\chi h^2 \beta^2 k + k\beta) \cos L \cos F - \cos L \right. \\ & \left. - \chi h \sin L \cos F - \chi \sin L^2 \cos F - (\chi h^2 \beta^2 k + k\beta) \sin L \sin F + (\chi h^3 \beta^2 + 2h\beta) \sin L \cos F \right] \end{aligned} \quad (5.60)$$

Grouping like terms and simplifying with trigonometric identities results in (5.61).

$$\begin{aligned} \frac{\partial L}{\partial h} = & \frac{a}{r} \left[-\chi \cos F (k \cos L + h \sin L + 1) - \chi h \beta^2 k \sin F (k \cos L + h \sin L) + k\beta (\cos F \cos L - \sin L \sin F) \right. \\ & \left. + \chi h^2 \beta^2 \cos F (k \cos L + h \sin L) + 2h\beta \sin L \cos F - \cos L \right] \end{aligned} \quad (5.61)$$

Utilizing the relation in Equations (5.62), and (5.31), and completing the square, results in Equation (5.63).

$$\left(\frac{a}{r} \right) \chi^{-2} = 1 + k \cos L + h \sin L \quad (5.62)$$

$$\begin{aligned} \frac{\partial L}{\partial h} = & \frac{a}{r} \left[-\chi^{-1} \cos F \left(\frac{a}{r} \right) - \chi^{-1} h \beta^2 k \sin F \left(\frac{a}{r} \right) + k \beta (\cos F \cos L - \sin L \sin F) \right. \\ & \left. + \chi^{-1} h^2 \beta^2 \cos F \left(\frac{a}{r} \right) + 2h \beta \sin L \cos F - \cos L + \chi h \beta^2 (k \sin F - h \cos F) \right] \end{aligned} \quad (5.63)$$

Reordering terms, substitution of (5.62), and again completing the square are shown in (5.64) and (5.65).

$$\begin{aligned} \frac{\partial L}{\partial h} = & \frac{a}{r} \left[-\chi^{-1} \cos F \left(\frac{a}{r} \right) + \left(\chi h \beta^2 - \chi^{-1} h \beta^2 \left(\frac{a}{r} \right) \right) (k \sin F - h \cos F) + \beta \cos F \left(\frac{a}{r} \right) \chi^{-2} \right. \\ & \left. - k \beta \sin L \sin F + h \beta \sin L \cos F - \cos L - \beta \cos F \right] \end{aligned} \quad (5.64)$$

$$\frac{\partial L}{\partial h} = \frac{a}{r} \left[\chi^{-1} \cos F \left(\frac{a}{r} \right) (\chi^{-1} \beta - 1) + \left(\chi h \beta^2 - \chi^{-1} h \beta^2 \left(\frac{a}{r} \right) - \beta \sin L \right) (k \sin F - h \cos F) - \cos L - \beta \cos F \right] \quad (5.65)$$

Using the following two relations, given in (5.66) and (5.67), the final form of the partial derivative of L with respect to h can be obtained, and is given in Equation (5.68).

$$\chi^{-1} \beta - 1 = \frac{\sqrt{1-h^2-k^2}}{1+\sqrt{1-h^2-k^2}} - \frac{1+\sqrt{1-h^2-k^2}}{1+\sqrt{1-h^2-k^2}} = \frac{-1}{1+\sqrt{1-h^2-k^2}} = -\beta \quad (5.66)$$

$$\lambda = F - k \sin F + h \cos F \quad (5.67)$$

$$\frac{\partial L}{\partial h} = \frac{a}{r} \left[-\cos F \left(\left(\chi^{-1} \beta \frac{a}{r} \right) + \beta \right) + \left(\chi h \beta^2 - \chi^{-1} h \beta^2 \left(\frac{a}{r} \right) - \beta \sin L \right) (F - \lambda) - \cos L \right] \quad (5.68)$$

The partial with respect to k can be obtained in a similar method to the partial with respect to h. Following the same steps as above, an equivalent of (5.60) for k is shown in Equation (5.69).

$$\begin{aligned} \frac{\partial L}{\partial k} = & \frac{a}{r} \left[-\beta^2 \chi h k^2 \sin F \sin L + \chi h \sin F \sin L - \beta h \sin F \sin L + \beta^2 \chi h^2 k \cos F \sin L - \beta^2 \chi k^3 \sin F \cos L \right. \\ & \left. + \sin L + \chi k \sin F \cos L - 2\beta k \sin F \cos L + \beta^2 \chi h k^2 \cos F \cos L + \beta h \cos F \cos L + \chi \sin F \right] \end{aligned} \quad (5.69)$$

After regrouping and completing the square, this becomes Equation (5.70).

$$\frac{\partial L}{\partial k} = \frac{a}{r} \left[\chi \sin F (1 + h \sin L + k \sin L) - \beta^2 \chi k^2 \sin F (1 + h \sin L + k \cos L) + \beta^2 \chi h k \cos F (1 + h \sin L + k \cos L) \right. \\ \left. + \sin L - \beta \sin F (1 + h \sin L + k \cos L) + \beta \cos L (h \cos F - k \sin F) - \beta^2 \chi k (h \cos F - k \sin F) + \beta \sin F \right] \quad (5.70)$$

Substitution of (5.62) results in Equation (5.71).

$$\frac{\partial L}{\partial k} = \frac{a}{r} \left[\left(\frac{a}{r} \right) \chi^{-2} (\chi \sin F - \beta^2 \chi k^2 \sin F + \beta^2 \chi h k \cos F - \beta \sin F) + (\beta \cos L - \beta^2 \chi k) (h \cos F - k \sin F) + \sin L + \beta \sin F \right] \quad (5.71)$$

Final regrouping, and substitution of (5.66) and (5.67), result in the final form of the partial derivative of L with respect to k, shown in (5.72).

$$\frac{\partial L}{\partial k} = \frac{a}{r} \left[\sin F \left(\left(\frac{a}{r} \right) \chi^{-1} \beta + \beta \right) + \left(-\chi k \beta^2 + \left(\frac{a}{r} \right) \chi^{-1} k \beta^2 + \beta \cos L \right) (F - \lambda) + \sin L \right] \quad (5.72)$$

5.4.2.4 Development of the Potential Function Partial Derivatives

The partial derivatives of the U function are fairly simple, and require no major substitutions. As such, they are simply presented in Equations (5.73a) through (5.73h). For a derivation of these partials, refer to the Macsyma file found at \$E53-BOB\D:\EVO-HEO\MACSYMA\J22\FILES\U-PART.MFE.

$$\frac{\partial U_{2,0}}{\partial a} = \frac{3J_2 R_c^2 \mu B^3 \left\{ \frac{3[(p^2 - q^2) \cos(2L) - 2pq \sin(2L)]}{(c+1)^2} - \frac{c^2 - 4c + 1}{2(c+1)^2} \right\}}{a^4} \quad (5.73a)$$

$$\frac{\partial U_{2,0}}{\partial p} = \frac{J_2 R_c^2 \mu B^3 \left\{ \frac{2(c^2 - 4c + 1)p - 12p[(p^2 - q^2) \cos(2L) - 2pq \sin(2L)]}{(c+1)^3} + \frac{6(2p \cos(2L) - 2q \sin(2L)) - (4cp - 8p)}{2(c+1)^2} \right\}}{a^3} \quad (5.73b)$$

$$\frac{\partial U_{2.0}}{\partial q} = - \frac{J_2 R_e^2 \mu B^3 \left\{ \frac{2(c^2 - 4c + 1)q - 12q[(p^2 - q^2)\cos(2L) - 2pq\sin(2L)]}{(c+1)^3} + \frac{6(-2p\cos(2L) - 2q\sin(2L)) - (4cq - 8q)}{2(c+1)^2} \right\}}{a^3} \quad (5.73c)$$

$$\frac{\partial U_{2.0}}{\partial h} = 0 \quad (5.73d)$$

$$\frac{\partial U_{2.0}}{\partial k} = 0 \quad (5.73e)$$

$$\frac{\partial U_{2.0}}{\partial \lambda} = 0 \quad (5.73f)$$

$$\frac{\partial U_{2.0}}{\partial \frac{a}{r}} = - \frac{3J_2 R_e^2 \mu B^2 \left\{ \frac{3[(p^2 - q^2)\cos(2L) - 2pq\sin(2L)]}{(c+1)^2} - \frac{c^2 - 4c + 1}{2(c+1)^2} \right\}}{a^3} \quad (5.73g)$$

$$\frac{\partial U_{2.0}}{\partial L} = - \frac{3J_2 R_e^2 \mu B^3 [-2(p^2 - q^2)\sin(2L) - 4pq\cos(2L)]}{a^3(c+1)^2} \quad (5.73h)$$

5.4.2.5 Summary of Necessary Partial Derivatives

In order to more efficiently wade through the multitude of equations presented above, Table 5.2 is presented below to afford the reader a reference to the locations of above partial derivatives. A corresponding equation number is given next to each partial symbol (as opposed to the equation itself) to minimize the space required for the reference table.

Table 5.2: Summary Reference of Necessary Partial Derivatives Equation Numbers

$\frac{\partial(\text{column})}{\partial(\text{row})}$	Potential ($U_{2,0}$)	True Longitude (L)	Eccentric Longitude (F)	$\frac{a}{r}$ Ratio
a	5.73a			
h		5.68	5.27	5.50
k		5.72	5.29	5.53
p	5.73b			
q	5.73c			
λ		5.41, 5.57	5.25	5.38
$\frac{a}{r}$	5.73g	5.72		
L	5.73h			

5.4.3 Development of the F Functions

The first step in determining the F functions is to develop the partials of the variables L and (a/r) , as shown in Section 4.2. Next, it is necessary to develop a simple representation of the gravitational potential, which was accomplished by separating the S function into its real (represented by C terms) and imaginary (represented by S terms) parts. The resulting Equation is given in (5.74):

$$U_{2,0} = -\left(\frac{\mu}{a}\right)\left(\frac{R_e}{a}\right)^2\left(\frac{a}{r}\right)^3 J_2 \left[-\frac{1}{2} S_4^{(0,0)} + 3(C_4^{(0,2)} \cos(2L) - S_4^{(0,2)} \sin(2L)) \right] \quad (5.74)$$

Next, the partials of this modified version of the geopotential are obtained, as shown in Equations (5.75a) through (5.75h).

$$\frac{\partial U_{2.0}}{\partial a} = -\frac{3}{a} U_{2.0} \quad (5.75a)$$

$$\frac{\partial U_{2.0}}{\partial p} = -3 \left(\frac{\mu}{a} \right) \left(\frac{R_e}{a} \right)^2 \left(\frac{a}{r} \right)^3 J_2 \left[\frac{\partial C_4^{(0.2)}}{\partial p} \cos(2L) - \frac{\partial S_4^{(0.2)}}{\partial p} \sin(2L) \right] \quad (5.75b)$$

$$\frac{\partial U_{2.0}}{\partial q} = -3 \left(\frac{\mu}{a} \right) \left(\frac{R_e}{a} \right)^2 \left(\frac{a}{r} \right)^3 J_2 \left[\frac{\partial C_4^{(0.2)}}{\partial q} \cos(2L) - \frac{\partial S_4^{(0.2)}}{\partial q} \sin(2L) \right] \quad (5.75c)$$

$$\frac{\partial U_{2.0}}{\partial L} = 6 \left(\frac{\mu}{a} \right) \left(\frac{R_e}{a} \right)^2 \left(\frac{a}{r} \right)^3 J_2 \left[(C_4^{(0.2)} \sin(2L) + S_4^{(0.2)} \cos(2L)) \right] \quad (5.75d)$$

$$\frac{\partial U_{2.0}}{\partial \left(\frac{a}{r} \right)} = \frac{3r}{a} U_{2.0} \quad (5.75e)$$

$$\frac{\partial U_{2.0}}{\partial h} = \frac{\partial U_{2.0}}{\partial L} \frac{\partial L}{\partial h} + \left(\frac{3r}{a} U_{2.0} \right) \frac{\partial \left(\frac{a}{r} \right)}{\partial h} \quad (5.75f)$$

$$\frac{\partial U_{2.0}}{\partial k} = \frac{\partial U_{2.0}}{\partial L} \frac{\partial L}{\partial k} + \left(\frac{3r}{a} U_{2.0} \right) \frac{\partial \left(\frac{a}{r} \right)}{\partial k} \quad (5.75g)$$

$$\frac{\partial U_{2.0}}{\partial \lambda} = \frac{\partial U_{2.0}}{\partial L} \frac{\partial L}{\partial \lambda} + \left(\frac{3r}{a} U_{2.0} \right) \frac{\partial \left(\frac{a}{r} \right)}{\partial \lambda} \quad (5.75h)$$

The reader should note that the partials of the C and S terms are given by Equations (5.76) and (5.77).

$$\frac{\partial C_4^{(0,2)}}{\partial p} = \frac{1}{q} S_4^{(0,2)} - 4p C_2^{(1,1)} C_4^{(0,2)} \quad (5.76a)$$

$$\frac{\partial C_4^{(0,2)}}{\partial q} = -\frac{1}{p} S_4^{(0,2)} - 4q C_2^{(1,1)} C_4^{(0,2)} \quad (5.76b)$$

$$\frac{\partial S_4^{(0,2)}}{\partial q} = \left(\frac{1}{p} - 4p C_2^{(1,1)} \right) S_4^{(0,2)} \quad (5.77a)$$

$$\frac{\partial S_4^{(0,2)}}{\partial p} = \left(\frac{1}{q} - 4q C_2^{(1,1)} \right) S_4^{(0,2)} \quad (5.77b)$$

In order to simplify the resulting expressions for the F functions, the following d terms have been defined to represent the summation over the Poisson brackets of the U partials, and are given in Equation (5.78).

$$\begin{aligned} d_1 &= \frac{2}{an} = -(a, \lambda) \\ d_2 &= \frac{1}{a^2 n(x+1)} = -\frac{1}{h}(\lambda, h) = -\frac{1}{k}(\lambda, k) \\ d_3 &= \frac{(c+1)x}{2a^2 n} = -\frac{1}{p}(\lambda, p) = -\frac{1}{q}(\lambda, q) \\ d_4 &= \frac{1}{a^2 nx} = -(h, k) \\ d_5 &= \frac{(c+1)^2 x}{4a^2 n} = -(p, q) \end{aligned} \quad (5.78)$$

Using these terms results in the following form of the F Functions, given in Equations (5.79a) through (5.79f).

$$F_1 = d_1 \frac{\partial U_{2,0}}{\partial \lambda} \quad (5.79a)$$

$$F_2 = -d_2 h \frac{\partial U_{2,0}}{\partial \lambda} + d_3 k \left(q \frac{\partial U_{2,0}}{\partial q} + p \frac{\partial U_{2,0}}{\partial p} \right) + d_4 \frac{\partial U_{2,0}}{\partial k} \quad (5.79b)$$

$$F_3 = -d_2 k \frac{\partial U_{2.0}}{\partial \lambda} - d_3 h \left(q \frac{\partial U_{2.0}}{\partial q} + p \frac{\partial U_{2.0}}{\partial p} \right) - d_4 \frac{\partial U_{2.0}}{\partial k} \quad (5.79c)$$

$$F_4 = -d_3 p \left(\frac{\partial U_{2.0}}{\partial \lambda} - h \frac{\partial U_{2.0}}{\partial k} + k \frac{\partial U_{2.0}}{\partial h} \right) + d_5 \frac{\partial U_{2.0}}{\partial q} \quad (5.79d)$$

$$F_5 = -d_3 q \left(\frac{\partial U_{2.0}}{\partial \lambda} - h \frac{\partial U_{2.0}}{\partial k} + k \frac{\partial U_{2.0}}{\partial h} \right) - d_5 \frac{\partial U_{2.0}}{\partial p} \quad (5.79e)$$

$$F_6 = d_3 \left(q \frac{\partial U_{2.0}}{\partial q} + p \frac{\partial U_{2.0}}{\partial p} \right) + d_2 \left(h \frac{\partial U_{2.0}}{\partial h} + k \frac{\partial U_{2.0}}{\partial k} \right) - d_1 \frac{\partial U_{2.0}}{\partial a} \quad (5.79f)$$

The partials of the F functions have been developed using Macsyma, but considerable simplification is necessary to make these enormous equations useful. The reader can access these partials at:

`$E53-BOB\D:\EVO-HEO\MACSYMA\J22\FILES\FPARTIALS.MFE.`

5.4.4 Macsyma Symbolic Algebra Utilities

The intensity of algebraic manipulation necessary for this problem make symbolic algebra programs a vital tool. Although several symbolic algebra programs exist on the market today, previous work done in this area (Zeis [32]) with Macsyma, coupled with superior technical support, led to the decision to use Macsyma version 2.6.1 for this investigation.

Eric Zeis (and Paul Cefola) used Macsyma block utilities for a 1978 investigation into the construction of nonsingular satellite theories. The utilities he developed, which will be termed the Zeis utilities, are capable of generating the special functions which appear in the expansion of the gravitational potential in terms of equinoctial variables, the

averaged potential, and equations of motion and short-periodics due to an arbitrary harmonic [32]. Specifically, the utilities were designed as an analytical tool to handle the critical inclination problem for small eccentricity orbits. Several of the functions computed by the Zeis utilities are needed in the development of the J_2^2 problem, which necessitated the update of these utilities to the current version of Macsyma, a task primarily done by Paul Cefola. Table 5.3 describes the functionality of the Zeis utilities, while Figure 5.12 provide a flowchart of the block heirarchy.

Table 5.3: Zeis Macsyma Execution Blocks [32]

Name	Function
HANSEN1 [T,N,M,MAXE] (h,k)	Computes the modified Hansen coefficient $Y_T^{N,M}(h,k)$ with an expansion through order MAXE in terms of the eccentricity
HANSEN2 [N,M,MAXE] (h,k)	Computes the modified Hansen coefficient $Y_0^{N,M}(h,k)$ with an expansion through order MAXE in terms of the eccentricity if MAXE ≥ 0 , or exactly if MAXE < 0
JACOBI [N,A,B] (Y)	Computes the Jacobi Polynomial $P_N^{A,B}(Y)$
LEGZ1 [N,R]	Computes the value of the associated Legendre function of degree N and order R at the origin
MDEL [N,M,MAXE]	Computes the short-periodics due to the M-daily potential of dgree N with an expansion through order MAXE in terms of the eccentricity if MAXE ≥ 0 , or exactly if MAXE < 0
NEWCOMB1 [R,S,N,M]	Computes the Newcomb operator $X_{R,S}^{N,M}$
POISSON ($I,J,MAXE$)	Computes the Poisson bracket (I,J) of the equinoctial elements I and J with an expansion through order MAXE in terms of the eccentricity if MAXE ≥ 0 , or exactly if MAXE < 0
RT [N,M,RATIO,MAXE] (a,h,k,p,q,λ)	Computes the averaged resonant potential due to the tesseral harmonic of degree N and order M for an orbit whose period is RATIO days with an expansion through order MAXE in terms of the eccentricity.
RTVOP [N,M,RATIO,MAXE]	Computes the element rates due to the resonant potential of degree N and order M for an orbit whose period is RATIO days with an expansion through order MAXE in terms of the eccentricity.
S1 [N,M,I,T,MAXE]	Computes the generator given by: $S_{NMIT} = \frac{\int^{TL-M\theta} REAL(U_{NMIT}^*) d(TL - M\theta)}{T_n - M\omega_e}$ with an expansion through order MAXE in terms of the eccentricity if MAXE ≥ 0 , or exactly if MAXE < 0

S2 [N,M,I,T,RATIO,MAXE]	Identical to S1, except that it conventionally assigns zero to the terms corresponding to the M-daily effect and the resonance effect for an orbit whose period is RATIO days.
SFUNCT1 [N,M,S] (<i>p,q</i>)	Computes the function $S_{2N}^{M,S}(p,q)$
SLZ [N,MAXE] (<i>a,h,k,p,q,λ</i>)	Computes the averaged zonal potential due to the zonal harmonic of degree N with an expansion through order MAXE in terms of the eccentricity if MAXE ≥ 0, or exactly if MAXE < 0
SLZVOP [N,MAXE]	Computes the element rates due to the averaged zonal potential of degree N with an expansion through order MAXE in terms of the eccentricity if MAXE ≥ 0, or exactly if MAXE < 0
TESDEL [N,M,RATIO,MAXE]	Computes the short-periodics due to the tesseral harmonic of degree and order M for an orbit whose period is RATIO days with an expansion through order MAXE in terms of the eccentricity
USTAR [N,M,I,T,MAXE] (<i>a,h,k,p,q,λ</i>)	Computes the complex potential: $U_{NMIT}^* = C_{NM}^* V_{N,I}^M S_{2N}^{(M,I)}(p,q) Y_T^{-N-1,I}(h,k) e^{j(TL-M\theta)}$ with an expansion through order MAXE in terms of the eccentricity if MAXE ≥ 0, or exactly if T = 0 and MAXE < 0
VFUNCT1 [N,R,M]	Computes the rational number $V_{N,R}^M$
VOPEQ (MAXE)	Computes the orbital element rates due to a disturbing potential R(<i>a,h,k,p,q,λ</i>) with an expansion through order MAXE in terms of the eccentricity if MAXE ≥ 0, or exactly if MAXE < 0
ZSIMP (EXPR)	Writes the EXPR under the form of a Poisson series in the trigonometric variables L and θ
ZDEL [N,MAXE]	Computes the short-periodics due to the zonal harmonic of degree N with an expansion through order MAXE in terms of the eccentricity

Several other blocks were also developed as a result of this investigation. A functionality listing and corresponding flowchart are in Table 5.4 and Figure 5.13 respectively. For a listing of these new blocks, please refer to Appendix E. For a complete listing of all Macsyma utilities currently available at Draper, please reference \$E53-BOB\D:\EVO-HEO\MACSYMA.

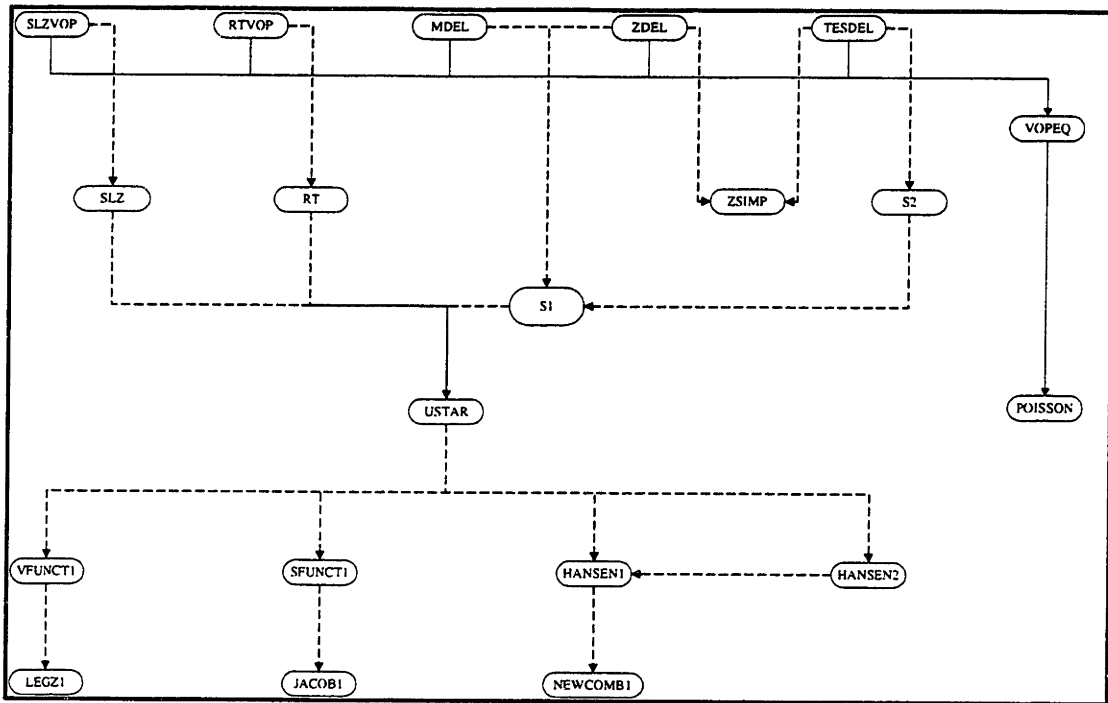


Figure 5.12: Zeis Macsyma Blocks Flow Diagram [32]

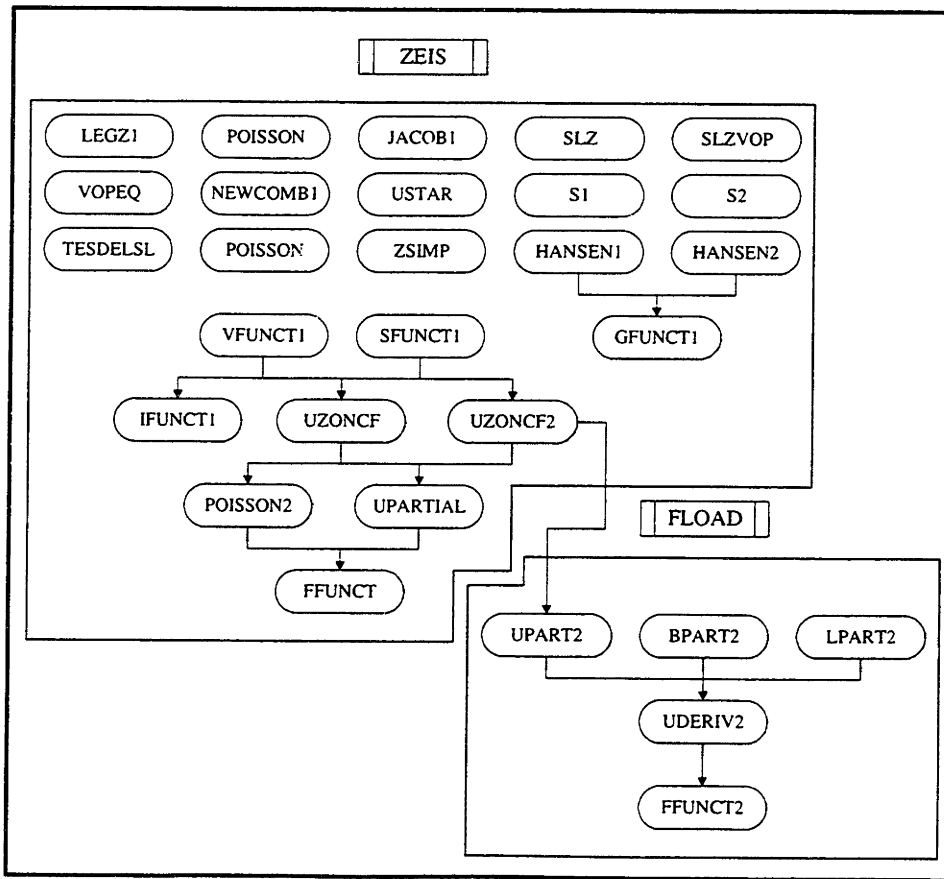


Figure 5.13: New Macsyma Blocks Flow Diagram

Table 5.4: New Macsyma Execution Blocks

Name	Function
BPART2(I)	Computes the partial derivatives of (a/r) with respect to the equinoctial element (a, h, k, p, q, λ) determined by I
FFUNCT(I)	Computes the F function, of equinoctial variable I, used in calculating the second order contributions of J2 by summing the product of the Poisson brackets and simplified symbolic partial derivatives of U.
FFUNCT2(I)	Computes the F function, of equinoctial variable I, used in calculating the second order contributions of J2 by using the d1-d6 functions defined in (5.78) to simplify
FLOAD	Loads necessary execution blocks into Macsyma for use with the development of the F functions.
F[L,M,P](I)	Computes the explicit inclination functions defined by Kaula (IFUNCT1)
G [L,P,Q,MAXE](E)	Computes the explicit eccentricity functions defined by Kaula (GFUNCT1)
LPART2(I)	Computes the partial derivatives of the True Longitude (L), with respect to the equinoctial element (a, h, k, p, q, λ) determined by I
POISSON2(I,J,MAXE)	Updated version of POISSON which accounts for new notation
UDERIV2(I)	<p>Computes the explicit derivatives of the gravitational potential for the zonal harmonic J2 with respect to the equinoctial element (a, h, k, p, q, λ) determined by I, using separation of variables in the following form:</p> $\frac{dU_{2,0}}{da_i} = \frac{\partial U_{2,0}}{\partial a_i} + \frac{\partial U_{2,0}}{\partial L} \frac{\partial L}{\partial a_i} + \frac{\partial U_{2,0}}{\partial \left(\frac{a}{r}\right)} \frac{\partial \left(\frac{a}{r}\right)}{\partial a_i}$
UPART2(I)	Computes the explicit partial derivatives of the gravitational potential for the zonal harmonic J2 with respect to the equinoctial element (a, h, k, p, q, λ), determined by I. If I = 7, returns the partial with respect to L and if I = 9, the partial with respect to (a/r).
UPARTIAL(I)	Computes simplified symbolic partial derivatives of the gravitational potential for the zonal harmonic J2 with respect to the equinoctial element (a, h, k, p, q, λ), determined by I.
UZONCF[n](a,p,q,B,L)	Computes the closed form representation of the Real portion of the gravitational potential expanded in terms of the n th zonal harmonic, using the form given in (5.20), and determining the derivative of B
UZONCF2(a,p,q,B,L)	Computes the closed form representation of the Real portion of the gravitational potential expanded in terms of the second zonal harmonic, using the form given in (5.20)
ZEIS	Loads all updated Zeis execution blocks into Macsyma for use

[This page intentionally left blank.]

Chapter 6

Decay Phase Analysis

6.1 Introduction

As an orbital object enters the decay phase, atmospheric effects become the dominant perturbations to two body motion. Unfortunately, the stochastic nature of these disturbances make modeling very difficult and introduce significant error into the orbit prediction process. Within this broad category of perturbations, some of the primary sources for error include:

- Atmospheric drag and Density modeling
- Atmospheric lift
- Atmospheric winds
- Speed ratio effect
- Transition region effect
- Structural Failure and Reentry heating
- Hypersonic flow considerations

Error sources specific to semianalytical theory include:

- Higher order drag and oblateness coupling
- Short periodics with drag

Rather than attempting to model all of these perturbative forces, Chapter 6 will outline a brief analysis on each in order to characterize the corresponding error signatures and gain a better understanding of the Molniya decay profile. The reader should note that while most analyses are applicable to any catastrophic decay cases, the analyses are done primarily for the NSSC-13390 rocket body.

6.2 Atmospheric Drag and Density Modeling

As the rocket body decays, drag quickly becomes the dominant perturbation on the orbital path. Acting in a direction opposite to the relative velocity vector, atmospheric drag robs the orbit of energy. The more energy drag takes away, the smaller the orbit becomes, exposing the satellite to more of the atmosphere and amplifying the effects of drag (Figure 1.8). Calculating the drag force on a satellite is directly related to the determination of the atmospheric density, which is in turn related to the altitude of the object. As the altitude increases, the average distance a molecule moves between successive collisions (mean free path) also increases, which creates three primary aerodynamic regimes: free molecular flow, continuum flow and the transition regime, each of which interact differently with the satellite.

Free molecular flow is simply defined as that portion of the atmosphere in which the mean free path is greater than the characteristic dimension of the spacecraft. Continuum flow is the portion of the atmosphere below the point where the mean free path is equal to the satellite's dimensions, and the transition regime is simply the imprecise boundary of the joining of free molecular and continuum flow. Before

analyzing the effect of density errors on the force of drag in free molecular flow, we must first understand some basic principles of this regime.

Understanding the interactions of the gas molecules of the atmosphere with the satellite surface is pivotal in predicting the amount of force the molecules impart on the satellite. In order to model these interactions, Gerald Karr developed the Gas Surface Interaction (GSI) model [104]. Figure 6.1 shows a general non-specular type of reflection where the reflected molecules leave the surface in the direction θ_j with an average velocity of U_j .

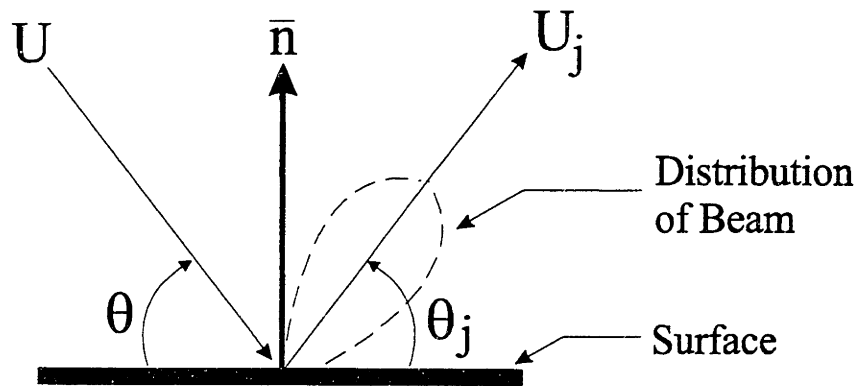


Figure 6.1: Gas Surface Interaction Model

θ_j and U_j are related to each other with Equations (6.1) and (6.2) where α_j is the thermal accommodation parameter and P_j is the reflection type parameter for the interaction.

$$\theta_j = \frac{\pi}{2} P_j + (1 - P_j)\theta \quad (6.1)$$

$$U_j = \sqrt{1 - \alpha_j} U \quad (6.2)$$

In Karr's experiments, the GSI parameters are determined through the reduction of data, but with no attitude data for the rocket body, values for the GSI parameters must be assumed using what is known of their values. The reflection type parameter, P_j , represents the type of reflection of the molecules dependent on the type of surface on the

satellite, where a value of zero represents a perfect specular reflection (smooth surface), a value of one represents a diffusive reflection (rough surface - random), and a value of two represents perfect back-scatter reflections (other forces deflect particles more than 90 degrees to the initial direction of travel). α_j , the thermal accommodation parameter, describes the thermal characteristics of the material and must also be determined from experimental data. Due to the concave skirt on the rocket body, typically resistant thermal properties of rockets bodies, and discussions with Dr. Karr, the following values were chosen for this analysis: $\alpha_j = 0.75 \pm 0.10$ and $P_j = 0.90 \pm 0.10$. With estimates for the GSI parameters, it was possible to determine the drag force acting upon the rocket body using these simplified forms of Karr's equations [105]:

$$Drag = \frac{1}{2} \rho U^2 C_D \bar{A} \quad (6.3)$$

$$C_{D_{cylinder}} = 2 + 2\sqrt{1-\alpha_j} \left(\frac{\cos\left(\frac{\pi}{2}\right) P_j}{(1-P_j)(3-P_j)} \right) \quad (6.4)$$

$$C_{D_{sphere}} = 2 + \sqrt{1-\alpha_j} \left[\frac{4 \left(1 - \cos\left(\frac{\pi}{2}\right) P_j \right)}{P_j (4 - P_j)} \right] \quad (6.5)$$

Both the sphere and a cylinder equations were used analyze the effect of GSI parameter and density variation. For reasons discussed in the atmospheric lift section, however, the rocket body will be modeled as a sphere. In order to determine the maximum error attributable to these variations, a NORAD element set from the "mission phase" of the rocket body, mean atmospheric density, and the spherical face area and cylindrical face area were first obtained. The analysis simply involved comparing variant bounds of error in the GSI parameters and 3% assumed density error. Under such conditions, and with the pre-determined values of the parameters, the resultant error in

drag force due to only the GSI parameter error and density error was found to be approximately 6% for the free molecular flow region.

Obviously, this estimate increases with increases in the assumed density error, and is accordingly sensitive to the atmospheric model employed. As such, it was extremely important to make use of a sufficiently developed atmosphere model for this investigation. The result, was the integration of the MSISE-90 atmospheric model into NT-GTDS, as described below.

6.2.1 MSISE-90 Atmospheric Model

In order to account for the drag perturbation, it is necessary to accurately model the density of the earth's atmosphere. Currently, GTDS contains the Harris-Priester and Jacchia-Roberts atmospheric density models that, while still effective for Low Earth Orbit (LEO) satellites, are ineffective for altitudes below 90 km. For a density model to be effective for modeling a satellite through decay, it must be capable of accurately predicting density between 60 and 90 km. Satellite drag based models, such as the Jacchia 77 model in GTDS, and models based on satellite mass spectrometer and ground-based incoherent scatter data (e.g. MSIS-86), have historically had a mesopause altitude limit due to inaccuracies in measurements in the lower atmosphere. MSISE-90 uses analytic models (rather than tabular format) to model the lower altitudes to account for disturbances such as solar activity, magnetic storms, and daily variations, as well as latitude, longitude, and monthly variations [131].

Of the high drag density models available for incorporation into GTDS, the MSISE-90 offered the most accuracy for the least amount of coding effort. The MSISE-90 model, developed by A. E. Hedin at NASA Goddard, models the atmosphere from 0 to 2000 km with an overall standard deviation of 3K for temperature and 2% for pressure

[131]. While offering this amount of accuracy, the model is relatively simple and more readily assimilated into the GTDS coding structure than the alternatives. Unfortunately these errors, magnified by other un-modeled perturbations and random variations, can create sizable inaccuracies in orbit prediction. In addition to density modeling errors, other atmospheric effects such as those examined below, and random variability caused by solar activity, magnetic storms, and other cosmic anomalies, can create drag estimate errors in excess of 15%. Although this estimate lumps all atmospheric and mis-modeled errors as drag estimation error, a recent paper by Chastain and Chao [106] identified density overestimation errors as high as 22% for small spheres in circular decays, which are typically less volatile than the catastrophic decay cases examined in this paper. In addition to these errors, future predictions must use predicted values of the solar flux and geomagnetic indices which fail to account for daily variations and produce more error.

One alternative was the Global Reference Atmosphere Model (GRAM) which offers impressive characteristics such as a “fairing” technique used to insure a smooth transition between 90 and 115 km [132]. Unfortunately, the GRAM model is prohibitively complex for this initial study and, as a result, the MSISE-90 model was chosen. It is important to keep in mind that there is very little data available for the behavior of objects in the 60-90 km region. As part of the development of the GTDS MSISE-90 model, the actual data available for this paper will be compared with predicted values in this region and may serve to enhance the model itself.

6.2.2 GTDS Integration

The MSISE-90 model has been coded in Fortran 77 and is publicly available on the Internet [108]. This code has been retrieved from the NASA anonymous FTP site to be incorporated into GTDS. Figure 6.2 shows the relationship of the various MSISE-90

subroutines and functions and Table 6.1 describes each of the MSISE-90 subroutines and what they do.

Table 6.1: MSISE-90 File Descriptions

<i>Name</i>	<i>Description</i>
GTD6	Main driver program for MSISE-90 software
GTD6BK	Block data corresponding to current version of the MSIS model
TSELEC	Initializes the control switch arrays for main terms (SW) and cross terms (SWC)
VTST	Function which tests to see if the geophysical variable and switches have been changed, then saves the values
GLATF	Function which calculates the latitude variable gravity and effective radius
GTS6	Neutral thermosphere model above 72.5 km
DENSM	Function which calculates temperature and density profiles for lower atmosphere
GLOBE6	Function to calculate upper thermosphere parameters
SPLINE	Calculates 2 nd derivative of cubic spline interpolation function
SPLINI	Integrates cubic spline over specified interval
SPLINT	Calculates cubic spline interpolation value
GLOB6S	Version of GLOBE6 for lower atmosphere
DENSU	Function which calculates temperature and density profiles for MSIS models using a new lower thermosphere model
DNET	Function for turbopause correction for MSIS models
CCOR	Function for chemistry/dissociation correction for MSIS models

Time has not been devoted to the task of understanding the theory of the MSISE-90 model. Instead, the MSISE-90 code has been treated as a black box. This allows a great deal of flexibility and modularity for future expansion of MSIS models within GTDS. As models improve, block-data, subroutines, or functions can simply be replaced with improved versions with a minimum of coding effort. The spare room in the MSIS data files also promotes this modularity by allowing for expansion if additional input variables are needed for future versions of the driver program.

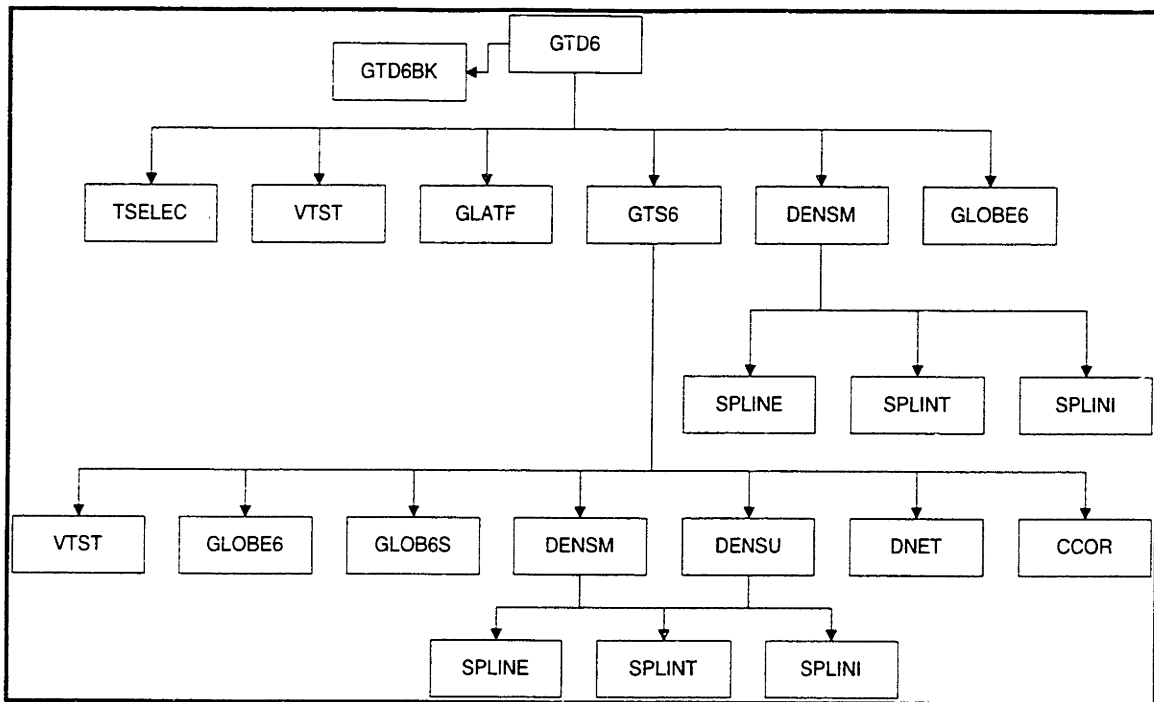


Figure 6.2: MSISE-90 Coding Structure

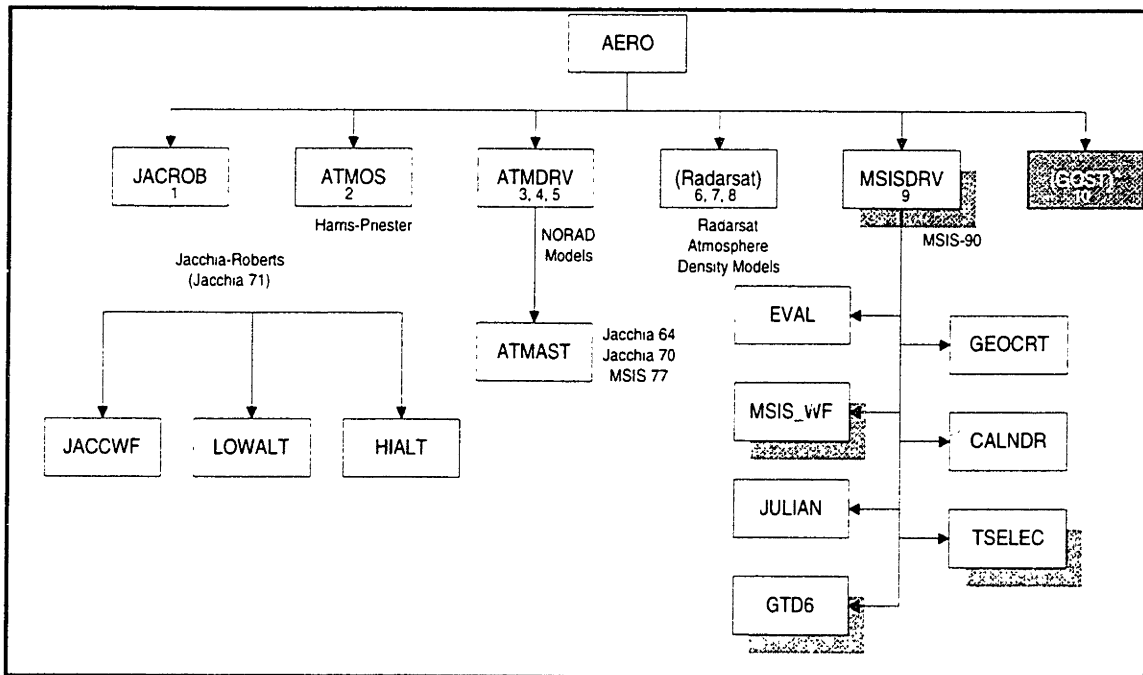


Figure 6.3: AERO Structure Diagram

Figure 6.3 shows the modular structure of the AERO subroutine which is responsible for using, among other things, the specified atmospheric model to calculate

the vector force on a spacecraft due to aerodynamic forces. Models are assigned a model number starting with Jacchia-Roberts 71 (model one) and continuing on with no theoretical limit. New models such as the Russian GOST [130] atmospheric model can simply be added with an interface routine analogous to MSISDRV which passes the necessary inputs into the model and retrieves the current value of atmospheric density, ρ .

MSISE-90 was incorporated by:

- Creating MSISDRV to pass the proper input parameters into GTD6,
- Creating MSIS_WF to read the necessary A_p and flux data from the binary file
- Including several routines from the VAX Radarsat version of GTDS to calculate geodetic latitude and longitude
- Placing the TSELEC subroutine outside the MSISE code
- Minor modifications to GTDS initialization routines.

The result has been a functional atmospheric model valid from 2000 km to the Earth's surface.

In order for the MSISE-90 model to function properly once integrated, it requires the inputs listed in Table 6.2. Please note that the 25 element array SW is a control switching array for the various options in the MSIS model. The array allows the user to turn on, off, or couple terms only for any of 23 different options (please refer to code for listing of options). Of particular interest is SW(9) which controls the A_p array option listed in Table 6.2. If SW(9) has a value of negative one (-1), the program will use the seven element array defined in Table 6.2 for its calculations. Otherwise (value of one or two), the program will use the A_p daily value. With the proper inputs, the model then calculates the number densities of He, O, N₂, O₂, Ar, H, and N, as well as the total mass

density, exospheric temperature, and temperature at altitude for the given input parameters.

Table 6.2: MSISE-90 Input Parameters

<i>Variable</i>	<i>Index</i>	<i>Description</i>
IYD		Year and day of input time as YYDDD
SEC		Seconds from UT midnight of the corresponding day
ALT		Altitude above Earth reference ellipsoid (km)
GLAT		Geodetic latitude (degrees)
GLON		Geodetic longitude (degrees)
STL		Local apparent solar time (hours).
F107A		3 month, 81 day average of F _{10.7} flux centered on day 41
F107		Daily F _{10.7} flux for previous day
AP		Daily magnetic index (default)
AP		Array containing: (SW(9) = -1)
	1	Daily Ap
	2	3 hour Ap index for current time
	3	3 hour Ap index for 3 hours before current time
	4	3 hour Ap index for 6 hours before current time
	5	3 hour Ap index for 9 hours before current time
	6	Average of eight 3 hour Ap indices from 12 to 33 hours prior to current time
	7	Average of eight 3 hour Ap indices from 36 to 57 hours prior to current time
MASS		Mass number (only density for selected gas is calculated. Mass = 0 for temperature only, Mass = 48 for all densities)

In order to most efficiently input the above parameters into the MSISE-90 program, a binary file structure was adopted to store the necessary atmospheric information (F_{10.7} and A_p information). The data itself came from a historical archive of actual atmospheric environment readings found on the National Geophysical Data Center web site [109], and combined with predictions of F_{10.7} and A_p values obtained from Dr. Kenneth Schatten [110]. The data was manipulated with Fortran routines developed from David Carter's (CSDL) original VAX utilities for creating binary atmospheric files. The new routines, however, take both actual and predicted values and combine them into one file using a smoothing, weighted interpolation process. Table 6.3 lists the temporary data

data file format used in the development process, while Figure 6.4 illustrates the flow of the process from input data files to the resulting binary files.

Table 6.3: MSISE-90 Temporary Data File Format

<i>Record</i>	<i>Column</i>	<i>Variable</i>	<i>Description</i>
1	001-002	ID	Record identifier
	003-008	IT	Modified Julian date of the first day in the record
	009-428	KP(21,4)	21 days of packed 3-hour geomagnetic indices, starting with IT-1
	429-548	TC(20)	20 days of night-time minimum global exospheric temperatures, starting with day IT-1
	549-674	F107A(21)	21 days of the 81 day average F10.7 centered on the required day, starting with day IT-1
	675-800	F107(21)	21 days of the daily F10.7, starting with day IT-1
2	001-002	ID	Record identifier
	003-086	APD(21)	21 days of the daily Ap, starting with day IT-1
	087-822	APH(23,8)	23 days of the 3-hour geomagnetic indices, starting with day IT-3
3	001-002	ID	Record identifier
	003-008	IDREC1	Modified Julian date of the first day of available data in the file
	009-014	IDRECE	Modified Julian date of the last day of available data in the file
	015-020	NREC	Number of records in the file
	021-026	LGD	Modified Julian date of the last good data point (last day of extrapolated data)

PROC_SCH.M is a Matlab script constructed by David Carter (CSDL) which interpolates the monthly Schatten predict values using a cubic spline and produces the corresponding data files for the variant bounds of solar activity (hot, nominal, cold) and cycle mis-modeling (early, nominal, late). These files, and the actuals, are fed into JRMSISSP.FOR and JRMSISRL.FOR respectively to create temporary data files of the format described in Table 6.3. As text files, they can be transferred to any operating system before the JRMSISIN process converts them into the usable binary files. This

process is run for each variant of the predicts (9) before sending the real and predicted temporary data files into JRMSISIN.FOR.

Table 6.4: MSISE-90 Binary Data File Format

<i>Record</i>	<i>Size</i>	<i>Bytes</i>	<i>Variable</i>	<i>Description</i>
1	2048	0001-0004	IDREC1	Modified Julian date of the first day of available data in the file
		0005-0008	IDRECE	Modified Julian date of the last day of available data in the file
		0009-0012	NREC	Number of records in the file
		0013-0016	LGD	Modified Julian date of the last good data point (last day of extrapolated data)
		0017-2048	spare	
2-NREC	2048	0001-0004	IT	Modified Julian date of the first day in the record
		0005-0172	KP(21,4)	21 days of packed 3-hour geomagnetic indices, starting with IT-1
		0173-0252	TC(20)	20 days of night-time minimum global exospheric temperatures, starting with day IT-1
		0253-0336	F107A(21)	21 days of the 81 day average F10.7 centered on the required day, starting with day IT-1
		0337-0420	F107(21)	21 days of the daily F10.7, starting with day IT-1
		0421-0504	APD(21)	21 days of the daily Ap, starting with day IT-1
		0505-1240	APH(23,8)	23 days of the 3-hour geomagnetic indices, starting with day IT-3
		1241-2048	spare	

The reader should note that the data files include both Kp and exospheric temperature values in order to maintain the same data structure found in the Jacchia-Roberts binary files. This inclusion necessitated the use of the BARTELS.FOR subroutine which converts Ap to Kp. Making the MSISE and Jacchia files similar allows the user to run permanent file reports (PFR) for different binary files and compare the results to Jacchia files. The COMP*.dat files allow the user to compare the Jacchia PFR to the MSISE file without even running the PFR for the MSISE file. The output, the

MS9*.DAT files, are used as GTDS file \$076 with the MSIS model. The format of these files are given in Table 6.4. The aforementioned similarity with Jacchia-Roberts files is apparent in that the first 252 bytes are the exact same as Jacchia-Roberts binary data files.

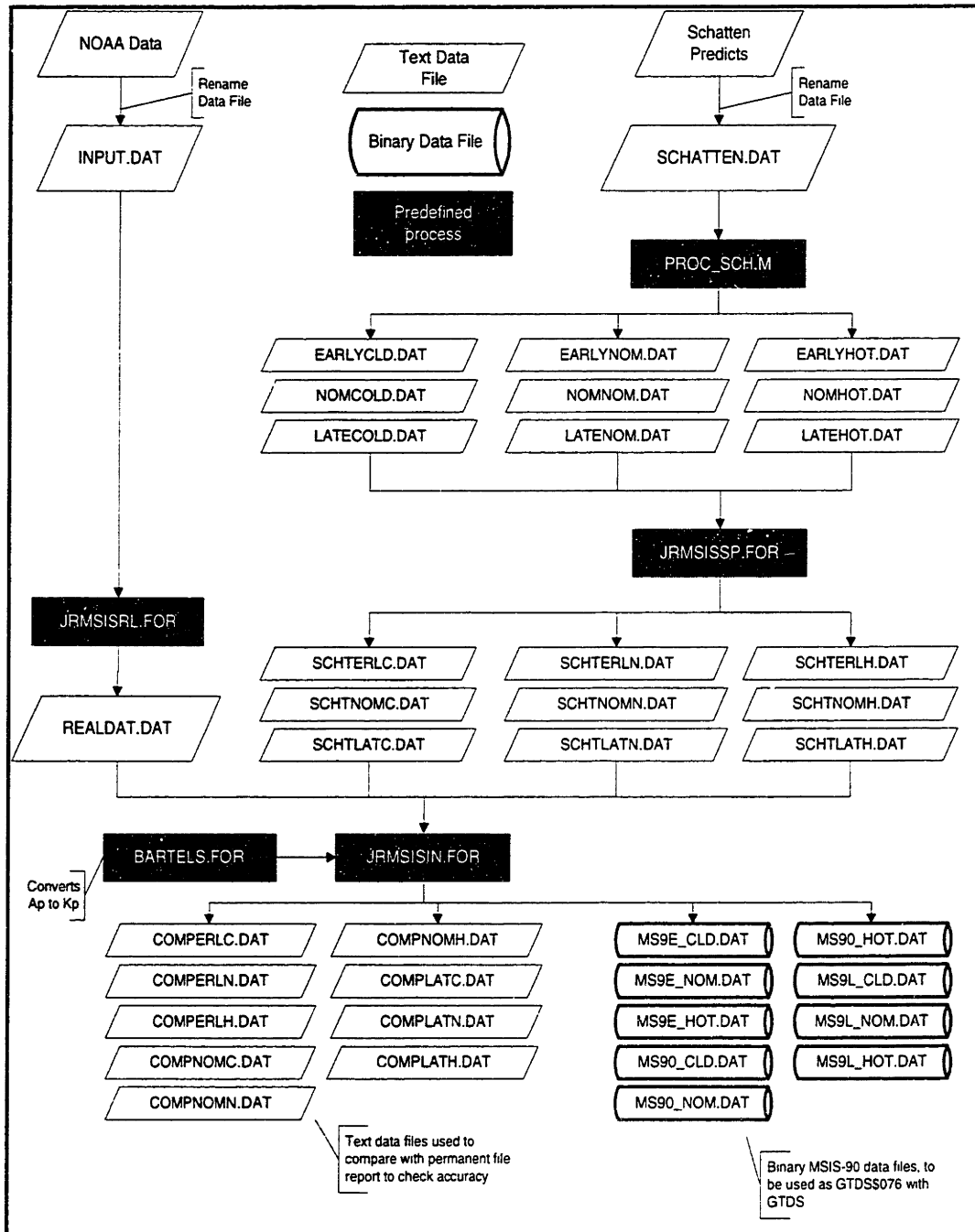


Figure 6.4: MSISE-90 Binary Data File Flowchart

The reader should also note the inherent variability associated with the real data used in these files, especially the geomagnetic indices (A_p). In order to combine these values with Carter's smooth, cubic spline interpolations for A_p from Dr. Schatten's data, a weighted averaging process which dampens the oscillations over an 80 day period was utilized. Figure 6.5 depicts this merging technique and demonstrates how it was possible to get data files spanning 1980 to 2008.

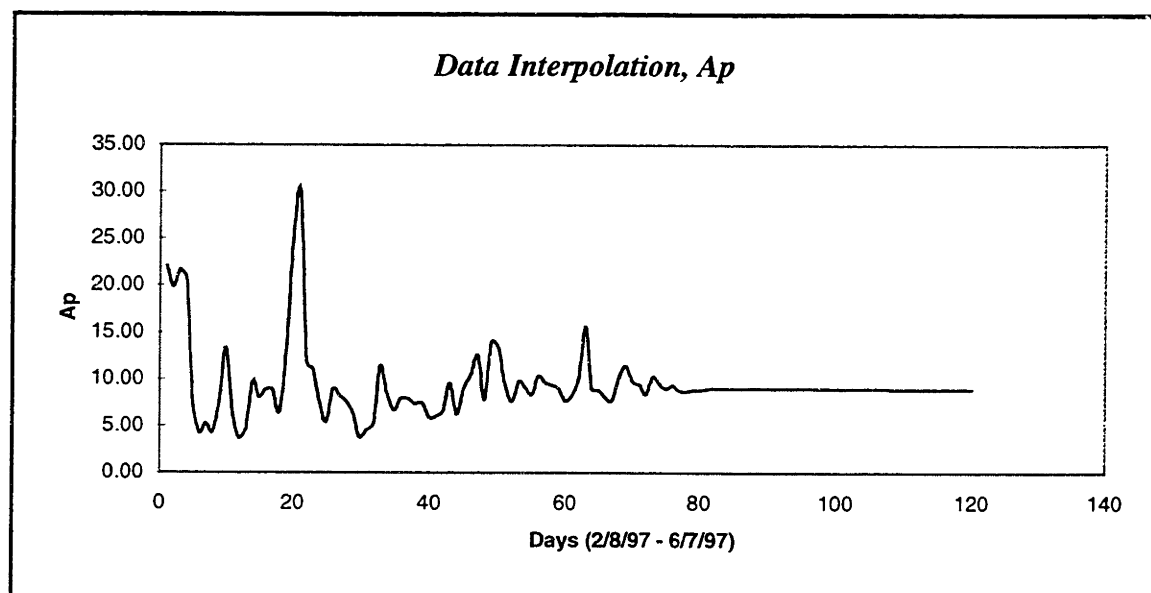


Figure 6.5: A_p Data Merging Technique

6.2.3 Testing and Validation

In order to validate the new MSISE-90 model within GTDS, several tests were performed and achieved satisfactory results. The first test was to use the supplied test driver program from the NOAA FTP site to confirm that the MSISE-90 code was working properly. The code produced identical results to the given control solutions and thus verified that the FTP process was successful.

The next test was to run the same test cases from within the GTDS structure. This was accomplished by modifying MSISDRV to hardwire the input values to GTD6 in

order to compare the output densities and temperatures with the control solutions. This test produced identical results as well.

The code was then compared to the Jacchia-Roberts 71 model (already operational in GTDS) using a testing run from a previous analysis done at Draper on an Ellipso™ satellite. The testing run involved a three month DC fit from a Jacchia produced PCE file, then used the converged DC element set to run a three month ephemeris prediction with MSISE-90 ($SW(9) = -1$). Figure 6.6 shows the differences in semi-major axis which has an average error of 1.821m, while Figure 6.7 simply shows the semi-major axis element history over the fit span. Albeit, over a long arc which is experiencing higher drag, the differences will become more sizable than those presented here due to the compounding differences in mean anomaly, this test was meant simply to see that the two models were comparable and that MSISE-90 was performing satisfactorily.

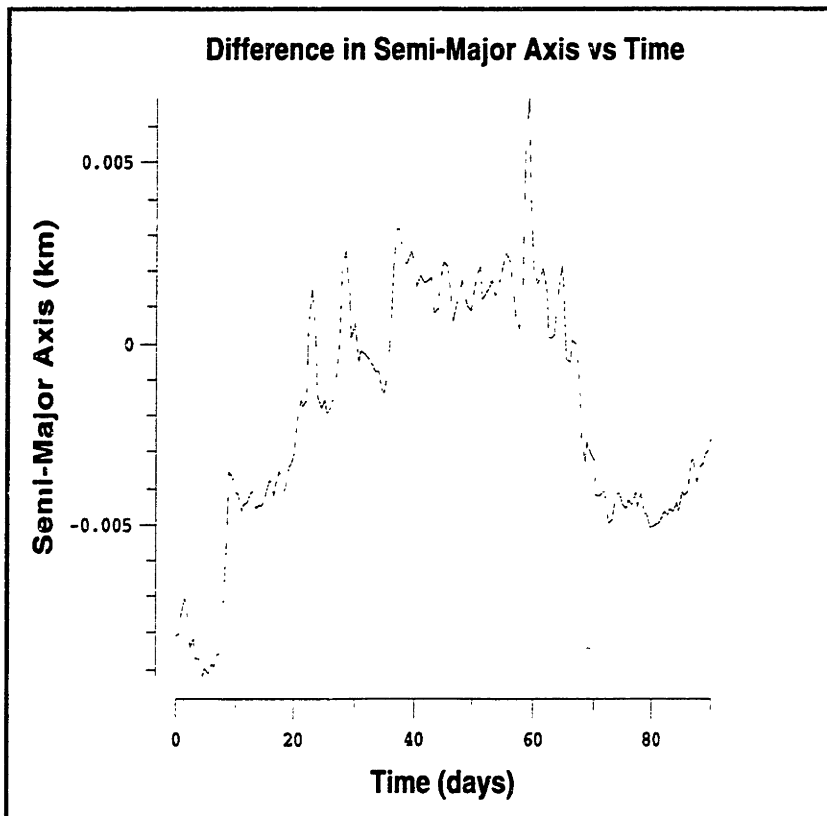


Figure 6.6: AtmosphericModel Semi-Major Axis Difference Comparison

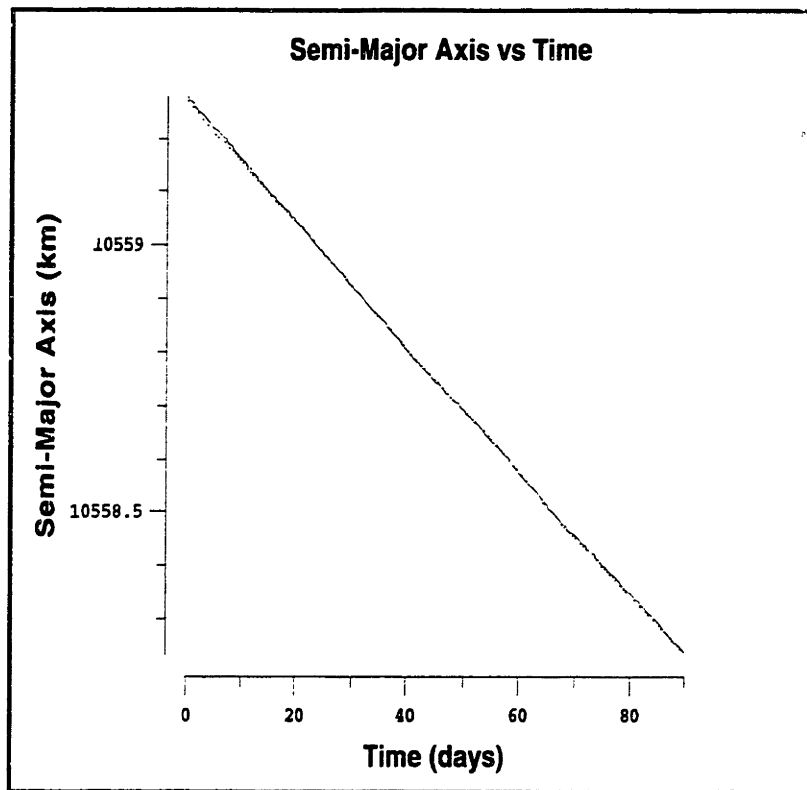


Figure 6.7: Semi-Major Axis Element History

A comparison run was also done on a long arc comparison test between Jacchia 71 and the MSISE-90 model. Using the aforementioned Ellipso™ test case, two ephemeris generation runs were conducted over a two year span. The only difference between the runs was the atmospheric model, and the magnitude of the errors was commensurate with those demonstrated by Hedin in his comparison of MSIS-86 (the basis for MSISE-90 upper atmospheric calculations - where this test case was) and Jacchia 71 [111].

In addition to these tests, the MSIS model has been successfully run with both the Semianalytic and Cowell integrators doing both DC and ephemeris generation runs. Given the results of these tests, the authors consider the MSISE-90 model operational in GTDS. Further testing will be conducted to insure that all components of GTDS work effectively with the model, but that will be an ongoing process, and not presented here.

6.3 Atmospheric Lift Modeling

For the most part, satellites entering the lower atmosphere are not functioning, and have no stabilization to maintain an orientation. As such, they randomly tumble and the net effect of lift will not be a factor. Non-spinning “dead” satellites, such as the rocket body being studied, were initially thought to fall into this category. With no attitude control and a concave rocket skirt (see Figure 1.12), the rocket body was thought to have no predictable orientation and net lift effect that would average out over the random angle of attack. A simplified study of a flat plate in continuum hypersonic flow demonstrates this concept. The following equations, developed in Anderson [51] using the modified Newtonian law, represent the coefficients of drag and lift for a flat plate:

$$L = N \cos \alpha = \rho_{\infty} V_{\infty}^2 A \sin^2 \alpha \cos \alpha \quad (6.6)$$

$$D = N \sin \alpha = \rho_{\infty} V_{\infty}^2 A \sin^3 \alpha \quad (6.7)$$

$$C_L = \frac{L}{\frac{1}{2} \rho_{\infty} V_{\infty}^2 A} = 2 \sin^2 \alpha \cos \alpha \quad (6.8)$$

$$C_D = \frac{D}{\frac{1}{2} \rho_{\infty} V_{\infty}^2 A} = 2 \sin^3 \alpha \quad (6.9)$$

When graphed as a function of the angle of attack, α , the reader should note in Figure 6.8 the net effect of lift is averaged to zero over the domain of α .

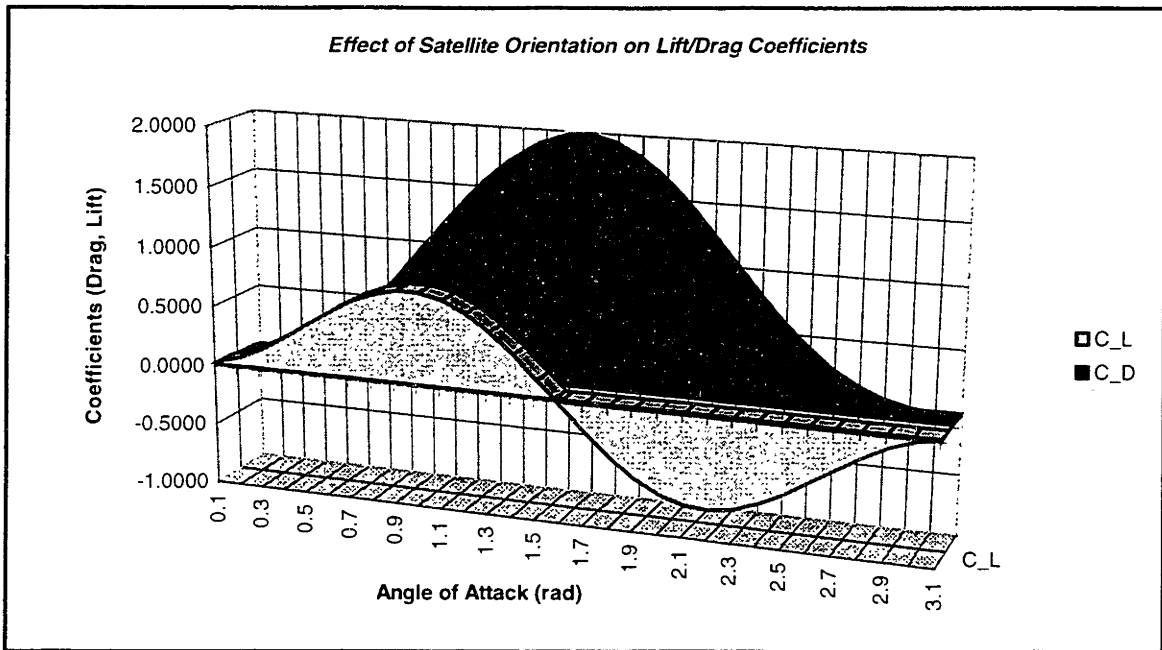


Figure 6.8: Effect of Satellite Orientation on Lift/Drag Coefficients

Analyzing atmospheric lift for Molniya orbits may seem trivial at this point, but the final results have proven otherwise. This investigation has shown that atmospheric lift does in fact play a part in modeling catastrophic decay objects. As such, lift averaging to zero is not an entirely valid assumption for the Molniya rocket bodies studied as part of this analysis. Figure 1.9 is a technical drawing of the Molniya spacecraft, and shows the increased complexity of the satellite structure which may contribute even more to the lift perturbation. The reasoning for the lift error identification can be found in Chapter 7, but suffice to say that in order to more successfully model catastrophic decays, it is necessary to develop some form of a lift model.

If attitude orientation can be assumed, or if attitude data becomes available, the process of finding the coefficients of drag and lift simply becomes a process of adding Karr's [104] free molecular flow equations for spheres, disks, cylinders, and cones.

The reader should note that satellite orientation can create a major source of error due to the effects of lift and drag. However with essentially dead objects, having no attitude history to analyze, another method had to be employed. The following section outlines the simplified lift model developed for use with NT-GTDS. The Macsyma symbolic algebra program was used to check all partial derivatives in Section 6.3.1, and Mathcad for Windows 6.0 was used to check numerical accuracy of the corresponding NT-GTDS generated values.

6.3.1 Simplified Lift Model

Initial testing indicated that atmospheric lift may play a large role in the final reentry of catastrophic decay cases. Unfortunately, without attitude data, determination of the impact of lift on the object's orbital motion became arbitrary at best. It therefore became necessary to develop a simplified lift model, which could function without attitude data and account for the variations due to atmospheric lift. A stochastic model would be ideal for this situation, but given that this model was designed and implemented in a little over one month, such a complex form was not feasible. Whereas the model chosen is physically unrealistic by assuming lift force for a sphere (although Karr has shown lift for spinning spheres [104]), it affords NT-GTDS PR6 with the dynamic ability to solve for accelerations in two lift directions, and therefore correct for the variations due to atmospheric lift. The following discussion outlines the coordinate system, model description, and corresponding mathematical derivation of the Simplified Lift Model (SLM) employed in NT-GTDS PR6.

Figure 6.9 represents the right-handed coordinate system chosen for the SLM. T is in the direction of the relative velocity vector (V_{rel}), defined in Equation (6.10) where ω

represents the Earth rotation vector, \mathbf{R} represents the object position vector, and \mathbf{V}_{rel} is defined as the velocity of the object relative to the atmosphere. This velocity is determined in the inertial reference frame by subtracting the motion of a rotating atmosphere [4]. The N and W vectors are then described by the cross products shown in Equations (6.11) and (6.12). The N and W unit vectors are shown in Equations (6.13) and (6.14).

$$\mathbf{V}_{rel} = \dot{\mathbf{R}} - (\boldsymbol{\omega} \times \mathbf{R}) \quad (6.10)$$

$$\mathbf{W} = \mathbf{R} \times \mathbf{V}_{rel} \quad (6.11)$$

$$\hat{\mathbf{W}} = \frac{\mathbf{W}}{|\mathbf{W}|} = \frac{\mathbf{R} \times \mathbf{V}_{rel}}{|\mathbf{R} \times \mathbf{V}_{rel}|} \quad (6.12)$$

$$\mathbf{N} = \mathbf{V}_{rel} \times \mathbf{W} \quad (6.13)$$

$$\hat{\mathbf{N}} = \frac{\mathbf{N}}{|\mathbf{N}|} = \frac{\mathbf{V}_{rel} \times \mathbf{W}}{|\mathbf{V}_{rel} \times \mathbf{W}|} \quad (6.14)$$

The model then makes the physically unrealistic assumption that lift in the N and W directions can be modeled for a spherical object using the form of the drag acceleration for a sphere given in Equations (6.15a) and (6.15b) as a standard form. Using the new coefficients for lift in the N and W directions, defined as C_{L-N} and C_{L-W} , and defining the accelerations as positive in the direction of the N and W unit vectors respectively, the accelerations due to lift in those directions are then defined by Equations (6.16) and (6.17).

$$\ddot{\mathbf{R}}_D = -S_S \rho \mathbf{V}_{rel} |\mathbf{V}_{rel}| \quad (6.15a)$$

$$S_S = \frac{1}{2} C_D \left(\frac{A}{m} \right) = \frac{1}{2} \left(\frac{\pi d^2}{4m} \right) \quad (6.15b)$$

$$\ddot{\mathbf{R}}_{L-N} = S_{L-N} \rho |\mathbf{V}_{rel}|^2 \hat{\mathbf{N}} \quad (6.16a)$$

$$S_{L-N} = \frac{1}{2} C_{L-N} \left(\frac{A}{m} \right) \quad (6.16b)$$

$$\ddot{\mathbf{R}}_{L-W} = S_{L-W} \rho |\mathbf{V}_{rel}|^2 \hat{\mathbf{W}} \quad (6.17a)$$

$$S_{L-W} = \frac{1}{2} C_{L-W} \left(\frac{A}{m} \right) \quad (6.17b)$$

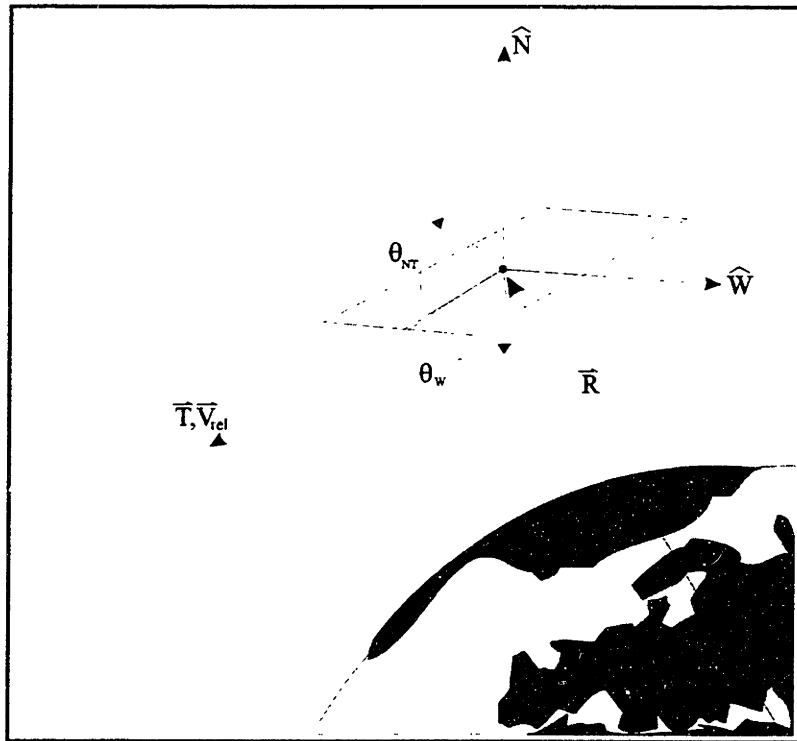


Figure 6.9: The SLM Coordinate System

6.3.1.1 Drag Model Partial Derivatives

In order to integrate lift forces into the NT-GTDS aerodynamic force model, it is necessary to compute the partial derivatives of the lift acceleration with respect to variations in the spacecraft local inertial state (position and velocity) and dynamic

parameters (coefficients of drag and lift) for use with the variational equations [4]. As an overview of the necessary partials needed for the SLM, a brief derivation of the partials for the drag model is presented below.

Before computing the partials of the acceleration due to drag, it is convenient to express the relative velocity in a different form, namely that given in Equation (6.18), where Ω is a rotation matrix defined in Equation (6.19).

$$\mathbf{V}_{rel} = \dot{\mathbf{R}} - \Omega \mathbf{R} \quad (6.18)$$

$$\Omega = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (6.19)$$

Given these equations, and in light of simplified dependencies, the partials of (6.15a) with respect to position and velocity are simplified to the forms given in (6.20) and (6.21). The necessary partial of the acceleration with respect to the relative velocity is simply attained, and given in (6.22).

$$\frac{\partial \ddot{\mathbf{R}}_D}{\partial \dot{\mathbf{R}}} = \frac{\partial \ddot{\mathbf{R}}_D}{\partial \mathbf{V}_{rel}} \quad (6.20)$$

$$\frac{\partial \ddot{\mathbf{R}}_D}{\partial \mathbf{R}} = \left(\frac{\partial \ddot{\mathbf{R}}_D}{\partial \mathbf{V}_{rel}} \right) \Omega \quad (6.21)$$

$$\frac{\partial \ddot{\mathbf{R}}_D}{\partial \mathbf{V}_{rel}} = -S_{s\rho} \left(\frac{\mathbf{V}_{rel} \mathbf{V}_{rel}^T}{|\mathbf{V}_{rel}|} + |\mathbf{V}_{rel}| \mathbf{I} \right) \quad (6.22)$$

The reader should note that because density is not dependent on velocity (relatively speaking), (6.22) is sufficient to describe the velocity partial. For position, however, of which density is dependent, (6.21) represents only one part of position partial. Equation (6.21) only accounts for Earth rotation with respect to the relative

velocity vector, and another term is necessary to account for density variation, which is given in Equation (6.23). The partial is separable because density depends only upon the spacecraft local position and not upon the local velocity.

$$\frac{\partial \ddot{\mathbf{R}}_D}{\partial \mathbf{R}} = \frac{\ddot{\mathbf{R}}_D}{\rho} \frac{\partial \rho}{\partial \mathbf{R}} \quad (6.23)$$

Obviously, the first term is trivial, but the partial of density with respect to position is not. The density partial is a complex analytical partial derivative which is completely dependent on the atmospheric model. Currently, GTDS calculates the partials for the Jacchia-Roberts and Harris-Priester models by separating variables and computing the partial of density with respect to altitude and multiplying by the simple partial of altitude with respect to position (please refer to Sections 4.5.5 and 4.5.7 of reference [4] for a discussion of the development of these density partials). Unfortunately, such a simplification ignores geodetic position dependencies of newer, more complex models, such as the MSISE-90 model used in this investigation.

The last necessary partial is that of the dynamic parameter, or in the case of the drag model, the coefficient of drag. Since the coefficient is a constant, the derivative can be found by simply dividing (6.15a) by C_D , as shown in Equation (6.24).

$$\frac{\partial \ddot{\mathbf{R}}_D}{\partial C_D} = -\frac{1}{2} \left(\frac{A}{m} \right) \rho \mathbf{V}_{rel} |\mathbf{V}_{rel}| \quad (6.24)$$

6.3.1.2 Lift Model Velocity Partial Derivatives

Due to the dependencies given in the definitions of the \mathbf{N} and \mathbf{W} vectors ((6.11) and (6.13)), the velocity partial derivatives are more complex than that given in (6.22), but they can be obtained using the simplifying assumption in (6.20). If the reader first remembers the form of a derivative of a vector with respect to a vector (given in (6.25)),

the first necessary partial derivative, of the \mathbf{W} unit vector, can then be obtained explicitly and simplified by defining the \mathbf{W}_{TEM} vector in (6.26), and substitution into the form given in Equation (6.27).

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix} \quad (6.25)$$

$$\mathbf{W}_{TEM} = \begin{bmatrix} R_3 W_2 - R_2 W_3 \\ R_1 W_3 - R_3 W_1 \\ R_2 W_1 - R_1 W_2 \end{bmatrix} \quad (6.26)$$

$$\frac{\partial \hat{\mathbf{W}}}{\partial \mathbf{V}_{rel}} = \begin{bmatrix} -\frac{W_1 W_{TEM_1}}{|\mathbf{W}|^3} & -\frac{W_1 W_{TEM_2}}{|\mathbf{W}|^3} - \frac{R_3}{|\mathbf{W}|} & \frac{R_2}{|\mathbf{W}|} - \frac{W_1 W_{TEM_3}}{|\mathbf{W}|^3} \\ \frac{R_3}{|\mathbf{W}|} - \frac{W_2 W_{TEM_1}}{|\mathbf{W}|^3} & -\frac{W_2 W_{TEM_2}}{|\mathbf{W}|^3} & -\frac{W_2 W_{TEM_3}}{|\mathbf{W}|^3} - \frac{R_1}{|\mathbf{W}|} \\ -\frac{W_3 W_{TEM_1}}{|\mathbf{W}|^3} - \frac{R_2}{|\mathbf{W}|} & \frac{R_1}{|\mathbf{W}|} - \frac{W_3 W_{TEM_2}}{|\mathbf{W}|^3} & -\frac{W_3 W_{TEM_3}}{|\mathbf{W}|^3} \end{bmatrix} \quad (6.27)$$

Next the partial of the \mathbf{N} unit vector can be obtained in a similar manner by defining the temporary vectors \mathbf{TA} in (6.28) and \mathbf{TB} in (6.29) to simplify the explicit partial, which is then given in Equation (6.30).

$$\mathbf{TA} = \begin{bmatrix} -R_3V_{rel_3} - R_2V_{rel_2} \\ 2R_2V_{rel_1} - R_1V_{rel_2} \\ 2R_3V_{rel_1} - R_1V_{rel_3} \\ -R_3V_{rel_3} - R_1V_{rel_1} \\ 2R_3V_{rel_2} - R_2V_{rel_3} \\ 2R_1V_{rel_2} - R_2V_{rel_1} \\ -R_2V_{rel_2} - R_1V_{rel_1} \\ 2R_2V_{rel_3} - R_3V_{rel_2} \\ 2R_1V_{rel_3} - R_3V_{rel_1} \end{bmatrix} \quad (6.28)$$

$$\mathbf{TB} = \begin{bmatrix} N_1TA_1 + N_2TA_2 + N_3TA_3 \\ N_1TA_6 + N_2TA_4 + N_3TA_5 \\ N_1TA_9 + N_2TA_8 + N_3TA_7 \end{bmatrix} \quad (6.29)$$

$$\frac{\partial \hat{N}}{\partial \mathbf{V}_{rel}} = \begin{bmatrix} \frac{TA_1}{|\mathbf{N}|} - \frac{N_1TB_1}{|\mathbf{N}|^3} & \frac{TA_6}{|\mathbf{N}|} - \frac{N_1TB_2}{|\mathbf{N}|^3} & \frac{TA_9}{|\mathbf{N}|} - \frac{N_1TB_3}{|\mathbf{N}|^3} \\ \frac{TA_2}{|\mathbf{N}|} - \frac{N_2TB_1}{|\mathbf{N}|^3} & \frac{TA_4}{|\mathbf{N}|} - \frac{N_2TB_2}{|\mathbf{N}|^3} & \frac{TA_8}{|\mathbf{N}|} - \frac{N_2TB_3}{|\mathbf{N}|^3} \\ \frac{TA_3}{|\mathbf{N}|} - \frac{N_3TB_1}{|\mathbf{N}|^3} & \frac{TA_5}{|\mathbf{N}|} - \frac{N_3TB_2}{|\mathbf{N}|^3} & \frac{TA_7}{|\mathbf{N}|} - \frac{N_3TB_3}{|\mathbf{N}|^3} \end{bmatrix} \quad (6.30)$$

With these partials, it is possible to obtain the partials of the lift accelerations with respect to the velocity vector. Equation (6.31), the partial of the magnitude of \mathbf{V}_{rel} with respect to \mathbf{V}_{rel} , is used in conjunction with the chain rule to differentiate (6.16a) and (6.17a), and produce Equations (6.32a) and (6.32b).

$$\frac{\partial |\mathbf{V}_{rel}|^2}{\partial \mathbf{V}_{rel}} = \frac{\partial (V_{rel_1}^2 + V_{rel_2}^2 + V_{rel_3}^2)}{\partial \mathbf{V}_{rel}} = [2V_{rel_1} \quad 2V_{rel_2} \quad 2V_{rel_3}] = 2\mathbf{V}_{rel}^T \quad (6.31)$$

$$\frac{\partial \ddot{\mathbf{R}}_{L-N}}{\partial \mathbf{V}_{rel}} = S_{L-N}\rho \left[|\mathbf{V}_{rel}|^2 \frac{\partial \hat{N}}{\partial \mathbf{V}_{rel}} + 2\hat{N}\mathbf{V}_{rel}^T \right] \quad (6.32a)$$

$$\frac{\partial \ddot{\mathbf{R}}_{L-w}}{\partial \mathbf{V}_{rel}} = S_{L-w}\rho \left[|\mathbf{V}_{rel}|^2 \frac{\partial \hat{W}}{\partial \mathbf{V}_{rel}} + 2\hat{W}\mathbf{V}_{rel}^T \right] \quad (6.32b)$$

6.3.1.3 Lift Model Position Partial Derivatives

The current drag model in GTDS uses a combination of partials, discussed above, to calculate the partial derivative of the acceleration due to drag with respect to position. Among these partials is the partial of the atmospheric density with respect to position, and its calculation make the determination of the position partial for the SLM (which primarily uses the MSISE-90 atmospheric model) very difficult. The MSISE-90 formulation has a significant number of terms, which makes analytical partials unpleasant. However, the MSISE-90 model is continuous in its arguments, allowing for accurate numerical computation of density partials [107]. As such, it was decided to numerically approximate the entire partial (not just the density with respect to position one) of the lift acceleration.

To develop the position partials, the first step is to define a process, $ACLF(\mathbf{R})$. The $ACLF(\mathbf{R})$ process involves recalculating \mathbf{V}_{rel} , \mathbf{N} , \mathbf{W} , ρ , and the corresponding lift accelerations as a function of the spacecraft position vector. If the position vector is then perturbed by a user defined δR , the perturbed position vector can be input into the process to develop the corresponding perturbed accelerations. Equations (6.33a) through (6.33c) define three perturbed accelerations in each component (the equations are generalize for both \mathbf{N} and \mathbf{W} accelerations, which are denoted in (6.34) and (6.35) with an additional subscript identifier).

$$\ddot{\mathbf{R}}_{LX} = ACLF(R_1 + \delta R, R_2, R_3) \quad (6.33a)$$

$$\ddot{\mathbf{R}}_{LY} = ACLF(R_1, R_2 + \delta R, R_3) \quad (6.33b)$$

$$\ddot{\mathbf{R}}_{LZ} = ACLF(R_1, R_2, R_3 + \delta R) \quad (6.33c)$$

The nominal acceleration is then subtracted from the above accelerations and divided by the δR to determine the 9 components of the position partials matrices shown in Equations (6.34) and (6.35)

$$\frac{\partial \ddot{\mathbf{R}}_{L-N}}{\partial \mathbf{R}} = \begin{bmatrix} \frac{\ddot{\mathbf{R}}_{LNX_1} - \ddot{\mathbf{R}}_{L-N_1}}{\delta R} & \frac{\ddot{\mathbf{R}}_{LNY_1} - \ddot{\mathbf{R}}_{L-N_1}}{\delta R} & \frac{\ddot{\mathbf{R}}_{LNZ_1} - \ddot{\mathbf{R}}_{L-N_1}}{\delta R} \\ \frac{\ddot{\mathbf{R}}_{LNX_2} - \ddot{\mathbf{R}}_{L-N_2}}{\delta R} & \frac{\ddot{\mathbf{R}}_{LNY_2} - \ddot{\mathbf{R}}_{L-N_2}}{\delta R} & \frac{\ddot{\mathbf{R}}_{LNZ_2} - \ddot{\mathbf{R}}_{L-N_2}}{\delta R} \\ \frac{\ddot{\mathbf{R}}_{LNX_3} - \ddot{\mathbf{R}}_{L-N_3}}{\delta R} & \frac{\ddot{\mathbf{R}}_{LNY_3} - \ddot{\mathbf{R}}_{L-N_3}}{\delta R} & \frac{\ddot{\mathbf{R}}_{LNZ_3} - \ddot{\mathbf{R}}_{L-N_3}}{\delta R} \end{bmatrix} \quad (6.34)$$

$$\frac{\partial \ddot{\mathbf{R}}_{L-W}}{\partial \mathbf{R}} = \begin{bmatrix} \frac{\ddot{\mathbf{R}}_{LWX_1} - \ddot{\mathbf{R}}_{L-W_1}}{\delta R} & \frac{\ddot{\mathbf{R}}_{LWY_1} - \ddot{\mathbf{R}}_{L-W_1}}{\delta R} & \frac{\ddot{\mathbf{R}}_{LWZ_1} - \ddot{\mathbf{R}}_{L-W_1}}{\delta R} \\ \frac{\ddot{\mathbf{R}}_{LWX_2} - \ddot{\mathbf{R}}_{L-W_2}}{\delta R} & \frac{\ddot{\mathbf{R}}_{LWY_2} - \ddot{\mathbf{R}}_{L-W_2}}{\delta R} & \frac{\ddot{\mathbf{R}}_{LWZ_2} - \ddot{\mathbf{R}}_{L-W_2}}{\delta R} \\ \frac{\ddot{\mathbf{R}}_{LWX_3} - \ddot{\mathbf{R}}_{L-W_3}}{\delta R} & \frac{\ddot{\mathbf{R}}_{LWY_3} - \ddot{\mathbf{R}}_{L-W_3}}{\delta R} & \frac{\ddot{\mathbf{R}}_{LWZ_3} - \ddot{\mathbf{R}}_{L-W_3}}{\delta R} \end{bmatrix} \quad (6.35)$$

6.3.1.4 Lift Model Dynamic Parameters Partial Derivatives

The last set of partials needed is the acceleration partials with respect to the dynamic parameters, which are by far the easiest to calculate. Since the coefficients of lift are constant, the partials with respect to those constants is simply the acceleration divided by the constant, as was shown in (6.24). These equations are given in (6.36) and (6.37).

$$\frac{\partial \ddot{\mathbf{R}}_{L-N}}{\partial C_{L-N}} = -\frac{1}{2} \left(\frac{A}{m} \right) \rho |\mathbf{V}_{rel}|^2 \hat{\mathbf{N}} \quad (6.36)$$

$$\frac{\partial \ddot{\mathbf{R}}_{L-W}}{\partial C_{L-W}} = -\frac{1}{2} \left(\frac{A}{m} \right) \rho |\mathbf{V}_{rel}|^2 \hat{\mathbf{W}} \quad (6.37)$$

6.3.2 NT-GTDS Integration

The incorporation of the SLM into NT-GTDS was a relatively extensive modification. As such, the majority of the details of the code modifications, additions, and updates have been relegated to Appendix C. This section is meant only to give the reader a brief overview of some of the routines which were modified, and the basic structure of the primary modifications. Figure 6.10 identifies the modified routines responsible for initialization, filter update, DC update, and output functionality. Figure 6.11 identifies the flow structure of the primary routines involved with the SLM. The reader should note that these modifications only reflect the changes made to create a functional lift model for use with the DC, FILTER, and EPHEM programs and the Cowell integrator. A listing of routines that will need to be modified to get the SLM working with DSST is also included in Appendix C.

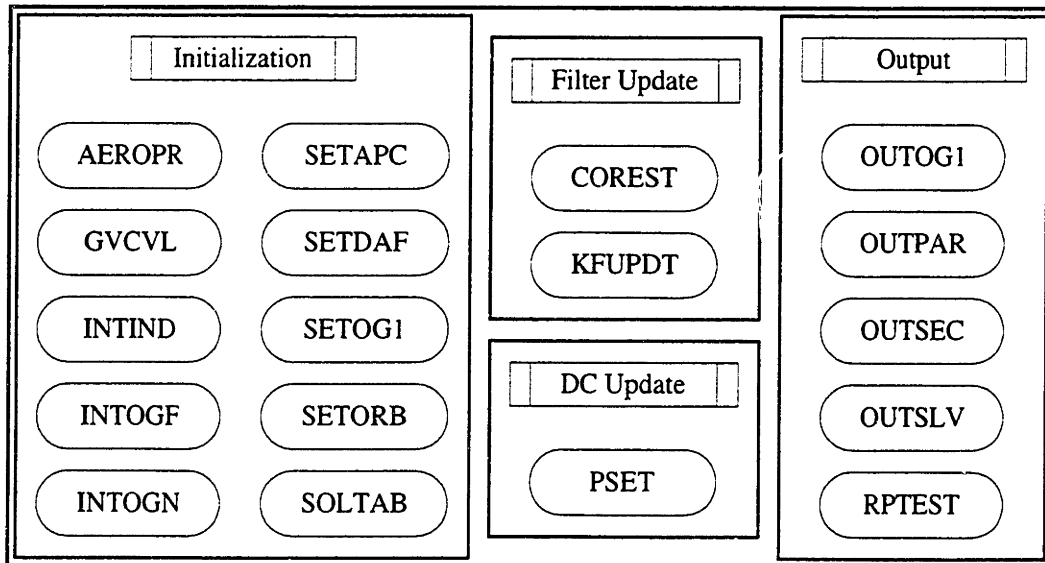


Figure 6.10: SLM Routine Modifications

The partials discussed in the previous section are calculated in AEROLIFT, which is called by AERO if the lift accelerations or partials are needed. AEROLIFT delegates the partials to VRELPART, which calculates the V_{rel} partials, and LIFTACC which represents the ACLF(R) function. The forces and partials calculated are then used by

VARFRC which then computes the partials of the total acceleration vectors with respect to position, velocity, and dynamic parameters. DRAGV combines the aerodynamic force vectors and partials for use with VARFRC. For a more detailed description of the SLM code modifications and additions, please refer to Appendix C.

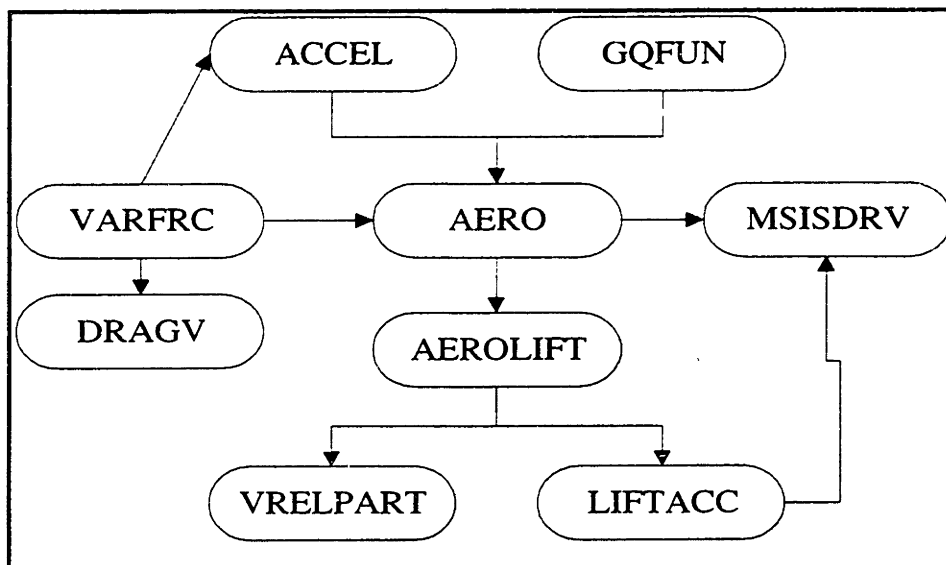


Figure 6.11: SLM Flow Diagram

6.3.3 Testing and Validation

Validation of a physically unrealistic model can be a difficult task. Testing of the model can be accomplished by simply conducting unit and regression tests, but the validation of the model must include extensive tests to even understand the behavior and characteristics of the lift model. In addition to the tests listed below, tests done on the lift model during the last week of 13390's decay are included in Chapter 7. The reader is encouraged to read that section for validation comments and model analysis.

- **Unit Tests:** each new routine, or modification of an old routine, which made any calculation in the SLM structure was tested with individual testing

programs to make sure that each unit was producing the correct values. These numbers were all compared against a Mathcad workbook and validated.

- **Regression Tests:** simply put, these tests were to insure that the SLM didn't break anything that worked before. With the exception of a few select output statements, results matched those obtained with PR5 exactly. The following programs and corresponding propagation methods were tested against PR5:
 - *DC:* DSST, Cowell, Time Regularized Cowell
 - *EPHEM:* DSST, Cowell, Time Regularized Cowell
 - *FILTER:* DSST, Cowell, Time Regularized Cowell, Runge-Kutta
- **Ideal PCE Fit:** As discussed in Chapter 4, GTDS has the ability to generate artificial observations with the EPHEM program for use as input data for a DC run. The process, identified as a Precise Conversion of Elements (PCE) truth data fit, creates a "truth" orbit with specified lift coefficients and acceleration effects. The DC is then used to fit over the PCE data to attempt to recover the values used as input for the EPHEM run. The first PCE run attempted was at a perigee height of approximately 400km, and the effect of the lift coefficients were barely noticeable in comparison with drag. As such, if a priori values were used that were close to the PCE values, the DC converged with impressive accuracy on the first run. By perturbing the input values for lift, due to the dominance of the drag effect, CD was adjusted more readily than the lift coefficients to fit the data. Further results, which will be presented in Chapter 7, show performance of the lift model as the satellite goes through final decay, which improves upon previous results and identifies characteristics of the SLM.

6.4 Other Decay Phase Perturbations

Although density modeling and lift were the only perturbations addressed as a part of this investigation, the highly dynamic environment of catastrophic decays are subject to many more, which will be analyzed in the following sections. All Macsyma files for these analyses can be found at: \$E53-BOBD:\EVO-HEO\MACSYMA\ATMOS.

6.4.1 Speed Ratio Error

One source of error is found in our earlier assumption of hypersonic flow in the initial lift analysis, known as the speed ratio error. The reader can indeed verify the existence of hypersonic flow by referencing Figure 6.12 which graphically demonstrates the typical velocity (in km/s - z axis) of a Molniya orbit about the Earth.

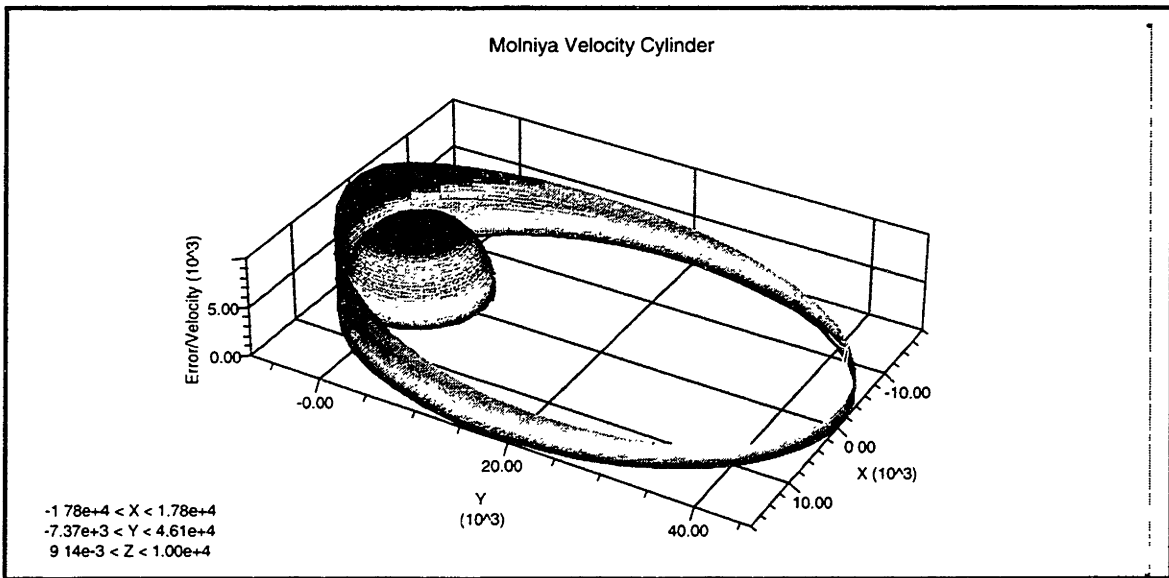


Figure 6.12: Molniya Velocity Cylinder

Loosely defined as flow above Mach 5 [51], hypersonic flow is clearly demonstrated for the atmospheric portion of the Molniya orbit with speeds on the order of Mach 30. Equation (6.38) [105] represents the speed ratio where S is the speed ratio, U is the satellite velocity, M is the Mach number, R is the universal gas constant, T is the

temperature of the gas, and ∞ refers to free-stream conditions [112] (hypervelocity flow is where S approaches infinity).

$$S_{\infty} = \frac{U_{\infty}}{\sqrt{\frac{2RT}{M}}} \quad (6.38)$$

The two main factors which affect S, and therefore contribute to its mis-modeling, are the velocity of the satellite (which decreases as altitude increases - see Figure 6.12), and the temperature of the atmosphere which increases with increasing altitude through the exospheric heating region (see Section 6.5.2). The net effect on S as altitude increases, is a considerable decrease, and corresponding deviation from the hypervelocity assumption.

Karr showed that speed ratio error can effectively be the largest source of error in atmospheric prediction, boasting errors from 2% - 10% for the coefficient of drag for a sphere (where irregular shapes can be assumed to be greater). Fortunately, the MSISE-90 atmospheric model uses data primarily consisting of direct measurement of composition and temperature, only adding accelerometer and drag data in future versions of the MSIS model. As a result, MSISE-90 is unaffected by the speed ratio effect, and this effect will be negated in the error analysis [107].

6.4.2 Transition Region Error

As was discussed earlier, for this analysis, the atmosphere has been stratified into two regions of flow analysis which are separated by a rough boundary where the characteristic dimension of the satellite is equal to the mean free path. Equation (6.39) identifies the mean free path equation given by Karr [113]:

$$\ell = \left(2\pi \frac{\rho}{W/N_A} \sigma^2 \right)^{-1} \quad (6.39)$$

Where σ is the diameter of the gas molecule (using an average of $3.72 \cdot 10^{-10}$ m for air), N_A is Avagadro's constant, and W is the average molecular weight, defined by Equation (6.40) where z is altitude:

$$W = 24.68 + 0.1235z - 0.000874z^2 \quad (6.40)$$

Due to the mean free path's dependence on density, and the inherent variance of density for any given satellite pass, the transition between free molecular flow and continuum flow is described as the transition region. Somewhere between 90 km and 120 km for objects of this size, the transition region is merely a loose boundary within which the mean free path eventually becomes smaller than the characteristic dimension of the object.

Several methods exist for the determination of the coefficient of drag within this region, including those presented by Regan [112], Gorenbukh [114], and Karr [113], [115], with varying degrees of accuracy. Error arises when the interpolation of the coefficients insufficiently matches the actual transition from free molecular flow (with C_D 's greater than 2), to continuum flow (with C_D 's ≤ 1). The atmospheric model in use with this study, the MSISE-90 model, uses a complex cubic spline formulation to model this transition, and incorporates the projected error values in the model error of 3%. It is for this reason that the authors choose to neglect this source of error. Please refer to the aforementioned sources for detailed analysis of this topic.

6.4.3 Higher Order Drag and Oblateness Coupling

As previously discussed, modeling the atmospheric density (drag) becomes very important as a satellite enters the atmosphere. Within this region, the perturbations due to J_2 (oblateness of the earth) and drag are significant. As a result, it is necessary to include second order terms for oblateness and drag in the DSST model. Since the motion of the artificial satellite is modeled as a non-linear dynamic system, coupling of these second order oblateness and drag terms must also be considered. Andrew Green, in his doctoral thesis "Orbit Determination and Prediction Processes for Low Altitude Satellites [95]," extends the semianalytic equations of motion to second order in small parameters when two perturbing forces are considered. He develops a general form of a second order averaged orbit generator for two perturbations, a general form of the oblateness-drag coupling, and then optimizes the theory using various assumptions. Green also developed a GTDS subroutine named HODRAG which incorporates the second order drag, J_2^2 , and coupling terms into DSST.

To determine the error contribution of drag/oblateness coupling, the effect of using various HODRAG options (and neglecting HODRAG altogether) in a very high drag regime was evaluated. For the test runs, an interval of 7 months before the end of the orbital life of satellite 13390, when perigee height is approximately 200 km, was chosen. Using the SST propagator and the optimal Molniya case settings [90], the weighted RMS and semi-major axis sigma values in Table 6.5 were obtained for a 20 day batch DC run on GTDS. For a detailed description of the various HODRAG options, please refer to Green [95]. Table 6.5 also includes the various card settings used for the different runs. Please note the AVRDRAG (I1) card controls the main HODRAG effects, SSTAPGFL (R2) controls the partial derivative options, and AVRHARM (R2) is the switch for the Izsak height correction (2 = Off, 1 = On).

Table 6.5: HODRAG Option Effects

Run #	SSTAPGFL	AVRHARM	AVRDRAG	Description	Weighted RMS	SMA sigma (m)
Cow				Cowell truth comparison run	2.1614057	7.49068
0	0	2	0	HODRAG off, first order drag effects only.	4.0952666	9.21112
1	6	1	0	First order drag effects with Izsak's J_2 height correction used in the determination of the density. This includes first order drag effects and an approximation to the J_2 -drag effects.	4.0937578	9.20785
2	1	1	1	Drag and J_2 -drag effects	4.0896242	9.19904
3	2	1	2	Drag, J_2 -drag, and drag ² effects.	4.0896142	9.19902
4	3	1	3	Drag, J_2 -drag, drag- J_2 , and drag ² effects. (J_2 and J_2^2 done analytically)	4.1696534	9.67055
5	4	1	4	Drag, J_2-drag, and drag² effects done numerically and drag-J_2 effects done analytically.	4.0896130	9.19893
6	5	1	5	First order drag effects with Izsak's J_2 height correction and analytical drag- J_2 terms.	4.0937578	9.66497
7	0	1	0	Same as Run #1, no partials	4.0937578	9.20785
8	0	1	1	Same as Run #2, no partials	4.0896242	9.19904

9	0	1	2	Same as Run #3, no partials	4.0896110	9.19901
10	0	1	3	Same as Run #4, no partials	4.1696514	9.67054
11	0	1	4	Same as Run #5, no partials	4.0896171	9.19895
12	0	1	5	Same as Run #6, no partials	4.1668708	9.66310

While these results do not show major differences between the options (which is due to the small fit span), the differences do correspond to the results obtained by Green. For example, Run 3 yields slightly better results than Run 2 since it is a more complete model (the same applies with Runs 8 and 9). Runs 5 and 6, which are approximations to Run 4, yield better results (the same applies when comparing 11 and 12 to 10) because of more efficient models. Green's same conclusions may also be drawn in that second order drag effects are very important for low altitude satellites. The major second order effect is the J_2 -drag coupling effect, and it appears that Izsak's height correction is a fairly good approximation of the J_2 -drag effect. Unfortunately, these results also share error sources with Green, including: sampling rate errors, lack of $J_2^2 e$ terms in the averaged orbit generator, quadrature noise, and possible instabilities within the integrator (please refer to Green for a discussion of these errors).

One method Green identifies for reducing these errors was altering the number of frequencies for the J_2 and drag short periodics used in computing the mean element rates. Two of the above runs have been identified Runs (5 and 9) as those which produce the lowest RMS, semi-major axis sigma, and smallest deviations from the orbital elements of both the Cowell run and NORAD element sets. Since Run 9 is considerably faster (without the partials calculations), it has been chosen to use in analyzing the effect of

altering the aforementioned number of frequencies. Table 6.6 compares the differences in changing the number of frequencies on both weighted RMS and semi-major axis sigma (default = 4).

Table 6.6: Frequency Effect Table

<i>Case</i>	<i>J₂- SP</i>	<i>Drag SP</i>	<i>Weighted RMS</i>	<i>SMA sigma (m)</i>
0	4	4	4.0896110	9.19901
a	4	2	4.0896173	9.19902
b	4	6	4.0896093	9.19901
c	4	10	4.0896102	9.19901
d	10	10	4.3318637	271.06000
e	10	4	4.3319138	8.42990
f	6	4	4.1756762	8.63054
g	2	4	4.0884998	9.19658
h	2	2	4.0885032	9.19658

Table 6.6 yields some interesting results. For instance, with an increase in the number of frequencies, there is no appreciable improvement in convergence. In fact, convergence values tend to increase with an increased number of frequencies and conversely decrease as the number goes down. It should also be noted that the sigma for semi-major axis is at its lowest with a higher number of frequencies for the J₂ short periodics (10) and the default number for drag short periodics, but at the cost of speed. It is for this reason, and the low RMS and semi-major axis sigma seen in test case h, that the highlighted settings have been chosen for use with DSST. Using these settings, drag and oblateness coupling errors are minimized for DSST, as well as Cowell methods.

6.4.4 Short Periodics with Drag

Another perturbation whose effects need to be quantified, is that of modeling short periodics with drag. The idea of DSST is to remove high-frequency terms from the equations of motion using the Method of Averaging. This removal of high-frequency terms, or short-periodics, allows the propagation of orbits with a significantly larger step size and increase speed. Currently DSST has the capability to model short periodics with drag via the Fourier coefficients in the short periodic expansions of the DSST ([93], [94], [95]). As drag increases, so will the effects of these short periodics, which leads to appreciable error.

Using the same run as the HODRAG test cases (20 day DC rit span), the effect of the SST drag short periodic options was analyzed to minimize error. The only options are the short periodic expansion longitude for drag (either mean longitude (1), eccentric longitude (2), or true longitude (3)), and the maximum number of frequencies for drag (default = 6). Table 6.7 demonstrates the effects of these options:

Table 6.7: Drag Short Periodic Options Effects

<i>Case</i>	<i>Longitude</i>	<i>Frequencies</i>	<i>Weighted RMS</i>	<i>SMA sigma (m)</i>
0	Cowell	Cowell	2.7510118	8.52311
1	Off	Off	5.9377080	37.7071
2	1	6	6.6413320	43.3000
3	2	6	5.0257408	32.4008
4	3	6	3.7878506	28.6908
5	3	2	3.7879446	28.6915
6	3	10	3.7878506	28.6908
7	3	11...		

For this setup, the best longitude variable for the short-periodic drag expansion was the true longitude with the default number of frequencies for drag (6). Using true longitude allows for a finite expansion (as opposed to a truncated infinite expansion), and therefore yields tighter convergence values. For most orbits (of smaller eccentricities), the mean longitude approximation is sufficient, but not for the Molniya's highly eccentric orbit. The reader should also note that increasing the maximum number of frequencies had no significant effect on the results and, in fact, caused the runs to crash if a value over 10 was used (due to a limitation defined in the code). As such, the highlighted drag short periodic options were chosen as the optimal DSST settings to minimize drag short periodics error.

6.4.5 Atmospheric Winds

Perhaps the most stochastic source of error is from the unpredictable forces of atmospheric winds. Using a simple model of the atmosphere [116] which assumes a rotation of the upper atmosphere wind structure with a rotation rate equal to that of the Earth, and considering an East-West wind which varies in both magnitude and direction (could be approximated with a sinusoid variation with a longitude angle α , measured from a reference point of zero wind) as described in Karr [113], this error can be analyzed. The ratio of extremes of the drag estimate ($\alpha=90$ and 270 corresponding to adding and subtracting wind velocity) can serve as a rough metric of the error magnitude. Equation (6.41) defines this ratio with the given assumptions (6.42).

$$\frac{D_{90^\circ}}{D_{270^\circ}} = \left[\frac{\sqrt{\frac{\mu}{r_p}} \sqrt{1+e} - r_p \Omega_e - V_{\max}}{\sqrt{\frac{\mu}{r_p}} \sqrt{1+e} - r_p \Omega_e + V_{\max}} \right]^2 \quad (6.41)$$

$$r_p \Omega_e \approx 0.5 \frac{km}{sec}$$

$$\sqrt{\frac{\mu}{r_p}} \approx 7.5 \frac{km}{sec}$$
(6.42)

Using (6.41) it is possible to do a simplified analysis of the magnitude of the error produced by these winds. Figure 6.13 shows a three dimensional plot with an eccentricity of 0.75, radius at perigee on the X axis (100 to 1000 km), wind velocity on the Y axis (in km/sec), and percentage error on the Z axis. Under these conditions, there can be up to 6% of the drag estimate error attributable to atmospheric winds (which increases inversely with eccentricity). Assuming an average of 200 m/sec winds, the error is approximately 3% for the Molniya orbit, and increases to almost 4% as the object decays.

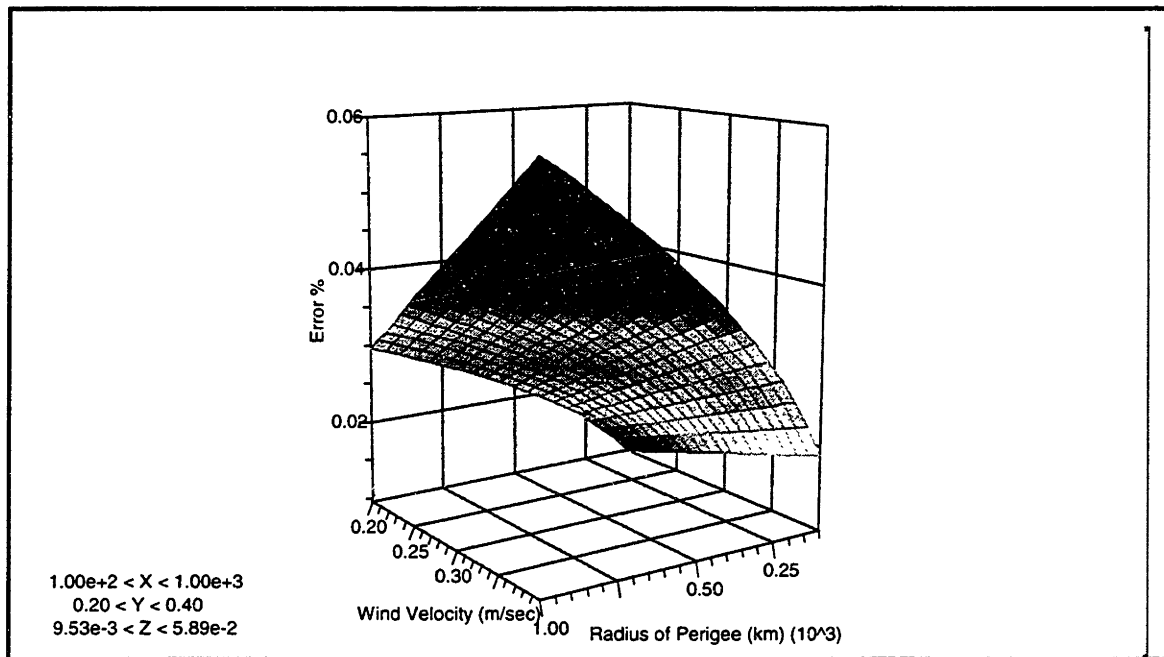


Figure 6.13: Atmospheric Winds Error Magnitude

6.4.6 Reentry Heating and Structural Failure

As catastrophic decay orbits plow through the atmosphere with each perigee passage, they do so at hypersonic speeds. As a result, the shock wave produced by the

interaction with atmosphere becomes very strong, and the temperature of the air behind the shock wave is extraordinarily high. These extreme temperatures, especially prevalent during final decay, cause the air itself to dissociate from N_2 and O_2 gases and create a chemically reacting gas [51]. The heat also imposes an enormous heat load onto the reentering object, which is followed by the extreme cold of space when outside of the atmosphere. This thermal cycling, when combined with gas effects, can lead to ablation of satellite surfaces or, in some cases, structural failure. The interested reader is encouraged to reference the countless numbers of reentry heating analyses, including Anderson [51], for a more detailed description of reentry heating. In the interest of brevity, however, an in depth analysis is not presented here because heating and structural failure have been shown not to play a significant role in modeling the decay of the objects studied in this analysis.

Typically, rocket bodies are built to withstand the rigors of launch and are made of high strength materials with a high tolerance to both heating and thermal cycling. As such, the initial analysis of the 13390 Molniya rocket body included the assumption that no significant ablation or structural failure would occur as the object reentered. During the analysis of 13390, however, a significant force was noticed at approximately 90 km (the start of continuum flow) which significantly altered the rocket body's trajectory, as seen in Figure 6.14 for the semi-major axis and eccentricity.

Initially, the change in trajectory was thought to be caused by an anomalistic burst of atmospheric wind that caused a significant lift force. Whereas lift may well play a role in the phenomenon, investigation of the other decaying rocket bodies proves that this phenomenon is not anomalistic or stochastic in nature. An analysis was done of each rocket body to determine if a similar change in trajectory was noticeable in the 90 km range. With the exception of NSSC-13112 (which had no data for the last four months of

decay), every object demonstrated a similar change in trajectory when going through a perigee height of between 105 and 80 km, with an average of 90 km. As an example, Figure 6.15 demonstrates the change in trajectory for NSSC-14830.

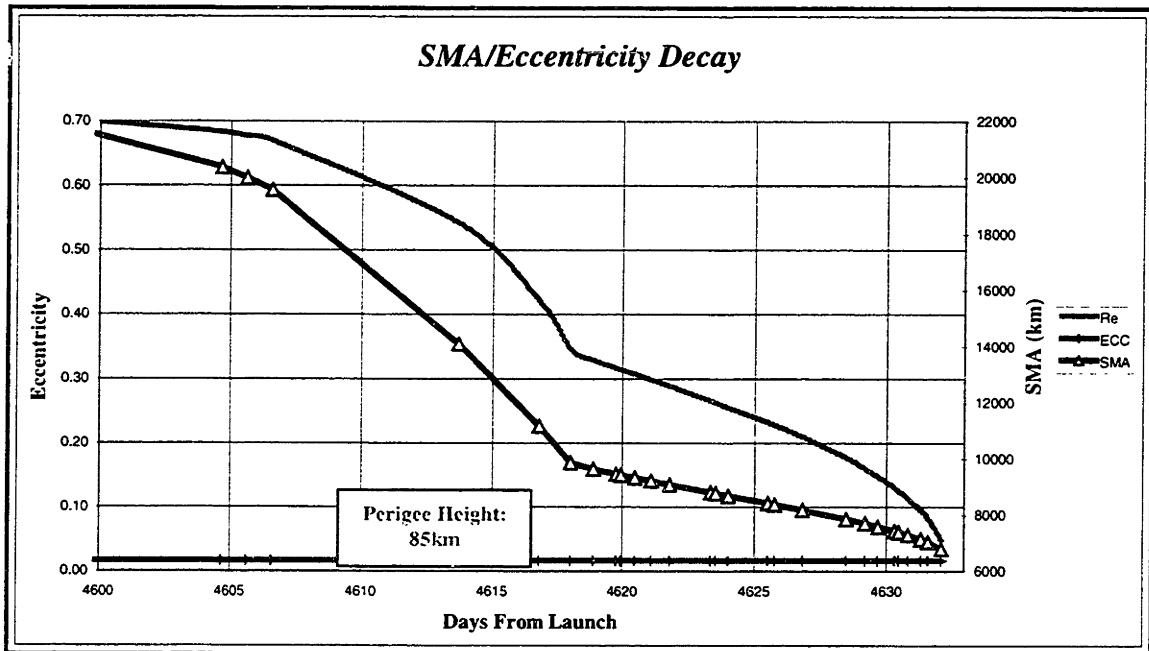


Figure 6.14: NSSC-13390 SMA/Eccentricity Decay History

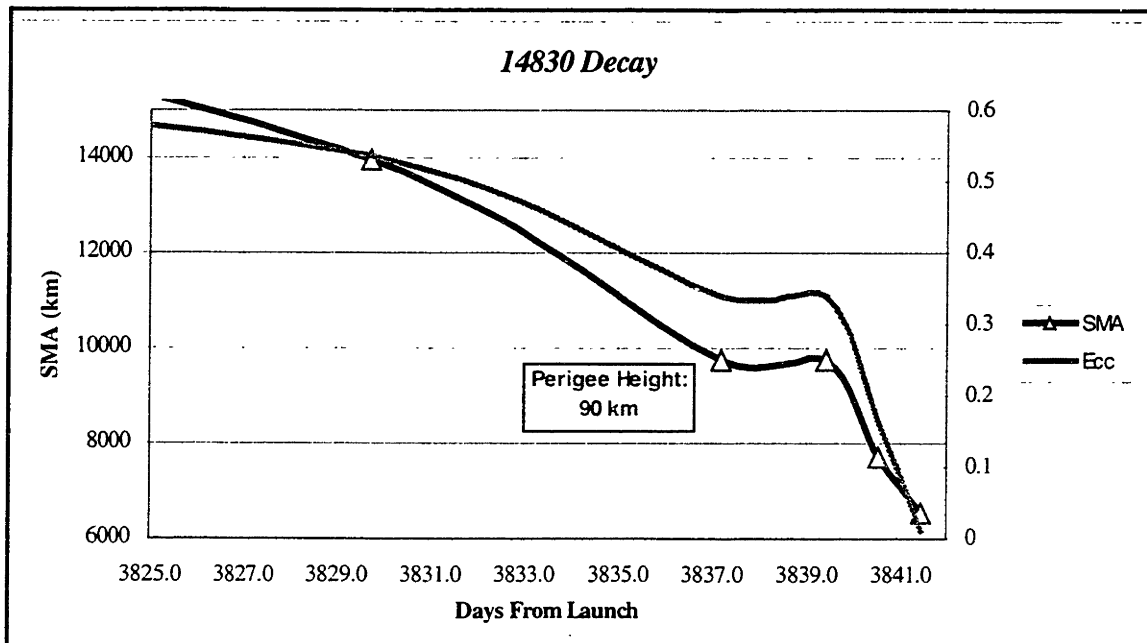


Figure 6.15: NSSC-14830 SMA/Eccentricity Decay History

Due to the fact that this anomaly was a predictable phenomenon, another explanation was sought, because the simple change to continuum flow does not necessarily explain the dramatic change in trajectory. In the search for predictable perturbation sources, a predictable ablation or structural failure was identified as a possible cause.

With no information regarding the materials or construction methods used with the Block L upper stage, a similar rocket body with known characteristics was identified. The only U.S. upper stage which has placed payloads in Molniya type orbits, and whose critical information was readily available, was the Lockheed Agena Space Vehicle. Built around a tubular truss-type aluminum frame, the Agena rocket body is comprised of a beryllium external skin, box type cast magnesium structure for the GNC equipment boom, aluminum tanks, and magnesium skin, monocoque design booster adapter skirt [117]. An in depth analysis of the reentry characteristics of the Agena rocket body was accomplished in reference [118], and concluded that with the addition of a simple ablation coating (Therma-Log), Agena would be able to survive any severe ablation and/or structural failure upon reentry. Considering the analysis was done in 1961, and all objects studied in this analysis were launched after 1980, it can safely be assumed that the Molniya objects had a similar coating. The Agena analysis [118] was investigating lifting reentry trajectories, rather than steep reentry, as they apply to manned reentry. As such, the maximum convective heating rate impulse of 700 BTU/ft/sec², and the low-peaked, long-duration heating pulses experienced by Agena are much less than what catastrophic decay orbits would encounter, and the increased heat loads may produce structural failure. Fortunately, this is not the case, as was shown in a 1995 joint U.S.-Russian analysis of Soviet/Russian satellite fragmentations [30].

With over 300 launches to its credit, the Block L has proven to be an extremely reliable final stage during the past 25 years. Only a total of three Block L's are believed

to have broken up in Earth orbit: one in 1962, and two more in the early 1980's [30]. Table 6.8 lists these breakups including the latter two which were successfully diagnosed as being caused by a rare construction defect in the main engine. While future breakups or structural failures are possible, they are highly unlikely given Block L's track record, and certainly don't explain the predictable change in trajectory upon entry into continuum flow. Accordingly, the effects of reentry heating are not included in the error profile.

Table 6.8: Molniya Final Stage Breakups [30]

Payload	Breakup Date	Breakup Altitude (km)
Sputnik 29	29 October, 1962	200
Cosmos 1305	11 September, 1981	600
Cosmos 1423	8 December, 1982	400

6.4.7 Hypersonic Flow Considerations

Another explanation for the trajectory change noted in the above section springs from not only atmospheric lift, but also from a class of perturbations known as hypersonic flow effects. Anderson [51] presents five primary aspects that characterize hypersonic flow in comparison with other flight regimes including [51]:

- **Thin shock layers:** High speeds cause the flowfield between the shock wave and the body surface to become very thin, thus the shock wave lies very close to the surface, which can cause physical complications. One example would occur at a low Reynolds number, when the boundary layer becomes large enough that it is nearly the same thickness as the shock layer and merges, creating a fully viscous, and very unstable, shock layer.

- ***Entropy layer:*** On blunt objects such as rocket bodies, the shock wave becomes slightly detached from the leading edge and curves around the rocket body. The high degree of curvature around the leading edge causes large velocity gradients in the flow behind the shock wave and strong thermodynamic changes in the flow. This region of strong gradients, or *entropy layer*, extends downstream, eventually interacting with the boundary layer and causing a significant increase in aerodynamic heating.
- ***Viscous Interaction:*** As a result of the hypersonic speeds, the boundary layer becomes so large that the flow outside the boundary layer, called the *inviscid flow*, is greatly affected by the resulting viscous interaction between the thick boundary layer and inviscid flow. The result is a disrupted inviscid flow feedback and increased boundary layer growth, which leads to increased surface pressure and skin friction and, ultimately, increased drag and aerodynamic heating.
- ***High Temperature effects:*** Hypersonic boundary layers are high-temperature regions of the flow due to viscous dissipation of the flow kinetic energy. As mentioned above, high temperatures cause chemical reactions to occur and, as a result, dissociation in nitrogen and oxygen. These dissociated molecules form nitric oxide, ionize and transform the flow to a chemically reacting flow which changes the flowfield properties and has an effect on aerodynamic heating of the spacecraft. This is the most dominant characteristic of hypersonic flow.
- ***Low density flow:*** As discussed in the atmospheric lift section with Karr's description of free-molecular flow, flow behaves differently at high altitudes.

Since these effects are accounted for by the atmospheric models employed in this analysis, they are not important to analyze here.

Combined, these effects can have a dramatic impact on an object during reentry. Although there isn't sufficient time to model these hypersonic flow effects, or the combined perturbative effect of the aforementioned perturbations, two functional error models have been developed to help account for their effects. The development of these error functions is discussed in the following section.

6.5 Error Modeling

The error sources outlined in the preceding sections can cause Kalman filters to diverge due to inaccuracy in the dynamic modeling of the orbital motion. Chapter 3 discusses some of the ways to account for these error sources, or process noise, such as non-adaptive methods, adaptive structural methods, and adaptive statistical methods. Without attitude data to decipher some of the discussed perturbation models, two non-adaptive noise models have been developed to compensate for the un-modeled perturbations of the catastrophic decay environment.

The Time dependent error function was developed specifically for the NSSC-13390 rocket body using NORAD element set semi-major axis decay data. This data was fit with a function representing the sum of four Gaussian normal distributions, and scaled with a true anomaly scaling factor to account for the position in the orbit. The function proved very successful for the analysis of 13390, but its specificity and decreased performance as the rocket body decayed, led to the development of the Altitude Dependent Error Function (ADEF). ADEF is a smoothed step function which models the atmosphere, independent of the body, and can therefore be used with any object. Increased performance has been noted for the ADEF, as presented in Chapter 7.

6.5.1 Time Dependent Error Function

Due to the complex nature of the perturbative effects encountered during perigee passage, a simplified, non-adaptive error function form was adopted. These effects combine to create forces in both up and down directions orthogonal to drag in a completely stochastic fashion. Incorrectly modeling the stochastic process could create more error than leaving the system alone, therefore, an anomaly isolation function using a form with four Gaussian normal distributions was employed. This function was created as a function of time, which in retrospect was not the best choice (which will be discussed in the next section), but sufficient for this first analysis in light of the high level of error inherent in the process.

The period function is directly correlated to the decay of the semi-major axis, which was believed to give a good approximation of the magnitude of the atmosphere's effect on the satellite. The function has a significant drop corresponding to the satellite's initial entry into significant free molecular flow, levels off, and then experiences catastrophic decay. This corresponds to the initial effects of free-molecular flow on the satellite (e.g. to increase tumbling), a leveling corresponding to the relatively quiet transition zone, and a rapid decay as the satellite reaches continuum flow, atmospheric winds, and other atmospheric phenomena. Figure 6.16 graphically demonstrates this decay. It is realized that this function was created from NORAD element sets that may be prone to some error. Dick Hujsak [119] identified a large number of the element sets for this satellite with no observational data for the orbit before or after the epoch given for the element set, which indicates that the sets may not be of the highest accuracy. Future revisions of the error functions must take this into account, but when creating an error function to approximate a nearly 15% error, accuracy of the element sets becomes almost irrelevant.

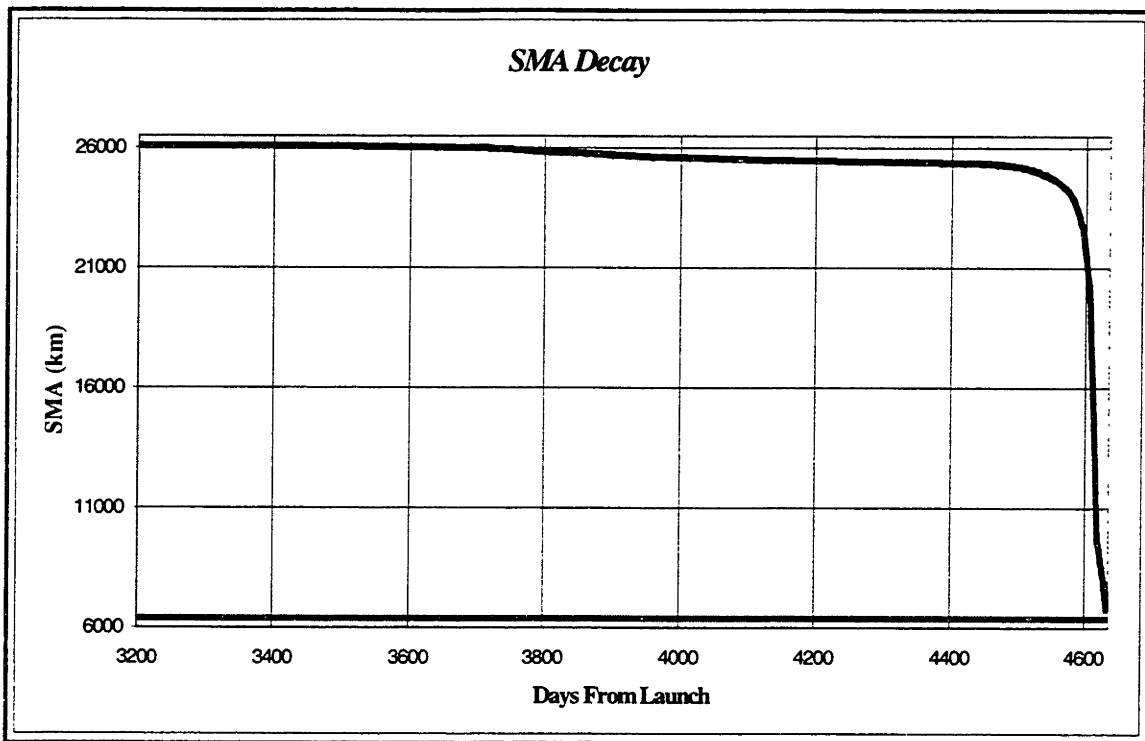


Figure 6.16: NSSC-13390 SMA Decay Profile Close Up

Assuming a form of the sum of four Gaussian normal distributions, the NORAD element set SMA data was utilized, in conjunction with the symbolic math program Macsyma, to solve for the various coefficients. Using the Macsyma Marquardt algorithm to obtain least squares fit for a first guess, then tuning the coefficients to achieve reasonable error limits, an error function was created. The function is accurate to within 0.005% up until the last month of its decay (indicated by 4600 days from launch), and 0.03% for the last month.

The sum of the Gaussians was then scaled approximately from 0 to 1 by multiplying by the inverse of the difference between the initial orbital period (P_0) and the final estimated period (P_f). This semi-major axis error function is the first piece of the error function and is defined as $G_4(ED)$, where ED is the day from launch. Equation (6.43) shows the G_4 function. The corresponding Macsyma file can be found at \$E53-BOB\D:EVO-HEO\MACSYMA\ERROR.MFE

$$G_n[k, \mu, \sigma, (t)] = k \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{t-\mu}{2\sigma}\right)^2} \quad (6.43a)$$

$$G_4(ED) \equiv \left(\frac{1}{P_0 - P_f} \right) \{ G_n[375, 4300, 350] + G_n[160, 4630, 90] + G_n[3200, 4706, 45] + G_n[79, 4612, 9] \} \quad (6.43b)$$

Another part of the function is the True Anomaly variation which focuses the effects of the error function on the portion of the orbit within the atmosphere. The first step in defining this function was to assign an altitude of 600 km as the function reference. Figure 6.17 shows the perigee altitude of the satellite as a function of the day from launch.

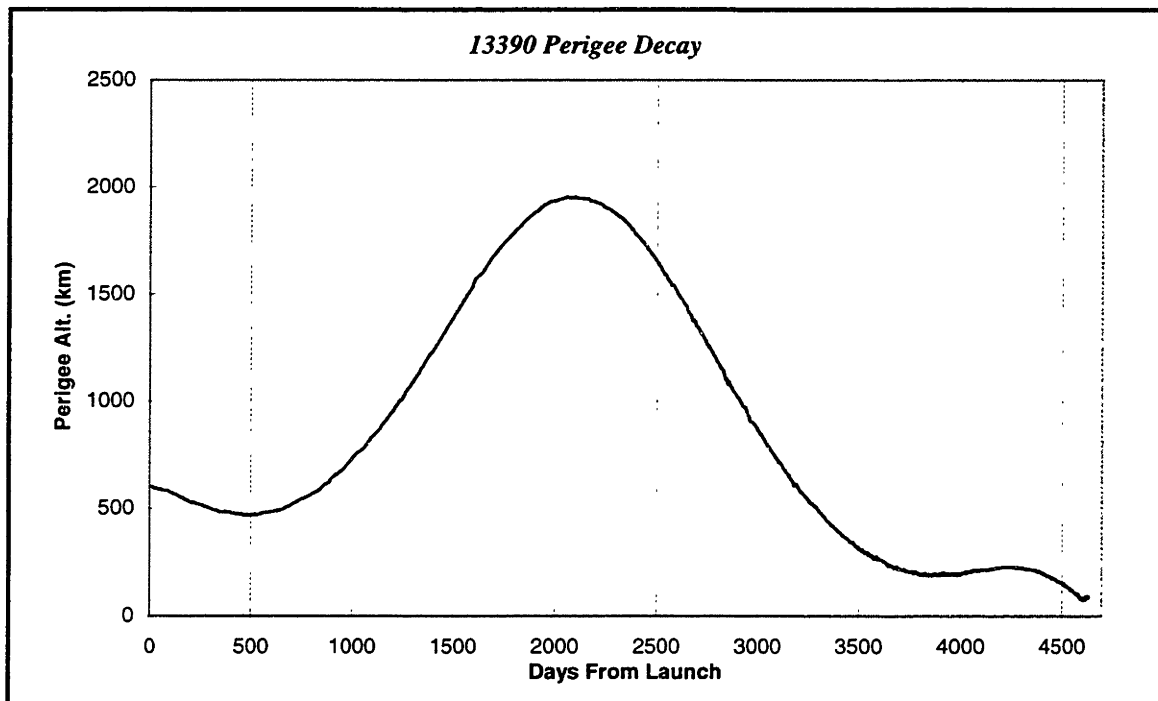


Figure 6.17: NSSC-13390 Perigee Decay Profile

Since the effects of solar radiation pressure and lunar-solar point mass which cause the temporary rise in altitude and eccentricity are fairly well modeled [19], the effects of the error function are focused on the final decay of the altitude. An arbitrary

altitude of 600 km was chosen, corresponding to approximately 3200 days from launch, as a starting point for the function. At that time, 13390 has an eccentricity of 0.74 and a semi-major axis of 26,081 km so that, by using the polar equation of orbit (6.44), the acceptable deviation of the true anomaly to be within 600 km of the surface was determined to be 20.991°, using Equations (6.44) and (6.45).

$$r = \frac{a(1-e^2)}{1+e \cos f} \quad (6.44)$$

$$f = \cos^{-1} \left[\frac{1}{e} \left(\frac{a(1-e^2)}{r} - 1 \right) \right] \quad (6.45)$$

Using half this value for σ_0 , it was then assumed that at that point the Gaussian normal distribution would model 95% of the un-modeled error within the bound established by $0 \pm 20.991^\circ$. Using this assumption, σ_0 , corresponds to the 0.025 percentile of the normal distribution, or 1.96 standard deviations. Solving for σ_0 yielded a value of 10.71°. Next it was necessary to define the basic True Anomaly function, identified as $T_A(f)$, as the sum of two normal distributions centered on 0 and 2π (360°), which is shown in equation (6.46).

$$T_A(f) \sim N(0, \sigma_f) + N(2\pi, \sigma_f) \quad (6.46)$$

As the satellite decays, the standard deviation must increase as more of the orbit falls prey to the atmosphere. To accomplish this, σ_f was scaled with the G_4 function and a multiplicative factor. That factor was determined by scaling the G_4 function with an exponential and manually determining the optimal additional scalar to use with the value $G_4(3200)$ (which corresponds to where the function is to begin). Equation (6.47) represents the final equation obtained for σ_f .

$$\sigma_f(ED) = e^{(2.64 * G_4(ED) - G_4(3200))} \sigma_0 \quad (6.47)$$

Finally, these equations were combined to create a scaling error factor (E_f), composed of the product of the T_A function and the G_4 function (which goes from 0 to 1) and a scaling factor E_k (set to 100 in the following figures), which allows the user to adjust the amount of error. Equation (6.48) shows E_f , while Figures 6.18 and 6.19 demonstrate the error function's behavior. The reader should note that the error function not only increases in magnitude, but also in standard deviation. This accounts for the fact that as the object decays, more of its orbit will fall inside the atmosphere. Figure 6.18 shows the initial error torus created by the error function, with an effective radius of 0 within the mission phase of the orbit. Figure 6.19 shows how the increasing standard deviation eventually causes the bulging error torus (which bulges around perigee) to converge to a large torus close to the Earth. During the decay, the error torus has a steadily increasing radius of error and bulge at perigee. The bulge steadily distributes to the remainder of the torus until it resembles Figure 6.19. Shown graphically in an artificial manner, Figure 6.20 demonstrates the representative shape of the error function at an arbitrary point in the orbit.

$$E_f(ED, f) = E_k T_A(f) G_4(ED) \quad (6.48)$$

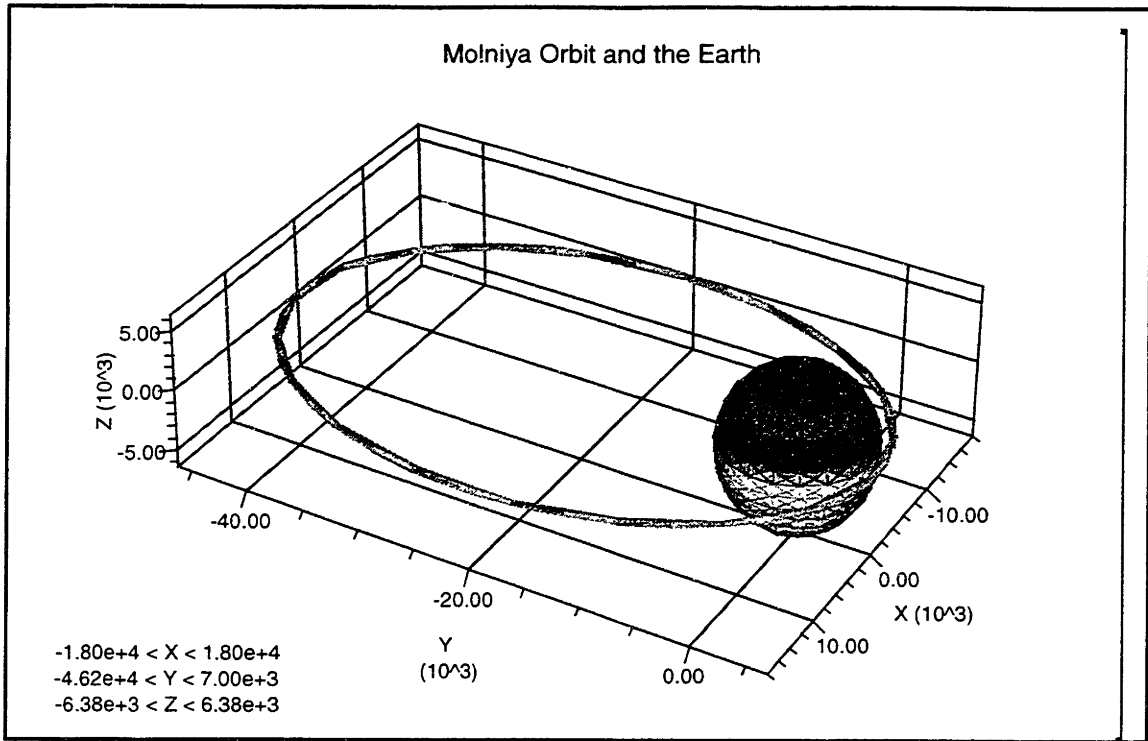


Figure 6.18: Error Torus at Epoch State

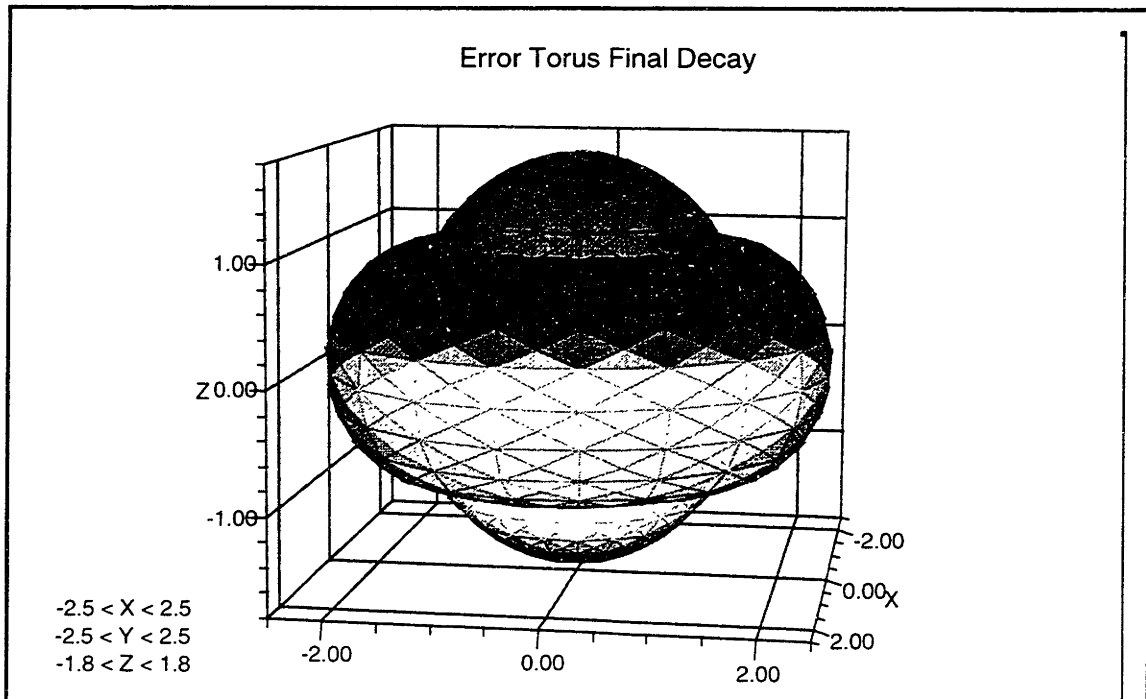


Figure 6.19: Error Torus at Final Decay

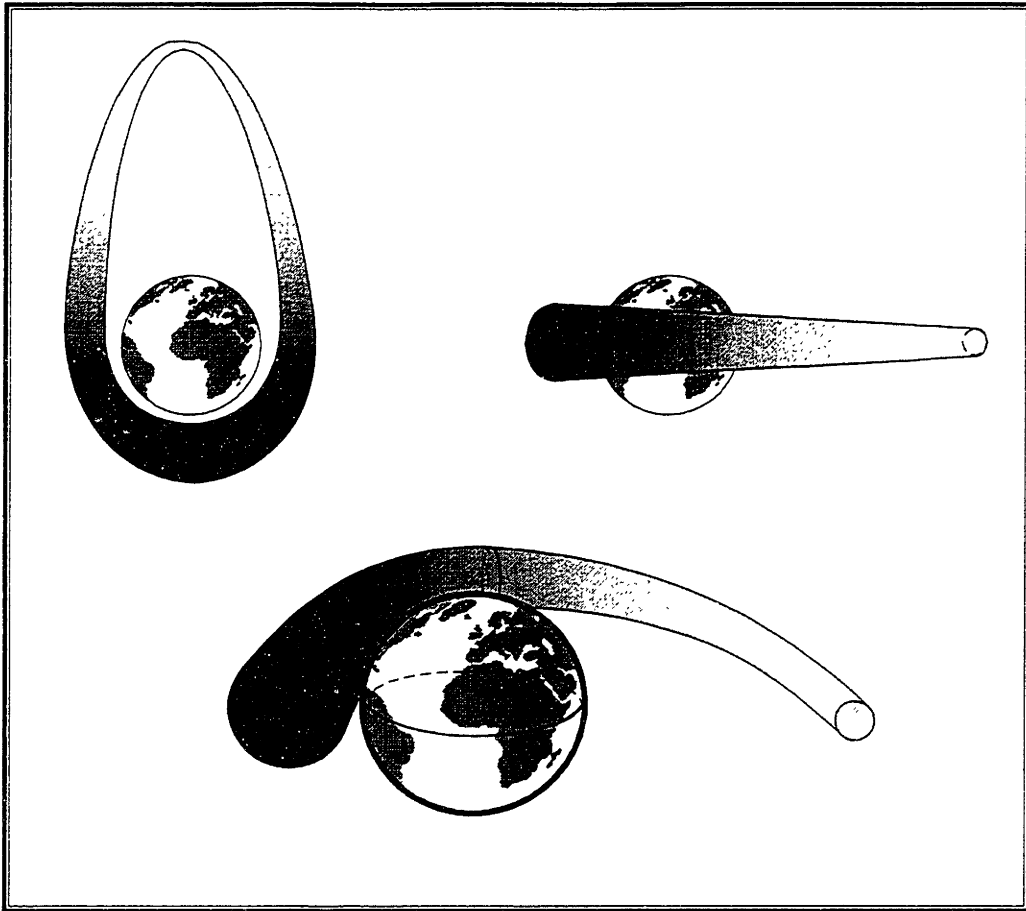


Figure 6.20: Error Torus Graphical Representation

This error function was intended to simply be a first attempt at modeling the process noise for this orbit type, and as such, has been successful. However, as was alluded to earlier, this error function is not without flaws. The most glaring problem with the function is that it is time dependent, as opposed to being a function of semi-major axis and eccentricity. Removing time dependence would make the function more applicable to a wider range of objects, and allow the function to adjust for anomalies with greater efficiency. The function is also susceptible to bad data (or no data) around perigee which can cause the filter to diverge. On the other side of the orbit, at apogee, un-modeled anomalies such as a solar storm are not covered by the function and therefore crash the filter. Highly dynamic data gaps (in which some large anomaly occurs) have a similar effect. One solution may be to add another Gaussian multiplier to add a spike at perigee,

but the centering of this spike, and the knowledge to use it in the first place, make it an ill-fated fix for an incomplete model. Despite its shortcomings, the error function did make the Molniya data reduction possible, while allowing the filter to use a large percentage of the observations and boast a surprisingly small semi-major axis covariance.

6.5.2 Altitude Dependent Error Function

Although apogee anomalies and bad or incomplete data still haunt the Altitude Dependent Error Function (ADEF), it successfully overcomes the dependence on time of its predecessor discussed in Section 6.5.1. It also allows the user to, after determining error estimates for given altitude ranges, use the error function with any similar orbit. The ADEF is more simple to develop, implement, and is more flexible by design. Acting in a similar manner to the time dependent function as demonstrated in Figure 6.**09, the ADEF stratifies the atmosphere into seven different regions, listed below.

- ***Continuum Flow Region (0-90 km)***: The portion of the atmosphere where continuum flow exists, as well as the majority of the atmospheric effects modeling error.
- ***Transition Region (90-120 km)***: Described above as the portion of the atmosphere which transitions between free molecular flow and continuum flow.
- ***Free Molecular Flow Region (120-200 km)***: Also described above as the portion of the atmosphere which begins to have a significant impact on the motion of the object, even though the mean free path is still larger than the characteristic dimension of the spacecraft.

- ***Exospheric Heating Region (200-250 km)***: The region from 120 to 600 km is known as the thermosphere and is categorized by its absorption of extreme ultraviolet radiation from the Sun which results in a very rapid increase in temperature with altitude (geomagnetic and magnetosphere activity also contribute to this increase). The increased heat increases atmospheric density in this region and therefore makes these regions more important to modeling. Within the 200-250 km region these high temperatures reach a limiting value, known as the exospheric temperature. Due to the concentration of radiation, geomagnetic, and magnetic forces, this region is defined as the exospheric heating region and will have a measurable impact on the orbital motion of reentering objects [9].
- ***Near Free Molecular Flow Region (250-600 km)***: This region represents the edge of the thermosphere and of free molecular flow. The atmospheric density is much less than in the previous regions, but the dynamics of molecule interactions are still constrained by free molecular flow.
- ***Near Space Vacuum Region (600-1000 km)***: This region represents a part of the atmosphere where molecules are so far apart that they have little effect on a satellite. The MSISE-90 model roughly considers the point at which diffusive equilibrium ends to be the end of the sensible atmosphere, and that occurs somewhere between 1000 and 2000 km. The reader should note, however, that density in this region is highly susceptible to solar activity. Larson showed that density variations between the solar maximum and solar minimum are approximately two orders of magnitude [9].

- **Space Vacuum Region (1000 km and above):** This region corresponds to the end of the atmosphere and corresponding effects. For this region the error is assumed to be zero.

Using the preceding error analysis as a baseline, default error values (on a 0 to 1 scale) for each region were determined, and are given in Table 6.9. The reader should note that these error levels can easily be changed, as well as the region definitions themselves, to adapt and optimize this function to any orbit determination problem.

Table 6.9: ADEF Atmospheric Region Error Estimates

Range (km)	Name	Error
0 – 90	Continuum Flow	1.00
90-120	Transition Region	0.90
120-200	Free Molecular Flow	0.92
200-250	Exospheric Heating Region	0.70
250-600	Near Free Molecular Flow	0.40
600-1000	Near Space Vacuum Region	0.15
1000 and above	Space Vacuum Region	0.00

The simplest approach to use with these regional error estimates, would be to create a simple step function to calculate the error estimate. Unfortunately, a step function lacks any transition between regions and could introduce new errors into the problem. To avoid having such severe transitions between regions, a Hermite cubic was used to integrate a cubic spline into the region transitions. The Hermite cubic, given in Equation (6.49) [63], is a special cubic polynomial which satisfies basic beam boundary conditions (simply supported ends, fixed ends, free ends, sliding clamped ends), and can be used for interpolation (construct a curve that goes through a given set of points).

$$u = a(x-1)^2(2x+1) + b(x-1)^2x + cx^2(3-2x) + dx^2(x-1) \quad (6.49)$$

The four terms are plotted from 0 to 1 in Figure (6.21) to demonstrate their basic form. The reader should note that the first (a) and third (c) terms of the polynomials represent interpolation forms that would sufficiently approximate a smooth transition between two different error levels in the ADEF.

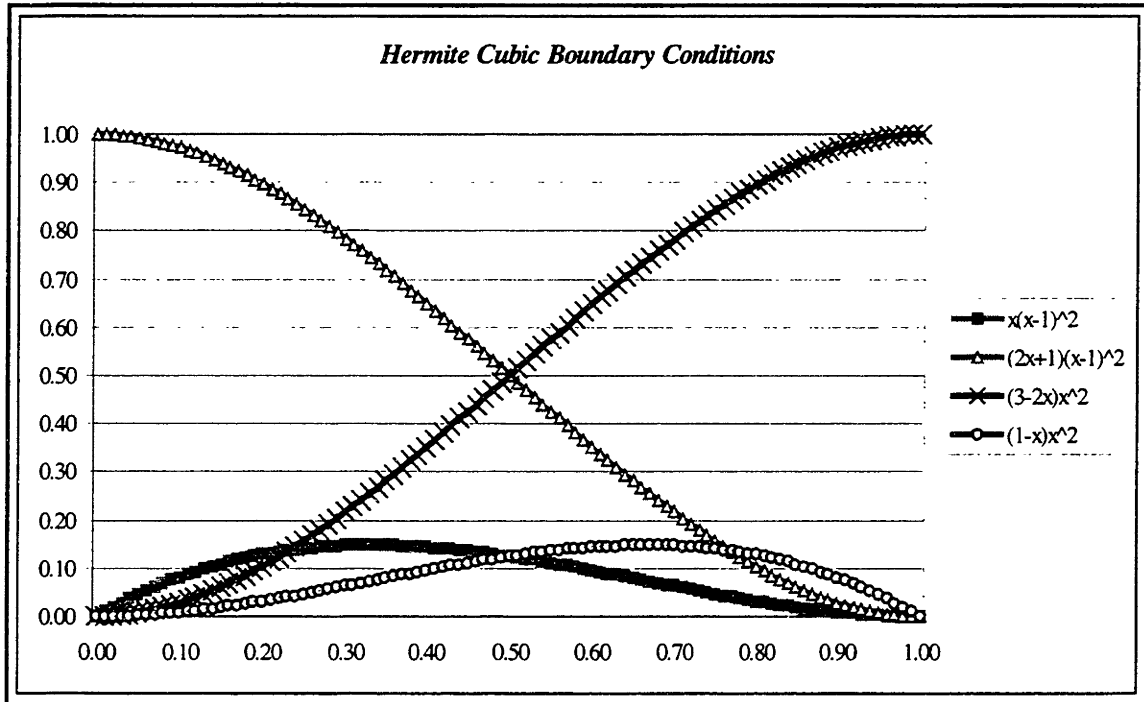


Figure 6.21: Cubics for the Four Boundary Conditions of the Hermite Cubic

The third term was arbitrarily chosen (over the first term) to incorporate into the ADEF to interpolate between steps. The next step was to determine the interval over which the interpolation would take place. To minimize overlaps, it was decided that the half-width of the smallest of the two intervals being joined would be the half-width of the interpolation. If ζ is the vector definition of region boundaries (shown in (6.50) with default values), and the half-width vector ψ is as defined in Equation (6.51). The determination of the proper half width a , is given by the function $\text{int}(a)$ in Equation

(6.52). The $\text{int}(a)$ function will start from the ground up, and determine the smallest interval around a given step to be used with the interpolation.

$$\zeta = \begin{bmatrix} 0 \\ 90 \\ 120 \\ 200 \\ 250 \\ 600 \\ 1000 \end{bmatrix} \quad (6.50)$$

$$\psi = \frac{1}{2} \begin{bmatrix} \zeta_2 - \zeta_1 \\ \zeta_3 - \zeta_2 \\ \zeta_4 - \zeta_3 \\ \zeta_5 - \zeta_4 \\ \zeta_6 - \zeta_5 \\ \zeta_7 - \zeta_6 \\ \zeta_7 - \zeta_6 \end{bmatrix} \quad (6.51)$$

$$\text{int}(a) \equiv \begin{cases} a, & \text{if } \psi_a \leq \psi_{a-1} \\ a-1, & \text{if } \psi_a > \psi_{a-1} \end{cases} \quad (6.52)$$

In order to use the Hermite cubic term properly, the next step is to parameterize over a given interval to maintain values between 0 and 1. This can easily be done by subtracting the start of the interval from the current position and dividing by the interval length. Using this parameterization for the interpolation intervals defined in Equation (6.54a), and the error function step values (defaults given in (6.53)) during the interim altitudes, as shown in (6.54b) for interval i , the ADEF can be defined over each interval. The reader should note that the last interval is slightly different, as shown in Equations (6.55a) and (6.55b).

$$\varepsilon = \begin{bmatrix} 1.00 \\ 0.90 \\ 0.92 \\ 0.70 \\ 0.40 \\ 0.15 \\ 0.00 \end{bmatrix} \quad (6.53)$$

$$\Xi(x, i) = \left[\varepsilon_{i-1} - (\varepsilon_{i-1} - \varepsilon_i) \left\{ \left[\frac{x - (\zeta_i - \Psi_{\text{int}(i)})}{2\Psi_{\text{int}(i)}} \right]^2 \left[3 - \frac{x - (\zeta_i - \Psi_{\text{int}(i)})}{\Psi_{\text{int}(i)}} \right] \right\} \right] \Rightarrow (\zeta_i - \Psi_{\text{int}(i)}) \leq x < (\zeta_i + \Psi_{\text{int}(i)}) \quad (6.54a)$$

$$\Xi(x, i) = [\varepsilon_i] \Rightarrow (\zeta_i + \Psi_{\text{int}(i)}) \leq x < (\zeta_{i+1} - \Psi_{\text{int}(i+1)}) \quad (6.54b)$$

$$\Xi(x, i) = \left[\varepsilon_{i-1} - (\varepsilon_{i-1} - \varepsilon_i) \left\{ \left[\frac{x - (\zeta_i - \Psi_{\text{int}(i)})}{2\Psi_{\text{int}(i)}} \right]^2 \left[3 - \frac{x - (\zeta_i - \Psi_{\text{int}(i)})}{\Psi_{\text{int}(i)}} \right] \right\} \right] \Rightarrow (\zeta_i - \Psi_{\text{int}(i)}) \leq x < (\zeta_i) \quad (6.55a)$$

$$\Xi(x, i) = [\varepsilon_i] \Rightarrow \zeta_i \leq x \quad (6.55b)$$

For the default function of 6 atmospheric regions, 13 definitions are necessary to define the ADEF, but the number of regions is completely scalable using the above generalized equations. The reader should also note that the function is scaled from 0 to 1 and must be modified by a scalar sensitivity factor which must be determined for each satellite (like E_x for the time dependent function), which is not shown in the above equations. Figure 6.22 represents the default definition of the error function (scaled from 0 to 1) over the various altitude regions, demonstrating its ability to transition smoothly between error regions and effectively represent process noise.

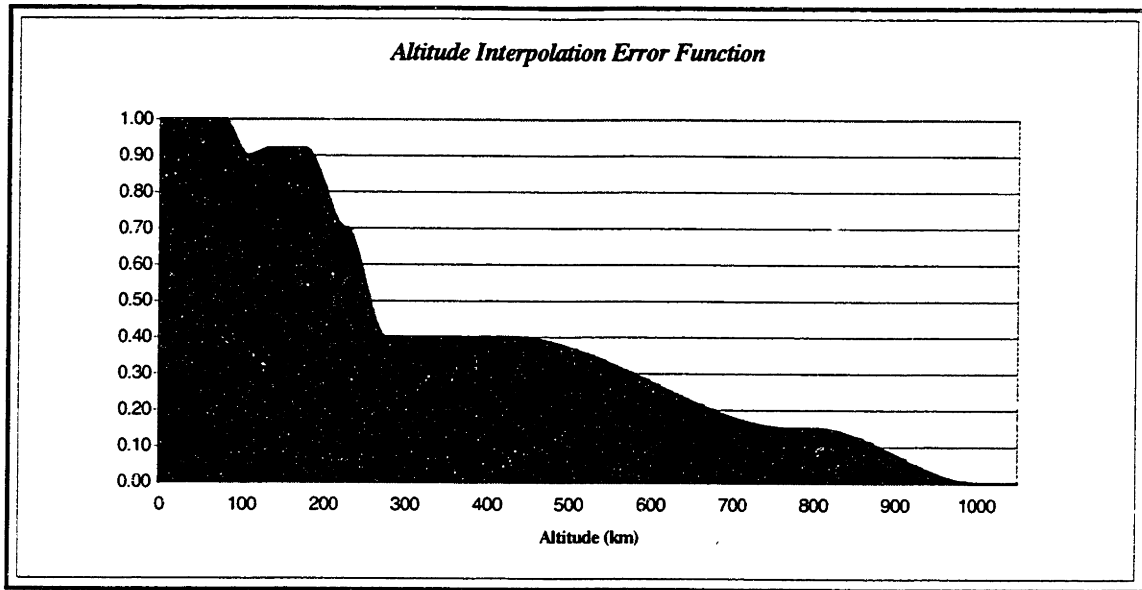


Figure 6.22: Altitude Dependent Error Function (Default)

6.5.3 GTDS Implementation

Implementation of the time dependent error function and ADEF into NT-GTDS was very simplistic, and involved relatively minor code modifications. The only major modification was to the subroutine ANOISE which calculates the analytic representation of process noise used in the NT-GTDS Kalman Filters. ANOISE is called only by KF (for initialization) and KFPRED (for prediction calculations). The subroutines ERRFUNCT, GNEF4, and ALTEF were created to calculate the process noise at a given point and then feed this information back into ANOISE for use in the prediction process. Subroutine TRUANO was also used to determine the True Anomaly at a given point for use with the time dependent function. The Figure 6.23 represents a flow diagram of these routines, and demonstrates the simple structure which was implemented. Several keyword cards were also developed in the SETFIL routine to use these functions, all of which can be found in Appendix C.

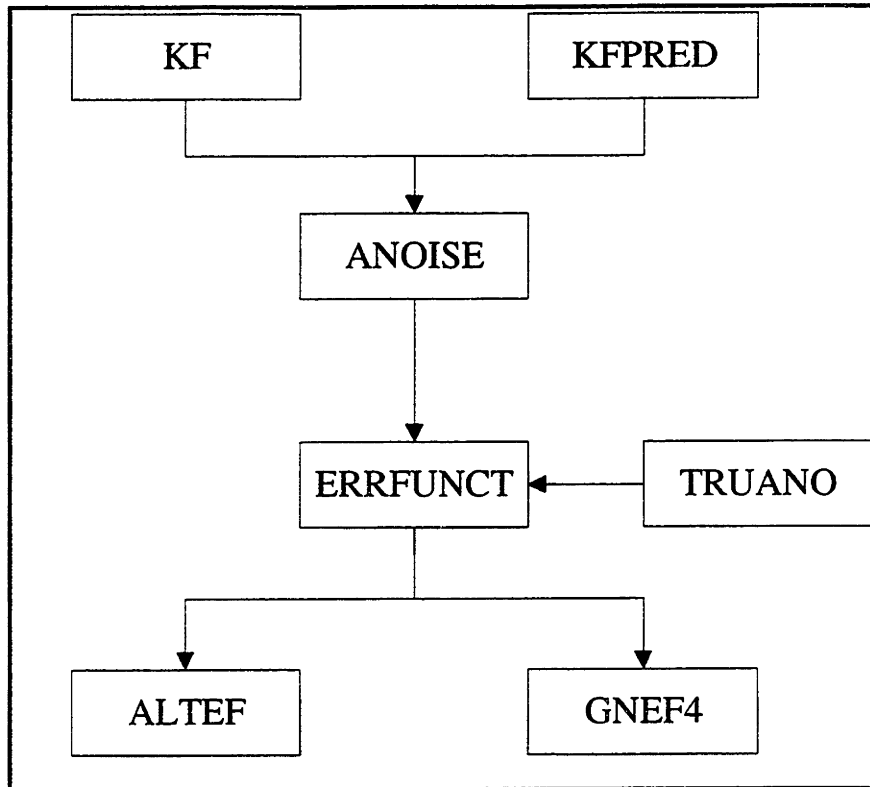


Figure 6.23: Error Function Implementation into NT-GTDS.

[This page intentionally left blank.]

Chapter 7

Decay Phase Testing and Results

7.1 Introduction

Molniya orbits experience a variety of regimes during their decay in which various perturbations are predominant. Transitions between these regimes require constant updates of the various force models, propagators, and interval lengths to be used in order to minimize errors in the predicted orbit. This thesis represents an evolution of thought in how the author approached the orbit determination problem for catastrophic decay cases, stratified into three primary phases. The first phase involved using a least squares batch filter to become familiarized with the catastrophic decay perturbative environment and identify areas for improvement. The next phase involved the use of Kalman Filters and the time dependent error function to process the data for the 13390 rocket body in its entirety. Finally, two improvements developed in the last month of this investigation, the Altitude Dependent Error Function (ADEF) and Simplified Lift Model (SLM), are tested against previous results to determine their effectiveness.

7.2 Differential Correction Batch Runs

After doing initial batch testing in the mission phase, discussed in Chapter 5, the first step in categorizing Molniya decays was to do long term predictions for several

Molniya objects. A nine month propagation was used to determine the effectiveness of current models and insure operation of the newly integrated MSISE-90 model through impact. Table 7.1 identifies the force models that were adopted at the start of the exercise (in accordance with the Fonte/Sabol optimal deck [90]). For each object, a DC fit over a one month time span ten months before decay was taken and the resulting element set used as the epoch value for an ephemeris generation over the same span. The corresponding state parameters, and solved-for values of the coefficient of drag and coefficient of solar radiation, are then used as a-priori values for another 30 day fit span nine months prior to decay. Finally, the converged values from the second DC are used as the a-priori for a nine month ephemeris generation, the results of which are listed in Table 7.2 for 12066, 13390, and 15481. These objects were chosen to show results for two of the best rocket bodies (in terms of data distribution) and a satellite. Please note that the other satellite NSSC-14199 was not included in this exercise due to non-standard behavior (burn maneuver) in 1988 [26].

Table 7.1: Molniya Optimized GTDS Input Card Options at Epoch [90]

<i>GTDS Keyword</i>	<i>Value</i>	<i>Description</i>
ORBTYPE	5 1 1 43200.0 1.0	SST propagator with half-day step size, M50, Runge-Kutta
DRAG	1 1.0	Include drag
ATMOSDEN	9	MSISE-90 Atmospheric Model
MSISOPT2	-1.0	SW(9) to -1 for MSIS hourly Ap option
SPDRAG	0	Do not include drag short periodics
SOLRAD	1 1.0	Include solar radiation effects
SPSRP	0	Do not include the solar radiation pressure short periodics

SPGRVFRC	1 1 3 1.0 1.0 1.0	J_2^2 optimized gravitational force model options for Fourier coefficients in short periodic expansions
SPZONALS	4 3 8	Truncated central body zonal harmonic short periodics
SPMDAILY	4 4 2	Truncated m-daily tesseral harmonic short periodics
MAXDEGEQ	8	8x8 geopotential field
MAXORDEQ	8	8x8 geopotential field
STATEPAR	3	Equinoctial unknowns
STATETAB	1 2 3 4.0 5.0 6.0	Solve for the mean equinoctial elements
DRAGPAR	1	Solve for ρ using pre-stored value as a priori
DRAGPAR2	1 1	Ignore adaptive atmosphere model, solve for C_D
SOLRDPAR	1	Solve for C_R
SSTESTFL	1 2 0 1.0	Partial derivative options: A (analytic), D (finite diff), SP's
SSTAPGFL	1 0 0 1.0 0.0 1.0	Element rate partials options - J_2^2 , drag, and coupling

Table 7.2: Long Term Decay Predictions

<i>NSSC</i>	<i>Actual Decay</i>	<i>Cowell Predicted Decay</i>	<i>SST Predicted Decay</i>
12066	18 November, 1995	7 November, 1995	25 November, 1995
13390	28 March, 1995	15 March, 1995	24 March, 1995
15481	24 April, 1995	21 April, 1995	28 April, 1995

The next step was to move the fit span closer to final decay and analyze the effects on convergence and decay prediction. Unfortunately the ever-increasing atmospheric density time variations (during decay) made the process of using periodic fits and model updates to improve the accuracy of the above predictions, somewhat ineffective. Fits were obtained for NSSC-13390 up to five months before decay, using a series of DC-EPHEM steps like the one described above. Steps were done for one month fit spans eight, seven, six, and five months before final decay using the optimal values for the HODRAG (2 2 2) and SPDRAG (1 3) options described above. The result was a ± 2 day range for both the Cowell and SST final decay predictions (14 - 18 March, 1995), and general trends in the fit span interval and convergence RMS. As the steps got closer to decay, the fit span necessary for any form of convergence became smaller. This necessitated fit spans as small as five days for the six to five month interval. As the fit spans shrunk, it became necessary to increase the values of observation noise for the observation sensors in order to account for the increasing math modeling errors. For example, observation noise was increased by a factor of ten by the six to five month interval. This improved both the convergence and RMS where they would normally degrade. The intent was to run the case again with these elements as a-priori and normal values for the sensor noise. Unfortunately, using such a setup, no acceptable fits were obtained.

The stepping exercise was discontinued after the six to five month interval due to a lack of measurable change in the final decay predictions. The growing disparity in the prediction orbits and observation data was partly responsible for the decreasing fit span interval. Eventually, using the prediction orbit element sets as a-priori elements for successive runs resulted in the DC being unable to converge. This explains the lack of change in the decay prediction in that, as the fit span decreased, the initial orbit prediction was less and less effected by observational data and was allowed to remain relatively

unchanged. Figure 7.1 demonstrates this graphically by showing how the predicted orbit becomes less effected by the fit spans as they decrease in size and increase in observation noise (A through F).

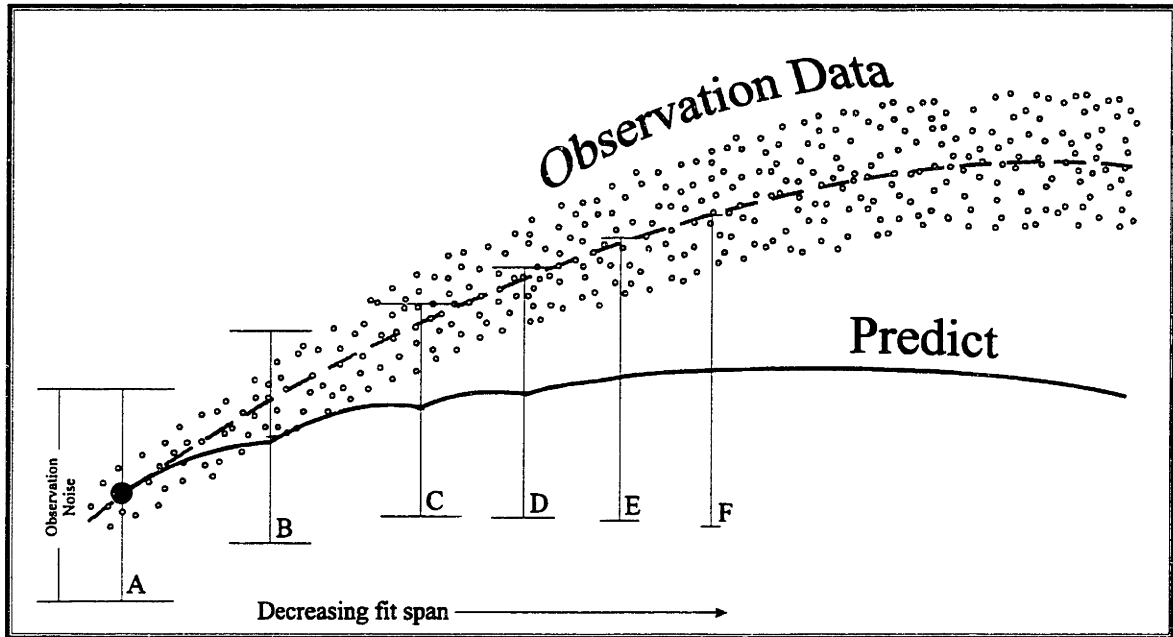


Figure 7.1: Step Fit Span Diagram

The stepping exercise was able to show several things. First, the prediction convergence effect delineated by Figure 7.1, showed that as the span moves closer to decay, there is more time variation in density which forces an increase of sensor observation noise (to account for math modeling errors), decrease in fit span size, and a minimization of short periodic effects. This in turn results in DC elements and ephemeris generations which deviate little from previous predictions. The ultimate result is a predicted orbit which eventually diverges completely and no longer accepts any observational data. It is for this reason that the use of a Kalman Filter technique was employed in subsequent processing.

7.3 Kalman Filter Runs

When analyzing the difference between batch and sequential filters, there is an obvious indication that a sequential filter such as a Kalman filter would be better suited for an object as it gets closer to Earth. As discussed in Chapter 3, a batch filter takes all the available data over the given period and determines the best estimate of the epoch conditions using a method such as weighted least squares. A sequential filter, however, determines the best estimate of the current conditions from the observations by processing them recursively. This is ideal for the decay phase where parameters such as the coefficient of drag, which are held constant in the batch filter and cause increasing error due to their rapidly changing values.

In order to do an accurate evaluation of catastrophic decay, it was necessary to choose an object with a sufficient data distribution to allow the filter to step through highly dynamic regions. Appendix A describes some of the utilities which were used in this determination process, and shows why NSSC-13390 was chosen as the primary test subject for this analysis. Beyond the fact that a Molniya rocket body is easier to model than a Molniya satellite, 13390 had by far the most well distributed data of all the objects analyzed in this analysis. 13390's normal phase (up until 1994), with the exception of a J_2^2 truncation error, was a relatively straightforward orbit which was propagated with high accuracy. As the orbit decayed however, the highly eccentric perigee passage encountered significant un-modeled perturbations, which made data reduction very difficult. As discussed in the previous section, the last 15 day DC fit span in September of 1994 was not sufficiently close enough to decay to produce an accurate impact estimate. The following section outlines the algorithm used to efficiently process the 13390 data in its entirety and make a much better impact estimate.

7.3.1 Analysis Variables

The purpose of creating the error function was to not only provide isolation of the process noise at perigee, but also to allow the user to tune the process noise for position and velocity by changing only two constants. The variable E_k in Section 6.5.1 represents the one dynamic variable in the error function model, which is tuned for both position (KXYZ) and velocity (KXYZDOT) for each run. Stratifying the error function for the position and velocity components makes the function more responsive to changes in eccentricity and semi-major axis than it would be if the same noise was used for all six state components. Other variables that proved useful and dynamic in the reduction of this data was the filter memory discussed previously, changing a-priori drag estimates, varying the gravity model complexity, switching to a linearized filter, and mean anomaly stepping. By observing the number and percentage of observations accepted, as well as the semi-major axis covariance as metrics, each run was manually tuned to obtain optimum performance.

The filter memory was primarily used in the first few months of the decay phase analysis (until February), and proved very effective in weighting the data. In that span, the filter memory should be manually tuned to its lowest working value to weight the data as much as possible. The user should start with a value larger than the largest gap in data during the fit. Then, if there are significant anomalies during the data gap, the memory can be further decreased to improve performance. The user should note that decreasing the memory to a value less than one orbital period usually results in significant error and a majority of the data being rejected.

Another variable available for this analysis was altering the a-priori estimate of drag. In some of the earlier, longer runs, changing C_D had little effect because the filter had enough time and data to correct inaccuracies. As the rocket body decays, however,

the fit span length decreases, as does the filter's ability to correct for invalid C_D approximations. In addition, as will be discussed later, the linearized filter had to be used during the last two months of analysis, and with short runs, the filter did little to correct the dynamic solve-fors (compared to the extended filter used in the previous months). Using basic approximations for given regions (2.0 and up for free molecular flow, 1.0 and lower for hypersonic flow), C_D was manually adjusted to account for un-modeled perturbations and match the data. This proved very useful in the last two months where an obvious un-modeled component (believed to be atmospheric lift) had a significant impact on the data reduction and drove C_D as low as 0.055.

During the initial decay, the semianalytical filter developed by Taylor [65] was used to process data. This was due to its low residuals, quick execution time, and generally better fit of the data than the Runge-Kutta extended filter (the only other option available for the Extended Kalman Filter). When the data started to become too volatile for the extended filter to handle (at the end of January), a simplified gravity model was employed for use with the filter (8x8 to a 2x2) in an attempt to average some of the perturbative effects bombarding the rocket body. This simplification was successful in extending the SST extended filters progression to decay until the end of January, but after that, the data became too erratic for the filter to reduce.

Also during initial decay, as was expected, observation residuals (and filter covariance) increased after data gaps and dropped during spans with measurements. Once the filter reached February, this relatively stable behavior was replaced with an exploding covariance as a result of an unusually high number of large data gaps within February, and a highly dynamic environment as the rocket body entered continuum flow. Figure 7.2 graphically demonstrates the numerous data gaps within the month of February which made data reduction so difficult.

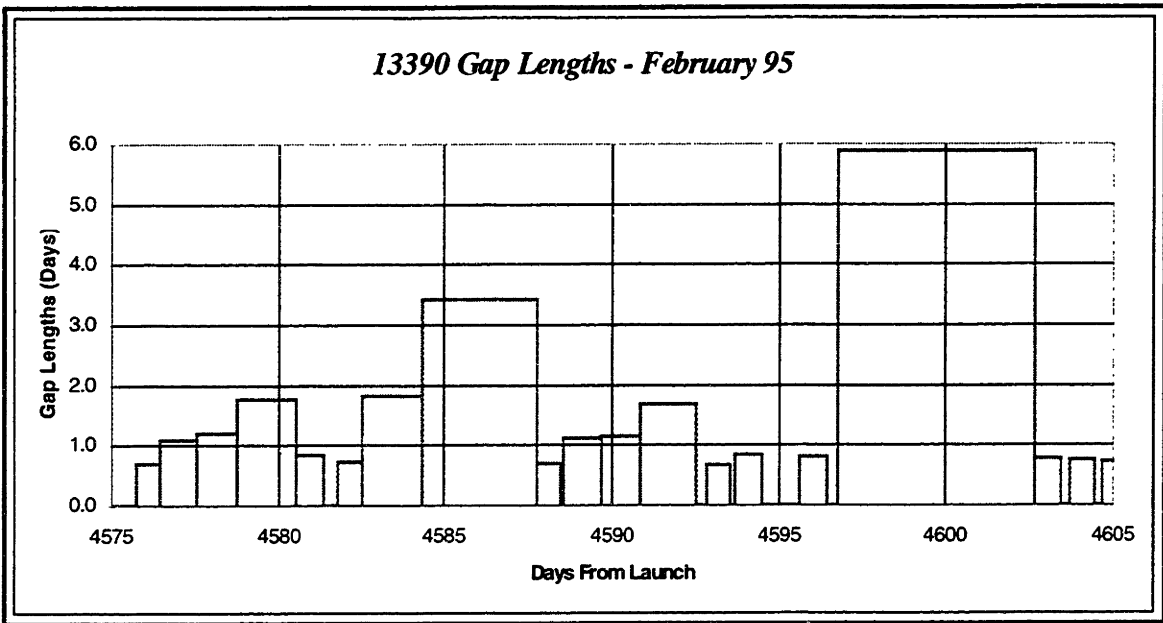


Figure 7.2: February 1995 Data Gaps Plot

Due to the large and frequent data gaps like those observed in Figure 7.2, the extended filter would diverge wildly about the extremely noisy observations and eventually diverge completely. As a result, the GTDS LKF Cowell linearized filter was used (initialized with elements from the SST runs) in order to reduce the data. By not correcting the estimate after every observation, but rather waiting until the end of the fit span, the linearized filter was able to wade through this region and through to final decay. To demonstrate this process graphically, Figures 7.3, 7.4 and 7.5 are presented below. Figure 7.3 shows the data gaps for the DEC_JAN run described in Table 7.4, and Figure 7.4 is a plot the corresponding EGLQ station residuals of the extended filter run over that region. The reader should note the pattern of increasing residuals after gaps followed by a return to stable oscillation within input station error bounds. In contrast, Figure 7.5 shows the semi-major axis residuals for station PRNQ (Princilik) during the last few hours of 13390's lifetime and exhibits a linear increase in error during the span until the final correction.

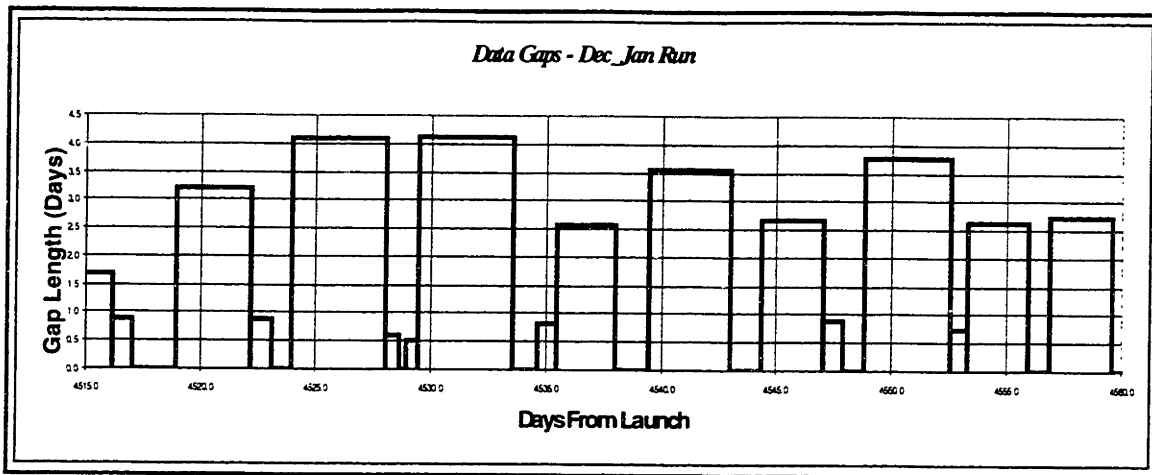


Figure 7.3: December – January 1995 – Data Gaps Plot

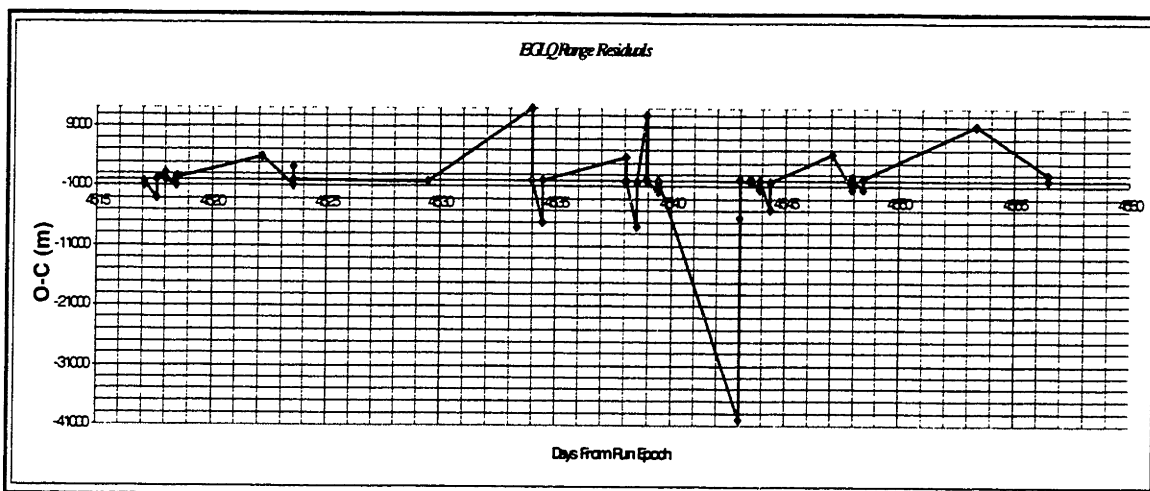


Figure 7.4: December – January 1995 – EGLQ SMA Residuals

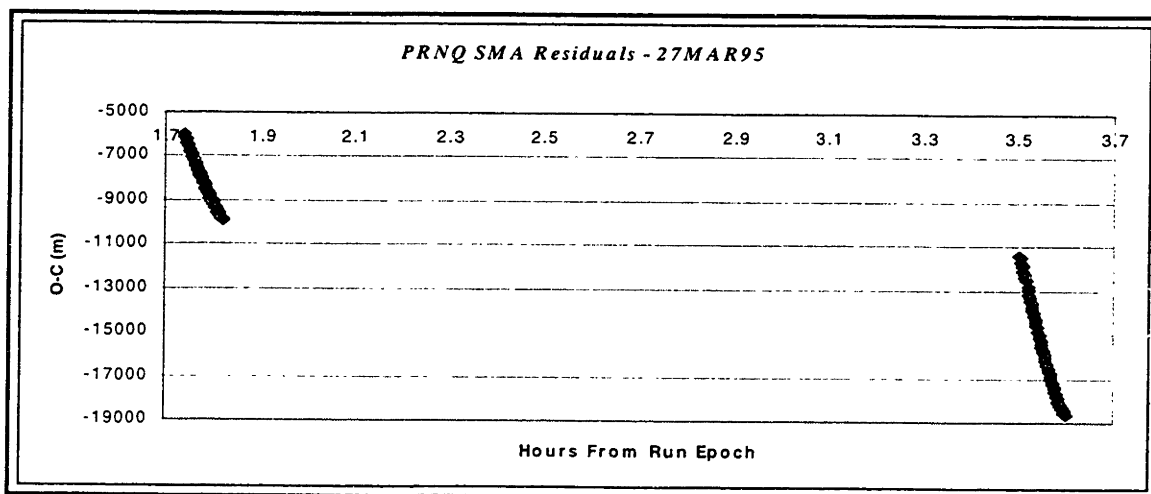


Figure 7.5: 27 March 1995 – PRNQ SMA Residuals

A final analysis variable, used primarily in conjunction with the linearized filter, was a process defined here as mean anomaly stepping. A major problem with stepping through the final months was initializing runs with estimates that were simply not close enough to truth to successfully start the filter. Whereas most of the elements encounter typical, smooth decays over the relatively short time spans in question, mean anomaly was a blaring exception. Figure 7.6 shows a plot of the NORAD Element Set approximation of $\frac{\ddot{n}}{6}$ to indicate the highly volatile nature of the mean motion acceleration (thus mean anomaly) in this region. Figure 7.7 reaffirms this conclusion by showing how the mean motion velocity is also highly volatile. Although the semi-major axis is directly related to the mean motion, corrections to SMA were not necessary in conjunction with the mean anomaly stepping because of the relatively minor effects on SMA.

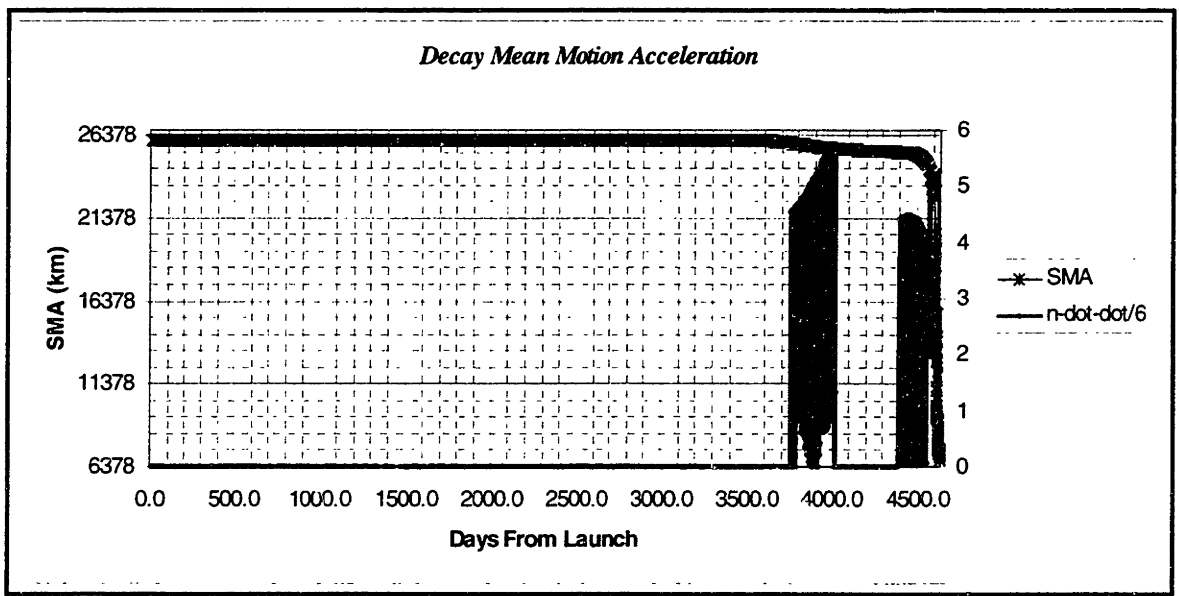


Figure 7.6: Decay Mean Motion Acceleration

This dramatic acceleration during both the entry to free-molecular and continuum flow, led to the conclusion that if one element were to blame for inaccurate ephemerides, it would be the mean anomaly. Accordingly, run a-priori mean anomaly values were stepped to fit the first few observations of the fit span and initialize the filter with a

decent estimate. Once initialized, the C_D was then adjusted to match the remaining data, that estimate was propagated over the data gaps, and the mean anomaly stepping process began again. This algorithm for stepping through the most dynamic region of the decay was extremely efficient and successful, the results of which are given in the following section.

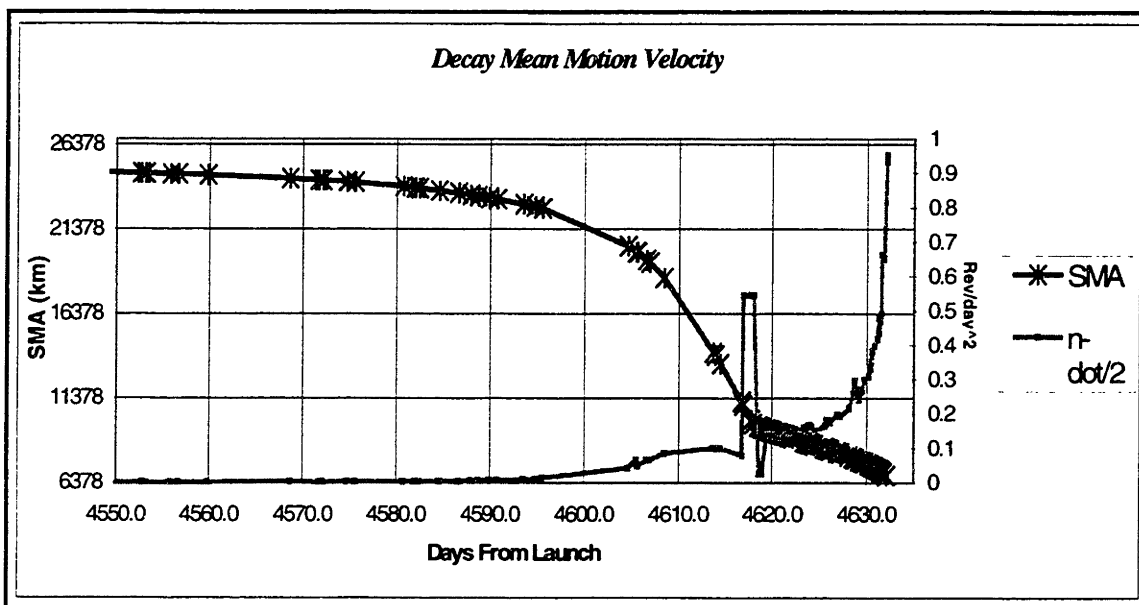


Figure 7.7: Decay Mean Motion Velocity

7.3.2 Runs Summary

Tables 7.3, 7.4, and 7.5 describe all the runs used to track 13390 to final decay, as well as all options and special considerations made with those runs. The reader should note that ephemeris generation runs were used to step across February and other data gaps in March where the frequent gaps coupled with perturbations made obtaining an acceptable fit very difficult. The reader should also note that the SST runs ended in February when the extended filter began to have troubles processing the highly variant data and a Cowell linearized filter (GTDS LKF) was used instead.

Table 7.3: DSST – Extended Filter Run Descriptions

<i>Run Name</i>	<i>June-July2</i>	<i>Aug-Sep</i>	<i>Oct-Nov</i>	<i>Dec-Jan</i>
<i>Period Start</i>	1-JUN-94	31-JUL-94	1-OCT-94	29-NOV-94
<i>Period Stop</i>	31-JUL-94	1-OCT-94	29-NOV-94	15-JAN-95
<i>KXYZ/KXYZDOT</i>	500, 1	1000, 50	1, 0.1	1125, 6
<i>Memory (sec)</i>	172800	160000	259200	864000
<i>Obs. Available</i>	1210	1045	1025	719
<i>% Obs. Accepted</i>	91	71	91	99
<i>SMA Covariance (m)</i>	1.91	0.28	8.44	209.00
<i>A-priori C_D</i>	2.0	2.0	1.0	1.0

Table 7.4: Cowell – Linearized Filter Run Intervals

<i>Name</i>	<i>Start</i>	<i>HHMMSS</i>	<i>Stop</i>	<i>HHMMSS</i>	<i>C_D</i>
<i>Dj_eph</i>	30JAN95	000000	18FEB95	115800	Dec-Jan
<i>Feb18</i>	18FEB95	115800	19FEB95	160000	0.055
<i>Feb_ephc</i>	19FEB95	160000	2MAR95	150000	2.000
<i>Mar_ccow</i>	2MAR95	150000	3MAR95	123000	1.925
<i>Mar3_eph</i>	3MAR95	123000	8MAR95	190000	1.000
<i>Mar3_cow</i>	8MAR95	190000	8MAR95	200000	1.000
<i>Mar4_eph</i>	8MAR95	200000	13MAR95	140000	0.950
<i>Turn1</i>	13MAR95	140000	13MAR95	200000	0.480
<i>Turn1eph</i>	13MAR95	200000	14MAR95	142500	0.480
<i>Turn2</i>	14MAR95	142500	14MAR95	171700	0.334

<i>Turn2eph</i>	14MAR95	171700	14MAR95	231000	0.334
<i>Turn3</i>	14MAR95	231000	16MAR95	055100	0.2535
<i>Turn4</i>	16MAR95	055100	16MAR95	205800	0.2292
<i>Turn5</i>	16MAR95	205800	18MAR95	000000	0.2181
<i>Turn6</i>	18MAR95	000000	18MAR95	100400	0.2466
<i>Turn6eph</i>	18MAR95	100400	18MAR95	165500	0.2466
<i>Turn7</i>	18MAR95	165500	20MAR95	182200	0.2060
<i>Turn8</i>	20MAR95	182200	21MAR95	070300	0.1836
<i>Turn8eph</i>	21MAR95	070300	23MAR95	232800	0.1836
<i>Turn9</i>	23MAR95	232800	24MAR95	154436	0.2673
<i>Turn9eph</i>	24MAR95	154436	25MAR95	001900	0.2673
<i>Turn10</i>	25MAR95	001900	25MAR95	201600	0.4099
<i>Turn11</i>	25MAR95	201600	26MAR95	223451	0.40036
<i>Turn12</i>	26MAR95	223451	27MAR95	034447	0.3630
<i>End_Eph</i>	27MAR95	034447	28MAR95	0850 GMT	0.3630

Table 7.5: Cowell – Linearized Filter Run Variables

<i>Name</i>	<i># Obs</i>	<i>% Used</i>	<i>MA</i>	<i>Adj-MA</i>	<i>KXYZ</i>	<i>KXYZDOT</i>
<i>Feb18</i>	176	80	42.564	42.000	0.000	0.000
<i>Mar_ccow</i>	141	91	204.699	330.500	0.000	0.000
<i>Mar3_cow</i>	60	100	158.208	279.500	0.001	0.000
<i>Turn1</i>	60	100	352.470	220.430	0.000	0.0005
<i>Turn2</i>	637	92	260.897	269.330	0.000	0.010
<i>Turn3</i>	4816	84	38.338	38.230	0.000	0.001

<i>Turn4</i>	3082	93	67.617	67.430	0.000	0.010
<i>Turn5</i>	7054	96	114.367	114.060	0.000	0.010
<i>Turn6</i>	1117	95	194.903	195.080	0.000	0.010
<i>Turn7</i>	5200	99	279.150	277.320	1.000	1.000
<i>Turn8</i>	1053	92	303.657	299.080	0.000	0.100
<i>Turn9</i>	2636	93	85.642	109.49	1.000	1.000
<i>Turn10</i>	3577	100	118.460	132.997	1.000	1.000
<i>Turn11</i>	6965	89	106.605	106.570	0.000	0.100
<i>Turn12</i>	1791	94	159.583	159.520	0.001	0.001

Figure 7.8 is a graphical representation of the above runs, and the stratification between extended filter runs, extended filter runs with a degraded gravity model, linearized filter runs, and ephemeris generations across data gaps. This figure is intended to give the reader a better perspective of the distance between data gaps and where fits actually took place.

Figures 7.9 and 7.10 show how the estimated time of decay improves after each run. Figure 7.9 shows the tail end of the SST run estimates in the January time frame, as well as the Cowell runs through decay. Figure 7.10 is a blow-up of the final two weeks in March. The reader should note that the ‘skipping’ anomaly identified in Figure 6.16 has a significant impact on impact predictions. If the filter had only been able to process observations into February, it would not have accounted for the anomaly and decay predictions would have been far too early (as the ones obtained with the batch fit in Table 7.2 were).

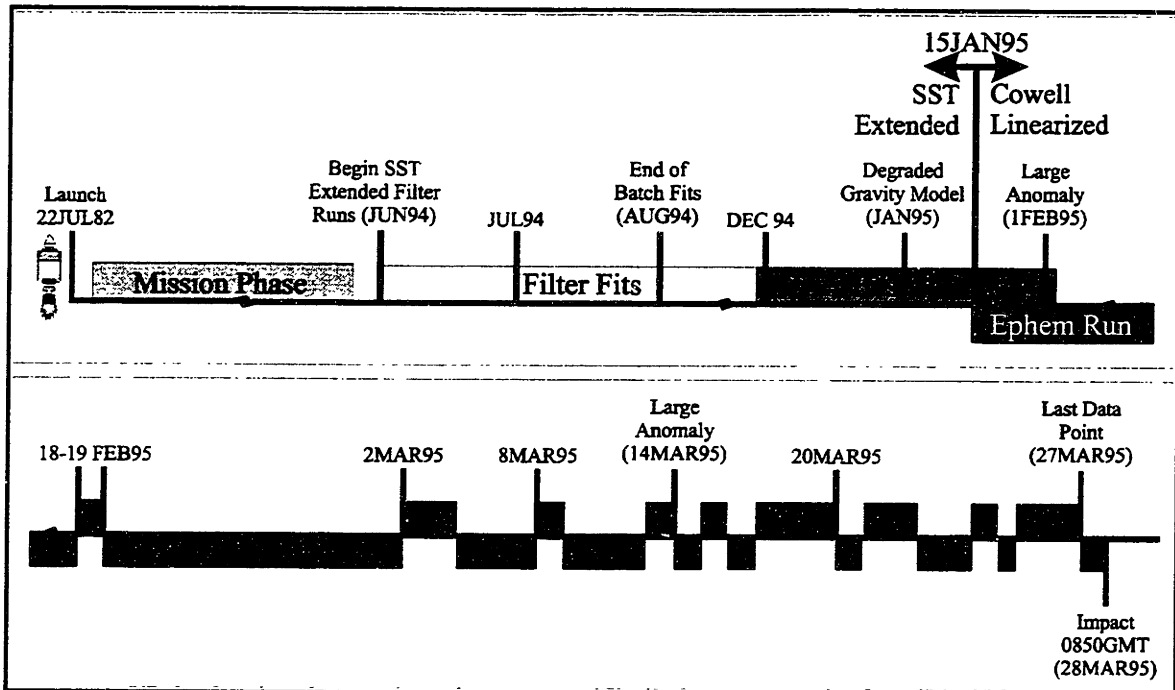


Figure 7.8: NSSC-13390 Decay Processing Diagram

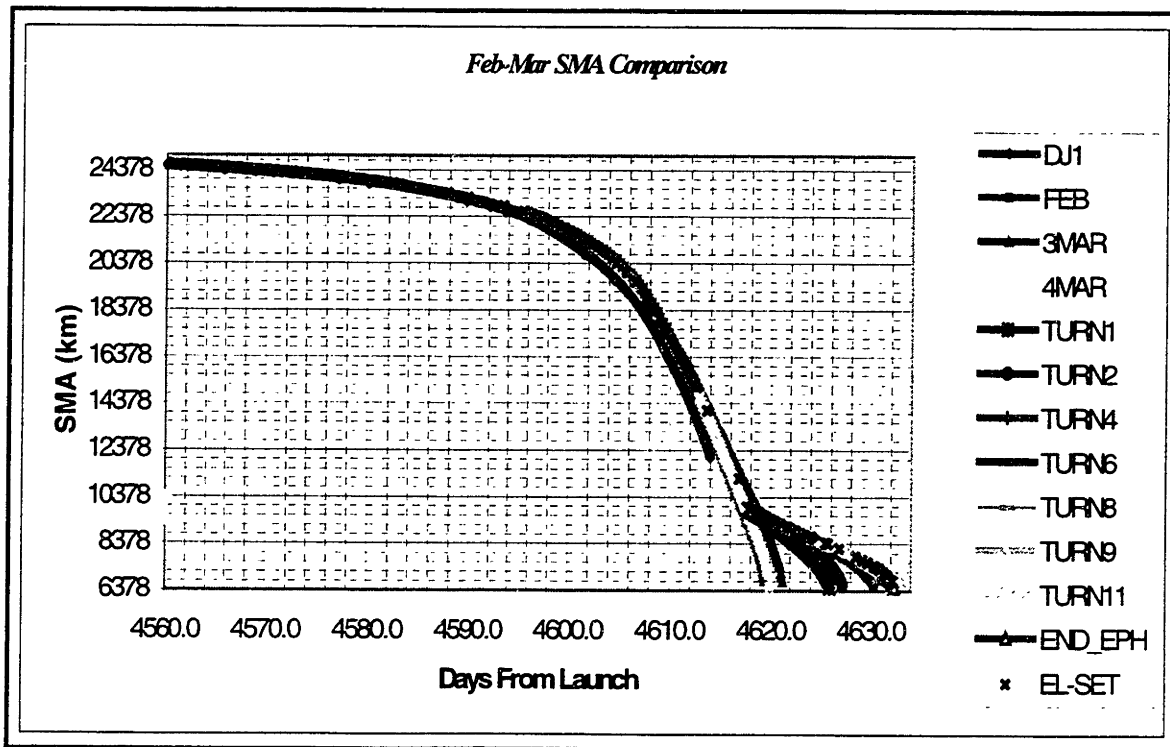


Figure 7.9: Cowell Epheris Generations: January - March 1995

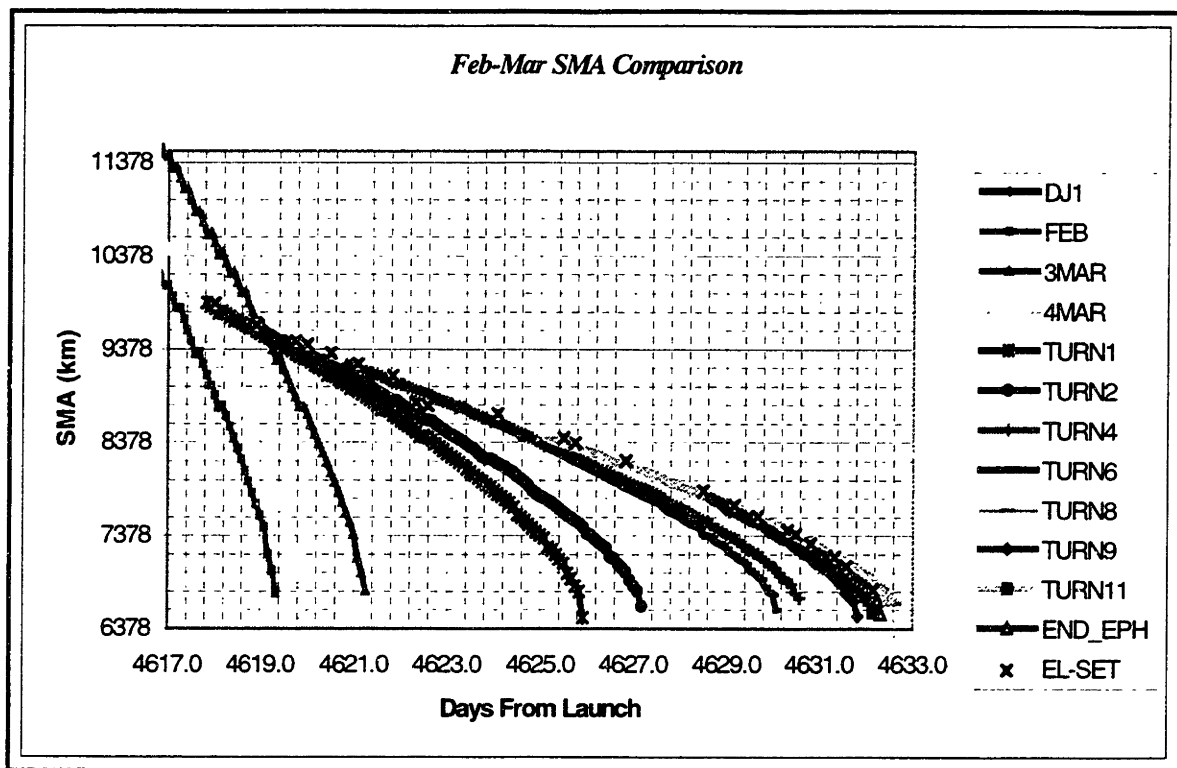


Figure 7.10: Cowell Ephemeris Generations: March 1995

The anomaly was initially thought to be caused by a combination of atmospheric winds and atmospheric lift. With a slowly tumbling object, its attitude would remain relatively constant for the perigee passage. If the rocket body passed through perigee at the critical angle of attack (maximum lift) and was boosted with a large gust of wind, it could produce an anomaly such as this. Unfortunately, anomalies like this are stochastic in nature and very difficult to model. In addition, as was shown in Chapter 6, this anomaly is a predictable event and is now believed to be caused by hypersonic flow effects and the entry into continuum flow.

To alleviate concerns that the anomaly was caused by a geomagnetic storm, the reader can reference Figures 7.11, 7.12, and 7.13 which are plots of the NOAA hourly K_p sum, Daily A_p values, and Daily $F_{10.7}$ indices respectively for the January to March time span, 1995. The reader should note that no significant activity occurs within this region.

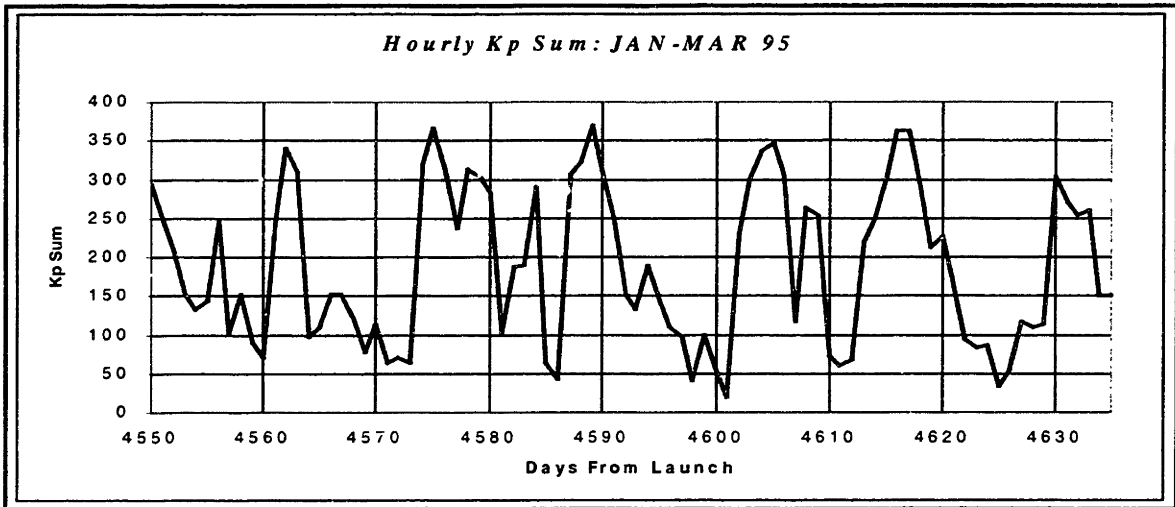


Figure 7.11: January-March 1995, Hourly K_p Sums

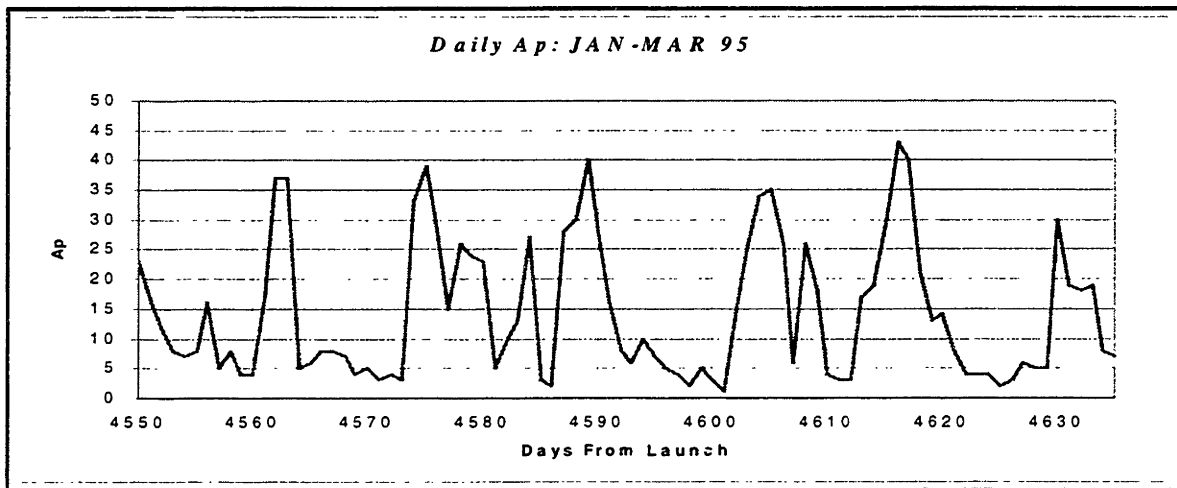


Figure 7.12: January-March 1995, Daily A_p Values

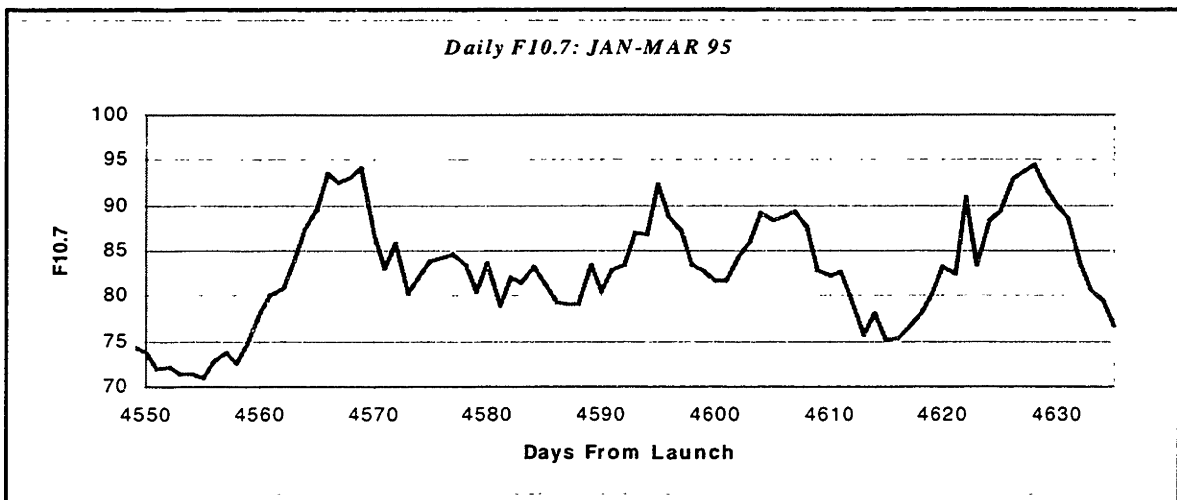


Figure 7.13: January-March 1995, Daily $F_{10.7}$ Values

7.3.3 Impact Prediction

As a low priority object, the exact time and location of impact for NSSC-13390 are not easily obtainable. As such, a direct comparison of actual and predicted impact locations and times is not possible. In light of this constraint, a brief comparison of impact predictions was accomplished using the Satellite Toolkit Lifetime analysis tool. A lifetime prediction was completed for three sets of orbital elements, including:

- **NORAD:** The last NORAD two line element set for NSSC-13390. The reader should note that on Figure 7.14, the label for the NORAD impact point is off the right side of the figure.
- **Lifetime:** The element set at the end of the last filter converged fit span which included the last observational data point for NSSC-13390 on 27 March, 1995 at 3:44:47.00 GMT.
- **NT-GTDS:** An element set taken from the end of the last ephemeris generation run (End_Eph), whose impact point matches that given by the highly accurate Cowell propagator in the EPHEM program.

Each run used the drag coefficient from the TURN12 run, a value of 0.363, and a 48 point Gaussian quadrature for the Lifetime calculations. The results of these runs are given in Table 7.6, where the latitude and longitude of impact are compared (using the NORAD estimate as a comparison).

Figure 7.14 represents the final orbit for each of these predictions, as presented by Satellite Toolkit. The reader should note that NT-GTDS impact prediction is essentially along the same ground track as the NORAD estimate. A mere three hours separate final impact estimates (28MAR95 08:51:00 GMT – NT-GTDS, and 28MAR95 12:09:35 –

NORAD) , but considering this discrepancy resulted in a nearly 20,000 km error, this error is unacceptable and highlights the need for further studies to improve the modeling of catastrophic decays. In comparison, the Lifetime run (which used the Lifetime program with the last converged filter run), was more than a day off in it's estimate, and nowhere near the orbital path.

Table 7.6: Impact Prediction Comparisons

<i>Name</i>	<i>Latitude</i>	<i>Longitude</i>	<i>Position Difference</i>	<i>Impact Time</i>
<i>NORAD</i>	60.884°	177.890°		1200, 28MAR95
<i>Lifetime</i>	60.652°	106.991°	3,659.870 km	1000, 27MAR95
<i>NT-GTDS</i>	-56.202°	41.459°	17,501.389 km	0900, 28MAR95

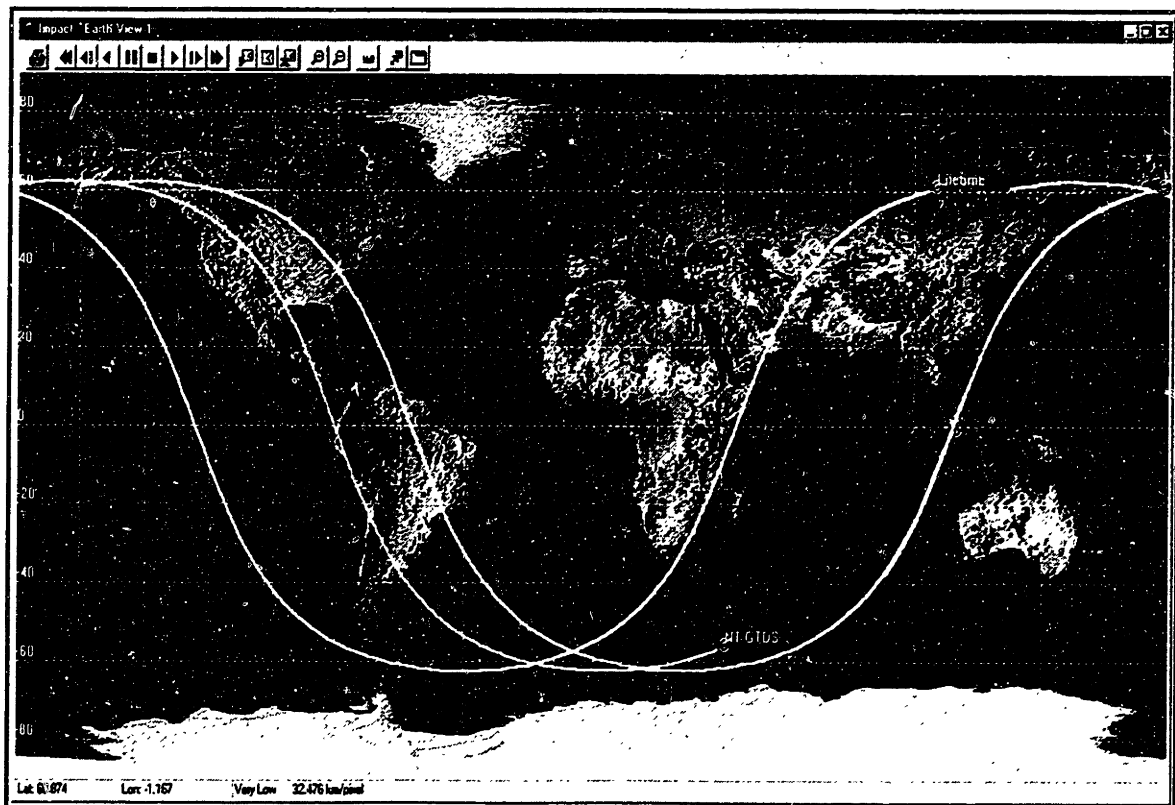


Figure 7.14: Impact Prediction Ground Tracks

7.4 Recent Filter Modifications

In the last two months, two major modifications were incorporated into NT-GTDS in order to improve the performance of the orbit determination process for catastrophic decay cases. The incorporation of the Altitude Dependent Error Function (ADEF) and the Simplified Lift Model (SLM) have shown promising results. While the SLM is not fully developed and tested, both modifications indicate the directions that future investigations into this orbit class should pursue.

7.4.1 Altitude Dependent Error Function (ADEF)

The time dependent error function was specifically designed for the 13390 rocket body. This design made use of the error function constrained to 13390 alone, and required a redevelopment of the Gaussian sum for the error function to be used with any other object. Figures 7.15 and 7.16 identify another weakness in the time dependent function by identifying how the error of the Gaussian approximation becomes larger as the object nears final decay. This error causes the error function to artificially inflate as the object decays and needlessly inflates the covariance.

Due to these errors the ADEF was developed to improve the NT-GTDS filter error modeling techniques for a broad range of orbit types. By modeling the errors attributable to specified altitude zones with a smoothed step function, the ADEF is highly adaptable to any orbit type, and very simple to modify. In short, it is superior to the time dependent error function in nearly every aspect and should certainly be used in any further investigations with NT-GTDS.

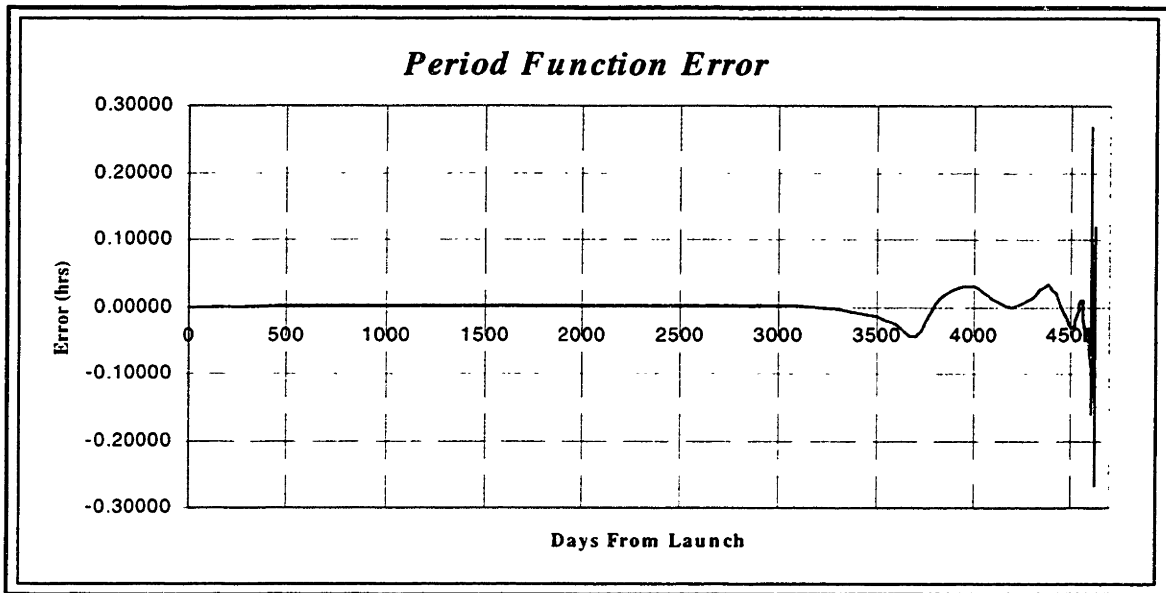


Figure 7.15: Time Dependent Gaussian Approximation Error

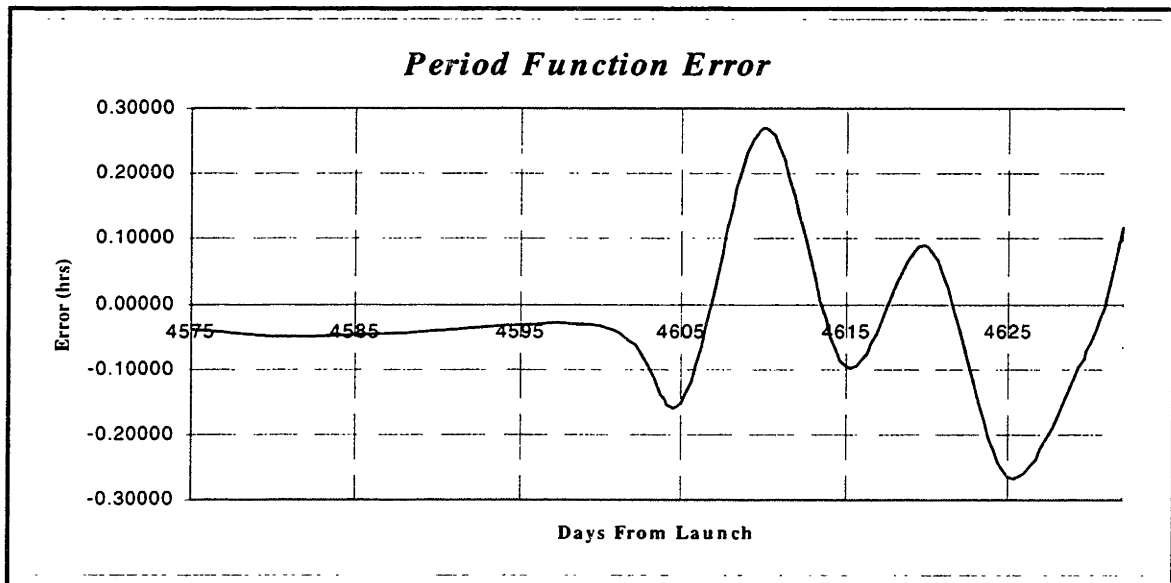


Figure 7.16: Time Dependent Gaussian Approximation Error – Close Up

Time constraints limited the testing that was accomplished with the ADEF, but every run in the last two months of 13390's decay was recreated to insure that the ADEF would produce comparable results to the time dependent function. The results were very successful and, as shown in Table 7.7, consistently produced similar or better results than the time dependent error function.

Table 7.7: ADEF Comparison Data

<i>Name</i>	Time D. SMA σ (km)	ADEF SMA σ (km)	Zone 6 Altitude (km)	Zone 6 Error Level
<i>Feb18</i>	3.247			
<i>Mar_ccow</i>	1.125			
<i>Mar3_cow</i>	1.509	1.440	5000	0.0008
<i>Turn1</i>	1.008	1.008	5000	0.00205
<i>Turn2</i>	1.211	1.210	11000	0.0012
<i>Turn3</i>	1292.000	1258.000	8000	0.00221
<i>Turn4</i>	3.258	3.270	10000	0.0022
<i>Turn5</i>	0.9609	0.9605	7000	0.00275
<i>Turn6</i>	1.692	1.698	6000	0.0029
<i>Turn7</i>	284.500	277.600	6000	0.00256
<i>Turn8</i>	0.7459	0.7459	6000	0.0025
<i>Turn9</i>	170.80	176.500	5000	0.00252
<i>Turn10</i>	108.500	38.680	4000	0.00025
<i>Turn11</i>	3.349	3.349	4000	0.00247
<i>Turn12</i>	0.6713	0.6781	4000	0.00253

The values in Table 7.7 were created by re-running the runs described in Table 7.4 and holding constant the scaling factors used by the time dependent error function. The residuals of the run were then analyzed to determine an average range of the observations within a given run, and zone six of the ADEF was then extended to this average. Finally, the error level within zone six was then adjusted so that the filter would accept the exact

same number and percentage of observations that it did with the time dependent error function. The reader should note that generally, the ADEF could be adjusted to accept more observations and lower the covariance further, but for the sake of comparison, observation statistics were matched exactly. In summary, Table 7.7 shows that the ADEF can be successfully used in place of the time dependent error function in future studies using NT-GTDS.

7.4.2 Simplified Lift Model (SLM)

In general, the coefficient of drag should be approximately 1.0 within continuum flow [51]. In Table 7.4, however, the drag coefficients used to match the data were as low as 0.055. The high variability evident in Figure 7.17 shows how the drag coefficient was used to account for un-modeled and anomalous perturbations the rocket body encountered during decay. Unfortunately, since the drag force could only really account for these perturbations by altering force in one direction, a stepping process was necessary so that the drag coefficient could be continually updated to catch un-modeled phenomenon. To avoid the continual restarts, and account for what is believed to be the primary source of un-modeled error for catastrophic decay cases, a simplified lift model was created.

Discussed in Chapter 6, the SLM creates artificial acceleration vectors in the cross plane and normal directions, thus creating a right-handed orthogonal system with the relative velocity vector. This affords the user the ability to model any acceleration in any direction as a combination of these three components. Table 7.8 lists the primary effects each coefficient has on the orbital elements by identifying if the element is adjusted in the same direction (when the coefficient is raised, so is the element) and inversely (when the coefficient is raised, the element is reduced). By listing the primary effects of each

coefficient, the reader should note that all elements, and hence nearly every orbital path, can be modeled with this acceleration triad.

Table 7.8: Primary Effects of Coefficient Alterations

<i>Element</i>	C_D	C_{L-N}	C_{L-W}
a	Inverse	Same	
e	Inverse	Same	
i			Inverse
ω	Inverse		
Ω			Same

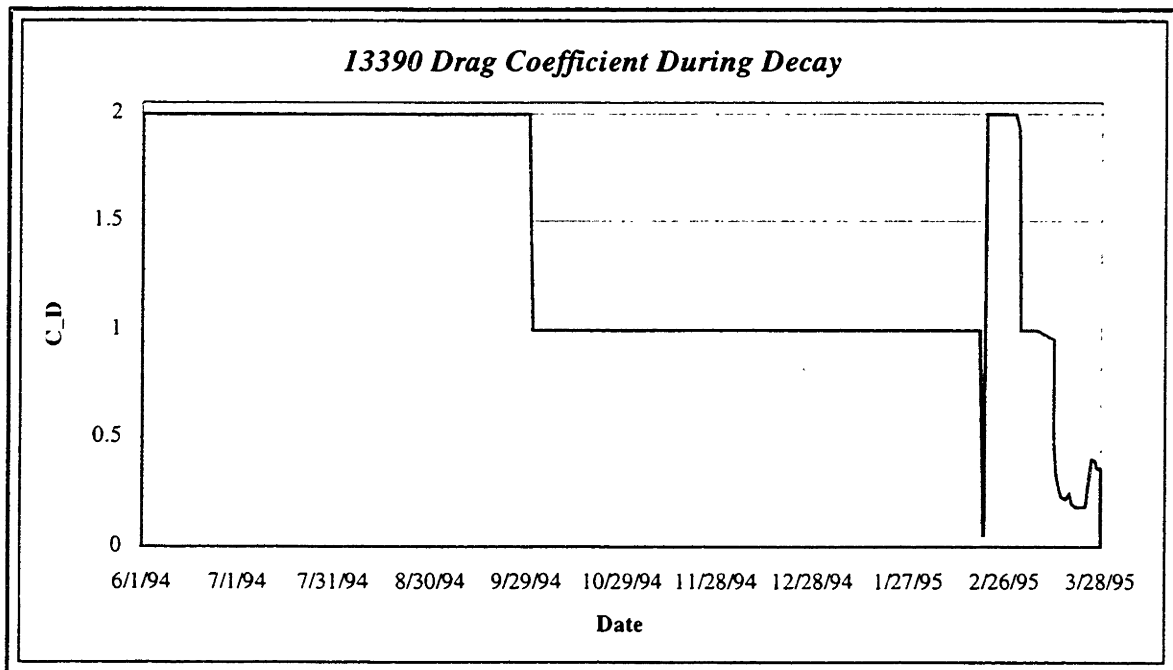


Figure 7.17: NSSC-13390 Drag Coefficient During Final Decay

The results from Table 7.8 were developed while trying to match orbits a simulated test orbit with actual values from the TURN_9EPH run in Table 7.4. The orbit

was manually tuned with the three coefficients to match the above elements. Using the values of 0.85 for C_D , 36.0 for C_{L-N} , and -0.54 for C_{L-W} , the orbits were nearly identical. Considering the logical range for the coefficient of drag in continuum flow is approximately 1.0, the value of 0.85 is much more reasonable than the 0.1863 used to process that time span with drag alone. The ability to match the orbits with physically meaningful values for C_D shows the importance of adding the accelerations of the orthogonal triad created with the lift drag accelerations. It also demonstrated that the lift model is fully functional when not using the solve-for options, but rather adding constant user-defined accelerations.

As was delineated in Chapter 6, there has been a substantial amount of testing done on the lift model to validate both the theory and code. Unit, regression, and PCE testing have all been completed successfully using elements taken from January of 13390's decay (when perigee height is still nearly 400 km and low atmospheric density minimizes the effect of the coefficients). Within this region, the coefficient of drag had a significantly larger impact on the orbit than its lift counterparts, nearly 100 times smaller for the N component, and another order of magnitude for the lift component. This disproportionate effect can lead to the lift effects being 'sucked in' to the drag coefficient if initial estimates are perturbed more than approximately 10% from the truth lift coefficient values, but within this region, the lift solve-for capability functioned quite well. Additional tests were then attempted to verify the solve-for capability using element sets for the end of March (only a week before 13390's decay) and the results have been extremely disappointing.

With the short time span in which this model was developed, coded, and tested, there was little room for error in order to develop any useful results with the SLM. Unfortunately, not enough room and as of now, the lift model solve-for capability is still

not functioning (the lift acceleration model is functional). Within this highly dynamic region, the lift coefficients become more sensitive and the combination of three sensitive solve-for variables causes the filter to produce wildly inaccurate estimates and quickly forces it to diverge. Dick Hujsak, an expert on filter use with orbit determination problems, said of the observed divergence that,

“Historically analysts have pointed to filter divergence as ‘proof’ that filters are unstable. However in recent years we have learned that filter divergence is useful, a symptom of incomplete or erroneous modeling. If a convergent filter suddenly diverges then the events at that time point to a stochastic or physical phenomenon which is incorrectly stated. If the NT-GTDS filter diverges then the filter is probably missing an important stochastic process or model or has incorrectly modeled a significant stochastic process [119].”

At this point, it is undetermined whether the NT-GTDS filter structure, SLM code, or SLM design itself are responsible for the uncontrollable sensitivity created by the SLM solve-for option.

Hujsak has successfully implemented an acceleration model similar to the SLM, with various improvements [119]. With the addition of lift states and lift process noise, Hujsak was able to process through all of the data (with the exception of the continuum flow entry turn) without restarting his filter. He implemented his lift accelerations in an orthonormal triad with the relative velocity vector to use the lift states as a mechanism to express uncertainty in each of the directions.

The most important difference was that Hujsak set very short decorrelation times for the lift states (on the order of 5-10 minutes) due to their high variability and stochastic nature. He also reduced the sigma for the components to be 0.001 and 0.02 of the drag

force for the N and W directions respectively, compared to 0.1 for drag. This differs from initial testing with the SLM in that the relative sensitivities indicated the N direction force was more volatile. Hujsak noticed that the filter actually pushed corrections into the lift states, in essence modeling any un-modeled perturbations.

The reader should note that the above discussion merely shows that a lift state model can be successfully applied to the catastrophic decay problem, not that the aforementioned characteristics are all that is necessary to create a robust filter for these orbits. Kalman Filters have been around for nearly 40 years, and in that time, significant advances in filtering techniques have driven the Kalman Filter to the state of the art. Hujsak, and his mentor Jim Wright, have spent a significant portion of their career developing and refining filters for use with orbit determination problems. As such, to compare Hujsak's filter with that of NT-GTDS is an unfair and incomplete comparison. To reproduce Hujsak's results, the NT-GTDS filter user must not only include an improved lift model, but also refine the filter structure to avoid uncontrollable divergence with highly dynamic cases like the catastrophic decays.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

Chapter 1 compares the complexity of catastrophic decay objects to that of other orbit classes, and concludes that catastrophic decays are among the most difficult to model and propagate. As such, by developing an algorithm to efficiently process observation data for these orbits, a robust system capable of tracking nearly any orbit type could be created. Refined to automatically execute routine operations, such a system could grow into an enhanced automated catalog updating system. Instead of an analyst manually wading through the data to try and find a given orbit, such a system could utilize parallel processing and/or genetic search algorithms to complete the task automatically, saving time and money, and improving efficiency. It is for this reason that this study was undertaken, and it is for this reason that future studies must continue.

Catastrophic decay orbits represent a complex problem which spans the disciplines of astrodynamics, aeronautics, physics, and mathematics. The high eccentricity and steep reentry paths of these orbits amplify model truncations and perigee disturbances to create one of the most highly dynamic perturbative environments of any Earth orbiting satellite. In order to model this enigmatic environment, efficiently reduce observational data for these orbits, and determine accurate reentry trajectories, an analyst must draw upon all the aforementioned disciplines.

While not a complete solution, this analysis has generated a wealth of information about the catastrophic decay problem, and different ways to approach it. It identifies the significant un-modeled perturbations and error sources, characterizes the decay environment, and develops an effective and efficient algorithm for reducing observational data and predicting final decay. To do so, several significant functionality improvements were added to the NT-GTDS operating system including:

- MSISE-90 atmospheric density model
- NT-GTDS filter input processor
- Various filter refinements
- Two functional process noise models
- Simplified lift model
- Several standalone data processing utilities

In order to properly describe the impact of these improvements and the accompanying analyses, the following chapter is stratified into three primary sections: data analysis techniques, mission phase, and decay phase. The data analysis techniques section will cover the data processing and pre-processing. The mission phase section will outline that phase's analyses and transition zone definition. Finally, the decay phase will outline the extensive analysis done on the decay environment and the decay phase NT-GTDS improvements.

8.1.1 Data Analysis Techniques

The pre-processing of the NORAD element sets into GTDS format proved very successful. Using significantly modified versions of previous mainframe utilities, and

creating the Station Location and Accuracy Database (SLAD), the processing had no major problems (refer to Appendix A for a listing of errors encountered). It is important to note, however, that a more comprehensive SLAD should be created which has the dates for which the various sensors were in service and assigned a particular number. Taft DeVere, of the SenCom Corporation, is currently involved in creating such a database, but more work needs to be done. With a more comprehensive listing, the pre-processing utilities (described in Appendix A) could be further improved to automatically choose the correct sensors given the observation time tag.

Attempts to actually process the observational data with a batch filter were relatively unsuccessful. First, the prediction convergence effect delineated in Chapter 7 showed that as the span moves closer to decay, there is more time variation in density which forces an increase of sensor observation noise, decrease in fit span size, and a minimization of short periodic effects. This in turn results in DC elements and ephemeris generations which deviate little from previous predictions. In short, the highly dynamic environment was not well suited for use with a batch least squares technique.

Processing of the observational data for these orbits with Kalman Filters was more successful, however, due to the severe perturbations common to this problem, errors were encountered. In addition to these perturbations, several error sources were also encountered that should be considered in any future work in this area. Resonance shifts, apogee anomalies, sensor switching, range gating, and data gaps are among the most influential of these error sources.

The Molniya orbit begins its life with a strong 2:1 resonance, which goes through a dramatic transformation as the satellite decays. As a result, the filter must be reinitialized periodically to account for the resonance shift or it will fail. For any attempts

to create a filter which can run automatically from cradle to grave, attention must be paid to updating tesseral resonance during the decay of the object.

The two functional process noise models, the time dependent error function and Altitude Dependent Error Function (ADEF), both performed quite well for this investigation. While the ADEF is much more adaptable and refined, both models allowed the filter to process observational data for the catastrophic decay cases. Without these models, processing the observations was not possible because a constant white noise type model exploded covariance in the attempts to capture observations in dynamic regions (such as perigee passage). The ADEF is recommended for use in future analyses to avoid deriving the necessary Gaussian sum for use with the time dependent model, and rather to simply define altitude dependent error zones.

As discussed earlier, the time dependent error function and ADEF are ill-suited to handle apogee centered perturbations. As a result, events such as solar storms require the user to utilize a more simplified model for process noise, which uses a constant value throughout the orbital period. For a detailed explanation of the process noise rate method and an in-depth algorithm for developing rate estimates, please refer to Taylor [65]. This method was utilized to go through the first few days of February, but due to the resulting large covariance, it is not recommend for routine use with this orbit type.

Another limitation on the analysis resulted from the station number constraint in GTDS. GTDS has a limit of 20 on the number of stations it will accept during one run. Considering 13390's decay includes data from over 70 stations, this poses another limitation as to the maximum length a run can be. Eventually the significant code modifications necessary to alleviate this problem will be addressed, but for now it will remain a limitation.

An error was also encountered with the GTDS range-gating feature. Designed to resolve ambiguities in observational data, range-gating had a significant impact on data reduction during the final month by improperly adjusting observations. The range-gating feature was accordingly turned off for the final month runs to correct the problem, and the RANGAT card was incorporated to control this feature in the future. Please refer to Appendix C for a detailed description of the RANGAT card. The reader may also reference Cefola's memorandum on range gating for observations in GTDS [120] for a more complete review of the range gating problem.

Perhaps the most significant sources of error, and of difficulty, were the numerous data gaps present throughout the data. During the mission phase, these gaps are almost inconsequential due to relatively stable behavior, but during final decay, where the orbit's behavior is highly dynamic, they can be fatal to the filter. Figure 7.2 is a graphical representation of the data gaps during the month of February (1 month before decay) which was by far the most difficult month to process through due to the large and frequent data gaps present. Suggestions to alleviate the problem will be included in the future work section.

Significant attention must be paid to the location of these data gaps with respect to the filter runs and ephemeris generations. By attempting to center filter runs over the gaps, or to simply do ephemeris runs across them, the 13390 rocket body was successfully tracked to decay. The algorithm for processing this data is comprised of generalized SST filter runs of one to two months until the gravity model is simplified to average errors. At that point, an ephemeris can be used to span data gaps using the solve-for drag coefficient from the previous run. The final element set of the EPHEM run can then be adjusted (mean anomaly) to match the initial observations of the next filter run, followed by adjusting the a-priori drag coefficient to match the data with the smallest

residuals possible. This element set can then be used in the next EPHEM run in order to continue the process through decay.

Finally, several small code modifications were made to the following routines to correct for small bugs in NT-GTDS Filter code, all of which are described in Appendix C.

- INTEGBD
- STAGE1BD
- RESTAT
- SECHEK
- ELEME
- RPTRES
- RANGAT
- SETESKF
- AERO

8.1.2 Mission Phase

Within the mission phase, most perturbations relevant to the environment were outlined by Fieger [19]. Fieger also showed how the unique characteristics of DSST were ideally suited for long term predictions for highly eccentric orbits. The fundamental perturbation which was not considered in Fieger's work was the truncation of higher order oblateness terms, most notably J_2^2 .

Since Kozai [103] pointed out that the main problem in the theory of artificial satellite movement lies in the determination of the effects due to the oblateness J_2 of the Earth, to the order 2 in this parameter, there have been many attempts to solve this problem. Some have met with success, most notably Metris [97]-[100], but none have developed the necessary closed-form analytical expressions in equinoctial elements,

which are necessary for use with DSST. Two methods are investigated in this work, including solving for the F functions directly, and grouping terms in a Fourier series. Both techniques exceeded the limitations of the current version of the Macsyma symbolic algebra program and refined techniques, or more powerful computers, need to be used if this problem is to be solved for use with DSST.

During the analysis, a definition of the transition from the mission or operational phase to the decay phase was obtained. It was found that the selected objects, with the exception of 14199 (which encounters an anomaly in 1988), behave normally while their mean motion remains at 2 ± 0.1 revolutions per day. At this point the eccentricity, which normally is about 0.72 ± 0.03 , begins a steady, and sometimes rapid, decay. The beginning of the decay phase (thus end of the transition zone), is defined as that portion of the orbit when perigee height drops below 250 km. It is at this point that the drag short periodics and drag-oblateness coupling effects must be taken into account. It also marks the beginning of the portion of most rapid decay during reentry and therefore necessitates the stepping or sequential filtering techniques to be used.

8.1.3 Decay Phase

Within the Decay Phase, the implementation of the MSISE-90 model was successful and demonstrated the black box concept for adding atmospheric density models to the NT-GTDS AERO sub-structure. By simply creating an interface to give the correct inputs (and retrieve the atmospheric density) to the model's main driver program, the model was successfully integrated. Once integrated, the model performance was consistent with standard behavior described by Hedin in his MSIS-86 paper [111], as well as the test runs supplied with the MSISE-90 model on the Internet. A common MSIS-

Jacchia data file was also investigated and implemented with the MSIS files. Slight modifications are still needed in order to make the JR-MSIS files useable for both density models.

Once the orbit was propagated into the decay phase (perigee height below 250 km), an analysis was accomplished on the effects of J_2 -drag coupling and drag short periodics on the orbit. It was found that when using the drag short periodics, it is necessary to expand the Fourier expansion in true longitude to avoid the errors associated with truncating an infinite series in mean or eccentric longitude. The two optimal HODRAG configurations presented in Chapter 6 represent the most efficiency and accuracy of the options for these orbits. This is due to fact that both options represent very good approximations of the complete second order/oblateness drag effects with only three numerical quadratures.

The anomaly encountered on 14 March, 1995, coupled with the extremely low coefficients of drag observed in the last two months of the orbit lifetime point to the fact that atmospheric lift is indeed a factor in this problem. Assuming that a concave rocket body would randomly tumble and produce a zero net lift was shown to be inaccurate, and this assumption has a significant impact on orbit predictions. Unfortunately, without attitude data on the rocket body, a complete analysis of the lift effect cannot be accomplished, and will require future investigators to resort to a simplified process noise model. This is due to the fact that the perturbations represent a stochastic process, and must either be modeled in a probabilistic fashion or purely as process noise.

The Simplified Lift Model (SLM) represents a simple method for modeling the effects of lift (and any un-modeled perturbations) for these orbits. The SLM acceleration model is operational and allows the user great control in morphing fit orbits to match observational data (thus solving for various coefficients manually). Unfortunately, as was

discussed in-depth in Chapter 7, the SLM solve-for capability is currently not functioning. Hujsak has shown that a model such as this can significantly improve data processing when properly implemented [119], and it is for that reason that the SLM should be the first area of concentration for any future studies into catastrophic decays.

8.2 Future Work

As stated, catastrophic decay objects do not represent a simple orbit determination problem, and much work remains to be done. The following section attempts to address some ideas that have been developed and encountered during this investigation, which might facilitate future progress in the study of these orbits.

8.2.1 Data Analysis Techniques

First and foremost, an effort needs to be made to find a better test objects to do this analysis on. Molniya objects are plentiful, but a more complete analysis needs to be done on an object with fewer data gaps and attitudinal data, all of which is not readily accessible for Molniya objects. An experiment should be conducted where a reentering rocket body (or other object) in a highly eccentric orbit can be saturated with SSN tasking. With a dense and distributed set of observations, the decay environment could be characterized, and as a result, better modeled.

In addition, the mass distribution and physical dimensions of the Molniya objects is at best an educated guess. Without this data, modeling lift effectively, will not be possible. Studies have been done with convex objects before, such as Karr's ([104], [105]) exhaustive research into the area of atmospheric lift. If an object could be identified, with attitudinal data available, Karr's equations could be utilized as a starting

block for the analysis. Without attitude data, however, any analysis would be superfluous. Additionally, an object should be studied whose exact impact time and coordinates are known. This will afford the investigator an important metric with which to measure the accuracy of their data processing algorithm.

Improvements in the error function also represent a significant opportunity for advancing this problem. Hujsak [119] has suggested using a stochastic functional form for the error function, but has warned of the special integration rules that would govern the solution (Ito Stochastic Calculus [121]). Through general calculus it is easily shown how eccentricity, mean motion, and semi-major axis are related and, using this formulation as a basis, it may be possible to create a stochastic model. If the model can be twice differentiated it may be possible to use acceleration error statistics (created from orbit to orbit fit comparisons) to fine-tune the model.

Capitalizing on the unique characteristics of equinoctial elements and DSST demonstrated by Fieger for long arc orbit determination of highly eccentric orbits, another alternative to advancing this problem involves developing a standalone filter program. While NT-GTDS does represent a highly functional and developed environment within which to operate many different types of analyses, its complexity can also make integration of a new model (such as the SLM) very difficult. Developing a new filter would require significant time and effort, but would allow the developer an opportunity to incorporate state of the art filtering techniques such as a dynamic state space or more complex noise models. A modularized design would allow future students to enhance and refine the filter without having to worry about how it effects the other NT-GTDS programs such as the DC, EPHEM, DATASIM, etc. For example, using Taylor's [65] semianalytical filters as a basis, the standalone could then be optimized for use with the DSST standalone.

Reasoning for this diversification is found in the basic nature of filters. Whereas a data analysis technique such as least squares has seen little development in recent years, work with filters is a continual process across many different disciplines and applications. Almost every technical journal published has a new article on how to improve a specific variant of a filter, and it is that continual research that makes it advantageous to develop a standalone filter so that advances may be efficiently incorporated into the code. Development of such a standalone will no doubt involve the use of a large number of NT-GTDS routines, modified and enhanced for specific filter use. If the PC is chosen as a development platform, thought should be given to converting the GTDS routines, and new standalone code, to Fortran 90 or an object oriented language such as C++. Beyond the obvious programming advantages, and 32-bit processing capability, these languages would offer a much better debugging environment to work in, which is by far the most significant limitation of the Lahey Fortran 77 environment currently being used. In addition, these languages could be incorporated into a Matlab or Macsyma environment to take advantage of existing mathematical tools and graphing utilities. Along the same lines, a filter smoother could easily be incorporated to improve data processing and provide a reference orbit for cases where there is no truth to compare results to. The smoother can also be used with a series of statistical tests (between it and the filter) to improve the filter modeling and tuning [119].

During this investigation, several implementation ideas have been developed for use with either NT-GTDS or a future standalone filter. One possibility, is a 'smart' filter which would separate the orbit lifetime into increasingly smaller intervals. The filter driver program could run the secondary filter (after optimizing the mean anomaly, drag coefficient, etc.) and analyze the residuals. After developing a sufficient stochastic model to be used with the driver, it could then analyze the conditional probability of an anomaly having occurred within the interval based on the residuals. This would trigger the driver

program to adjust input variables accordingly, or even solve for an impulse to account for the perturbation. Using the SLM, or a similar model, the filter could match state elements with high precision by implementing a search algorithm (e.g. genetic algorithm) to optimize lift and drag coefficients. Though not physically meaningful, these coefficients would allow the filter to account for any un-modeled perturbations including significant anomalies such as micrometeorite impacts or un-planned maneuvers. A simpler option would be to create another driver program which utilizes the existing genetic algorithm software (currently operational at Draper Lab in an MPI multi-tasking environment, and publicly available on the web), in a sliding batch scheme to create an orbital ephemeris. By creating an optimization function, such as a weighted six-dimensional (to account for the orbital elements) representation of observational data points, the genetic algorithm can reduce the observational data within a given interval to one point. Sliding down the orbit with several passes, the filter can increase interval size and eventually smooth the points to create a smooth orbital representation. This option would offer the benefit of not crashing regardless of data quality or data gaps, but would also be an averaged representation and perhaps not very accurate.

8.2.2 Mission Phase

. During the mission phase, a detailed analysis on the effects of lunar-solar point mass perturbations is warranted due to their dramatic impact on perigee height as presented by King-Hele [18]. There is also some indication that refinements in the solar radiation pressure model and inclusion of an earth albedo pressure model will improve accuracy in this phase. A new albedo model, developed by 2 Lt. Jim Smith for use with the VAX version of GTDS [122], could easily be incorporated to improve albedo pressure modeling.

An obvious area of future work lies in the development of a closed-form, second order, equations of motion in J_2 for use with DSST. The numerous sources listed in Chapter 5 ([32], [42], and [91] through [102]) identify several methods and techniques for developing the necessary formulations, and are encouraged reading for the interested reader. McClain recently developed revised versions of Zeis [32] equations for J_2 e short-periodic variations in the equinoctial elements, which may eventually lead to simplified forms of the J_2^2 expansions, but are currently not forthcoming.

Although not the least bit elegant, an interesting approach to divide and conquer the derivation would be to employ the use of a parallel processing environment such as the Parallel Virtual Machine (PVM) or Message Passing Interface (MPI) software packages. Depending on the method used (explicit development of the F functions, or Fourier series representation for the gravitational potential), a driver block (Macsyma program) could separate blocks of terms and spawn their generation to a slave computer. These components could then be combined and reduced using current Macsyma algorithms on a powerful master computer to develop the necessary equations. Considering the excellent technical support Macsyma has provided throughout this project, there is little doubt that they would be able to refine Macsyma for use with a parallel processing environment, and make such a derivation possible.

8.2.3 Decay Phase

In addition to the processing technique improvements (filter designs, error functions), there are several decay phase anomalies that should be studied in more depth to accurately model this region. The most simple would be an analysis of the HODRAG code to insure that the True Longitude expansions are consistent. This would guarantee

that the True Longitude expansion options chosen in Chapter 6 are in fact the optimal settings.

Another area of research pertinent to the decay phase is continuing upgrades of the atmospheric density model. Mike Picone is currently in charge of an NRL effort to improve the MSISE-90 model with actual atmospheric data [107]. With the black box structure currently in NT-GTDS for atmospheric models, any advances in either the MSISE-90 model or data files, can be easily incorporated into the system and improve performance. The black box structure also allows for the incorporation of more complex atmospheric models such as those of Nazarenko and Lafontaine.

Recently demonstrated by Professor Andrey Nazarenko, adaptive density estimation has shown to be an extremely effective method of improving atmospheric density calculation, specifically in the 500-600 km range. The estimation process includes the following four steps [55], which are demonstrated graphically in Figure 8.1 as the interaction of Nazarenko's atmosphere variation model with the Space Surveillance System (SSS).

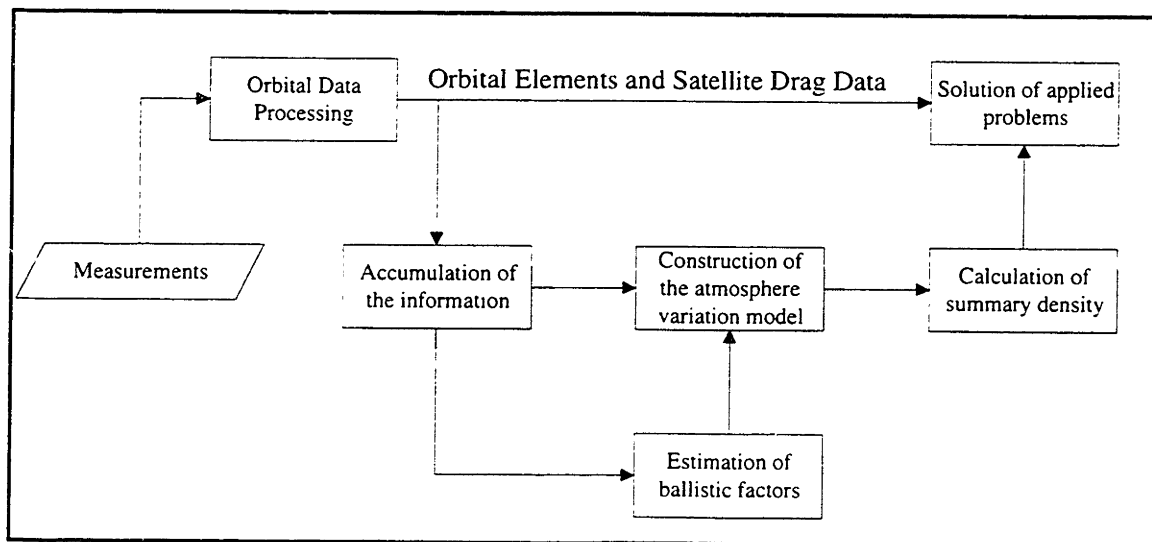


Figure 8.1: The Interaction of the Atmosphere Variation Model with the SSS Algorithms [123]

- *Accumulation of data*
- *Construction of the atmosphere variation model:* the regular (every 2-3 hours) operative updating of variation model parameters.
- *Calculation of summary density:* the calculation of summary atmosphere density in the required space-time domain.
- *Estimation of ballistic factors:* periodic (about monthly) updating of ballistic factors and their variability characteristics for all used objects with a-priori unknown ballistic factors.

Jean de Lafontaine, in his doctoral thesis entitled “Orbital Dynamics in a Stochastic Atmosphere and a Non-spherical Gravity Field,” develops an algorithm for analyzing the orbital dynamics of a near-Earth satellite subjected to the random perturbations of a stochastic atmosphere [124]. He uses a random process to represent the short-term density fluctuations and geopotential errors. The Method of Averaging is then used to simplify expansions of the corresponding deterministic density function into a stochastic Taylor series. This process maintains non-linearity of the equations of deterministic motion and avoids the typical limitation to small disturbances that constrain linearized formulations. Lafontaine also developed a computer program which utilizes the above expressions to generate confidence intervals that bound, in a probabilistic sense, the trajectory of a satellite [124]. If his utility were integrated into a driver program for a filter, the program could generate necessary process noise bounds using the confidence intervals, and create a filter which could process through large perturbations such as the one observed on 14 May, 1995.

Future research into catastrophic decays should also study the effects of hypersonic shock waves with various satellite configurations, combined with orientation data from future observational taskings, to determine the forces on the object due to lift

and other aerodynamic forces. The reader should also note that this paper was done with the Molniya rocket body as opposed to the Molniya spacecraft, in order to simplify the analysis. The satellite necessitates a much more comprehensive drag model due to its more complex design. Hypersonic shock waves, a stochastic model of solar array life (before being ripped off by the atmosphere), and the increased amount of lift created by the arrays are just a few examples of variables which will make the satellite problem more difficult.

Despite its difficulty, once an effective generalized method is created for dealing with catastrophic decay cases, it could be useful with nearly any orbit type, and should therefore be pursued.

Appendix A

Useful Program Utilities

Data, by itself, is useless. When presented in an effective manner, however, data becomes information, and useful. Appendix A is a compilation of seven utilities which were used as a part of this investigation to wade through a veritable mountain of data and create meaningful information. The NORAD Pre-Processor and Pre-Processor Shell can be found in \$E53-BOB at: BOB\D:\MOLY_DATA\DATA-ANALYSIS\PRE-PROCESSOR. All other utilities can be found at: BOB\D:\PROG_UTILS.

A.1 NORAD Pre-Processor (NPP)

In order to correctly process the observational data for this thesis, it was necessary to create the NORAD Satellite Surveillance Network Tracking Data Pre-Processor (NPP) and convert it to the IBM Compatible PC. The Processor takes the data, in NORAD B3 format, and converts the data to the FRN-15 OBSCARD format used by NT-GTDS. It was necessary to convert this utility from its original version, written for the Draper IBM Mainframe, to a PC compatible version in order to compliment the PC version of GTDS.

The NORAD Pre-Processor was first written around 1980 for the IBM Mainframe system at Draper. The code was written in Fortran 66 by Joe Lombardo and converted by Leo Early to Fortran 77. Early also improved the code structure, streamlined the 'find station' process and created a print file to output the known, unknown, used, and unused observations. In December of 1986, he modified the program to output to standard FRN-

15 format and print unknown station data. Finally, with the addition of the subroutine ASTRON.FOR, he added the ability to process optical observations.

ASTRON.FOR transforms the coordinates of optical observations (right ascension and declination) from the NORAD coordinate system (or others, depending on what is specified in the subroutine call) to the GTDS standard Mean-of-1950.0 coordinate system. Shortly after Early's modifications, in 1987, Marty Fieger used the processor for his thesis entitled "An Evaluation of Semi-Analytical Satellite Theory Against Long Arcs of Real Data for Highly Eccentric Orbits [19]." He formatted data with the processor and analyzed the performance of the Draper Semi-Analytic theory for Molniya orbits with perigee outside the atmosphere.

As a result of this investigation, the NPP was modified to include I/O capabilities, operational structure, and dealt with various conversion problems that will be discussed later. These additions culminated in NORADPP.EXE, a stand-alone pre-processing utility. NORADPP can be altered to print from the desired input file to the desired output file in any time frame, without re-compiling or linking.

A.1.1 Problems Encountered

As the pre-processor evolved from a simplistic IBM mainframe utility to an I/O capable PC utility, several problems were encountered. The problems are listed below in hopes that future efforts to convert Mainframe utilities can avoid making the same mistakes. The problems are stratified into two primary categories, IBM Mainframe Compatibility, and comparison data issues, both are discussed below.

A.1.1.1 IBM 3090 Mainframe Compatibility

There were two primary differences between the pre-processor code on the IBM 3090 and what would run on Fortran 77. The first, relatively minor, problem was the include statements. Include statements on the mainframe use parentheses to indicate an include module, while F77 for the PC uses apostrophes.

- Mainframe: INCLUDE (#OBSDAT)
- PC: INCLUDE 'OBSDAT'

A more substantial problem could be found in the character code differences between the two environments. The main program of NPP is the RUNADCOB.FOR subroutine which calls the ICHAR function (intrinsic in Fortran) to convert an overpunch 11 character code into an integer value. The other subroutines use this value to calculate the output and they were written for the IBM EBCDIC character code system, which the PC does not use. I had to convert these character code calls to their ASCII equivalents in order to make the subroutines work. Basically, the overpunch 11 code is used so that value from 9 to -9 can fit into a single space. Conserving spaces was necessary in the construction of the NORAD B3 format because they wanted to keep the field width to 80 columns for each observation. These numbers represents the tens place holder in the elevation angle and range rate. Table A.1 lists the input character, what it represents, and the corresponding ASCII and EBCDIC values to delineate this problem.

Table A.1: ASCII and EBCDIC Conversion Value Table [125]

<i>Input Character</i>	<i>Actual Value</i>	<i>EBCDIC Character Code</i>	<i>ASCII Character Code</i>
9	9	249	57
8	8	248	56
7	7	247	55
6	6	246	54
5	5	245	53
4	4	244	52
3	3	243	51
2	2	242	50
1	1	241	49
0	0	240	48
space	0	64	32
-	-1	96	45
J	-2	209	74
K	-3	210	75
L	-4	211	76
M	-5	212	77
N	-6	213	78
O	-7	214	79
P	-8	215	80
Q	-9	216	81

In order to simplify the code and remove machine dependency, a new subroutine called NUMCONV.FOR was created. NUMCONV does a direct translation of the overpunch characters and returns to the main program the complete values for the elevation angle and range rate values. Modifications were also necessary for ELEVAT, RANGER, and ASTRON to reflect the NUMCONV changes.

A.1.1.2 Comparison Data

Since there were no NORAD B-3 data files from Fieger's results that could be used for comparison, a truncated Fieger GTDS FRN-15 format file was employed. In comparing the output to the OBSCARD data, the following problems emerged:

- *FORMAT STATEMENTS*: Data obtained from Fieger's MVS directory 'mef2627' (ADC13964.OBS.DATA, renamed ADCOM.DAT) had an F17.2 format field for the output of the time tag in contrast with the F21.6 field used by all of the RUNADCOB.FOR subroutines. The format statements were then changed to produce output in the proper format (the comparison program would not work otherwise) while debugging the data. They have since been changed back to the standard F21.6 number field for the FRN-15 format. Each entry contains 80 bytes in the following format:

- Bytes 1-8: SNAME First Record - 'OBSCARD '
 - Last Record - 'END '
 - All Others - Station Name
- Bytes 9-11: MTYPE - observation type number

- Bytes 12-14: IGATE - range-gating indicator
 - Bytes 15-17: blanks
 - Bytes 18-38: Date of observation (YYMMDDHHMMSS.SSSS)
 - Bytes 39-59: OM1 - Uncorrected observation
 - Bytes 60-80: OM2 - Corrected observation
- STATION CODES: Another problem was the station codes issue. Several of the stations didn't have station codes at all, and others were wrong. Table A.2 lists the GTDS Tracker Type key to assign the new station codes for the stations listed in Table A.3. All stations which contributed to the Molniya data are included.

Table A.2: GTDS Station Tracker Type Key

Value of Last Digit	Tracker Type	Type #
V	GRARR-VHF	1
M	Minitrack	2
T, Q, F	C-Band	3
G	GRARR S-Band	4
3, A, W	USB30	5
8, A, W	USB85	6
X, Y, Z	SRE-VHF	7
R	ATSR	8
B	ATSR Ground Transponder	9
D	DSN	10
S	SRE	11
L	LASER	12
C	Optical	13
E, 4	X-Y Parabolic	14

Table A.3: NT-GTDS Station Name Modifications and Additions

ST #	Old Code	New Code	Description
201	****	CRSC	Cross Optical Site
202	****	EAGC	ETS Site A, Optical
204	****	EBGC	ETS Site B, Optical
206	****	GAGC	GTS Site A, Optical
207	****	GBGC	GTS Site B, Optical
241	****	DAGC	Diego Garcia A, Optical
242	****	DBGC	Diego Garcia B, Optical
243	****	DCGC	Diego Garcia C, Optical
330	****	FLDF	Fylingdales BMEWS
337	DYBF	PRNQ	Princlik, Turkey
344	****	FLYT	Fylingdales Phased Array
345	SHMQ	FLYQ	Fylingdales Phased Array
346	KOKQ	FLYF	Fylingdales Phased Array
355	****	ACSF	Ascension Island
370	****	MILF	Millstone, UHF
398	****	EGLF	Eglin Deep Space Radar
400	****	PRNF	Princlik, Turkey
401	DYBT	PRNT	Princlik, Turkey
952	****	AMSC	Amos, Hawaii - Optical

- **ROUNDING ERROR:** This F17.2 field also presented a problem in the form of a rounding error. The ADCOM.DAT data would always round down when faced with a 5 in the thousandth of a second digit, whereas my output always rounded up. This problem was solved when the data field was expanded to include 6-digit precision for the time tag.

A.1.2 Operation of the NORAD Pre-Processor (NPP)

The NPP is a small, portable utility which can be very useful in not only preparing NORAD B-3 data for use with NT-GTDS, but also in analyzing the observation distribution and composition of the object in question. Figure A.1 is a tree diagram to demonstrate the structure of the NPP where shaded blocks indicate modified Mainframe routines, and black blocks represent new or I/O code.

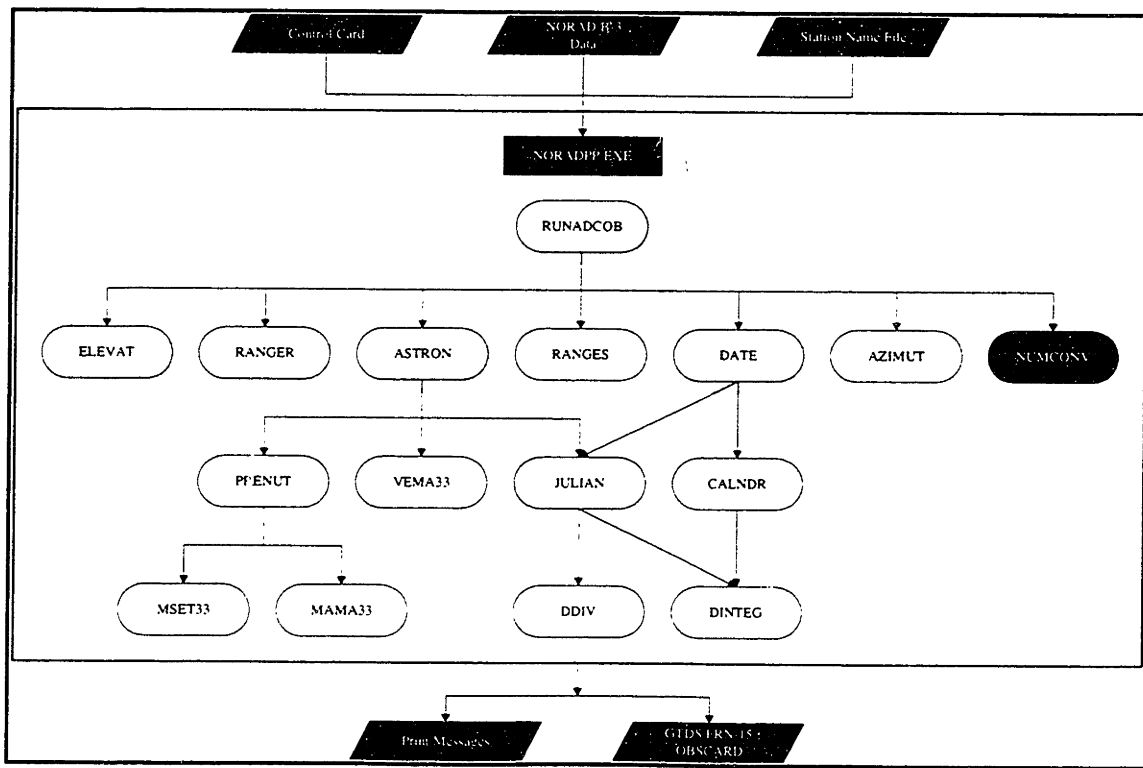


Figure A.1: Tree Structure Diagram of the NORAD Pre-Processor

One of the main features added to this processor was the ability to control the I/O environment with a control card to avoid the need to utilize the makefile every time you need to analyze different data. The control card, shown in Figure A.2, tells the processor which input data file to use, what to name the output file, and when to start and stop reading the data corresponding to the given time tags

SATELLITE DESIGNATOR	:	NSSC13390
START DATE (YYMMDD.)	:	950325.
STOP DATE (YYMMDD.)	:	950327.
NORAD OBS DATA INPUT FILE	:	13390.obs
GTDS OBS DATA OUTPUT FILE	:	13390ed2.OBS
OUTPUT PRINT FILE	:	11390ed2.msg

Figure A.2: NORAD Pre-Processor Input Control Card

The NORAD B-3 data input file will be discussed in Section A.1.3, and not addressed here. The final input file, the station names file, is simply a listing of the four letter GTDS code and corresponding station identification number for each station. Figure A.3 lists the first few stations of this file.

5	RMLC
25	PULC
27	STMC
28	SNVC
29	MTJC
30	EDWC
32	COLC

Figure A.3: Truncated NORAD Pre-Processor Station Names File

There are two output files for the NPP, a print messages file and a GTDS FRN-15 OBSCARD file. The print messages (PRINT.MSG) file contains:

- A listing of all of the names of the stations whose data was used, and not used
- A listing of the numbers and types of data used for each station
- A listing of any data entries omitted for faulty data

The OBSCARD simply lists the station name, time tag, and observation. Figures A.4 and A.5 show examples of the PRINT.MSG and OBSCARD files respectively.

```

SATELLITE DESIGNATOR      : NSSC13390
START DATE (YYMMDD.)     : 820722.
STOP DATE (YYMMDD.)      : 950226.
NORAD OBS DATA INPUT FILE : 13390.obs
GTDS OBS DATA OUTPUT FILE : 13390ST.OBS
OUTPUT PRINT FILE         : 11390ST.msg

```

Number of obs left unchanged
due to unknown reference frames: 0

Number of observations used: 10766

***** Known Stations Used *****

This list includes all known stations in the ADCOM observation file which generated GTDS-usable observations

Station Data			ADCOM TYPE (# Observations)										
Number	Code	Total Obs	0	1	2	3	4	5	5-0	6	7	8	9
27	STMC	70	0	0	0	0	0	70	0	0	0	0	0
28	SNVC	20	0	0	0	0	0	20	0	0	0	0	0
30	EDWC	36	0	0	0	0	0	36	0	0	0	0	0
201	CRSC	63	0	1	0	0	0	62	0	0	0	0	0
202	EAGC	7	0	0	0	0	0	7	0	0	0	0	0
204	EBGC	6	0	0	0	0	0	6	0	0	0	0	0
211	CAGC	862	0	12	0	0	0	675	175	0	0	0	0
212	CBGC	819	0	12	0	0	0	694	113	0	0	0	0
213	CCGC	343	0	0	0	0	0	332	11	0	0	0	0
221	KAGC	95	0	0	0	0	0	95	0	0	0	0	0
222	KBGC	154	0	7	0	0	0	147	0	0	0	0	0
223	KCGC	106	0	0	0	0	0	106	0	0	0	0	0
231	MAGC	507	0	13	0	0	0	405	89	0	0	0	0
232	MBGC	488	0	8	0	0	0	353	127	0	0	0	0
233	MCGC	359	0	0	0	0	0	351	8	0	0	0	0
241	DAGC	81	0	0	0	0	0	73	8	0	0	0	0
242	DBGC	100	0	0	0	0	0	86	14	0	0	0	0
243	DCGC	63	0	0	0	0	0	63	0	0	0	0	0
328	THUQ	2	0	0	0	0	2	0	0	0	0	0	0
329	CLEQ	1	0	0	0	0	1	0	0	0	0	0	0
330	FLDF	3	0	0	0	3	0	0	0	0	0	0	0
334	ALTF	1804	0	0	9	1795	0	0	0	0	0	0	0
341	FYLQ	1	0	0	0	1	0	0	0	0	0	0	0
342	FYLF	1	0	0	0	1	0	0	0	0	0	0	0
343	FYLT	1	0	0	0	1	0	0	0	0	0	0	0
348	THUF	1	0	0	0	0	1	0	0	0	0	0	0
354	ASCQ	60	0	0	0	0	60	0	0	0	0	0	0
363	ANTQ	1	0	0	0	0	1	0	0	0	0	0	0
369	MILQ	479	0	0	334	145	0	0	0	0	0	0	0
382	GOOQ	170	0	0	4	0	166	0	0	0	0	0	0
383	GOOF	73	0	0	0	0	73	0	0	0	0	0	0
385	ROBF	180	0	0	4	0	176	0	0	0	0	0	0
386	PPEQ	1	0	0	0	0	1	0	0	0	0	0	0
387	PPEF	1	0	0	0	0	1	0	0	0	0	0	0
388	PPWQ	24	0	0	0	0	24	0	0	0	0	0	0
393	CBDF	1	0	0	0	1	0	0	0	0	0	0	0
394	THPQ	1	0	0	0	0	1	0	0	0	0	0	0
396	PARQ	1	0	0	0	0	1	0	0	0	0	0	0
398	EGLF	1025	0	0	999	26	0	0	0	0	0	0	0
399	EGLQ	1620	0	0	1620	0	0	0	0	0	0	0	0

```

741  SNDQ      28      0  28      0      0      0      0      0      0      0      0      0
742  ELBQ      62      0  62      0      0      0      0      0      0      0      0
743  SILQ      40      0  40      0      0      0      0      0      0      0      0
744  FTSQ      36      0  36      0      0      0      0      0      0      0      0
745  NAVQ     794      0   0    794      0      0      0      0      0      0      0

746  RDRQ      10      0  10      0      0      0      0      0      0      0      0
747  HAWQ      74      0  74      0      0      0      0      0      0      0      0
951  MAUC      92      0   0      0      0      0     31     61      0      0      0

***** Unknown Stations Used *****

This list includes all unknown stations in the ADCOM observation file which generated GTDS-
usable observations.

      Station Data                                ADCOM TYPE (# Observations)
Number  Code  Total Obs   0    1    2    3    4    5  5-0    6    7    8    9

***** Non-GTDS Observations Found *****

508 Elevation rate, azimuth rate, and range acceleration triples were found.
0 Direction cosine pairs were found on range/range-rate/direction-cosine cards.
0 Direction cosine only cards were foundn

All non-GTDS observations found were ignored.

***** ADCOM Type Descriptions *****

Type #      Description          GTDS Type      Total Obs
Type 0: Range Rate          (GTDS Type 9)      0
Type 1: Azimuth             (GTDS Type 4)     303
      Elevation             (GTDS Type 5)
Type 2: Range               (GTDS Type 1)     3764
      Azimuth               (GTDS Type 4)
      Elevation             (GTDS Type 5)
Type 3: Range               (GTDS Type 1)     1973
      Azimuth               (GTDS Type 4)
      Elevation             (GTDS Type 5)
      Range Rate           (GTDS Type 9)
Type 4: Range               (GTDS Type 1)     508
      Azimuth               (GTDS Type 4)
      Elevation             (GTDS Type 5)
      Range Rate           (GTDS Type 9)
      Range Acceleration
      Azimuth Rate
      Elevation Rate
Type 5: Right Ascension     (GTDS Type 6)     4218
      Declination          (GTDS Type 7)
Type 6: Range               (GTDS Type 1)      0
Type 7: Direction Cosines
Type 8: Range               (GTDS Type 1)      0
      Direction Cosines
Type 9: Range               (GTDS Type 1)      0
      Range Rate           (GTDS Type 9)
      Direction Cosines

**Type 5 Reference Frame Listing:
      NORAD True of Date      606
      Space Identifier        3612
      Unknown Identifier       0

```

Figure A.4: NORAD Pre-Processor Print Messages File

```

OBSCARD
NAVQ      1      950102004515.926030  9509.29900000000    9509.29900000000
NAVQ      4      950102004515.926030  4.7369649615820    4.7369649615820
NAVQ      5      950102004515.926030  0.73256006562257   0.73256006562257
EGLQ      1      950102010555.662960  6540.55800000000   6540.55800000000
EGLQ      4      950102010555.662960  4.0364648489893    4.0364648489893
EGLQ      5      950102010555.662960  0.22787018714038E-01 0.22787018714038E-01
END

```

Figure A.5: NT-GTDS FRN-15 OBSCARD File

In summary, the following checklist identifies all necessary steps to execute a NPP run.

- Copy the necessary NORAD B-3 data file into the same directory as the NPP executable.
- Edit the CONTROL.DAT file.
 - Change the input data file name
 - Change start date
 - Change stop date
 - Save the file
- Exit to the DOS prompt and enter the NPP directory.
- Type 'NORADPP' at the prompt to begin execution.
- View output data through any text editor.

A.1.3 NORAD B-3 Format

In order for the reader to fully understand the code, or write a new processor, it is necessary to first understand the NORAD B-3 format. The following three tables attempt to explain B-3 for the user by identifying the format of the actual 80 space B-3 data file (Table A.4), a description of the observation types (Table A.5), and a description of the NORAD coordinate systems (Table A.6).

Table A.4: NORAD B-3 Observation Card Data Format

Variable	Variable	Variable	Start	Stop	Special
Description	Name	Type	Index	Index	Notes
Classification			1	1	U (Unclassified) or C (Classified)
NORAD Satellite ID			2	6	
Tracking Station Number	ISTNUM	I3	7	9	NORAD station number
Year of Observation	IYR	I2	10	11	YY
Day	IDAY	I3	12	14	DDD (1 to 365) - Day of year
Hour	IHR	I2	15	16	HH
Minute	IMIN	I2	17	18	MM
Second	SEC	F5.3	19	23	SS.SSS
Declination	LEL	A1	24	24	First Digit with Overpunch
Declination	ELEVR	F5.4	25	29	Remaining Digits
SPACE			30	30	
Right Ascension	IAZORA	I7	31	37	HHMMSS.S
SPACE			38	38	
Range Scale Factor	RNGE	F7.5	39	45	
Range exponent Factor	IRX	I1	46	46	Range = RNGE * (10.0 ^ IRX)
SPACE			47	47	
Range Rate	LRA	A1	48	48	First Digit with Overpunch
Range Rate	RATR	F6.5	49	54	Remaining Digits
SPACE			55	55	
Elevation Rate	LELR	A1	56	56	First Digit with Overpunch
Elevation Rate	ELRR	F4.4	57	60	Remaining Digits
SPACE			61	61	
Azimuth Rate	LAXR	A1	62	62	First Digit with Overpunch
Azimuth Rate	AZRR	F4.4	63	66	Remaining Digits
SPACE			67	67	
Range Acceleration	LAC	A1	68	68	First Digit with Overpunch
Range Acceleration	ACC	F4.4	69	72	Remaining Digits
SPACE			73	73	
Data Format Indicator	ITYPE	I1	74	74	See Data type table below
Coordinate System	EQNYR	A1	75	75	See Coordinate System Table below
Unused Indicators			76	85	GTDS does not use the remaining indicators (e.g. line number).

Table A.5: NORAD Observation Type Format

ADCOM Type Identifier (NORAD)	GTDS Type Identifier	Description	Variable's Used	Special Notes
0	9	Range Rate	ILRA, RATR	
1	4 & 5	Azimuth and Elevation	IAZORA, ILEL, ELEVR	
2	1, 4, & 5	Azimuth, Elevation, and Range	IAZORA, ILEL, ELEVR, RNGE, IRX	
3	1, 4, 5, & 9	Azimuth, Elevation, Range, and Range Rate	IAZORA, ILEL, ELEVR, RNGE, IRX, ILRA, RATR	
4	1, 4, 5, & 9	Azimuth, Elevation, Range, Range Rate, Range Acceleration, Azimuth Rate, Elevation Rate	IAZORA, ILEL, ELEVR, RNGE, IRX, ILRA, RATR, ACC, ELRR, and AZRR	Un-useable by GTDS
5	6 & 7	Right Ascension and Declination	IAZORA, ILEL, ELEVR, EQNYR	
6	1	Range	RNGE, IRX	
7		Direction Cosines	NCOS, and others	Un-useable by GTDS
8	1	Range Direction Cosines	NCOSR and others	Un-useable by GTDS
9	1 & 9	Range Direction Cosines and Range Rate	NCOSR, ILRA, RATR, and others	Un-useable by GTDS

Table A.6: NORAD Coordinate System Type

Identifier	Coordinate System	Notes
0	True of Date	At time of Observation
1	Mean of 1900.0	
2	Mean of 1920.0	
3	Mean of 1950.0	
4	Mean of 1975.0	
5	Mean of 2000.0	
6	Mean of 1850.0	
7	Mean of 1855.0	
8	Mean of 1875.0	
9	Mean of 1960.0	
X	True of Date	At date beginning of Besselian year of observation
Y	True of Date	At midnight, Jan 0, of year of observation
Else	Unrecognized	Unrecognized

A.1.4 Station Location and Accuracy Database (SLAD)

The NPP was utilized to analyze the data from all objects used in this investigation. The data was analyzed to determine not only the data distribution, but also the stations used in collecting data for these objects. Over their lifetimes, the Molniya objects had observations from over 70 different stations, and in order to use that data, it was necessary to compile a Station Location and Accuracy Database (SLAD) for use by NT-GTDS. The SLAD includes for each station, the station's longitude, latitude, height, and the accuracy for each type of measurement that the station produces.

Unfortunately, there isn't a single source to find this data from and a compilation of data and conversations with 6 different sources were utilized to create the SLAD. . These sources include conversations with Mike Gaposchkin, Taft DeVere, Dan Rypysc, Bill Craig, and Dan Fonte. We also used the attached spreadsheets from Mike Gaposchkin (Millstone Hill), NASA's NDOSL, Taft DeVere's spreadsheet, and the Kaman Sciences Sensor Location and Limits Revised Final Report (KSPACE 95-3).

Due to the potentially sensitive nature of the information contained within the SLAD, it is not reproduced in this work. The interested reader can contact the thesis advisors for this investigation, Dr. Cefola and Dr. Proulx for access to the full SLAD Excel spreadsheet, NT-GTDS card deck, or SLAD memorandum which presents the most complete listing of what values were used and why. The following is an example of the description style used in the SLAD memorandum (with accuracy and location information edited) to give the reader an idea of the information available. Additionally, Table A.7 lists the non-sensitive data from the SLAD Excel worksheet to identify stations which were used in this analysis.

369	MILQ	Millstone Hill	Radar
-----	------	----------------	-------

Station	Data	Station	Accuracy
Height		Range	
Latitude		Azimuth	
Longitude		Elevation	
Coord. Sys.		Range Rate	

Figure A.6: Station Location and Accuracy Database Description Block

- The numbers for the DeVere Database and Gaposchkin Spreadsheet were nearly identical and differed by about 3m Lat and 17m Lon from the Kaman Sciences Corporation value. This can be explained by Kaman Sciences listing in the WGS-84 reference frame. This combined with the fact that Gaposchkin works at Millstone, led to the choice of the Gaposchkin values for this station.
- The accuracy values we used we taken directly from a Millstone employee, Mike Gaposchkin.
 - Variation in these numbers has to do with the processing mode. Coherent mode on provides better accuracy.
 - These numbers are analogous to values used previously at Draper

Table A.7: Primary Station Data for Stations Used

ST. #	GTDS Code	Name	Type
5	RMLC	Malibar Telescope	Baker-Nunn
25	PULC	Pulmosan Korea	Baker-Nunn
27	STMC	St. Margaret, Canada	Baker-Nunn
28	SNVC	San Vito, Italy	Baker-Nunn
29	MTJC	Mt. John, New Zealand	Baker-Nunn
30	EDWC	Edwards AFB, California	Baker-Nunn
32	COLC	Cold Lake, Canada	Baker-Nunn
201	CRSC	Cross	Optical
202	EAGC	ETS Site A	Optical
204	EBGC	ETS Site B	Optical
206	GAGC	GTS Site A (GTS-1-GEODSS)	Optical
207	GBGC	GTS Site B (GTS-2-GEODSS)	Optical
211	CAGC	GEODSS CONUS (Stallion 1)	Optical

212	CBGC	GEODSS CONUS (Stallion 2)	Optical
213	CCGC	GEODSS CONUS (Stallion 3)	Optical
221	KAGC	GEODSS KOREA (Korea Cam 1)	Optical
222	KBGC	GEODSS KOREA (Korea Cam 2)	Optical
223	KCGC	GEODSS KOREA (Korea Cam 3)	Optical
231	MAGC	GEODSS MAUI (Maui Cam 1)	Optical
232	MBGC	GEODSS MAUI (Maui Cam 2)	Optical
233	MCGC	GEODSS MAUI (Maui Cam 3)	Optical
241	DAGC	GEODSS DIEGO GARCIA-1	Optical
242	DBGC	GEODSS DIEGO GARCIA-2	Optical
243	DCGC	GEODSS DIEGO GARCIA-3	Optical
328	THUQ	Thule, Greenland	Radar
329	CLEQ	Clear, Alaska	Radar
330	FLDF	Fylingdales BMEWS	Radar
334	ALTQ	Altair, Kwajalein	Radar
337	PRNQ	Prinlik, Turkey	Radar
341	FYLQ	Fylingdales, England	Radar
342	FYLF	Fylingdales, England	Radar
343	FYLT	Fylingdales, England	Radar
344	FLYT	Fylingdales, England	Phased Array
345	FLYQ	Fylingdales, England	Phased Array
346	FLYF	Fylingdales, England	Phased Array
348	THUF	Thule, Greenland	Radar
349	CLEF	Clear, Alaska	Radar
354	ASCQ	Ascension Island	Radar
355	ACSF	Ascension Island, 12.18	Radar
359	CLET	Clear, Alaska	Radar
363	ANTQ	Antigua Island	Radar
369	MILQ	Millstone Hill	Radar
370	MILF	Millstone UHF	Radar
382	GOOQ	Goodfellow AFB, Texas	PAVE-PAWS
383	GOOF	Goodfellow AFB, Texas	PAVE-PAWS
384	ROBQ	Robbins AFB, Georgia	PAVE-PAWS
385	ROBF	Robbins AFB, Georgia	PAVE-PAWS
386	PPEQ	PAVE-PAWS East	PAVE-PAWS
387	PPEF	PAVE-PAWS East	PAVE-PAWS
388	PPWQ	Beale AFB, California	PAVE-PAWS
389	PPWF	Beale AFB, California	PAVE-PAWS
393	CBDF	Cobra Dane, Alaska	Phased Array
394	THPQ	Thule AFB, Greenland	Phased Array

395	THPF	Thule AFB, Greenland	Phased Array
396	PARQ	PAR (Safeguard) Grand Forks, ND	Radar
398	EGLF	Eglin Deep Space (XR, [FPS-85])	Radar
399	EGLQ	Eglin AFB, Florida	Radar
400	PRNF	Princlik, Turkey	Radar
401	PRNT	Princlik, Turkey	Radar
404	PRKQ	Princlik, Turkey	Radar
741	SNDQ	San Diego, California	NAVSPAR
742	ELBQ	Elephant Butte, New Mexico	NAVSPAR
743	SILQ	Silver Lake, Mississippi	NAVSPAR
744	FTSQ*	Ft. Stewart, Georgia	NAVSPAR
745	NAVQ	NAVSPAR	NAVSPAR
746	RDRQ	Red River, Arkansas	NAVSPAR
747	HAWQ	Hawkinsville, Georgia	NAVSPAR
932	KAEQ	Kaena Point, Hawaii	Radar
951	MAUC	Maui, Hawaii	Optical
952	AMSC	Amos, 1.6m, Hawaii	Optical

A.2 NORAD Pre-Processor Shell

The NPP offers a great deal of flexibility when analyzing data of this type, but it can become very tedious to use when analyzing a large number of objects and trying to create density profiles. Another inefficiency exists when making runs with data when it is necessary to run the NPP to determine the stations used, copy the corresponding station information to a control card, and finally make the run. It is because of these inefficiencies that the NPP Shell program was written. As its name implies, the NPP shell program is simply a shell around the NPP which incorporates new functionality. The three main functions of the NPP Shell are listed below next to the number corresponding to the given option in the NPP Shell:

1. Generate a station matrix file.
2. Generate a standard GTDS FRN-15 OBSCARD file.

3. Generate an enhanced print messages file which lists the necessary station data for a run over the given interval, in control card format.

The OBSCARD file is standard fare, but the other two options warrant discussion. The station matrix file is designed to give the user the following information in an Excel compatible format given in truncated form in Figure A.7 (the last eight months are truncated so that the file will fit on one page).

- Total number of observations per year for every year of the satellite's lifetime
- Total number of observation per year for each station for every year of the satellite's lifetime.
- Total number of observations per month for the final year of the satellite's lifetime
- Total number of observation per month for each station for the final year of the satellite's lifetime.

	TOTAL	86	87	88	89	90	91	92	93	94	95	96	9512	9601	9602	9603
TOTAL	7862	443	474	654	586	658	762	393	791	626	893	1582	131	122	110	108
STMC	46	9	8	9	4	0	16	0	0	0	0	0	0	0	0	0
SNVC	11	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
CRSC	20	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0
EBGC	9	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0
GBGC	3	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
CAGC	481	22	12	46	26	91	86	26	39	25	41	67	0	5	0	5
CBGC	436	24	24	18	33	47	106	37	26	25	33	61	0	0	5	5
CCGC	794	31	81	96	71	5	6	1	0	0	98	405	20	46	42	54
KAGC	70	8	8	20	7	24	3	0	0	0	0	0	0	0	0	0
KBGC	102	3	0	30	4	43	4	18	0	0	0	0	0	0	0	0
KCGC	32	12	3	8	9	0	0	0	0	0	0	0	0	0	0	0
MAGC	433	43	46	26	0	37	70	44	23	33	41	70	23	0	15	5
MBGC	345	39	22	41	4	3	18	27	31	16	51	73	26	8	14	0
MCGC	671	29	53	80	121	90	9	0	0	0	38	251	21	15	5	20
DAGC	89	0	14	9	4	10	10	18	12	4	8	0	0	0	0	0
DBGC	82	0	5	3	4	8	7	11	6	0	8	30	0	0	0	0
DCCG	126	0	13	13	26	23	8	6	12	0	0	25	0	0	0	0
ALTO	1091	32	158	209	202	86	114	44	171	25	27	23	10	0	0	0
PRNQ	69	0	0	0	0	0	0	0	0	0	0	69	0	0	0	0
PYLC	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
PLYT	57	0	0	0	0	0	0	0	0	0	0	57	0	0	0	0
PLYU	39	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0
PLYF	57	0	0	0	0	0	0	0	0	0	0	57	0	0	0	0
MILQ	224	83	0	15	35	37	5	0	0	0	0	48	0	0	0	0
CBDF	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PARQ	6	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
ZGLF	723	0	0	0	0	73	250	105	293	0	0	0	0	0	0	0
ECLQ	895	7	26	10	2	6	0	0	101	318	252	172	13	27	16	0
PRKQ	22	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NAVQ	792	0	0	0	31	73	47	48	63	160	268	100	12	20	11	0
HAWQ	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KACQ	41	0	0	0	0	0	0	0	0	0	27	14	0	0	0	0
AMSC	71	71	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure A.7: Truncated 16805 Station Matrix File

These files proved invaluable when doing observation density profiles for each of the ten objects used in this study. All of the data distribution plots given in Appendix B were created in Excel using these files.

Another useful option (3) for the NPP shell is an enhanced print messages file which includes a listing of all stations used in the specified time span in GTDS control card format. The remainder of the print messages file remains the same, but at the end of the file, the station data is listed as shown in Figure A.8.

```

***** CONTROL CARD STATION INFORMATION *****
DMOPT
/ALTQ  100334  3    62.700                92413.0781        1672856.9910
/EGLQ  100399  3     0.380                303420.7790       2734706.5526
/FLYF  100346  3   338.900                542142.8299       3591947.6900
END
DCOPT
EDIT           2           3.0
TRACKELV  3 13           5.0                5.0
/ALTQ  001004005      8.0                28.8                28.8
/ALTQ  009           500.0
/EGLQ  001004005     30.0                45.0                45.0
/EGLQ  009           500.0
/FLYF  001004005     35.0                54.0                54.0
ELLMODEL  1           6378.135            298.26
ELLMODEL  2           6378.137            298.257223563
/ALTQ  200002
/EGLQ  200001
/FLYF  200001
END

***** UNUSED STATIONS *****

```

Figure A.8: Print Messages File Trailer Data

This feature saves an enormous amount of time when trying to step through the observation data with a large series of runs. The reader simply copies this section into a GTDS control card, and the run is ready to go, with almost no effort (compared to the painful process of cutting and pasting each individual station). The reader should note the ‘Unused Stations’ section, which is unused in the above example. The unused stations section lists, in the same format, whatever stations were edited by the NPP shell program. The shell program uses a simple insertion sort [126] to order the stations by the

number of observations attributed to them. The top 20 stations (the limit imposed by GTDS on the number of stations it will accept) are then printed in the control card station information section, and the edited stations in the unused stations section. This allows the user to override the NPP shell if it edits a highly accurate sensor such as Millstone Hill in favor of a less accurate sensor with only a few more observations.

Figure A.9 is a tree structure diagram of the NPP shell and the necessary input files used by the executable. The reader should note that the shell uses the same three input files as the NPP, with the addition of the station locations, accuracy, and coordinate frames files which are used by option three to create the control card listings. The first few lines of these files are shown in Figures A.10, A.11, and A.12 respectively.

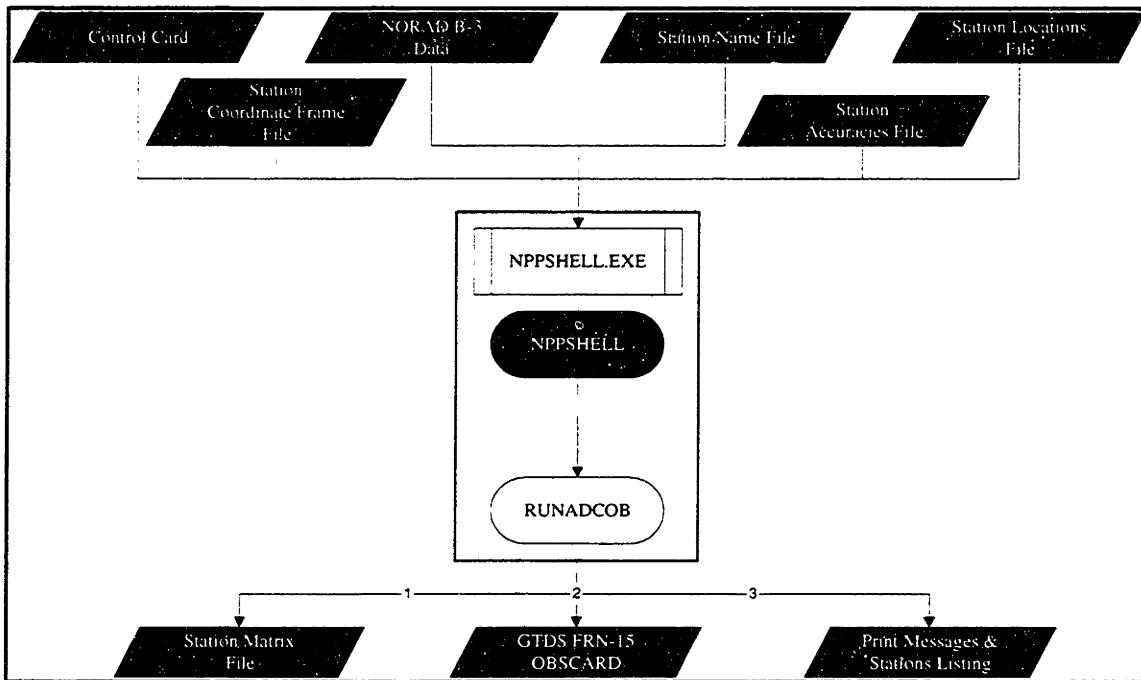


Figure A.9: Tree Structure Diagram of the NPP Shell Program

RMLC					
/RMLC	100005	13	-4.300	280128.5960	2791851.8040
PULC					
/PULC	100025	13	768.800	351002.3160	1284520.1600
STMC					
/STMC	100027	13	26.200	465357.8689	2944723.2829

Figure A.10: Truncated Version of the Station Locations File

RMLC			
/RMLC	006007	144.0	144.0
PULC			
/PULC	006007	144.0	144.0
STMC			
/STMC	006007	144.0	144.0

Figure A.11: Truncated Version of the Station Accuracy File

RMLC	
/RMLC	200001
PULC	
/PULC	200001
STMC	
/STMC	200001

Figure A.12: Truncated Version of the Station Coordinate Frame File

A.3 Single Orbit Observation Density Analysis Program (SOODAP)

When searching for a method to process through the Molniya data, one idea was to piecemeal single orbit determinations together to determine the effects of the perigee passage, and eventually model the perigee perturbations. Although this method proved ineffective, there were two important developments that came from this idea, the Single Orbit Observation Density Analysis Program, and eventually, the time dependent error function.

SOODAP was created in order to choose individual orbits with a high number of high quality observations on which to obtain fits. These fits would then be compared to a fit on the next orbit, and the elements of the two orbit compared to determine the effect of the passage through perigee. In order to accomplish this, it was necessary to model the orbital period and then count the number of observations within each given period. The resulting model was the sum of four Gaussian normal distributions, and was eventually used as the basis for the time dependent error function, developed in Chapter 6.

Figure A.13 shows the structure of the SOODAP program and corresponding I/O files. As input files, SOODAP uses a standard OBSCARD file and a control card of the format given in Figure A.14. The result is an Excel compatible data file which gives the following information for each orbit, and is shown in Figure A.15:

- The Gregorian date of the start of the given orbit
- The epoch day (days since launch) of the given orbit
- The number of azimuth/elevation observations for the orbit
- The number of range/azimuth/elevation observations for the orbit
- The number of right ascension/declination observations for the orbit
- The total number of observations for the given orbit
- The total number of observations for the lifetime of the object

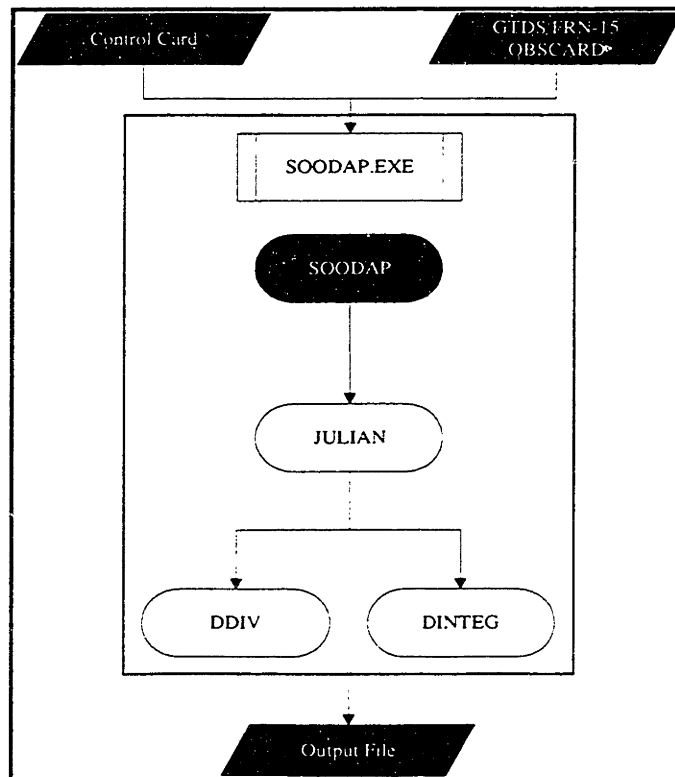


Figure A.13: Tree Structure Diagram of the Single Orbit Observation Density Analysis Program

```

GTDS OBSCARD DATA INPUT FILE : 13390all.OBS
OUTPUT DATA FILE : ORB_OBS#.dat
PERIOD FUNCTION DATA:      K0 : 11.6547
                             K1 : 375.0000
                             MU1 : 4300.0000
                             SIGMA1 : 350.0000
                             K2 : 160.0000
                             MU2 : 4630.0000
                             SIGMA2 : 90.0000
                             K3 : 3200.0000
                             MU3 : 4706.0000
                             SIGMA3 : 45.0000
                             K4 : 79.0000
                             MU4 : 4621.0000
                             SIGMA4 : 9.0000
GAUSSIAN TOLERANCE (HOURS) : 2.0000
LAUNCH DATE (YYMMDDHHMMSS.SS) : 820722000000.00
END ANALYSIS (YYMMDDHHMMSS.SS) : 950226000000.00

*END ANALYSIS DATE SHOULD BE APPROXIMATELY 30 DAYS
  PRIOR TO LAST DATA POINT.

```

Figure A.14: Single Orbit Observation Density Analysis Program Control Card

YR	MN	DY	HR	MN	SEC	ED	AZ/EL	RNG/AZ/EL	RA/DEC	OBS/ORB	TOTAL OBS
82	7	22	0	0	0.00000	0.0000	0	0	0	0	0
82	7	22	22	49	26.41003	0.9510	0	16	0	16	16
82	7	25	20	3	36.17603	3.8358	0	10	0	10	26
82	7	27	5	22	51.48401	5.2242	0	9	0	9	35
82	7	27	18	33	25.85803	5.7732	0	2	0	2	37
82	7	29	4	33	56.03503	7.1902	0	1	0	1	38
82	7	30	3	48	12.89502	8.1585	0	1	0	1	39
82	7	30	16	23	28.29895	8.6830	0	1	0	1	40

Figure A.15: Single Orbit Observation Density Analysis Program Output File

A.4 Gap Analysis Program (GAP)

Another useful utility is the data Gap Analysis Program (GAP) which is used to determine the size and location of gaps in data, as well as the number of observations between gaps. The gap program became a vital tool when trying to process the final months of data for NSSC-13390 because it identified where large data gaps and

observation concentrations were located so that filter runs and ephemeris generations could be optimally placed.

Figure A.16 shows the structure of the GAP program and corresponding I/O files. As input files, GAP uses a standard OBSCARD file and a control card of the format given in Figure A.17. The result is two data files, one of which describes the analysis interval, number of data gaps over the input tolerance level, and the start, stop, and length of the longest gap in the analysis interval. This is the summary output file, and is shown in Figure A.18.

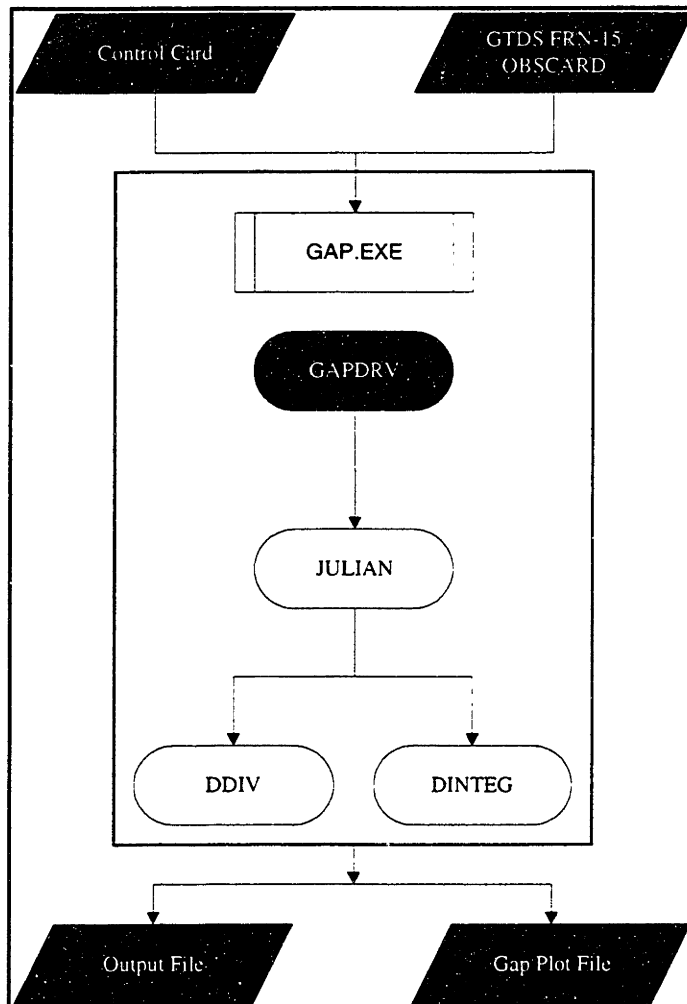


Figure A.16: Tree Structure Diagram of the Gap Analysis Program

```

GTDS OBSCARD INPUT FILE      : 13all.obs
OUTPUT FILE                   : GAP.OUT
INTERVAL START TIME          : 820722000000.00000
INTERVAL STOP TIME           : 950331000000.00000
GAP PLOT FILE NAME           : GAPLT.OUT
OBJECT LAUNCH DATE           : 820722000000.00000
MINIMUM GAP LENGTH (DAYS)    . 0.125

```

Figure A.17: Gap Analysis Program Control Card

```

*****
*                               INTERVAL DESCRIPTION                               *
*****

                YR MN DY HR MT  SECOND
START: 82   7 22 22 49 26.41003
STOP  : 95   3 27  3 44 46.46594

                LENGTH:      4630.2051 DAYS

OBSERVATIONS:                7087

*****
*                               GAP DESCRIPTION                               *
*****

                YR MN DY HR MT  SECOND
START: 88  10  4 16  1 20.92603
STOP  : 88  11  3  6 10 13.92004

                LENGTH:                29.5895 DAYS

                2556532.9940 SECONDS

*****
*****

```

Figure A.18: Gap Analysis Program Summary Output File

An additional output file is created by the GAP program to be used by Excel for creating a graphical representation of the gap data. Figure A.19 shows the format of these plotting files, and Figure A.20 is a sample plot. The dashed line represents a period with

no observations, or a data gap with a height (in days) of the gap length. The solid lines represent intervals with observations and a height corresponding to the number of observations within that interval.

EPOCH DAY	GAP	# OBS
0.00000	0.00000	0
0.00000	0.00000	1
0.00000	0.00000	1
0.00000	0.95100	0
0.95100	0.95100	0
0.95100	0.00000	48
0.97882	0.00000	48
0.97882	2.85701	0
3.83584	2.85701	0
3.83584	0.00000	30
3.83903	0.00000	30
3.83903	1.38518	0
5.22421	1.38518	0

Figure A.19: Gap Analysis Program Plotting Data Output File

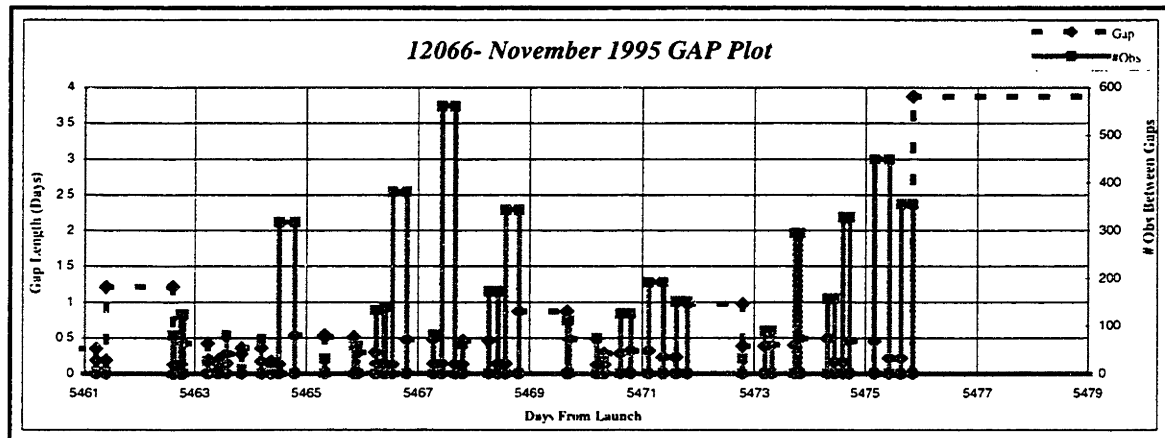


Figure A.20: NSSC-12066 November 1995 Gap Analysis Program Plot

A.5 Generated Ephemeris Analysis Program (GEAP)

The Generated Ephemeris Analysis Program is a simple program which reads an EPHEM output file and reorders the data in an Excel compatible format for plotting. The GEAP allows the user to specify which EPHEM elements are included in the output file

with a descriptor sum. The user simply sums the following option numbers of each set of parameters desired, and GEAP reorders the corresponding data. For example, if the user wanted to plot all parameters, a descriptor sum of 7 would be used.

1. Position and Velocity parameters (X, Y, Z, XDOT, YDOT, ZDOT, RMAG, VMAG)
2. Orbital Elements (SMA, ECC, INC, LAN, AP, MA, EA, TA, PERIOD)
4. Miscellaneous Data (PH, APH, C3, RA, DEC, VPA, AZ, LAT, LON, HEIGHT)

Figure A.21 shows the structure of the GEAP program and corresponding I/O files. As input files, GEAP uses a GTDS EPHEM file and a control card of the format given in Figure A.22. The result is an Excel compatible data file which gives the specified information as described above. These files are too wide to reproduce here, so Figure A.23 shows the first few lines of a sample output file 4-5 columns at a time.

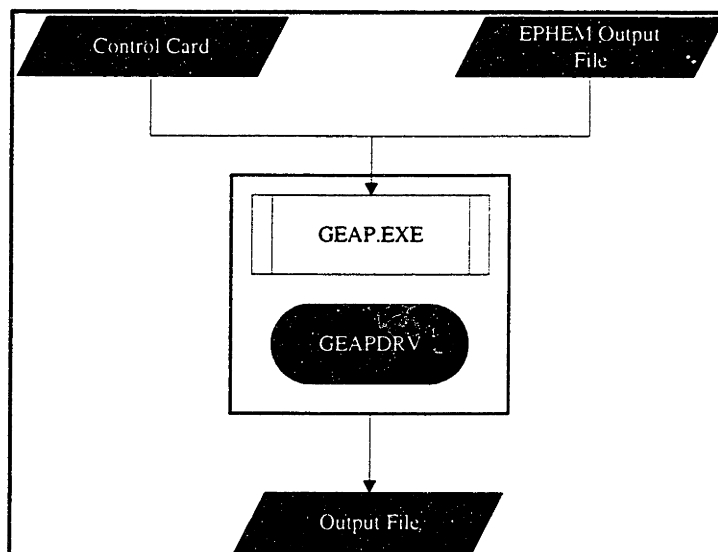


Figure A.21: Tree Structure Diagram of the Generated Ephemeris Analysis Program

```

GTDS EPHEM DATA INPUT FILE : eph_run2.out
OUTPUT DATA FILE           : decayeph.dat
DESCRIPTOR SUM              : 2

```

Figure A.22: Generated Ephemeris Analysis Program Control Card

MONTH	DAY	YEAR	TIME	X	Y	Z	RMAG
MA	31	1994	2449504 2803000000	-11639 7564600000	-8645 4689240000	-296 9673045000	14502 2844100000
				ADOT	YDOT	ZDOT	VMAG
				-1 5899893980	-4 5027960120	3 8766970870	6 2661000660
				SMA	ECC	INC	LAN
				25377 7319200000	0 7421459700	58 5145395300	217 3219808000
	AP			MA	EA	TA	PERIOD
	251		8723037000	20 0111810400	54 7294878400	106 7517443000	11 1760332500
	PH			APH	CJ	RA	DEC
	165		6141158000	37833 5769200000	-7 8533504170	216 6032451000	-1 1733435970
	VPA			AZ	LAT	LON	HGHT
	47		8855507300	31 4928202100	-1 3753300690	47 2318580200	8124 1603680000

Figure A.23: Generated Ephemeris Analysis Program Output File

A.6 Filter Control Card Generator (FCCG)

Another simple program is the Filter Control Card Generator (FCCG). Simply put, the FCCG takes the converged solution from a FILTER output file and creates a control card for use in the next run in a sequential processing of the data. The only option for the program is to specify whether or not the drag solve for option is used during the run (lift solve-for options are not supported). This program is diagrammed in Figure A.25, and requires only a FILTER output file and control card (given in Figure A.24). An example of the output file is given in Figure A.26.

```

GTDS FILTER DATA INPUT FILE : TURN COW.dat
OUTPUT DATA FILE           : FCCG.OUT
CD SOLVE FOR (1/0 = YES/NO) : 1

```

Figure A.24: Filter Control Card Generator Program Control Card

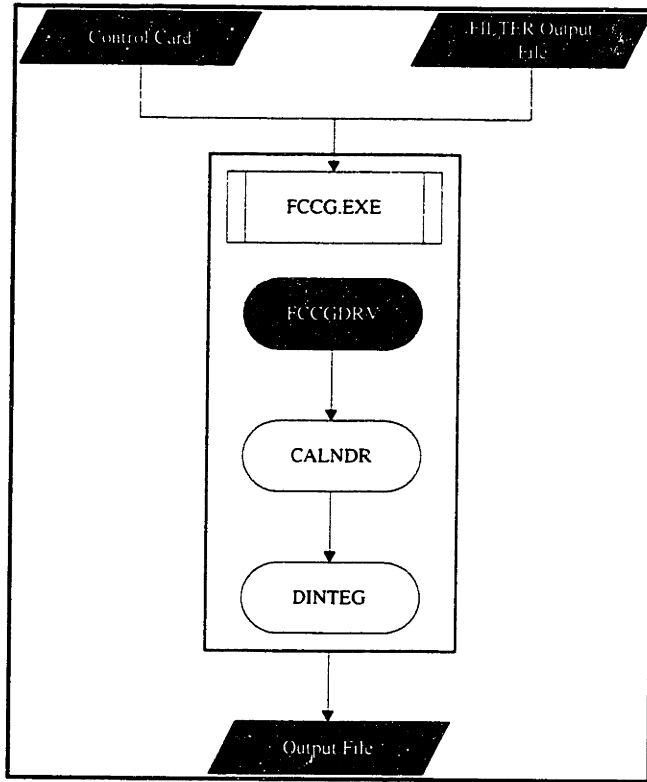


Figure A.25: Tree Structure Diagram of the Filter Control Card Generator Program

EPOCH				950318.00		0.0000000	
ELEMENT1	1	2	1	8968.264380804656	0.2800120620865360	62.11215385915079	
ELEMENT2				151.49067416246450	256.43820020471179	216.35285032817649	
COVARNC	11	12	13	0.4650066E+00	-0.3506771E-04	-0.2637596E-02	
COVARNC	14	15	16	-0.7212854E-02	0.9244152E-03	-0.3207854E-02	
COVARNC	22	23	24	0.8552225E-08	-0.1277589E-06	0.1147801E-06	
COVARNC	25	26	33	0.2674086E-06	-0.5685209E-07	0.7781132E-04	
COVARNC	34	35	36	0.8086327E-04	-0.8673012E-04	0.1303385E-03	
COVARNC	44	45	46	0.2197975E-03	-0.2269413E-04	-0.2088229E-04	
COVARNC	55	56	66	0.1306651E-03	-0.2273506E-03	0.4690330E-03	
DRAGPAR	3	0		0.5002416E+00			

Figure A.26: Filter Control Card Generator Program Output File

A.7 Residual Data File Generator (RDFG)

The least developed utility is the Residual Data File Generator (RDFG), designed to analyze the residuals of two specific FILTER runs, one Extended and one Linearized.

The residual plot files were then used to create plots such as Figures 7.4 and 7.5 which were used to verify the characteristic behavior of the two filter types.

The structure of the RDFG is shown in Figure A.27. Initially the program would allow the user to specify a specific station and residual type, but the current version outputs the specified residual type for all stations used, as shown in Figure A.28. The epoch day corresponding to every observation processed is included in the first column, and the corresponding residual (of the specified type) are then printed in the column of the station which produced it. Inputs for the program include a control card, shown in A.29, and a modified residuals file, as shown in Figure A.30. The input residuals file is created by reading in the residuals from a Filter output file, adjusting the data format with an Excel spreadsheet, using Excel to define the epoch day column, and saving to a data file. Obviously this utility is poorly designed, and future revisions are necessary to make it a truly useful utility.

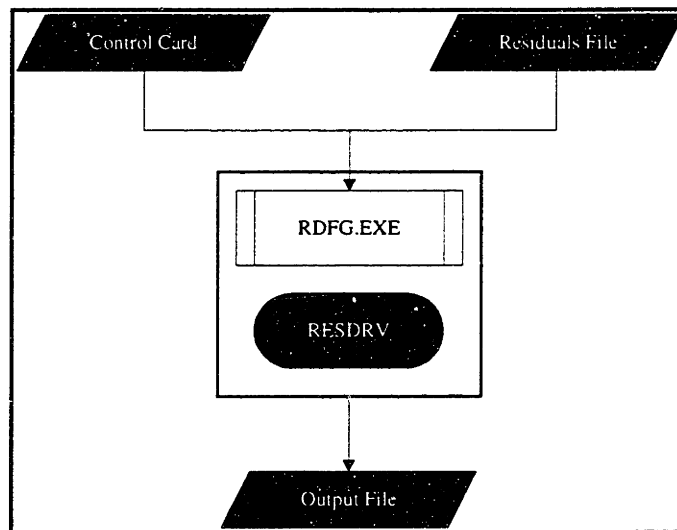


Figure A.27: Tree Structure Diagram of the Residual Data File Generator Program

EP DAY	PRNQ	FLYT	FLYQ	ANTQ	GOOQ	ROBQ	ROBF
0.0637800							-1601.47
0.0639100							-1561.55
0.0641800							-1553.19
0.0643000							-1537.70

Figure A.28: Residual Data File Generator Program Output File

GTDS RESIDUAL DATA INPUT FILE	:	T2_RESD2.dat
RESIDUAL OUTPUT FILE	:	T2_RDFG.OUT
STATION FOR ANALYSIS	:	PRNQ
OBSERVATION TYPE	:	RANG

Figure A.29: Residual Data File Generator Program Control Card

A.8 Utility Coding Summary

Table A.8 lists the existing Fieger and Leo Early routines that were modified for use with the new NPP, and Table A.9 lists the new routines created for all utilities. In all 287 lines were modified or added in four existing routines, and 1870 lines were added in 7 new programs.

Table A.8: Existing Code Modifications for the NORAD Pre-Processor

Name	Changes Overview	Lines
ASTRON	Comment out all EBCDIC format reference and replace with ASCII format and added the COUNTFR variable to count number of unchanged obs due to unknown reference frame.	
	The optical data had, for the value of EQNYR, some 0's and some blank spaces. The blank space, denoting the data was already in Mean of 1950.0, was fine for the Baker-Nunn sites but not the GEODSS sites. In order to correct for that problem, ASTRON then inputs the station number and compares that when looking at EQNYR. If the station number is greater than 210, it treats a space as a 0 and rotates from the NORAD true of date	75
ELEVAT	Comment out all EBCDIC format reference and replace with ASCII format	25
RANGER	Comment out all EBCDIC format reference and replace with ASCII format	25
RUNADCOB	Numerous modifications to support the new counters, data structures, NUMCONV subroutine, ASCII format, and several new write statements.	162
TOTAL		287

Table A.9: New Utilities Code

Name	Routine Description	Lines
GAPDRV	GAP analysis utility main program	244
SOODAP	SOODAP analysis utility main program	296
NUMCONV	ASCII/EBCDIC conversion utility	107
NPPSHELL	NPPSHELL analysis utility main program	521
FCCGDRV	FCCG utility main program	213
GEAPDRV	GEAP utility main program	294
RESDRV	RDFG utility main program	202
TOTAL		1877

[This page intentionally left blank.]

Appendix B

Data Analysis Plots

Appendix A outlines the various utilities used to process and analyze the substantial amount of Data used in this investigation. Appendix B is designed to present selected results from the data analysis in order to:

- Characterize a typical decay using the selected test object NSSC-13390
- Show the observation density profiles and final gap analyses for each object studied
- Delineate why NSSC-13390 was chosen over the other objects for study

B.1 NSSC-13390

NSSC-13390 is the rocket body for the Molniya 1-55 satellite, launched on 22 July, 1982. Before it's final decay on 28 March, 1995, there were 21,488 observations collected on NSSC-13390. The last observation taken was on the day before final decay, 27 March, 1995.

B.1.1 Element History Plots

The following element history plots are plots of the NORAD two-line element sets and are presented to characterize a typical catastrophic decay. Both lifetime, and final three months plots are given for each parameter.

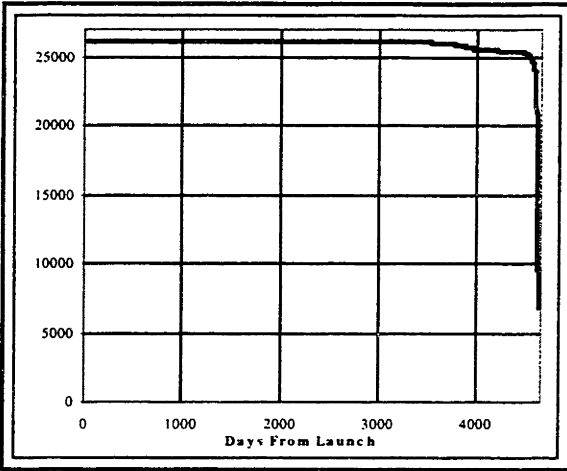


Figure B.1: SMA Lifetime History

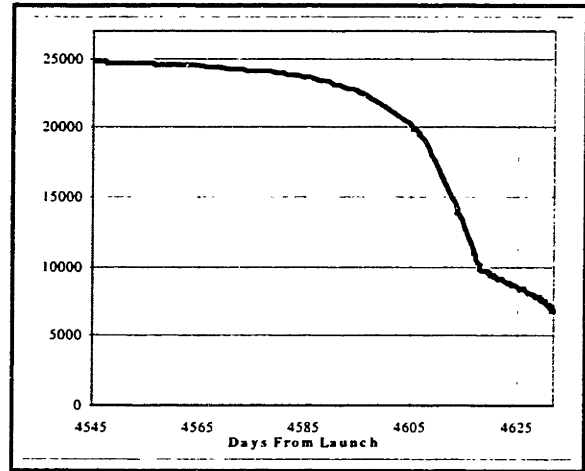


Figure B.2: SMA Final 3 Months History

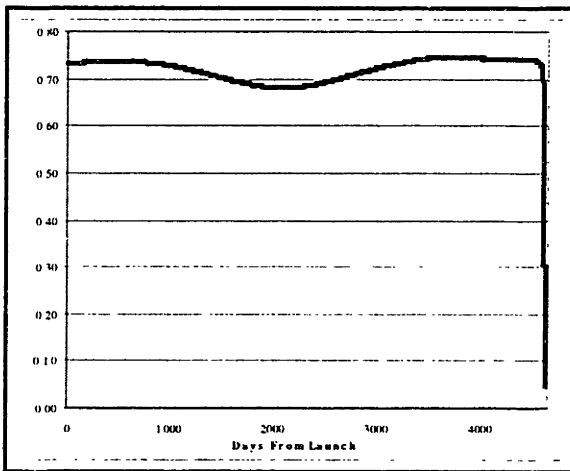


Figure B.3: Eccentricity Lifetime History

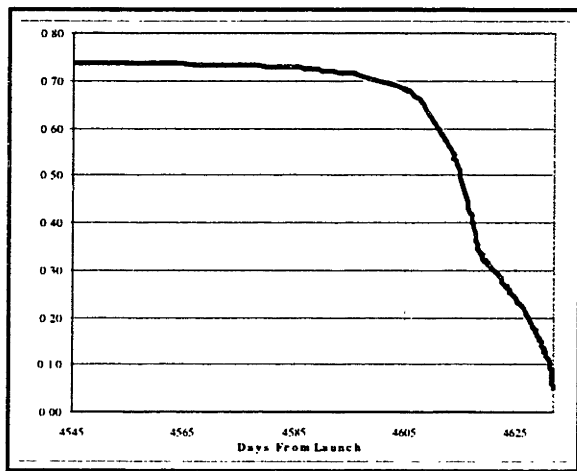


Figure B.4: Eccentricity Final 3 Months History

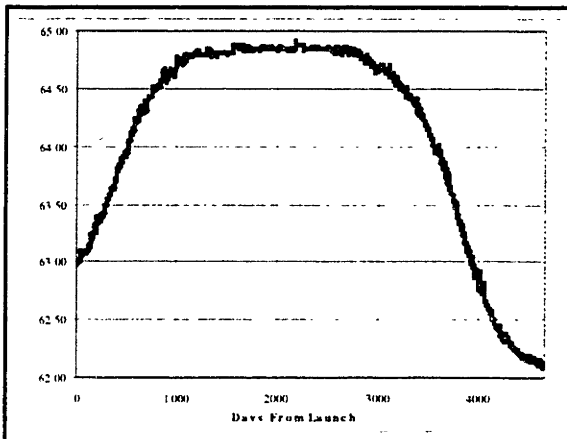


Figure B.5: Inclination Lifetime History

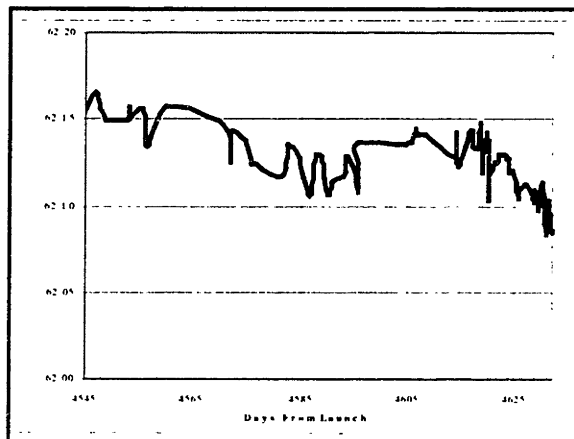


Figure B.6: Inclination Final 3 Months History

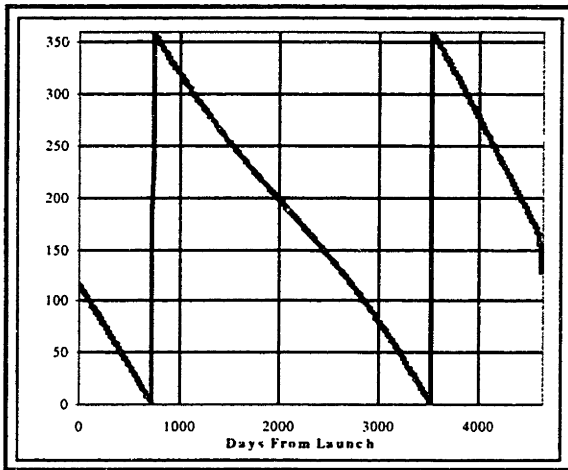


Figure B.7: RAAN Lifetime History

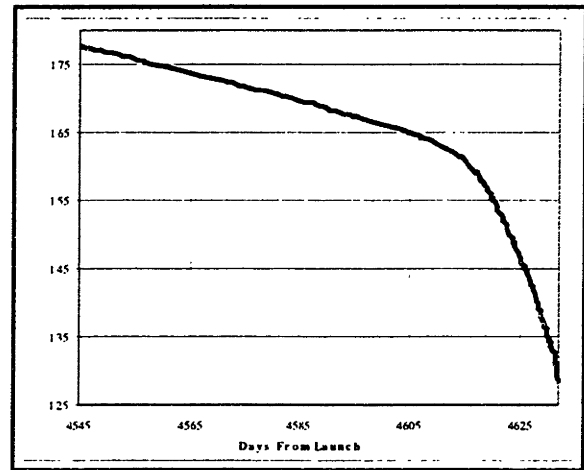


Figure B.8: RAAN Final 3 Months History

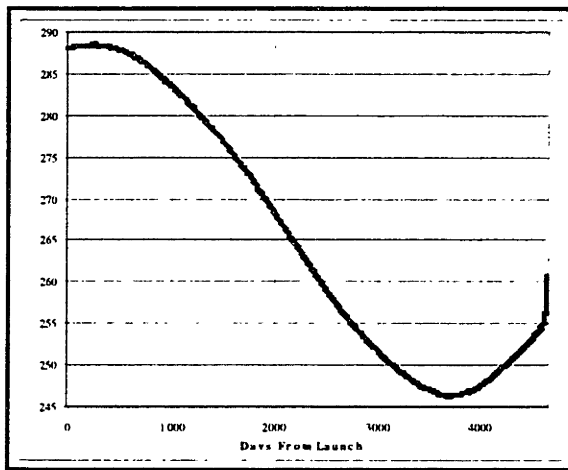


Figure B.9: ω Lifetime History

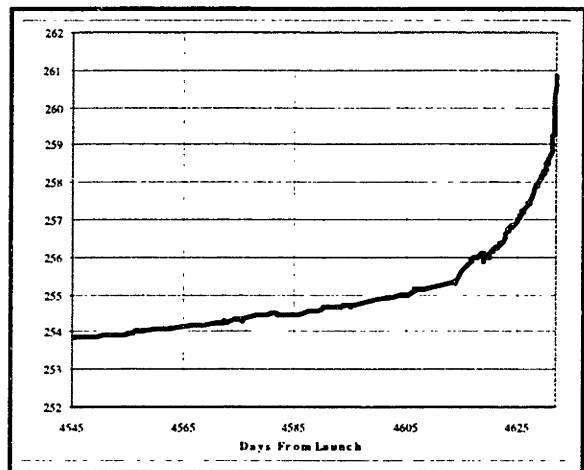


Figure B.10: ω Final 3 Months History

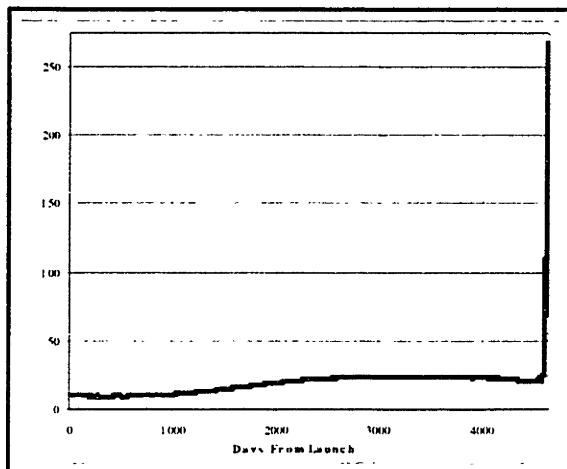


Figure B.11: Mean Anomaly Lifetime History

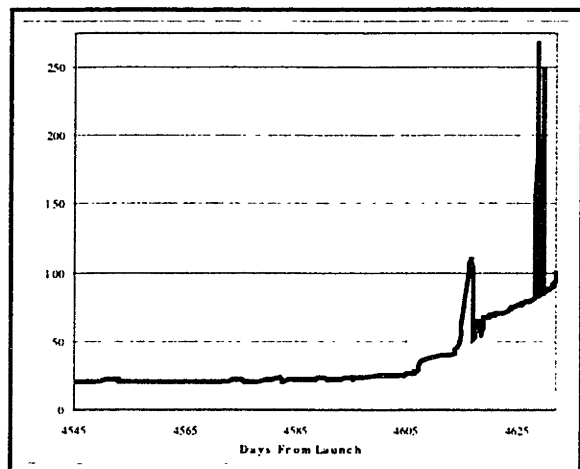


Figure B.12: Mean Anomaly Final 3 Months History

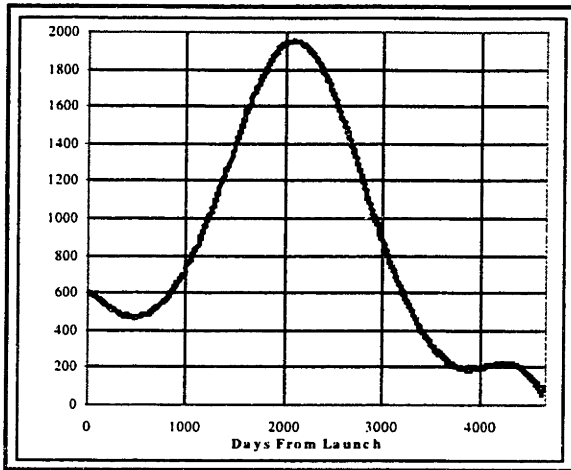


Figure B.13: Perigee Height Lifetime History

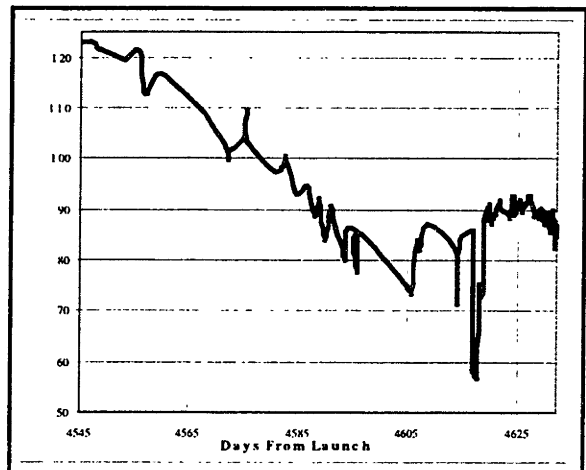


Figure B.14: Perigee Height Final 3 Months History

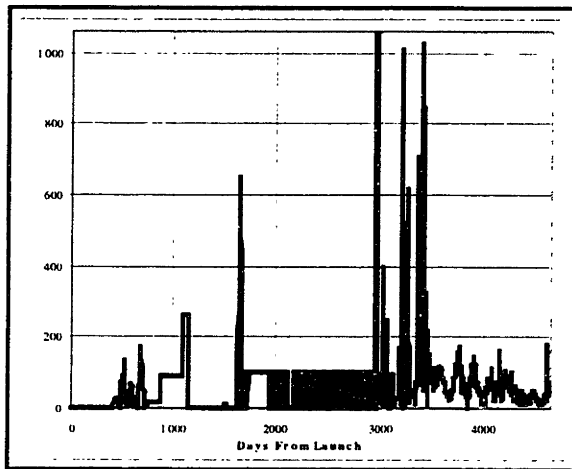


Figure B.15: B* Lifetime History

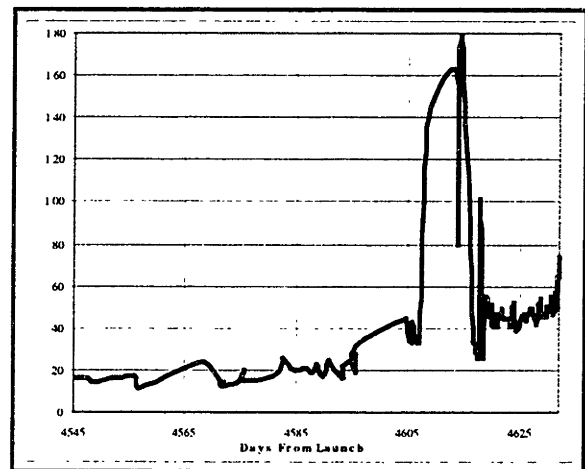


Figure B.16: B* Final 3 Months History

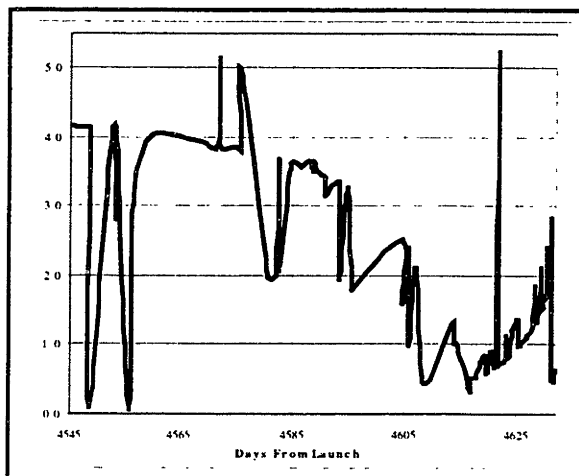


Figure B.17: $\dot{n}/6$ Final 3 Months History

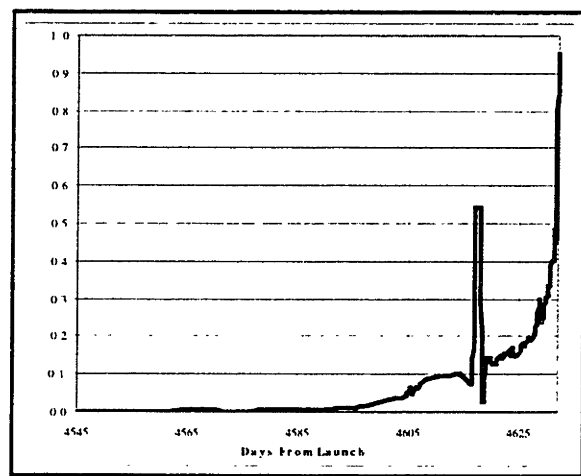


Figure B.18: $\dot{n}/2$ Final 3 Months History

B.1.2 Observation Density Analysis

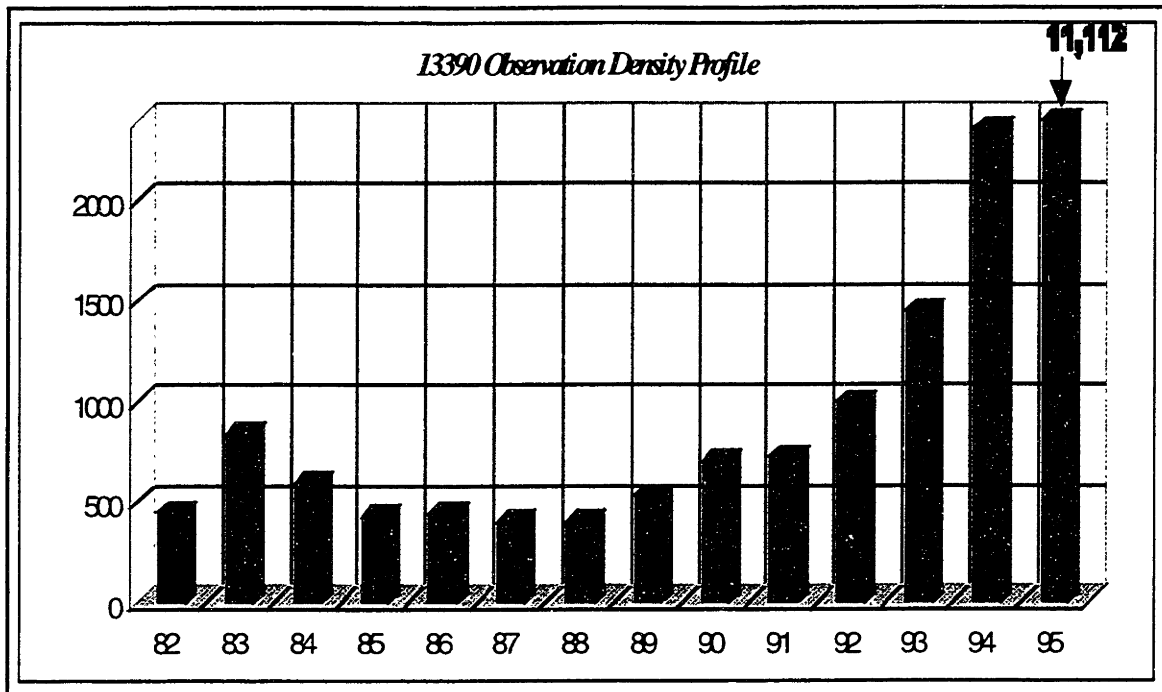


Figure B.19: NSSC-13390 Yearly Observation Density Profile

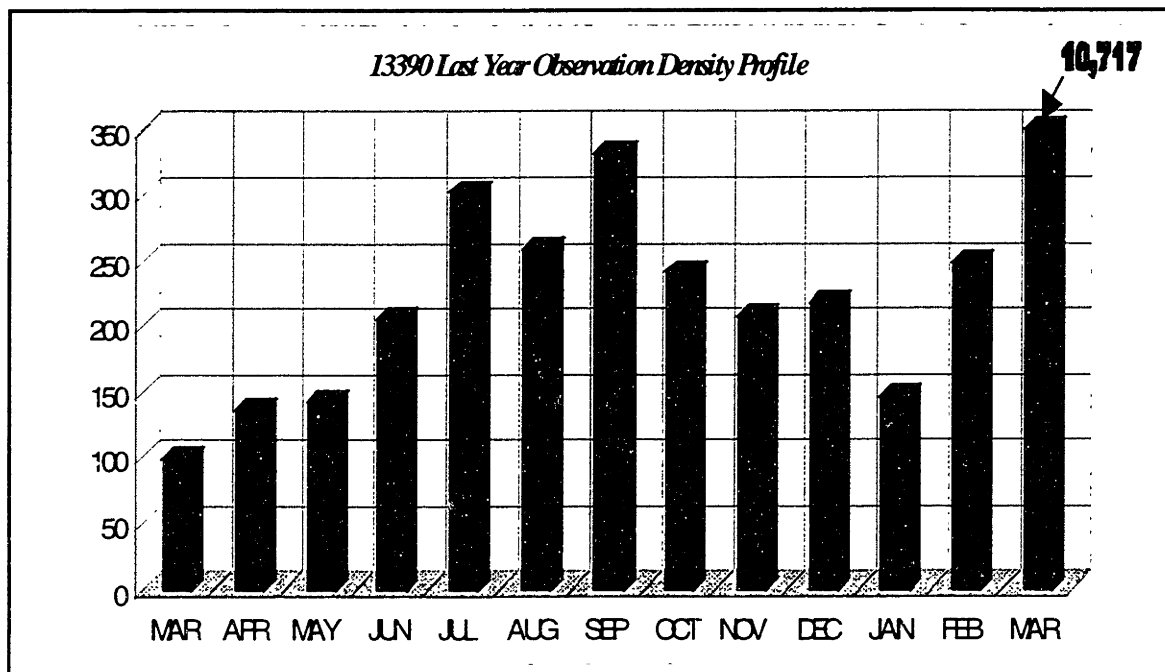


Figure B.20: NSSC-13390 Final Year Monthly Observation Density Profile

B.1.3 Single Orbit Observation Density Analysis

In order to demonstrate the uses of the SOODAP program, the following five plots are presented. B.21 represents the observation per orbit for the lifetime, B.22 represents the three months before the final month of decay (SOODAP constraint). Figures B.23 – B.25 represent the contributions to B.21 of each of the three observation types.

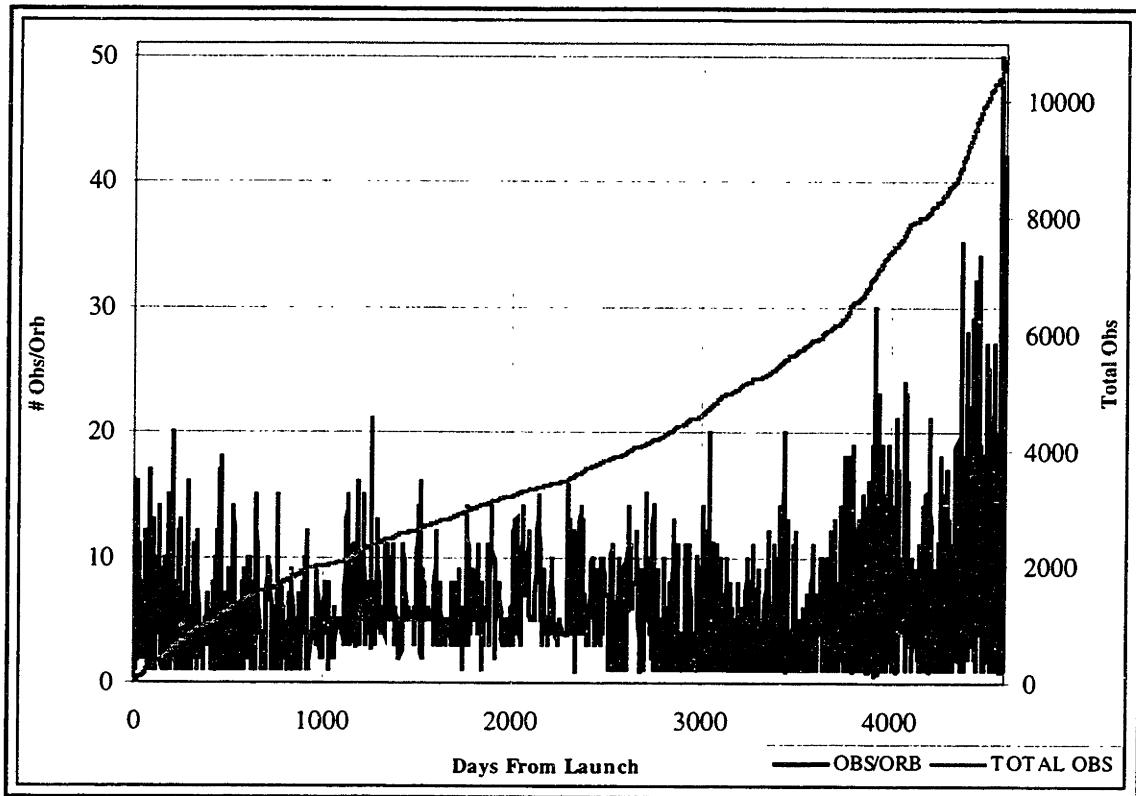


Figure B.21: NSSC-13390 Lifetime Single Orbit Observation Density Profile

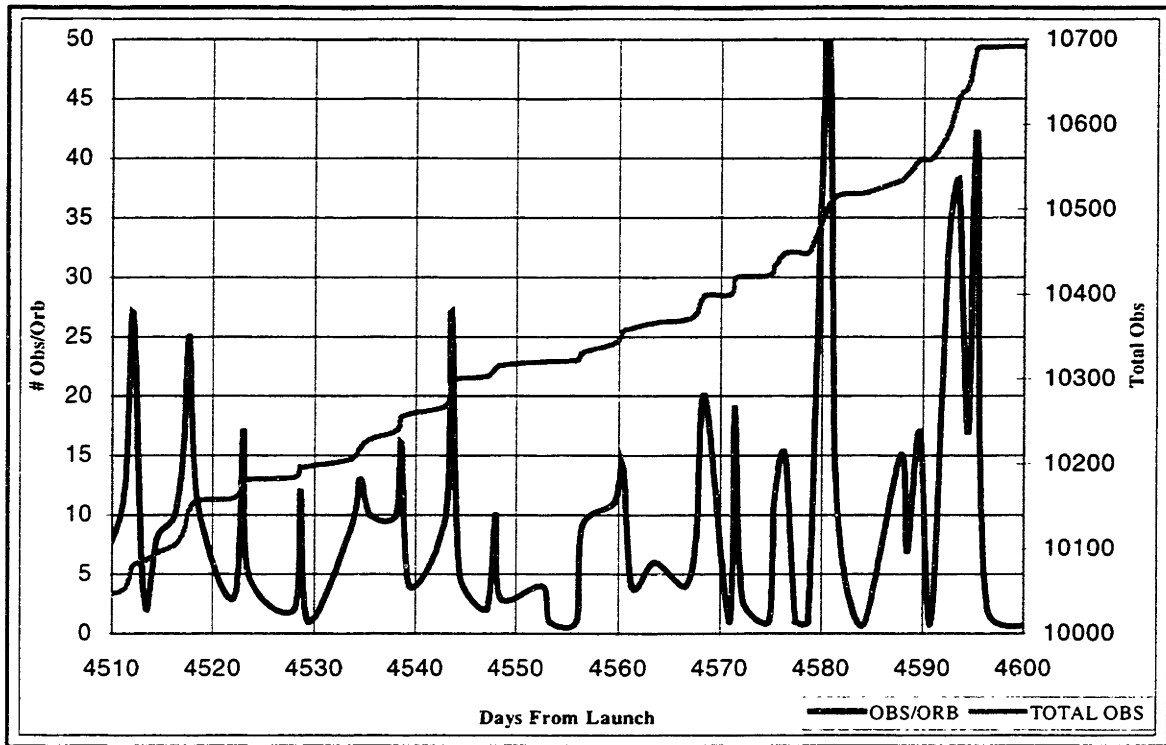


Figure B.22: NSSC-13390 Single Orbit Observation Density Profile: Three Months Prior to Decay Month

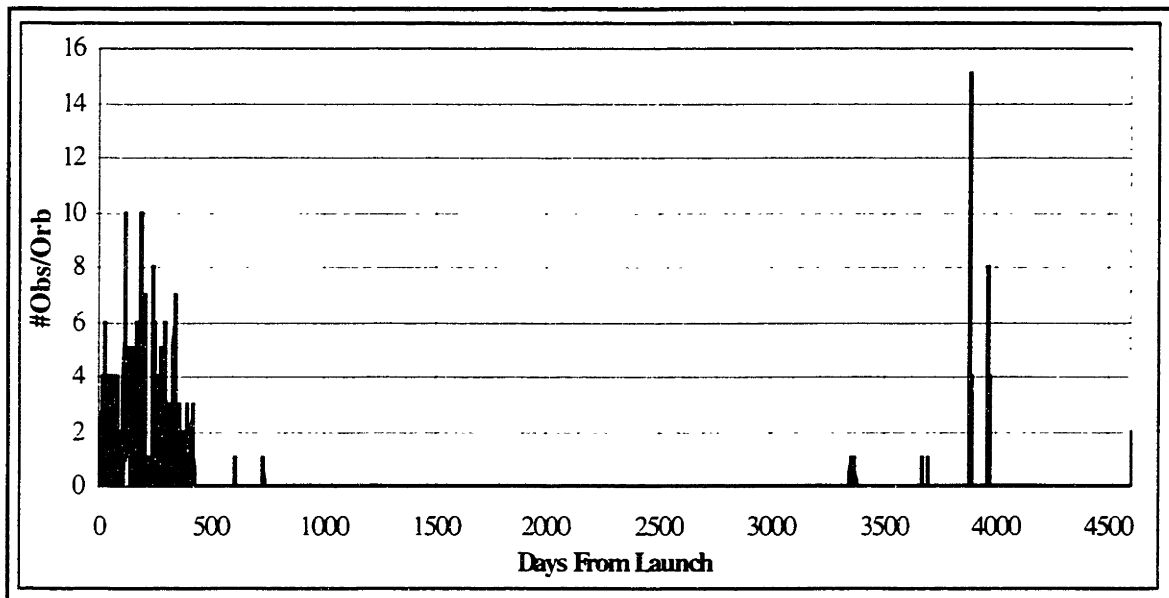


Figure B.23: NSSC-13390 Lifetime Single Orbit Azimuth-Elevation Observation Density Profile

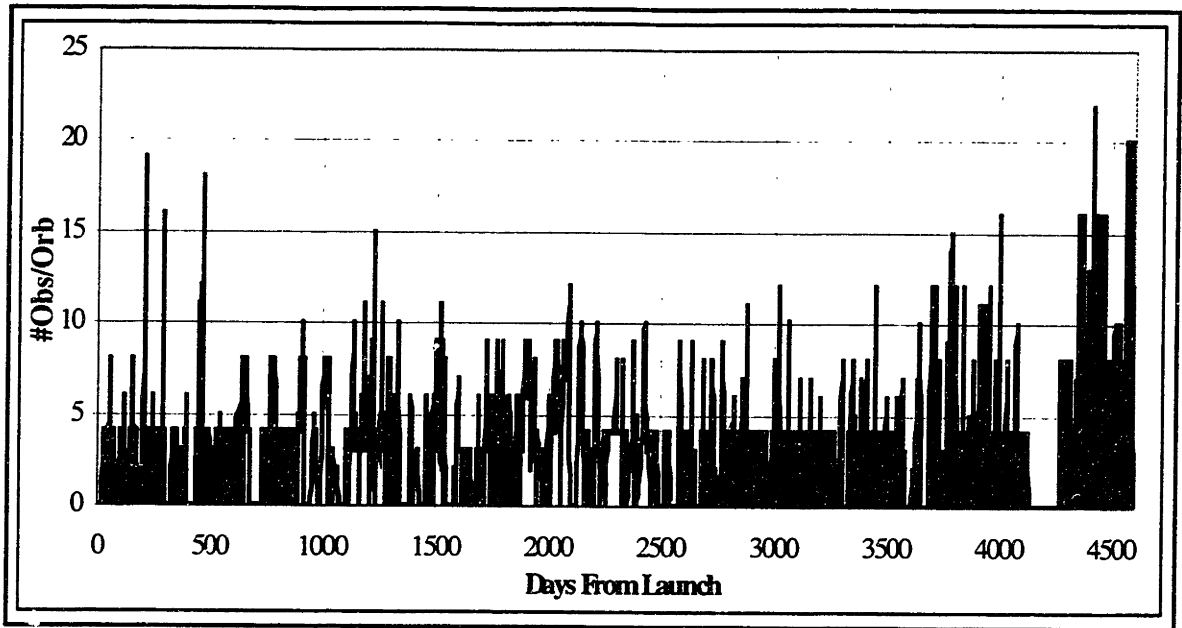


Figure B.24: NSSC-13390 Lifetime Single Orbit Range-Azimuth-Elevation Observation Density Profile

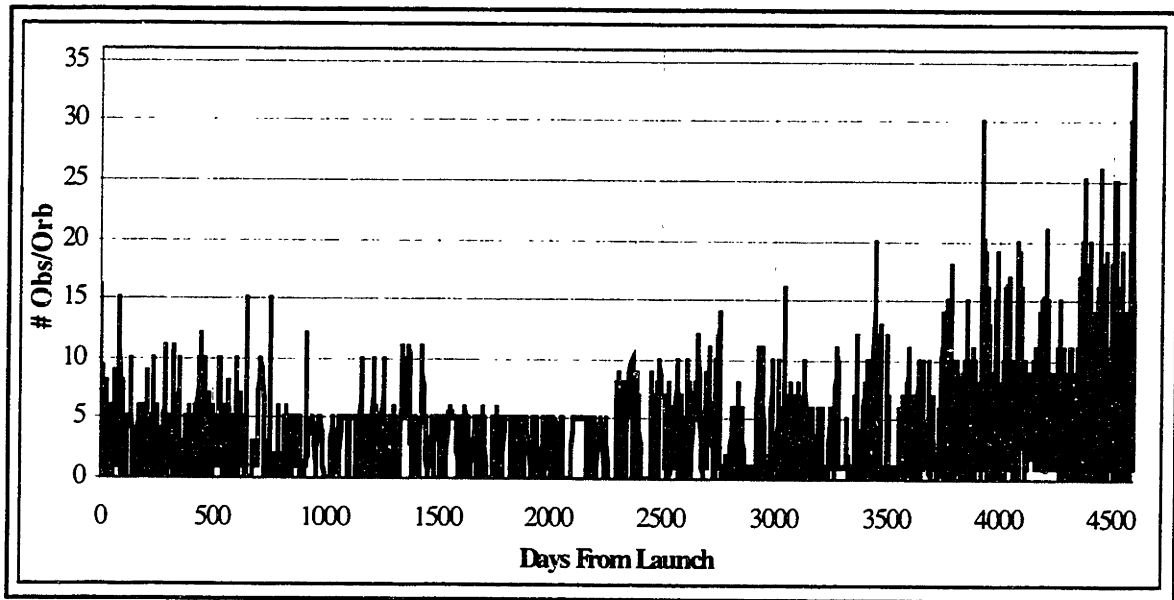


Figure B.25: NSSC-13390 Lifetime Single Orbit Right Ascension-Declination Observation Density Profile

B.1.4 Observation Gap Analysis

With the exception of the month of February, NSSC-13390's best quality is that the observations are very well distributed in time. In comparison with the other objects, the number and size of data gaps for NSSC-13390 clearly identify it as the most suitable test object. Figure B.26 is a lifetime gap analysis plot, B.27 plots the last nine months before decay, truncated two weeks before decay when the majority of the observations were taken. Figures B.28, B.29, and B.30 show the gap analyses for the final three months respectively. Scales are kept constant (with the exception of # obs for March) to promote comparison.

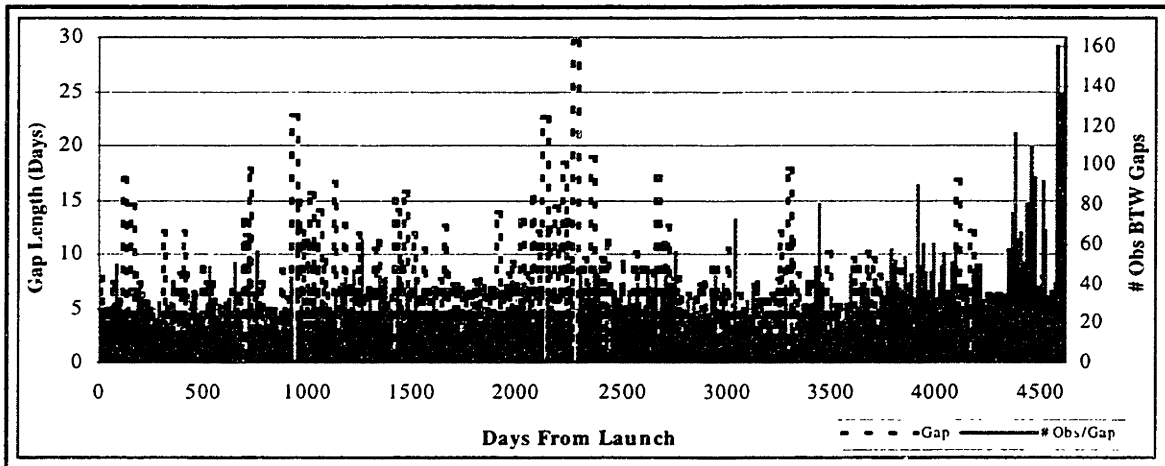


Figure B.26: NSSC-13390 Lifetime Gap Analysis Plot

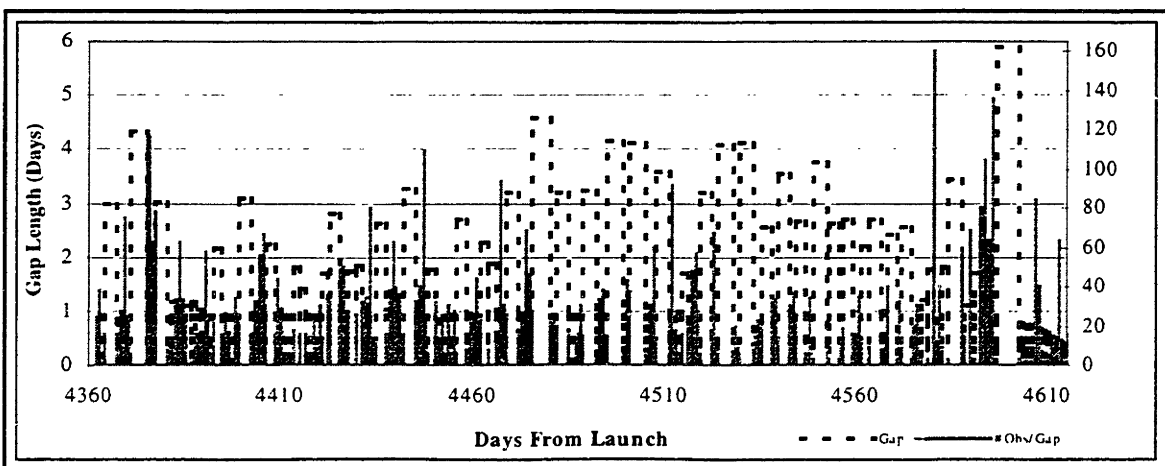


Figure B.27: NSSC-13390 Final Nine Months Gap Analysis Plot

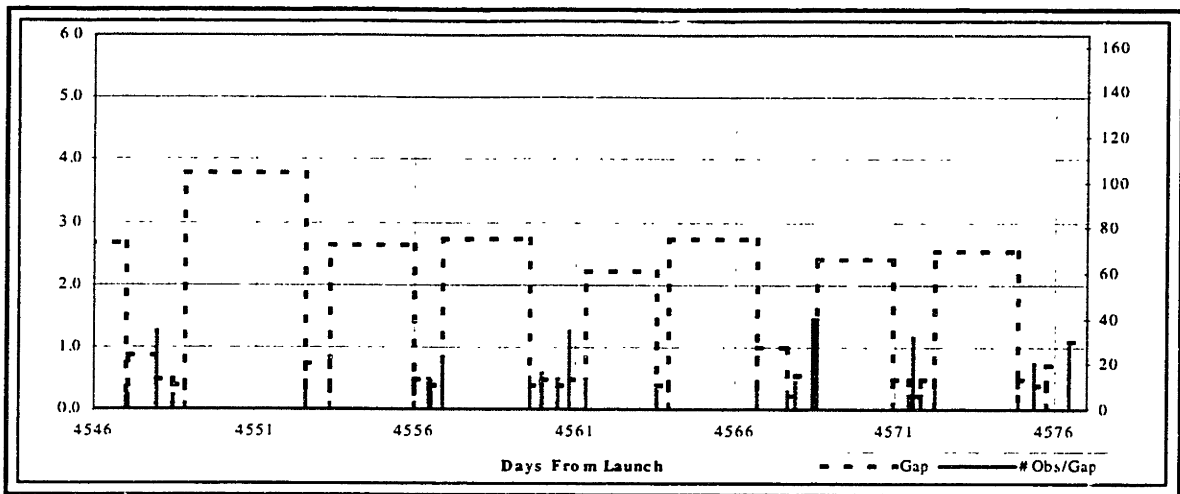


Figure B.28: NSSC-13390 January 1995 Gap Analysis Plot

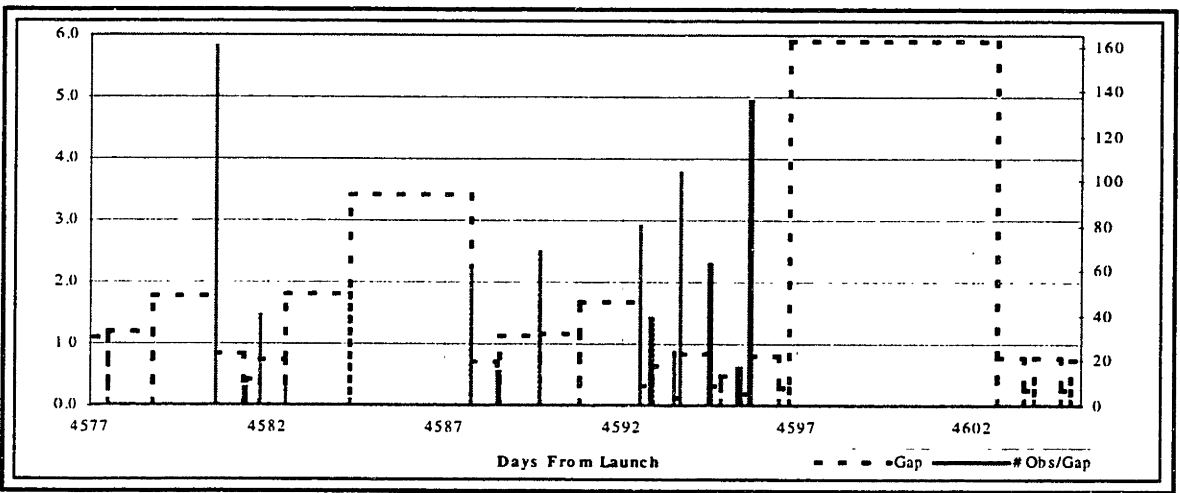


Figure B.29: NSSC-13390 February 1995 Gap Analysis Plot

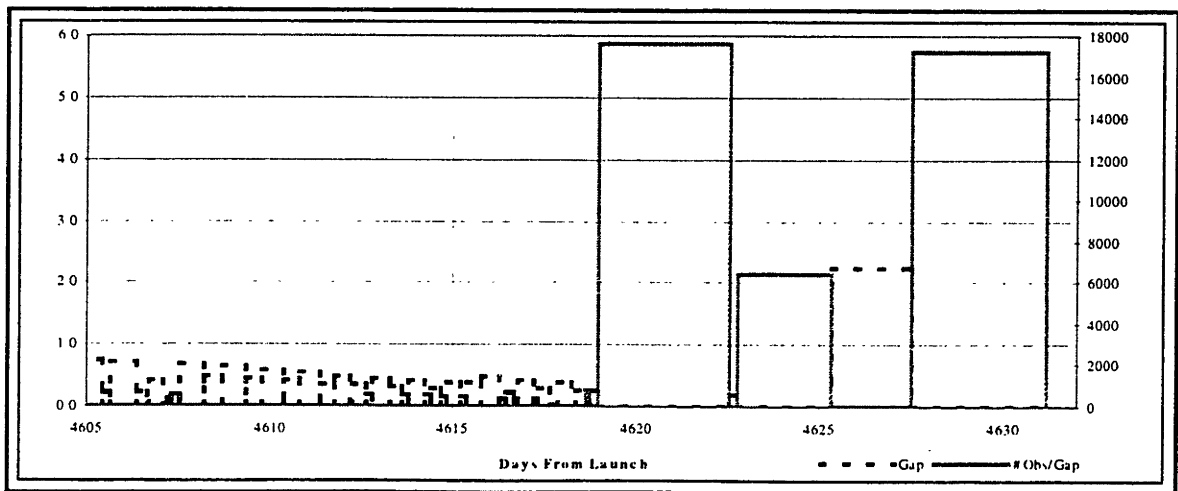


Figure B.30: NSSC-13390 March 1995 Gap Analysis Plot

B.2 NSSC-12066

NSSC-12066 is the Molniya 1-48 satellite, launched on 18 November, 1980. Before its final decay on 18 November, 1995, there were 26,480 observations collected on Molniya 1-48. The last observation taken was on the day of final decay, 18 November, 1995.

B.2.1 Observation Density Analysis

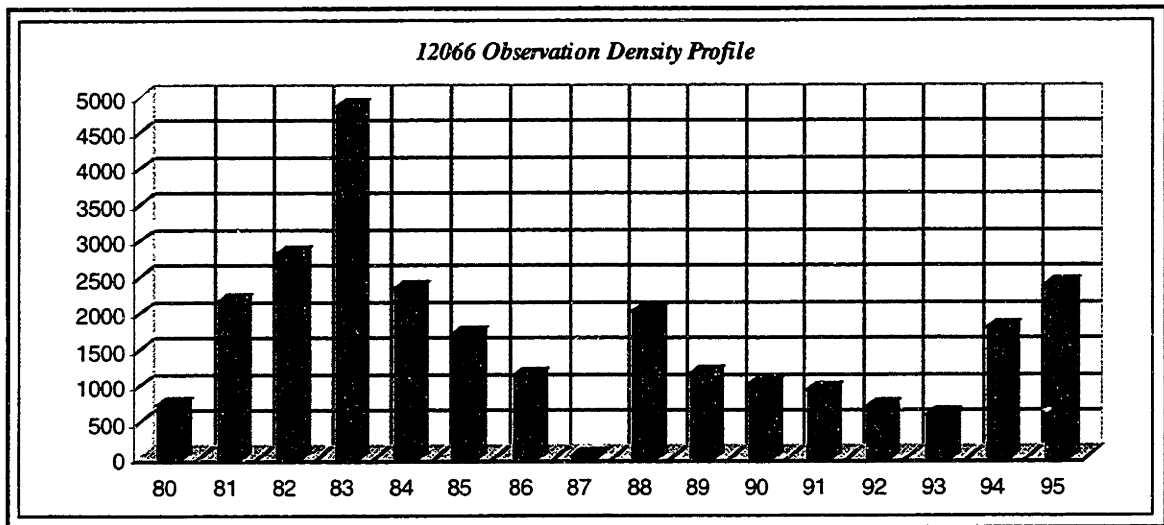


Figure B.31: NSSC-12066 Yearly Observation Density Profile

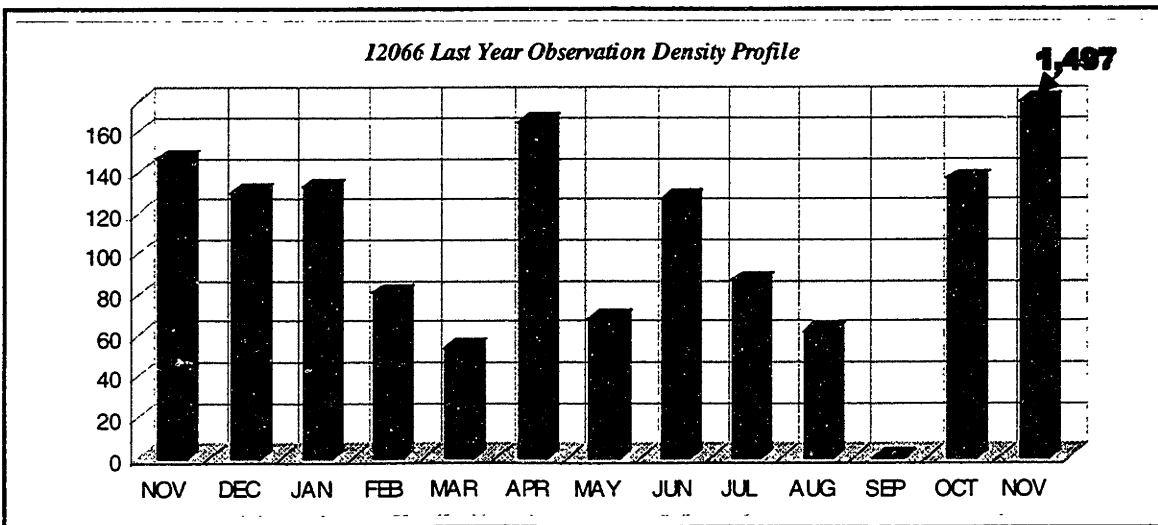


Figure B.32: NSSC-12066 Final Year Monthly Observation Density Profile

B.2.2 Observation Gap Analysis

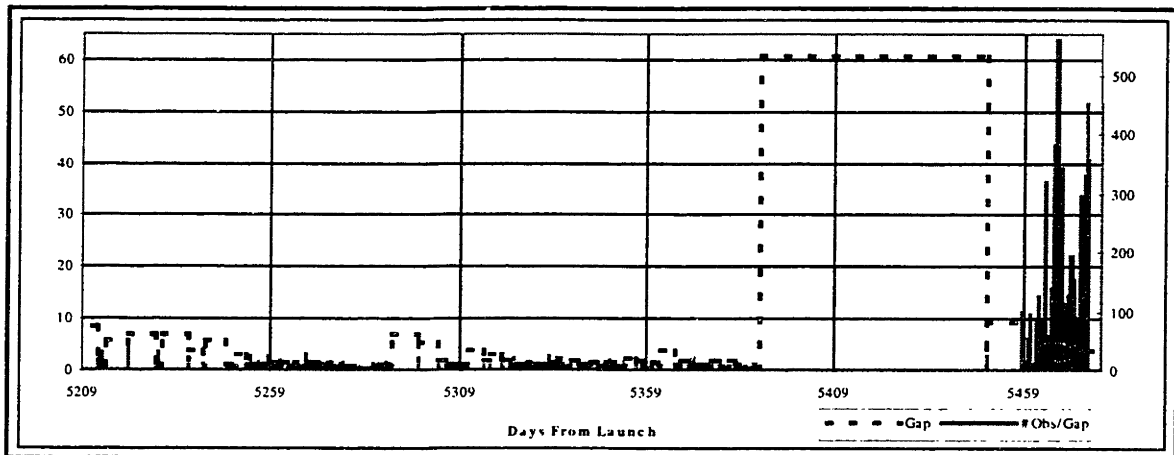


Figure B.33: NSSC-12066 Final Nine Months Observation Gap Analysis Plot

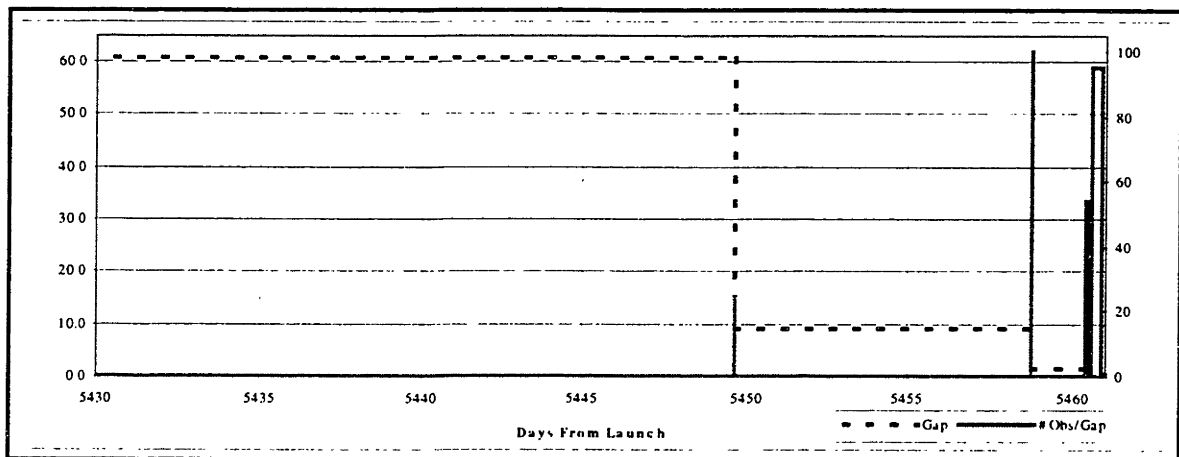


Figure B.34: NSSC-12066 October 1995 Observation Gap Analysis Plot

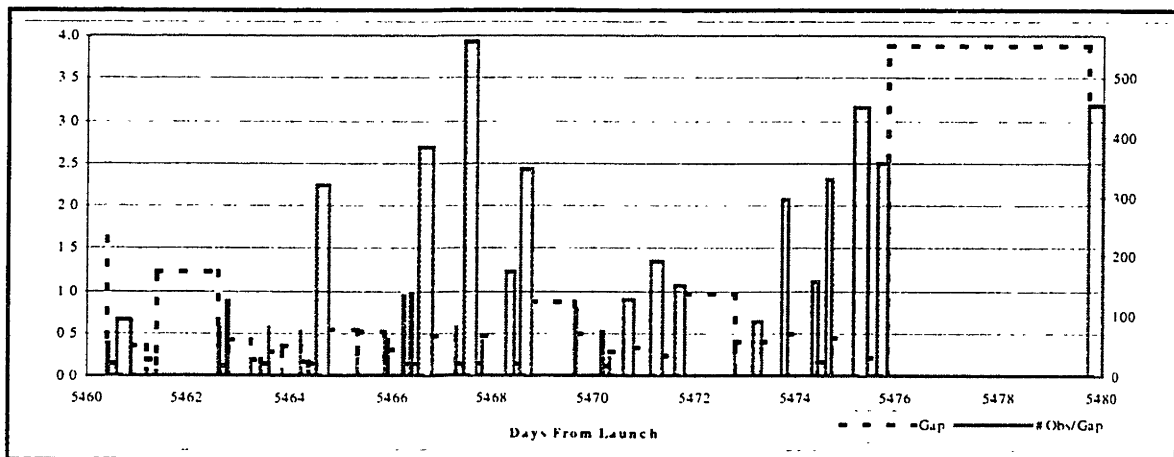


Figure B.35: NSSC-12066 November 1995 Observation Gap Analysis Plot

B.3 NSSC-13112

NSSC-13112 is the rocket body for the Molniya 3-18 satellite, launched on 24 March, 1982. Before its final decay on 14 November, 1992, there were 6,166 observations collected on NSSC-13112. The last observation taken was on 17 May, 1992, almost six months before decay.

B.3.1 Observation Density Analysis

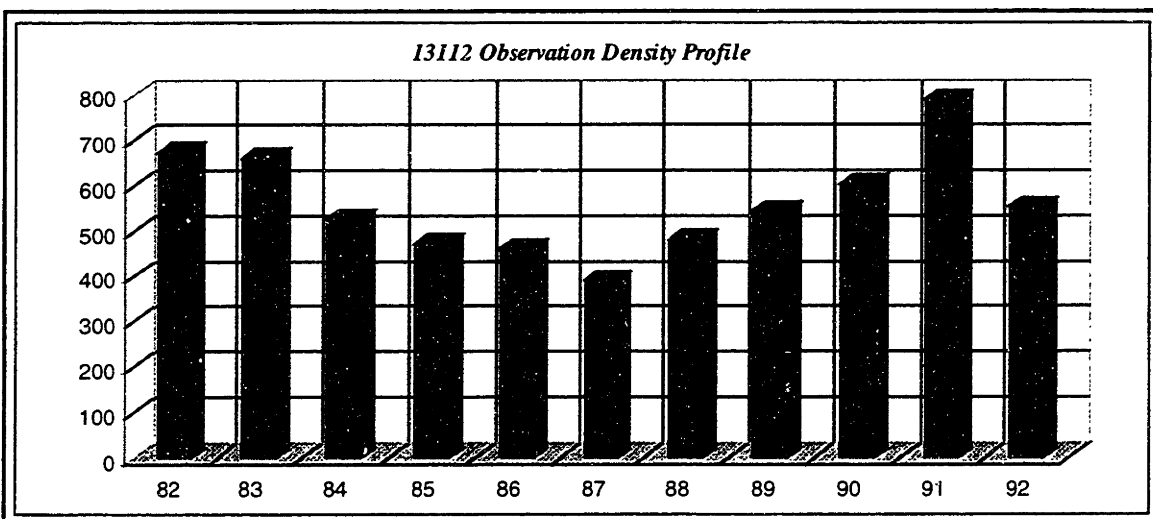


Figure B.36: NSSC-13112 Yearly Observation Density Profile

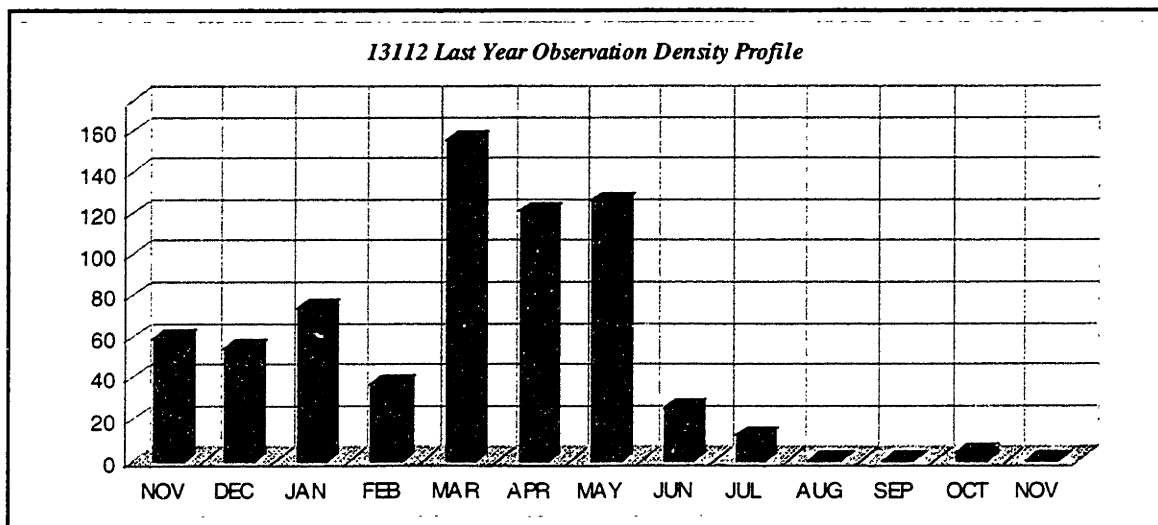


Figure B.37: NSSC-13112 Final Year Monthly Observation Density Profile

B.3.2 Observation Gap Analysis

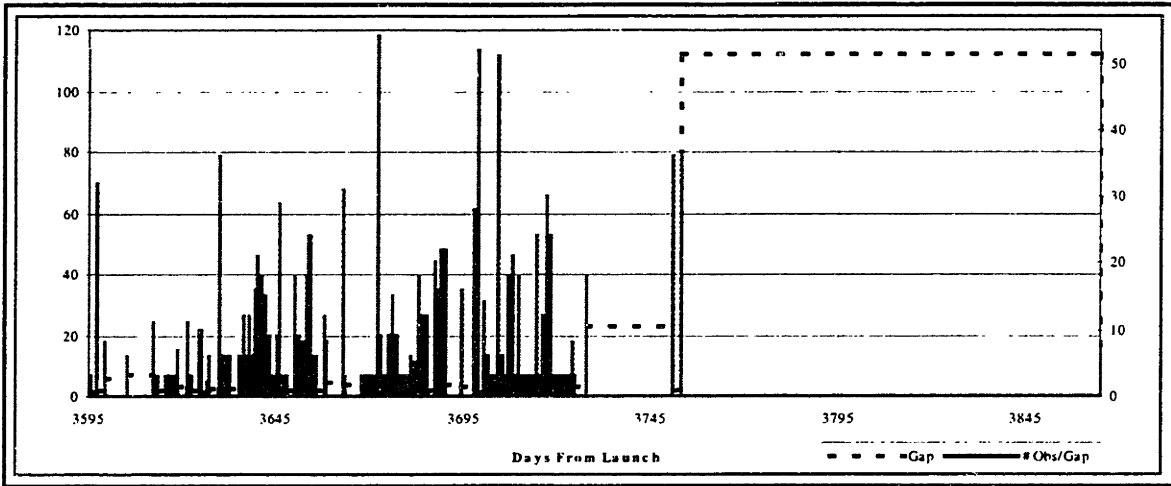


Figure B.38: NSSC-13112 Final Nine Months Observation Gap Analysis Plot

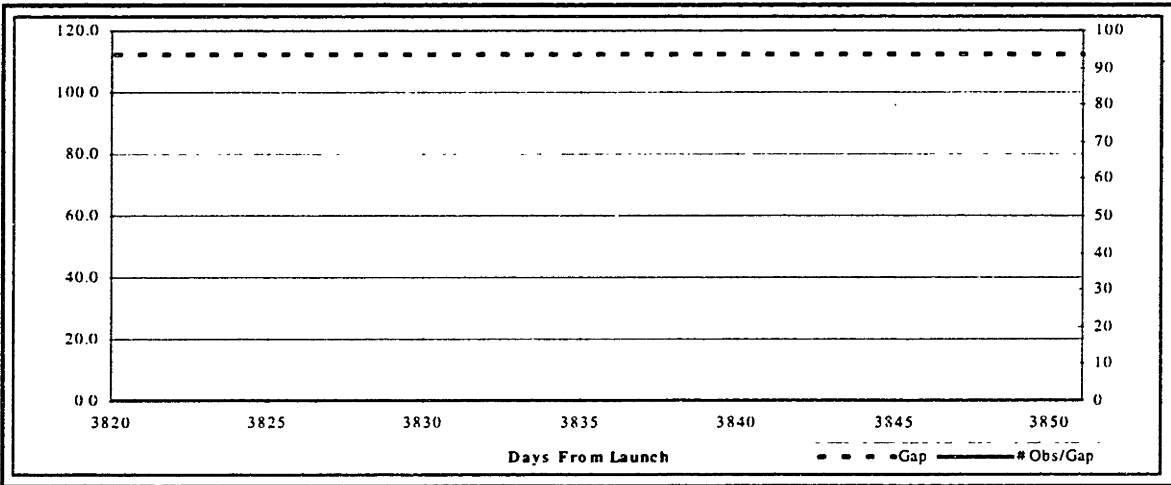


Figure B.39: NSSC-13112 Month Before Decay Observation Gap Analysis Plot

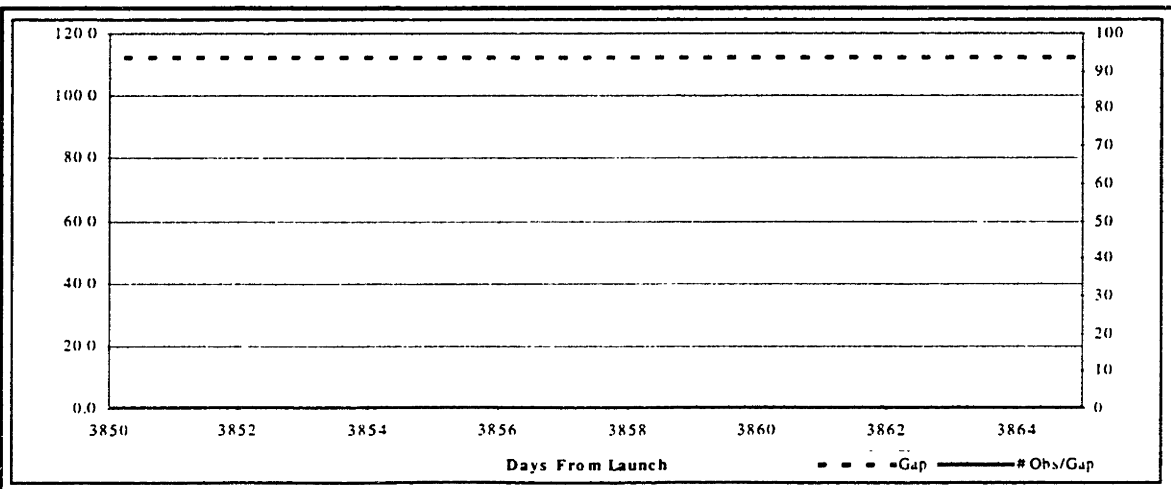


Figure B.40: NSSC-13112 Final Month Observation Gap Analysis Plot

B.4 NSSC-13253

NSSC-13253 is the rocket body for the Molniya 1-54 satellite, launched on 28 May, 1982. Before its final decay on 2 April, 1993, there were 7,879 observations collected on NSSC-13253. The last observation taken was on 2 April, 1993, the day of final decay.

B.4.1 Observation Density Analysis

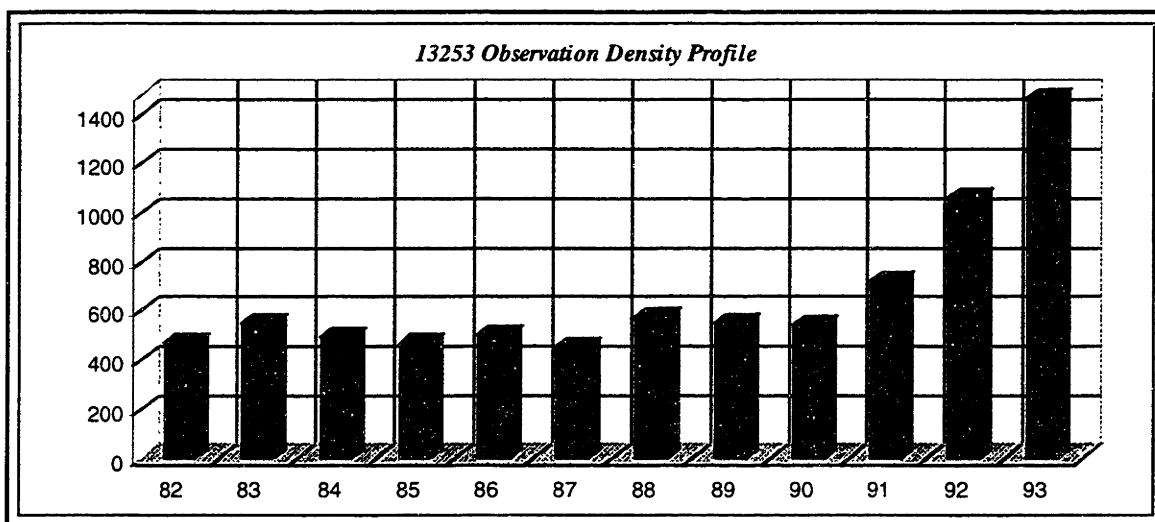


Figure B.41: NSSC-13253 Yearly Observation Density Profile

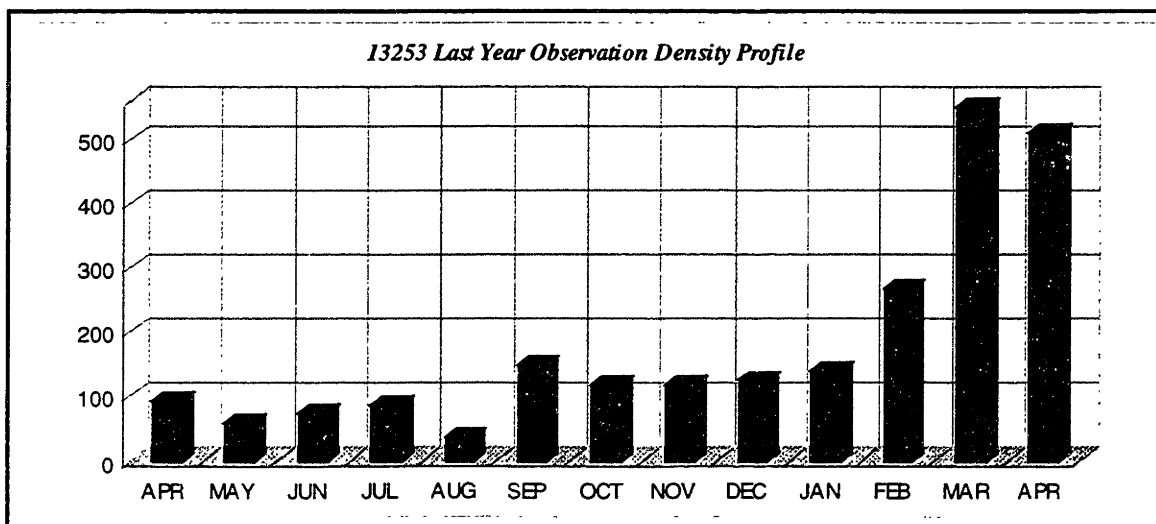


Figure B.42: NSSC-13253 Final Year Monthly Observation Density Profile

B.4.2 Observation Gap Analysis

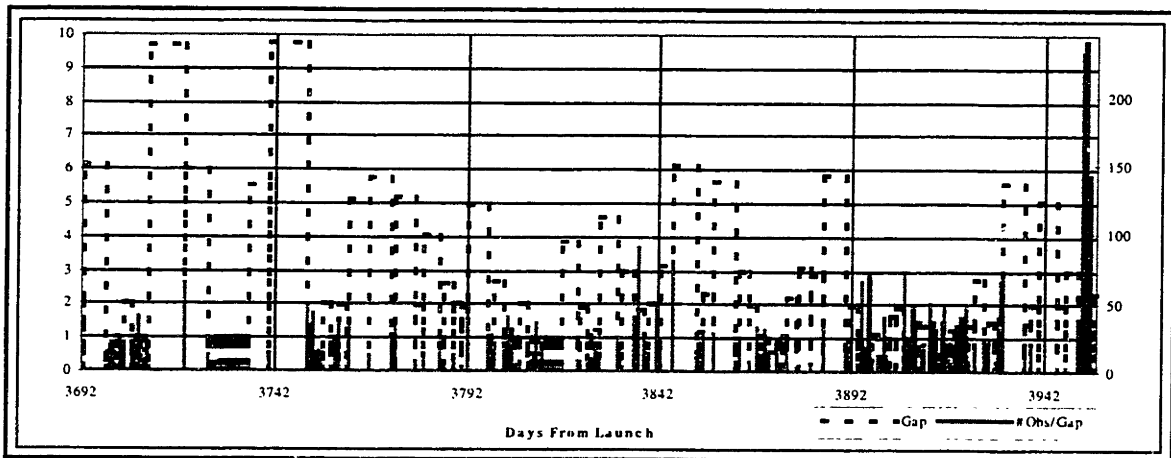


Figure B.43: NSSC-13253 Final Nine Months Observation Gap Analysis Plot

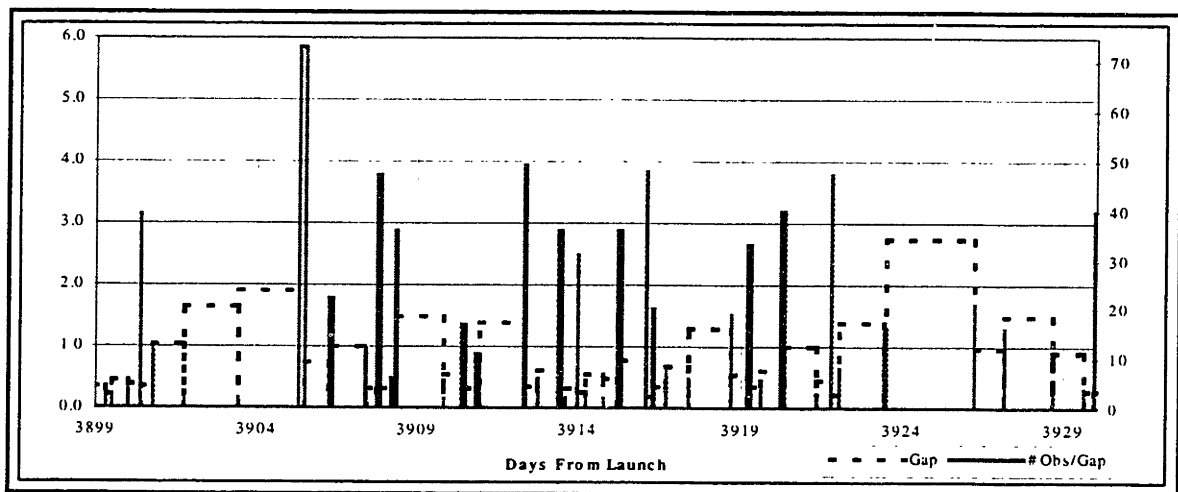


Figure B.44: NSSC-13253 February 1993 Observation Gap Analysis Plot

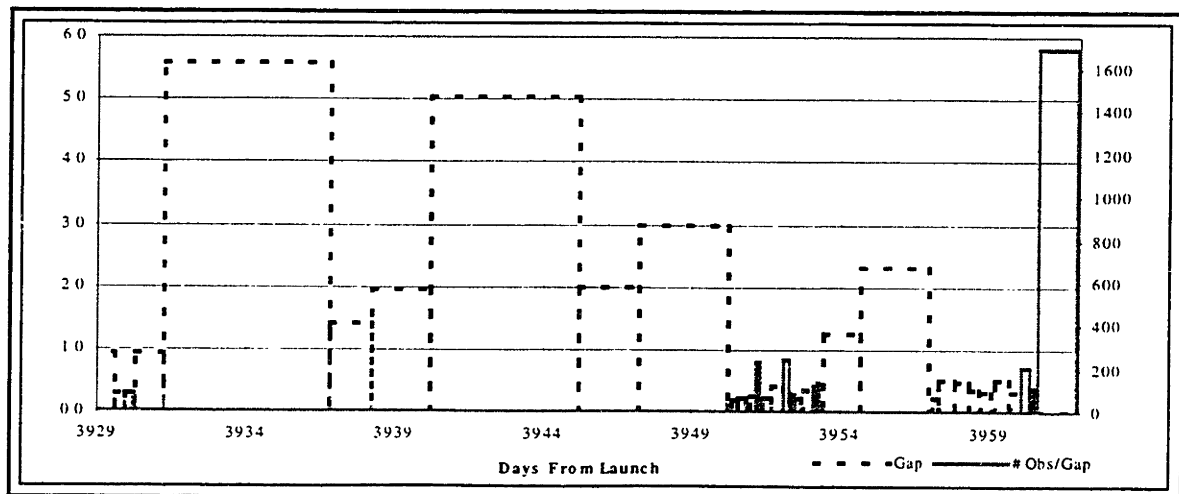


Figure B.45: NSSC-13253 March - April 1993 Observation Gap Analysis Plot

B.5 NSSC-14199

NSSC-14199 is the Molniya 1-58 satellite, launched on 19 July, 1983. Before its final decay on 23 February, 1995, there were 35,644 observations collected on Molniya 1-58. The last observation taken was on the day of final decay, 23 February, 1995.

B.5.1 Observation Density Analysis

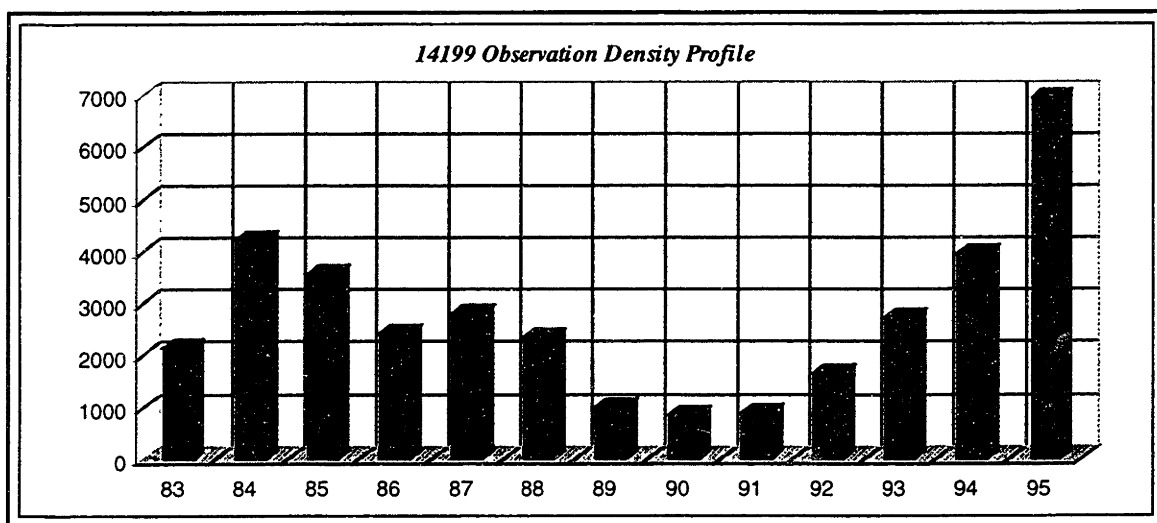


Figure B.46: NSSC-14199 Yearly Observation Density Profile

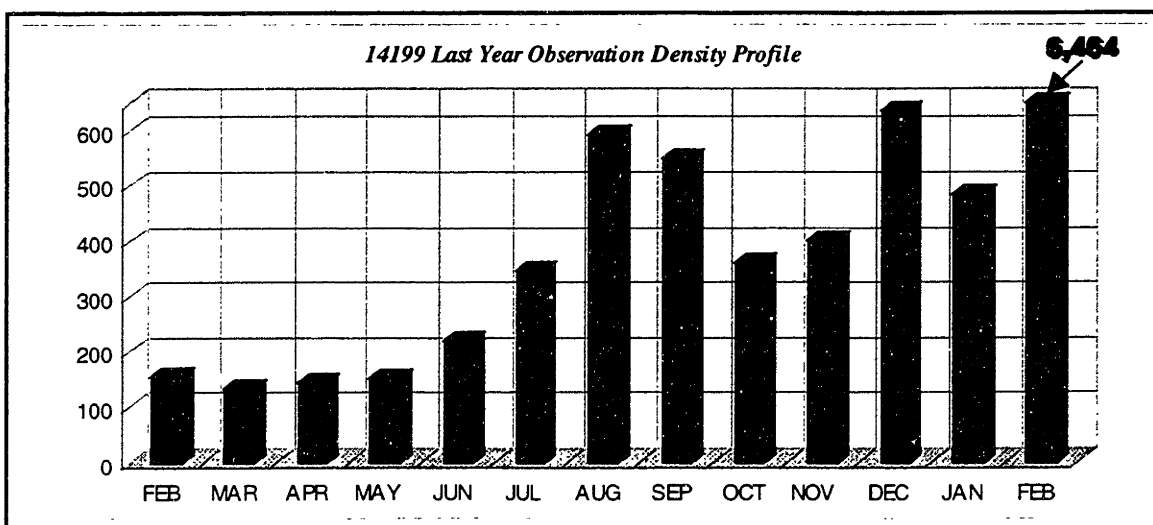


Figure B.47: NSSC-14199 Final Year Monthly Observation Density Profile

B.5.2 Observation Gap Analysis

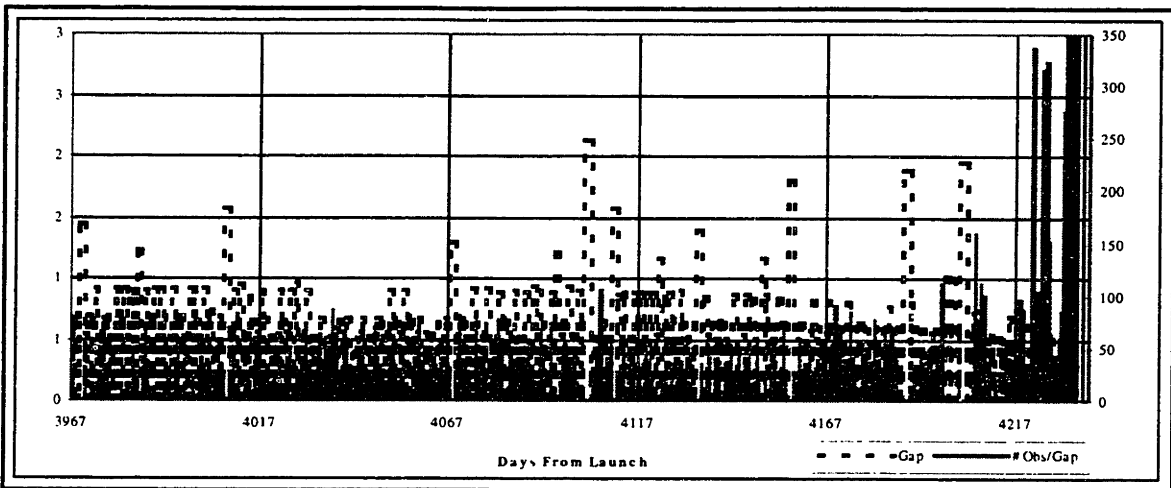


Figure B.48: NSSC-14199 Final Nine Months Observation Gap Analysis Plot

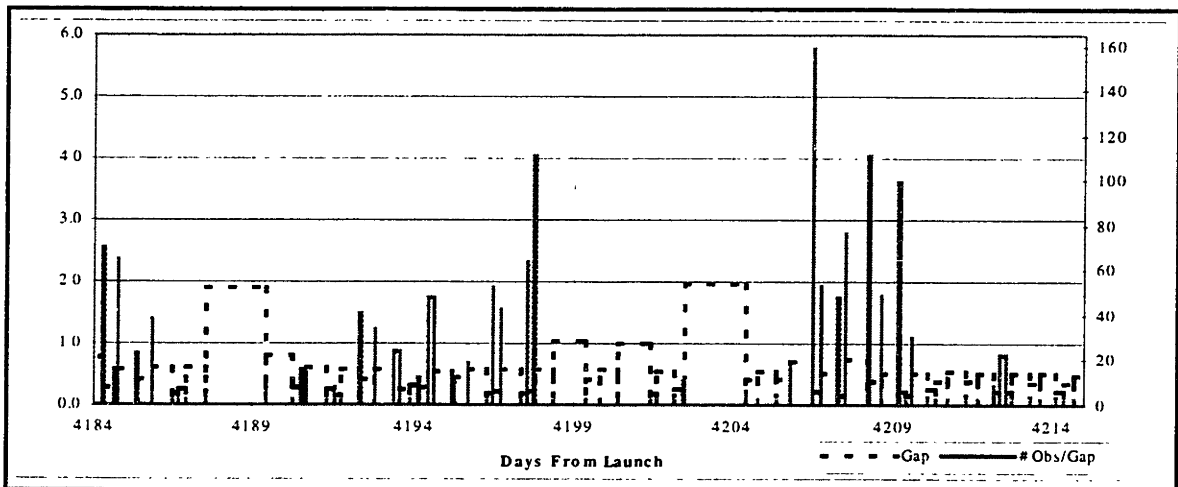


Figure B.49: NSSC-14199 January 1995 Observation Gap Analysis Plot

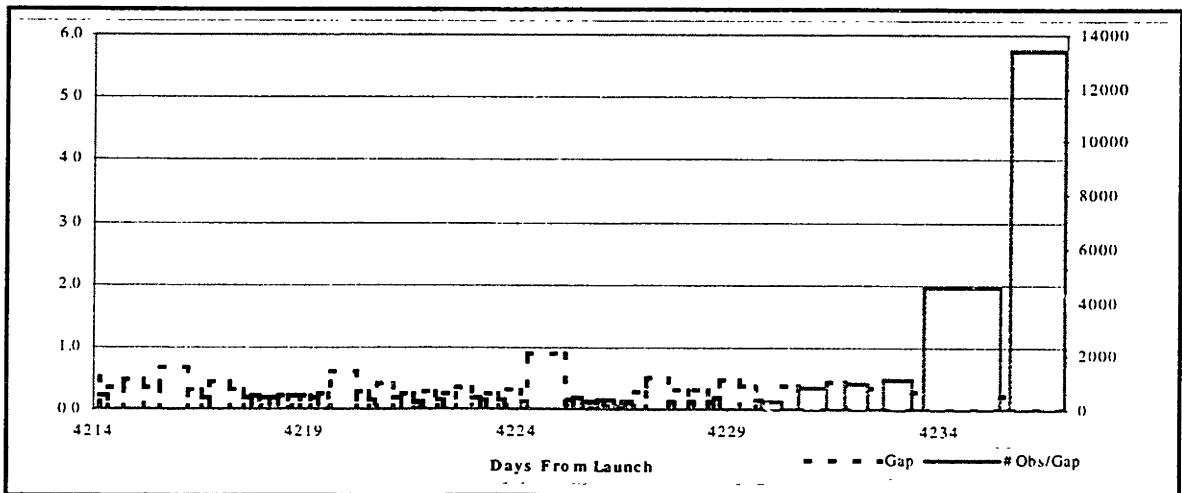


Figure B.50: NSSC-14199 February 1995 Observation Gap Analysis Plot

B.6 NSSC-14582

NSSC-14582 is the rocket body for the Molniya 3-22 satellite, launched on 21 December, 1983. Before its final decay on 24 April, 1994, there were 19,042 observations collected on NSSC-14582. The last observation taken was on the day of final decay, 24 April, 1994.

B.6.1 Observation Density Analysis

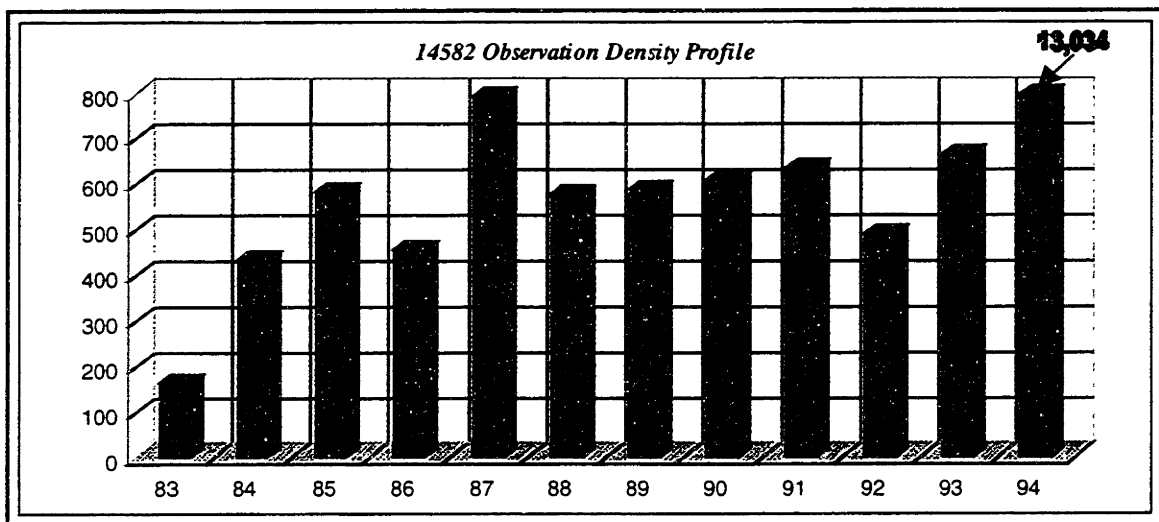


Figure B.51: NSSC-14582 Yearly Observation Density Profile

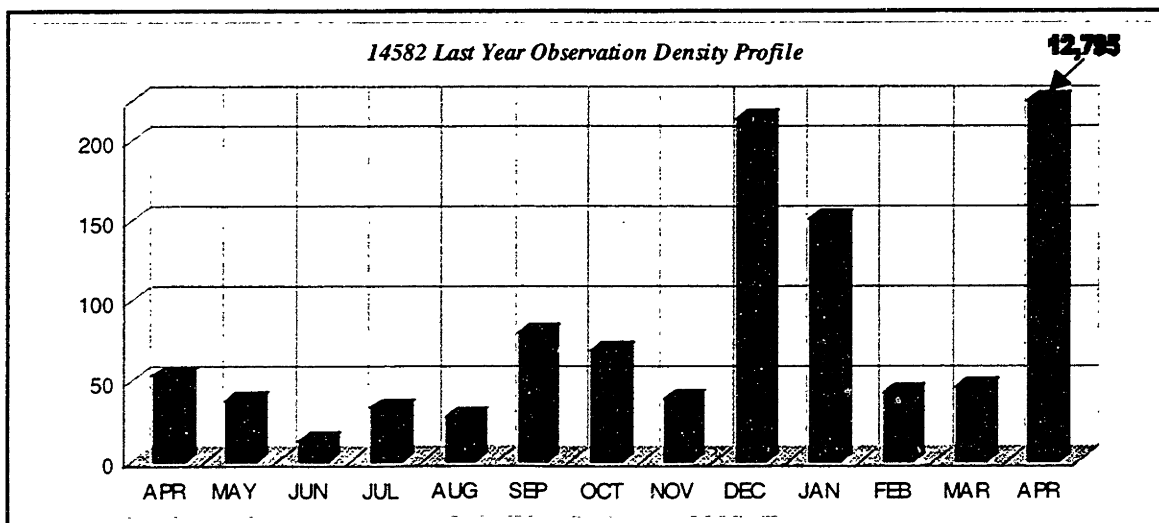


Figure B.52: NSSC-14582 Final Year Monthly Observation Density Profile

B.6.2 Observation Gap Analysis

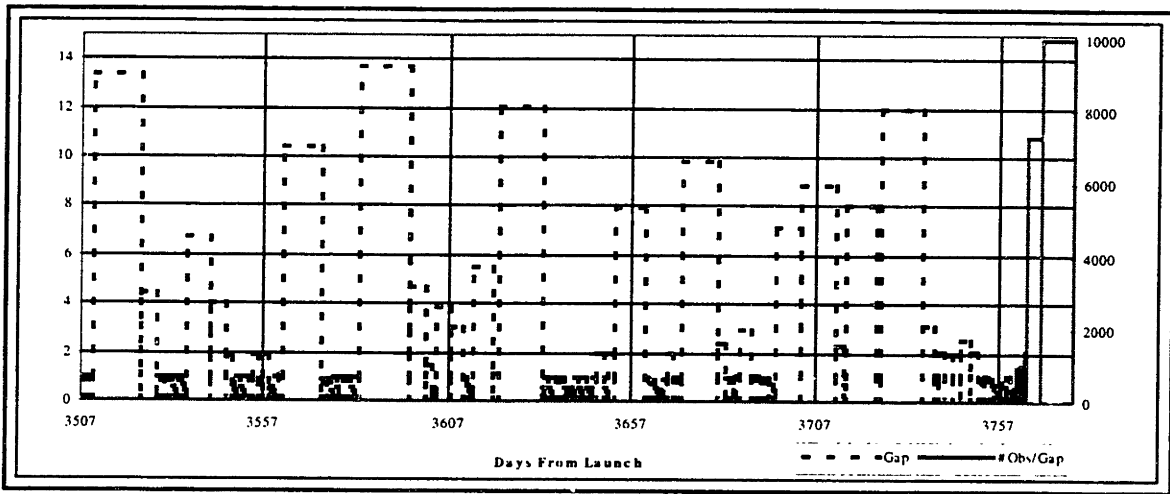


Figure B.53: NSSC-14582 Final Nine Months Observation Gap Analysis Plot

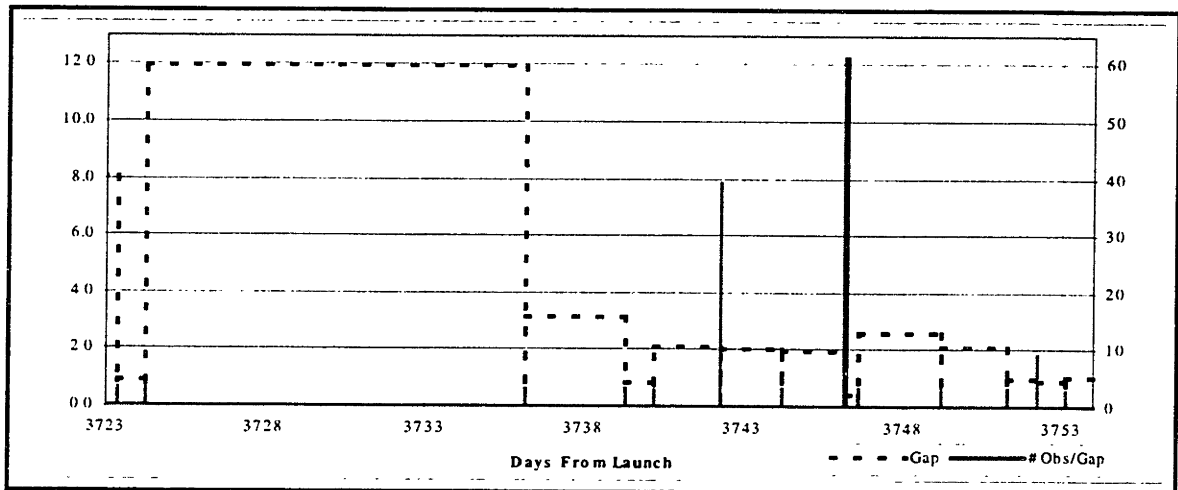


Figure B.54: NSSC-14582 March 1994 Observation Gap Analysis Plot

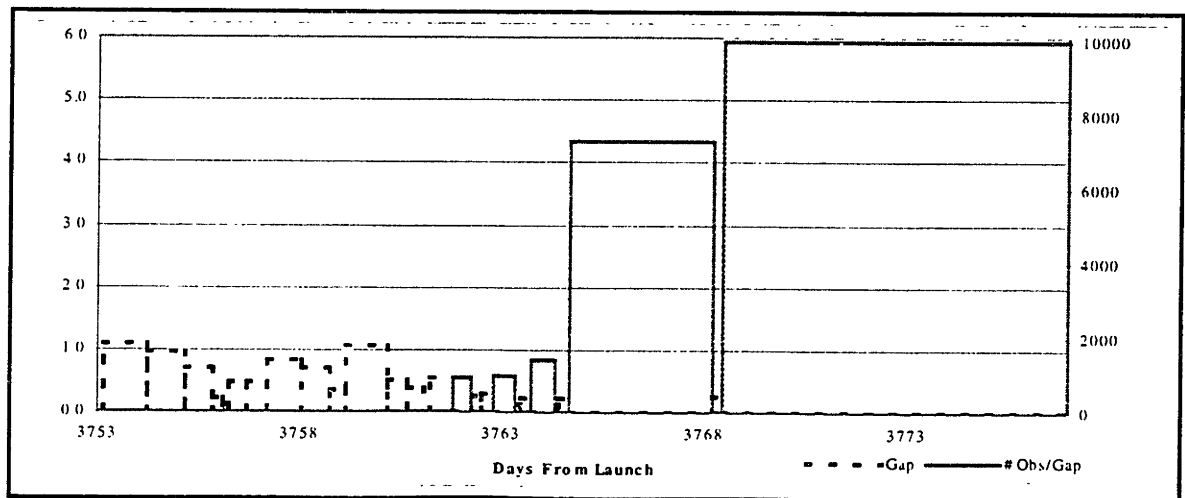


Figure B.55: NSSC-14582 April 1994 Observation Gap Analysis Plot

B.7 NSSC-14830

NSSC-14830 is the rocket body for the Molniya 1-60 satellite, launched on 16 March, 1984. Before its final decay on 22 September, 1994, there were 9,733 observations collected on NSSC-14830. The last observation taken was on the day of final decay, 22 September, 1994.

B.7.1 Observation Density Analysis

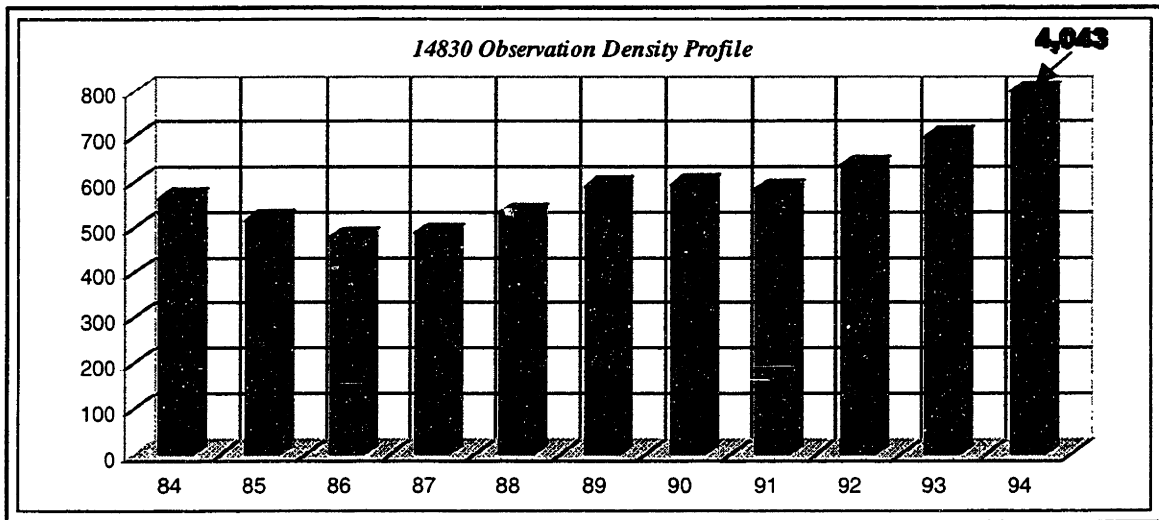


Figure B.56: NSSC-14830 Yearly Observation Density Profile

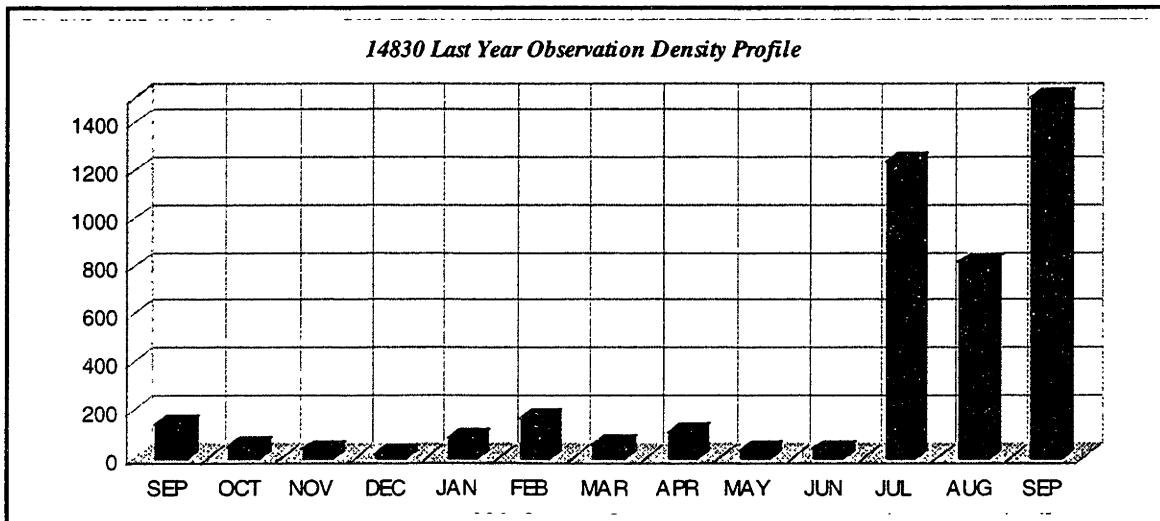


Figure B.57: NSSC-14830 Final Year Monthly Observation Density Profile

B.7.2 Observation Gap Analysis

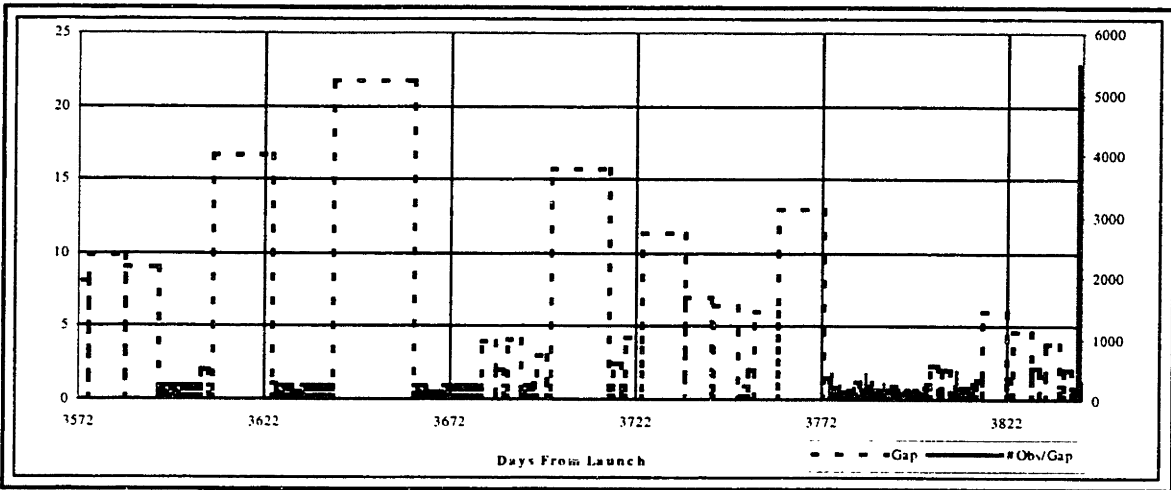


Figure B.58: NSSC-14830 Final Nine Months Observation Gap Analysis Plot

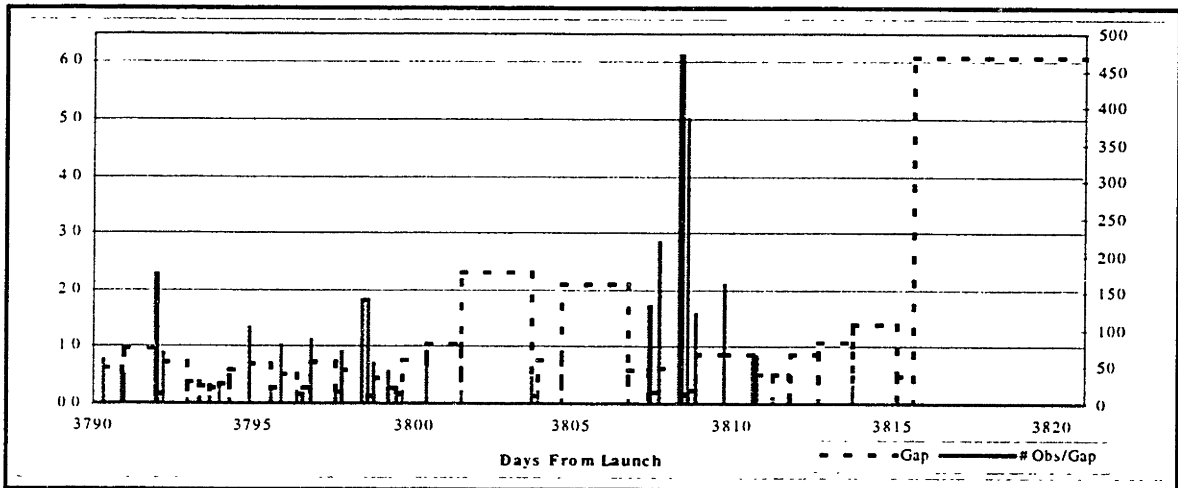


Figure B.59: NSSC-14830 August 1994 Observation Gap Analysis Plot

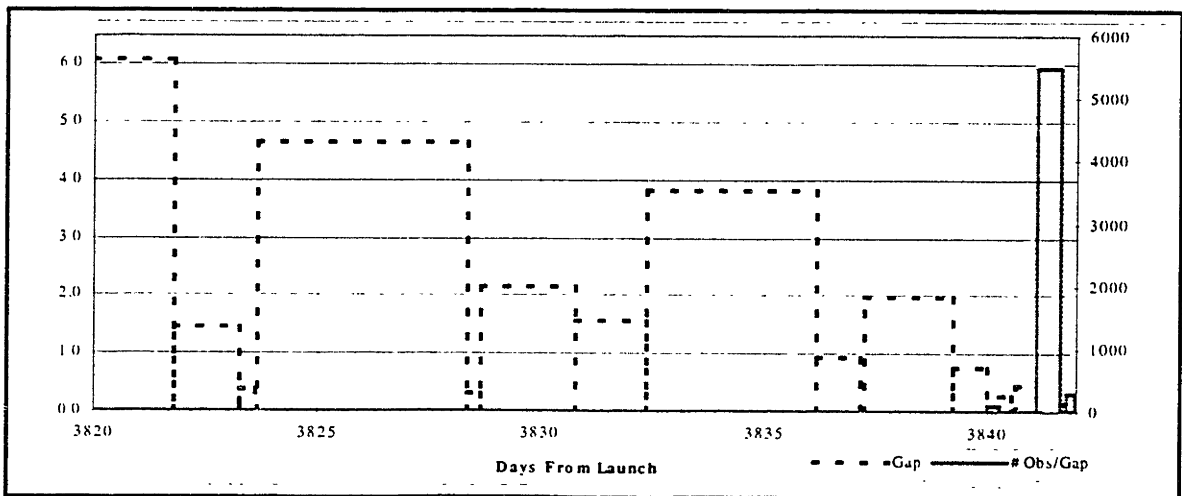


Figure B.60: NSSC-14830 September 1994 Observation Gap Analysis Plot

B.8 NSSC-15481

NSSC-15481 is the rocket body for the Molniya 3-23 satellite, launched on 19 July, 1983. Before its final decay on 24 April, 1995, there were 26,342 observations collected on NSSC-15481. The last observation taken was on the day before final decay, 23 April, 1995.

B.8.1 Observation Density Analysis

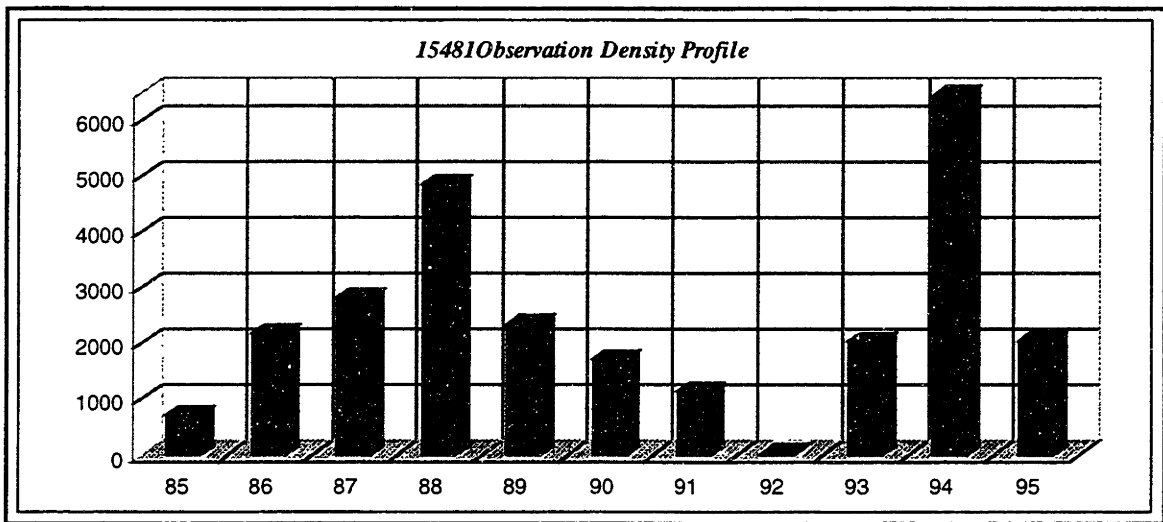


Figure B.61: NSSC-15481 Yearly Observation Density Profile

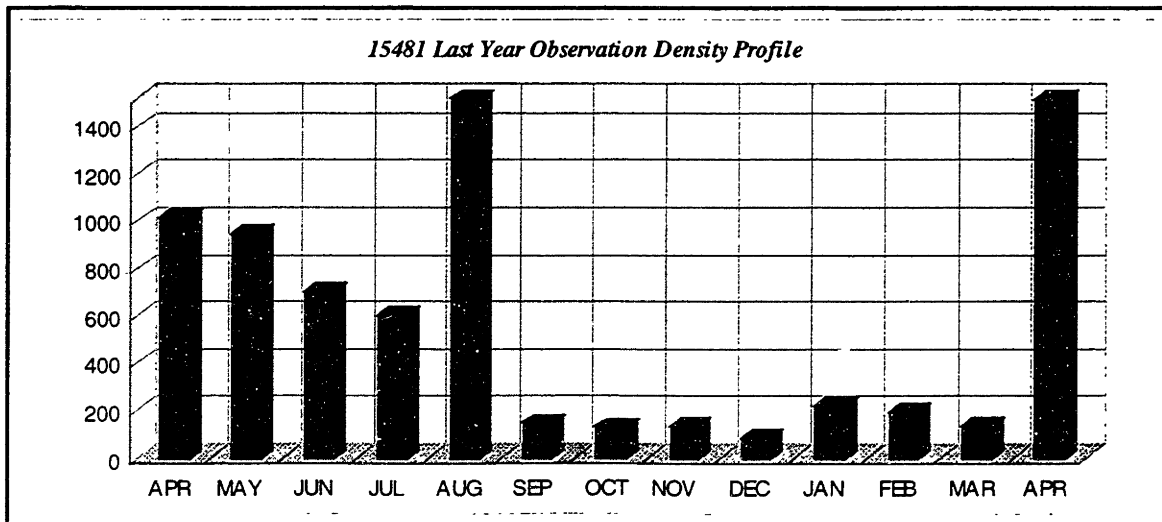


Figure B.62: NSSC-15481 Final Year Monthly Observation Density Profile

B.8.2 Observation Gap Analysis

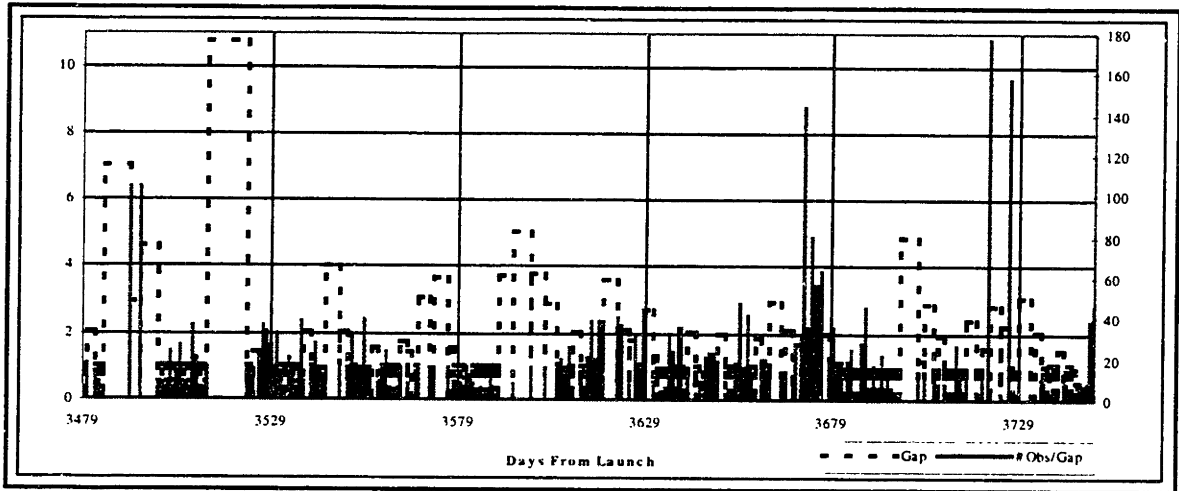


Figure B.63: NSSC-15481 Final Nine Months Observation Gap Analysis Plot

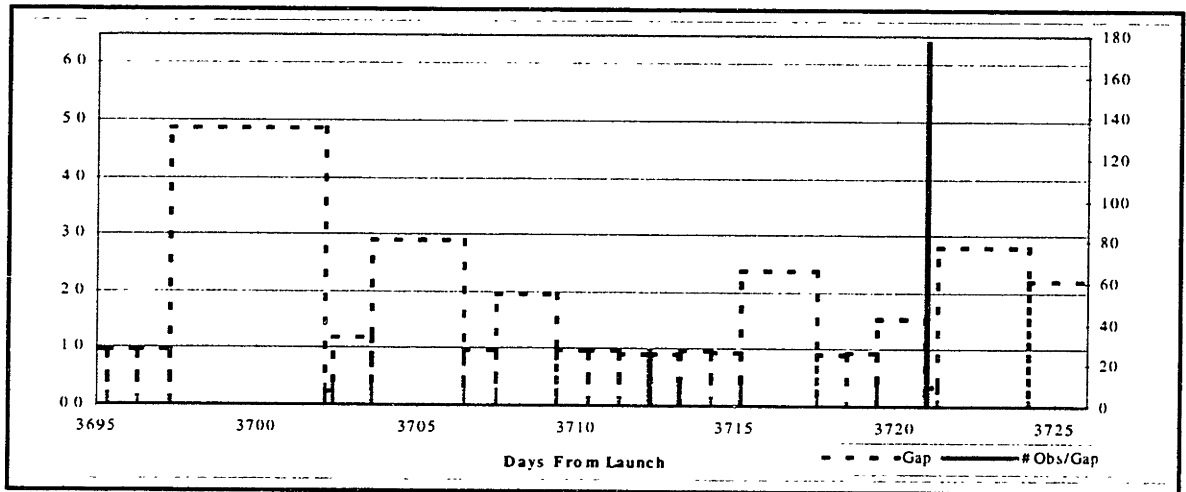


Figure B.64: NSSC-15481 March 1995 Observation Gap Analysis Plot

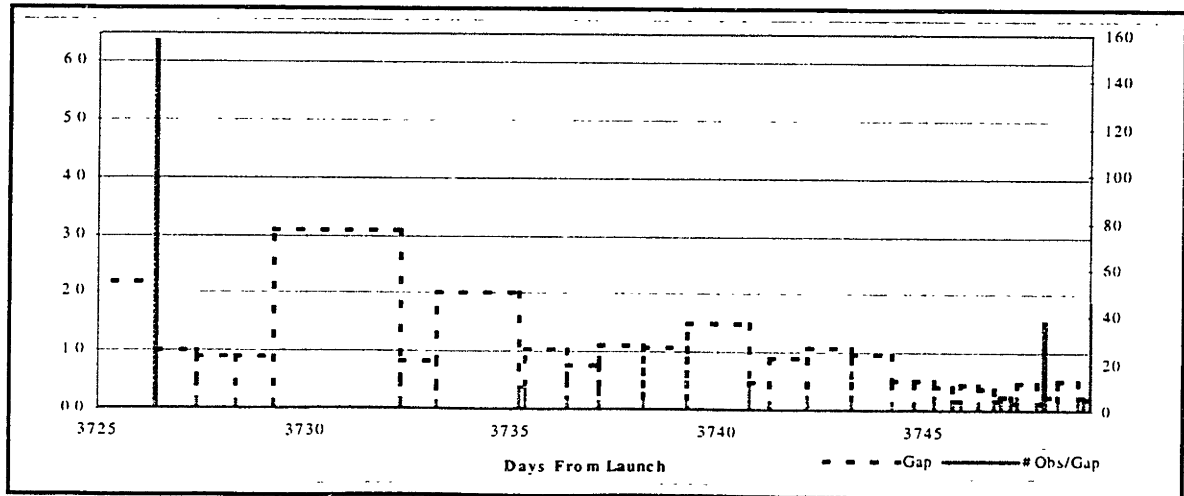


Figure B.65: NSSC-15481 April 1995 Observation Gap Analysis Plot

B.9 NSSC-16805

NSSC-16805 is the rocket body for the Molniya 3-29 satellite, launched on 19 June, 1986. Before its final decay on 21 December, 1996, there were 7,862 observations collected on NSSC-16805. The last observation taken was on the day of final decay, 21 December, 1996.

B.9.1 Observation Density Analysis

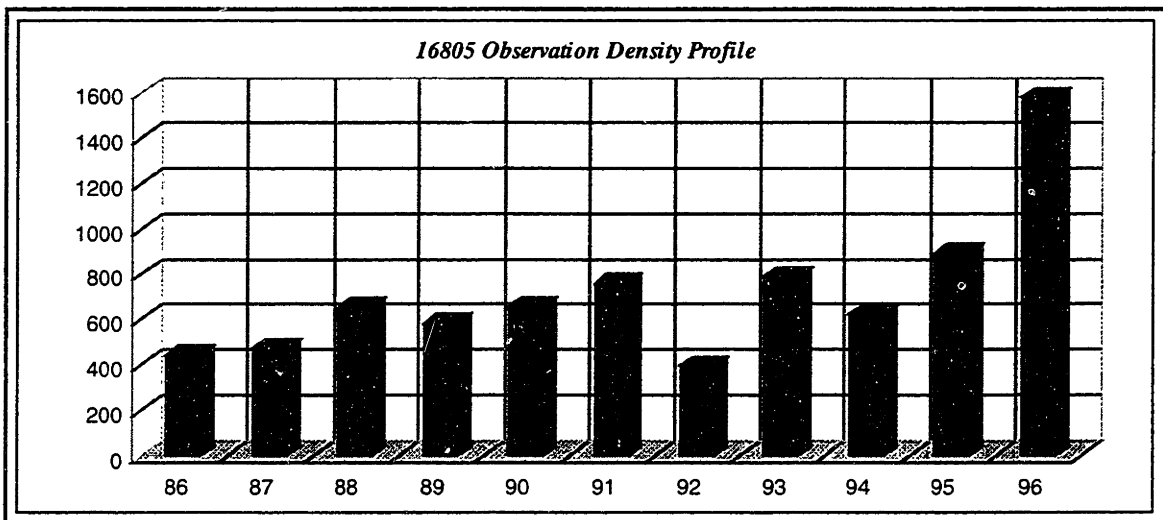


Figure B.66: NSSC-16805 Yearly Observation Density Profile

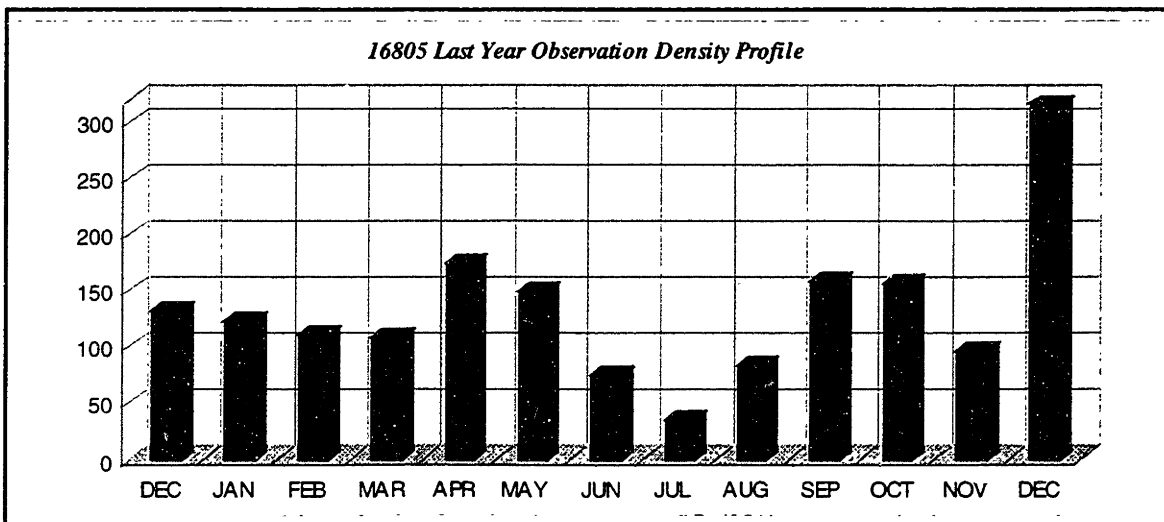


Figure B.67: NSSC-16805 Final Year Monthly Observation Density Profile

B.9.2 Observation Gap Analysis

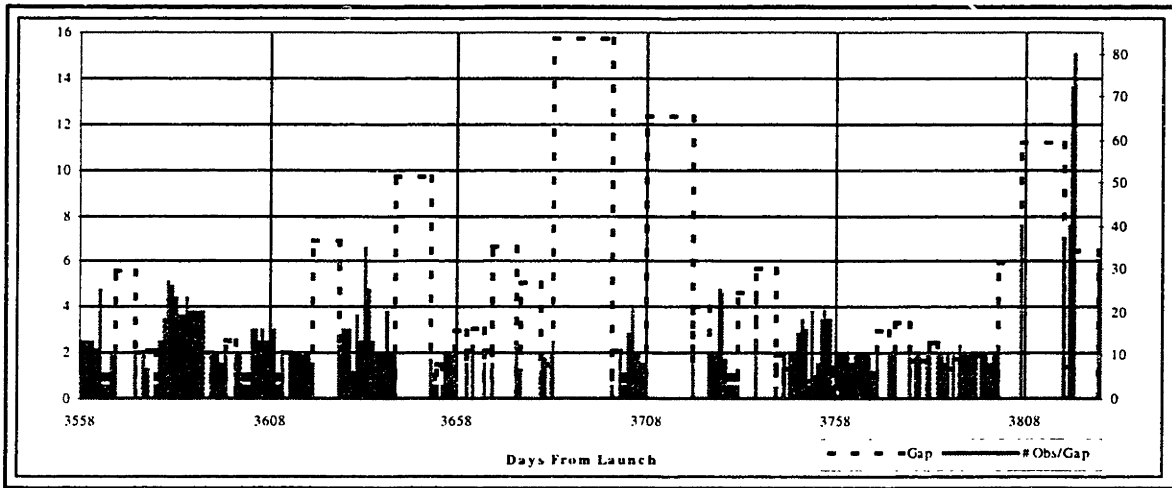


Figure B.68: NSSC-16805 Final Nine Months Observation Gap Analysis Plot

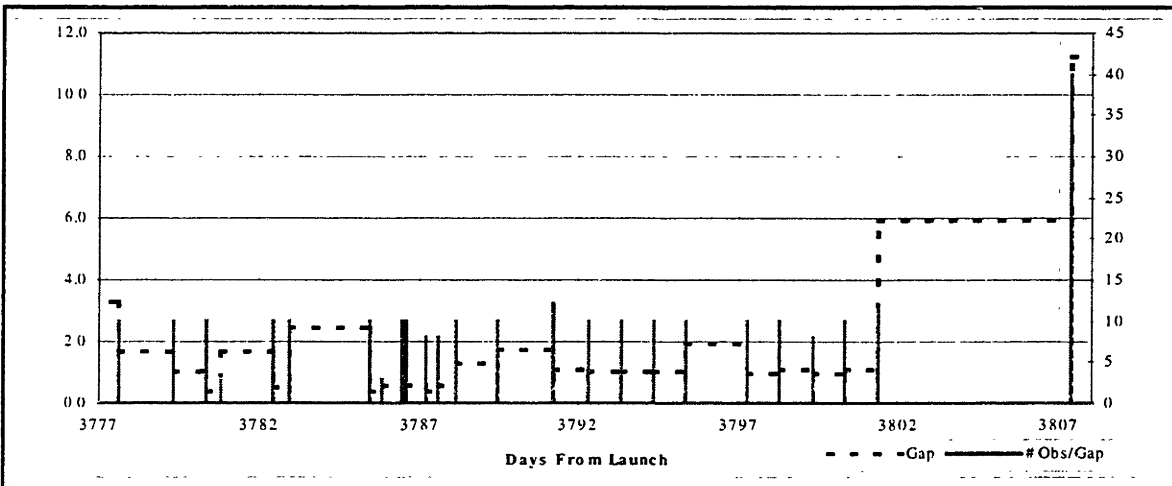


Figure B.69: NSSC-16805 November 1996 Observation Gap Analysis Plot

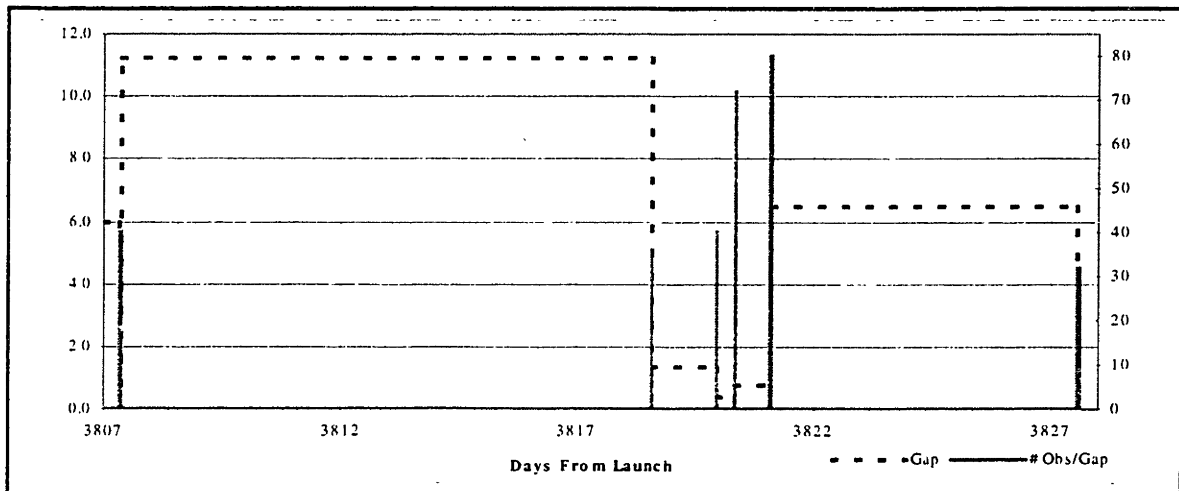


Figure B.70: NSSC-16805 December 1996 Observation Gap Analysis Plot

B.10 NSSC-16889

NSSC-16889 is the rocket body for the Molniya 1-67 satellite, launched on 30 July, 1986. Before its final decay on 14 October, 1996, there were 13,292 observations collected on NSSC-16889. The last observation taken was on the day of final decay, 14 October, 1996.

B.10.1 Observation Density Analysis

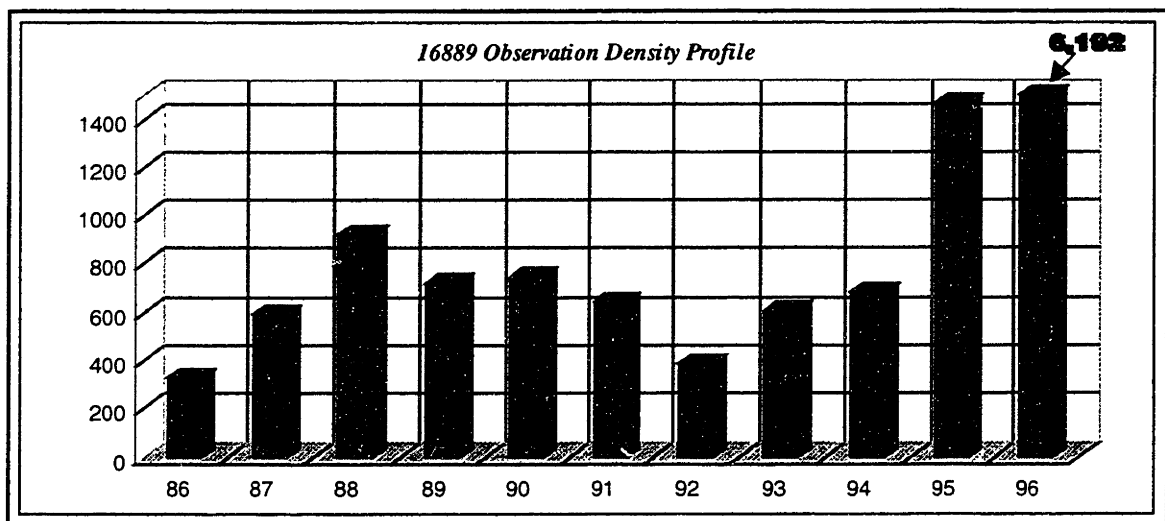


Figure B.71: NSSC-16889 Yearly Observation Density Profile

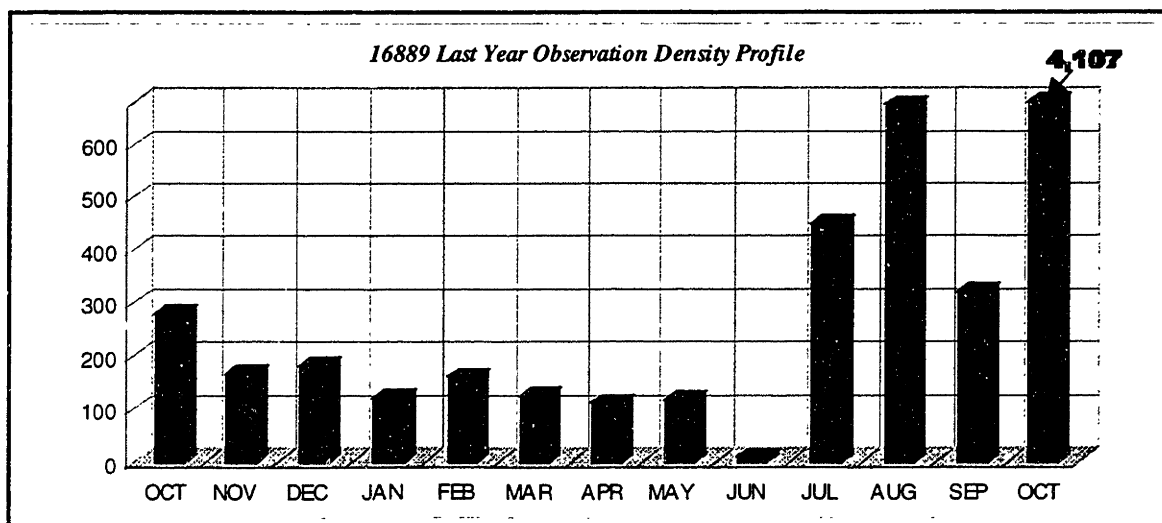


Figure B.72: NSSC-16889 Final Year Monthly Observation Density Profile

B.10.2 Observation Gap Analysis

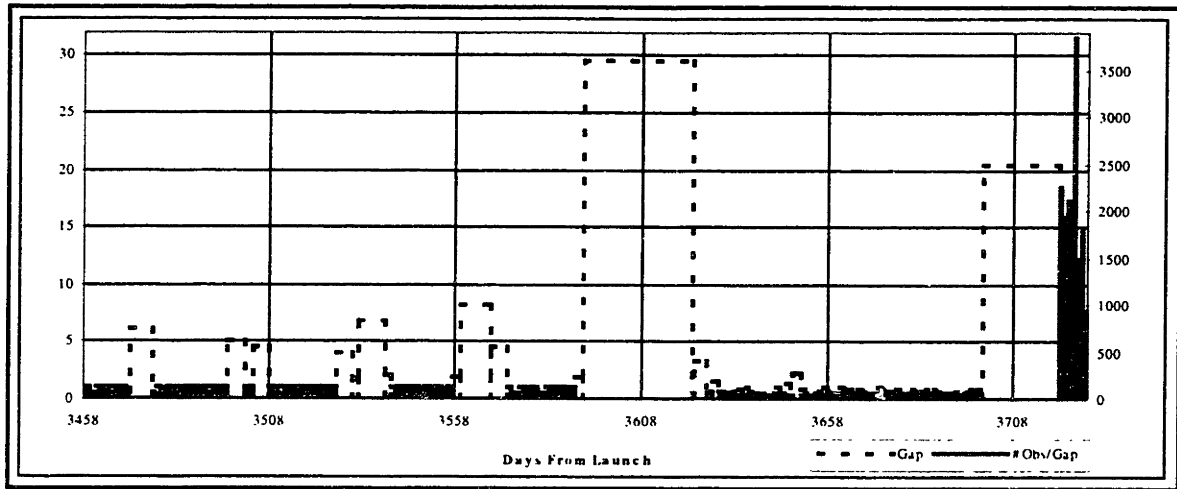


Figure B.73: NSSC-16889 Final Nine Months Observation Gap Analysis Plot

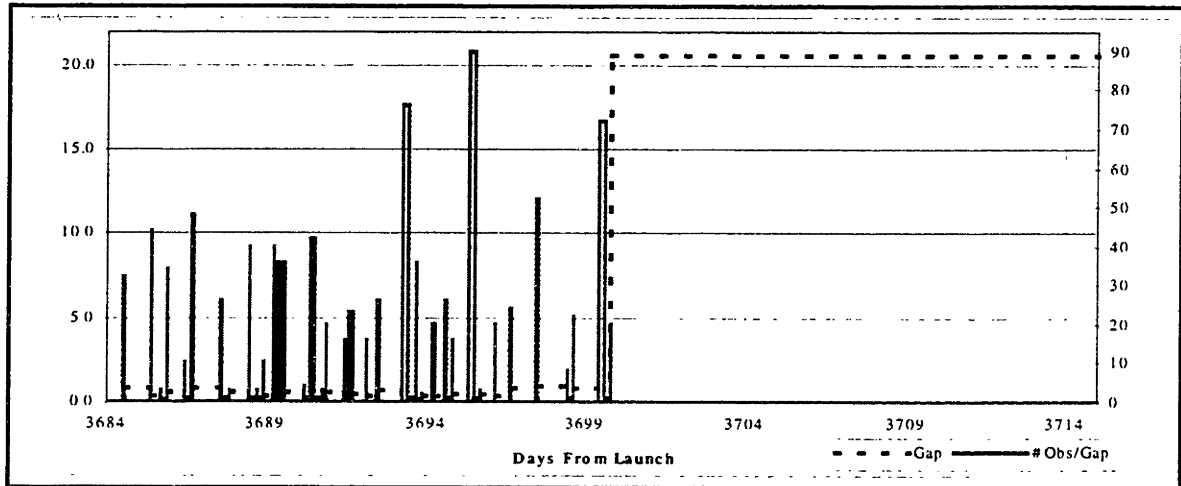


Figure B.74: NSSC-16889 September 1996 Observation Gap Analysis Plot

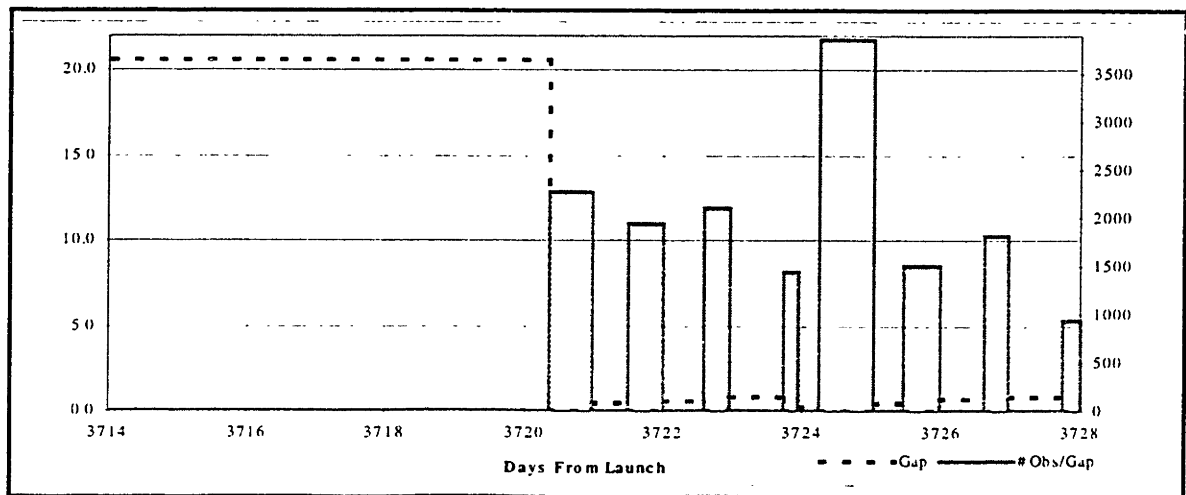


Figure B.75: NSSC-16889 October 1996 Observation Gap Analysis Plot

B.11 Data Analysis Summary

After analyzing the observation gap and density distributions, and given that a rocket body was more desirable than a satellite for study (due to the simplified aerodynamics during reentry), the NSSC-13390 rocket body was chosen as the primary test object for this investigation. The Molniya 1-58 satellite (NSSC-14199) demonstrated a better data distribution than any of the objects but, given that it was a satellite and executed a non-standard station keeping maneuver while still in the mission phase (see Chapter 1), it was not used. The reasoning behind using NSSC-13390 over the remaining eight objects is listed in Table B.1.

Table B.1: NSSC-13390 Object Suitability Comparison

<i>NORAD ID #</i>	<i>Description</i>	<i>Number Obs</i>
<i>NSSC-13390</i>	<i>Chosen for analysis due to well distributed and dense observation distribution</i>	<i>21,488</i>
NSSC-12066	Satellite; No observations for 1987; Very few observations during last year (best seen in Figure B.*7)	26,480
NSSC-13112	Low number of observations, specifically, 4 total observations for the last four months of the object's lifetime	6,166
NSSC-13253	Low total number of observations and low number of observations per month during the final year.	7,879
NSSC-14199	Satellite; Good density distribution but with non-standard station keeping maneuver	35,644
NSSC-14582	Although it has a large number of observations in the last month, D-2 and D-1 months have less than 50	19,042
NSSC-14830	Only 4,000 obs during the final year; extremely poor distribution during the final year	9,733
NSSC-15481	No obs for 1992; very few obs during the SEP 94 – MAR 95 time frame	26,342
NSSC-16805	Only 1,600 obs during the final year; extremely poor distribution during the final year	7,862
NSSC-16889	Poor final year distribution; Only 9 obs for June 96, four months before decay	13,292

To better delineate the differences in observation distribution, Figures B.76, B.77, B.78 and B.79 display comparative bar charts of the yearly and monthly

distribution. Figure B.80 is a close up of B.79 which truncates the satellites with a large number of observations in the last six months (listed in Table B.2, where shaded cells are truncated) to highlight the differences in preceding months between the satellites. NSSC-13390 is in black to accentuate contrast and identify a standard by which the other objects can be compared. In addition, truncated (at 4,000 and 700 observations respectively) contour plots are shown in Figures B.81 and B.82 to afford the reader a different view of the data.

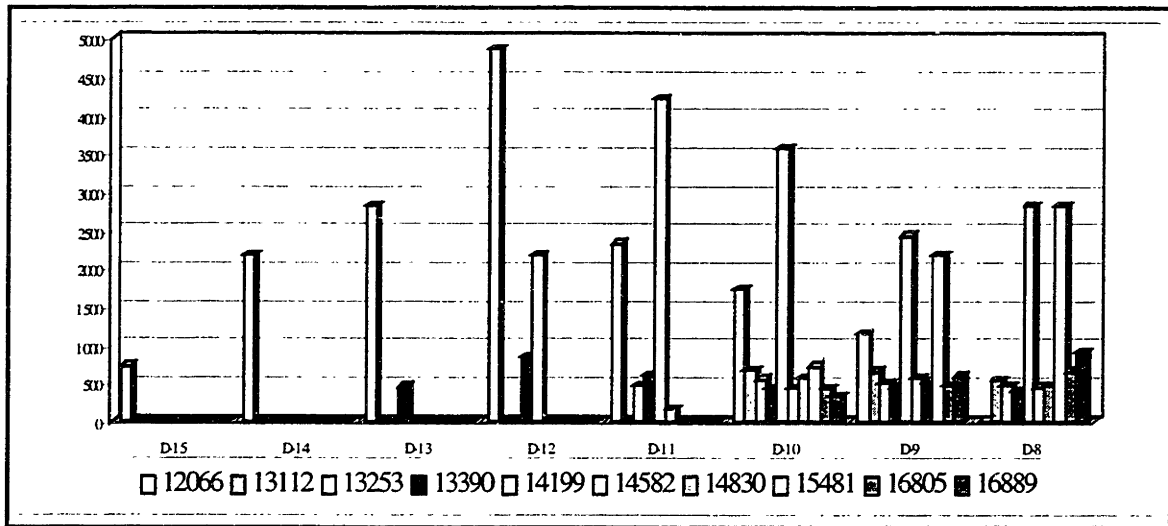


Figure B.76: Yearly Observation Density Comparison Plot: 15 - 8 Years Before Decay

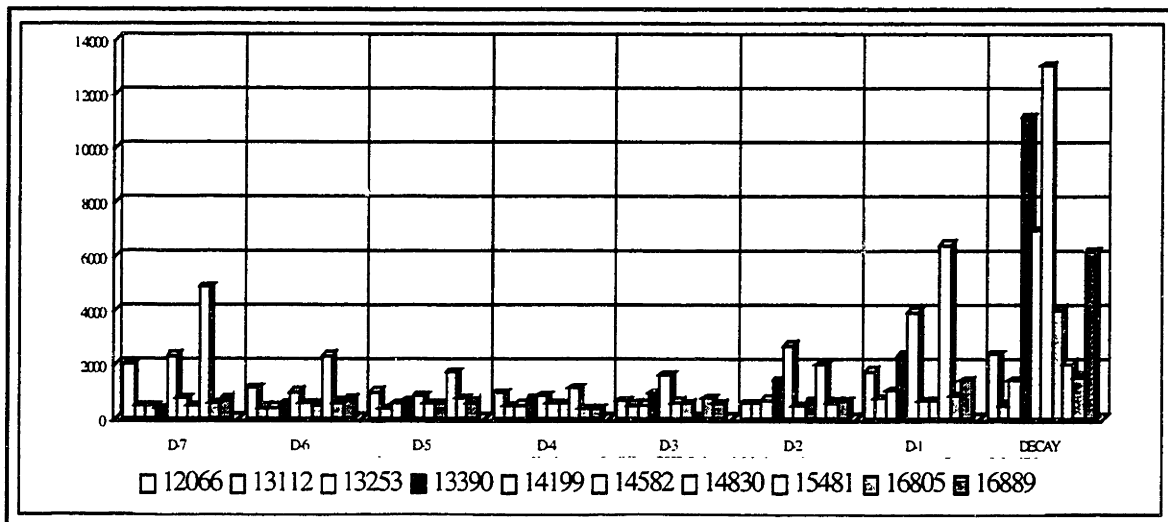


Figure B.77: Yearly Observation Density Comparison Plot: Last 7 Years of Decay

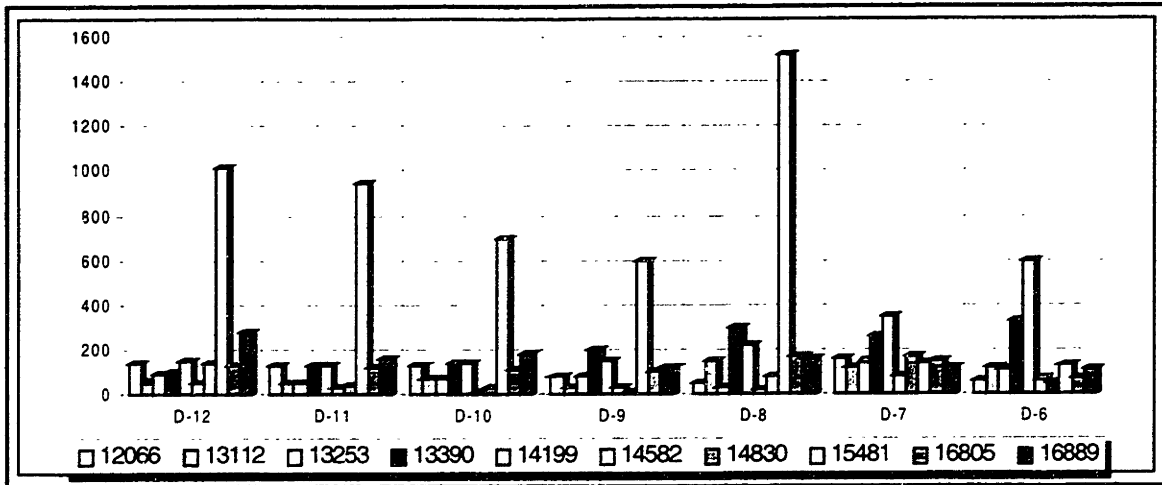


Figure B.78: 12 – 7 Months Before Decay Monthly Observation Density Comparison Plot

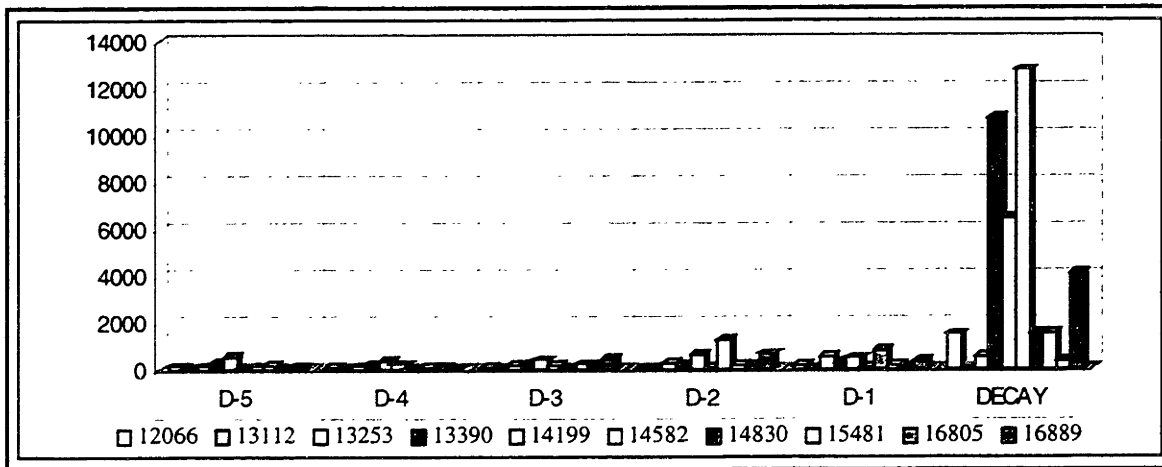


Figure B.79: Final Six Months Monthly Observation Density Comparison Plot

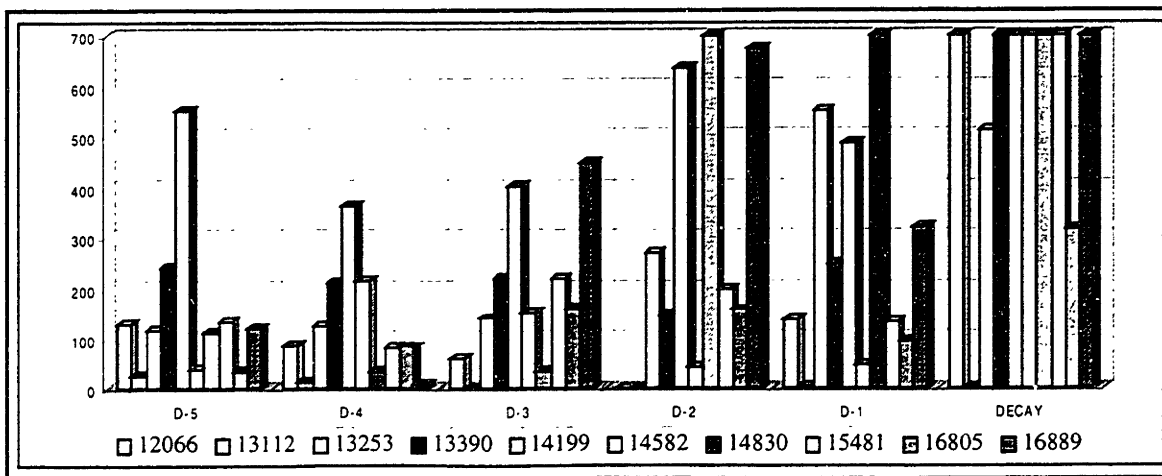
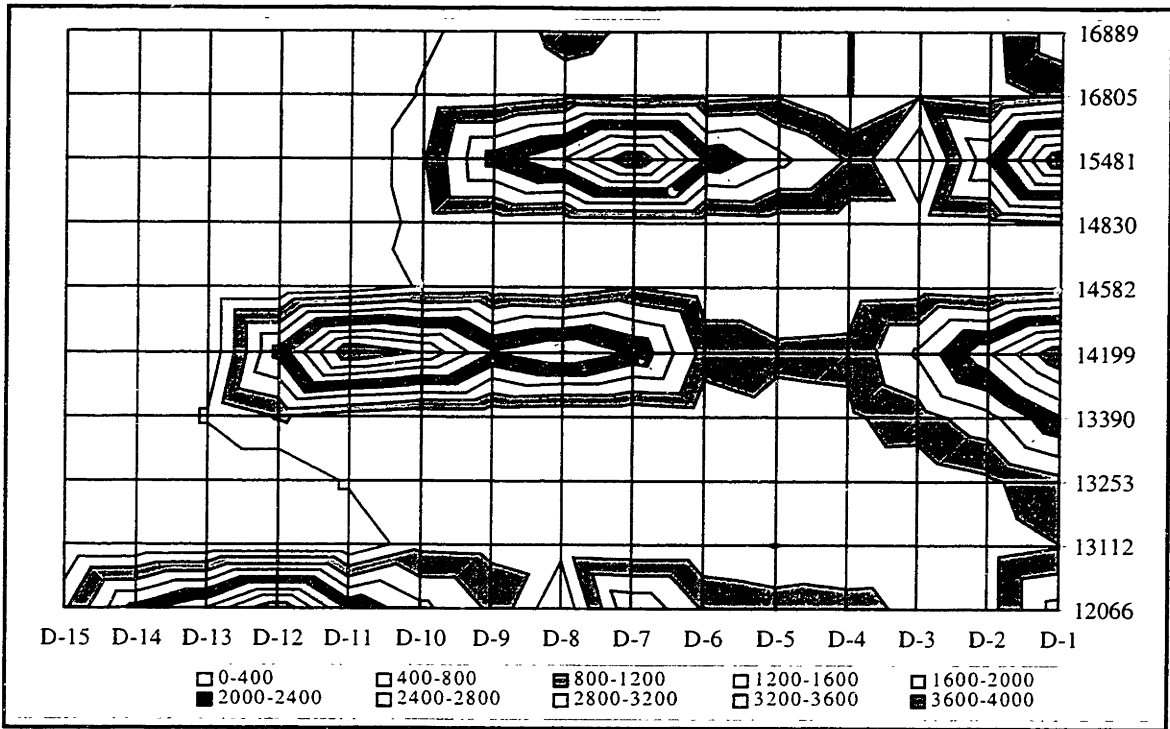
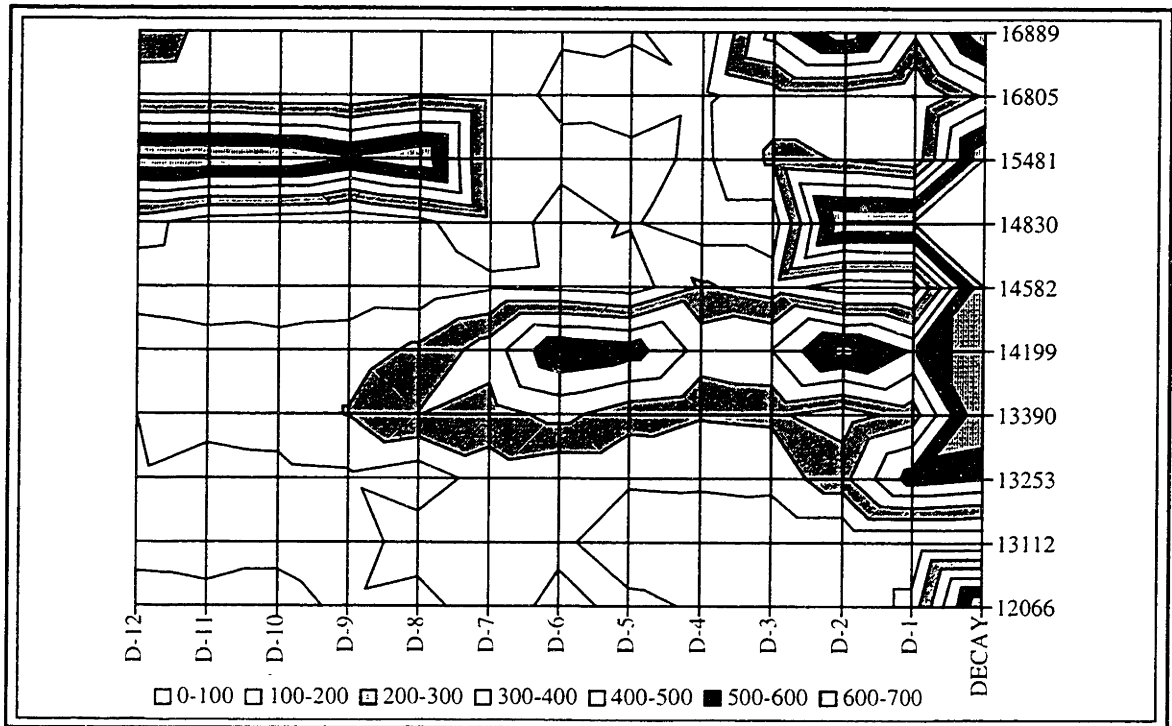


Figure B.80: Truncated Final Six Month Monthly Observation Density Comparison Plot



B.81: Truncated Contour Plot of Yearly Observation Density



B.82: Truncated Contour Plot of Final Year Monthly Observation Density

Table B.2: Final Data Comparison

NORAD ID #	2 Months Prior	1 Month Prior	Last Month	Last Year
NSSC-12066	0	137	1,497	2,411
NSSC-13112	0	4	0	555
NSSC-13253	268	551	511	1,469
NSSC-13390	147	248	10,717	11,112
NSSC-14199	636	485	6,464	6,949
NSSC-14582	42	46	12,795	13,034
NSSC-14830	1,239	816	1,504	4,043
NSSC-15481	195	134	1,511	2,059
NSSC-16805	156	96	316	1,582
NSSC-16889	675	320	4,107	6,192

[This page intentionally left blank.]

Appendix C

R&D GTDS Modifications and Additions

Substantial modifications and additions to R&D GTDS were made to support this thesis. This Appendix serves as a guide to those changes. The changes made in NT-GTDS, PR-5 and PR-6, can easily be incorporated into other versions of GTDS. Included in this appendix are the control card modifications and additions, code modifications and additions, and a modification summary.

C.1 Control Card Modifications and Additions

This Section lists the modifications to R&D GTDS control cards due to the introduction of the following items:

- MSISE-90 atmospheric model
- Kalman filter input processor
- Lift acceleration model
- Time-dependent error function model
- Altitude-dependent error function model
- Range gating control card

Additional and/or modified options have been added to the following pre-existing control cards:

ATMOSDEN	ATMRPTCD	DRAGPAR2
----------	----------	----------

The following new control cards have been added to support the new functionality of NT-GTDS, PR-5 and PR-6. The five *italicized* cards were designed by Dr. Paul Cefola in 1995, and incorporated into NT-GTDS by the author as part of this investigation.

ALTERROR	GAUSERR2	<i>LKFCNTL</i>
ALTZERR	GAUSERR3	MSISOPT1
<i>ESKFCNTL</i>	GAUSERR4	MSISOPT2
<i>ESKFOUT</i>	GAUSSERR	MSISOPT3
<i>ESKFPAR1</i>	GAUSSPER	MSISOPT4
<i>ESKFPAR2</i>	LIFTPAR	RANGEGAT
GAUSERR1	LIFTPAR2	TAERROR

In addition, the following DSST input processor cards are presented in order to have a complete listing of all cards representing specific NT-GTDS functionality. For a complete description of these input cards, and their integration into PC-GTDS, please reference Cefola [127].

AVRDRAG	SPMDAILY	SPZONALS
SPDRAG	SPNUMGRV	SSTAPGFL
SPGRVFRC	SPOUTPUT	SSTESTFL
SPINTCOF	SPSHPER	SSTESTOU
SPINTPOS	SPSOLARA	SSTSPPGF
SPJ2MDLY	SPSRP	
SPLUNARA	SPTESSLC	

All cards have been listed alphabetically, with modifications and additions noted in bold print. The reader should use the following cards in conjunction with *the R&D GTDS User's Guide* [82] when utilizing any of the new functions presented in these cards.

**ALTERROR
(FILOPT)**

ALTERROR

- **Card format: (A8, 3I3, 3G21.14)**
- **Applicable programs: FILTER**
- **Detailed format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	ALTERROR – Input card to turn on the Altitude Dependent Error Function to calculate functional process noise for the Kalman Filter. This card turns the function on and defines the position and velocity scaling factors used with the function.
9-11	I3	ADEF Function control switch variable = 0, Off = 1, On
12-17	2I3	Not Used
18-38	G21.14	Position Scaling Factor (KXYZ)
39-59	G21.14	Velocity Scaling Factor (KXYZDOT)
60-80	G21.14	Not Used

**ALTERRZN
(FILOPT)**

ALTERRZN

- **Card format: (A8, 3I3, 3G21.14)**
- **Applicable programs: FILTER**
- **Detailed format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	ALTERRZN^{1, 2}– This card is used to modify the default³ error zone stop altitudes and error levels of the Altitude Dependent Error Function.
9-11	I3	ADEF Adjustment variable = 1, Adjust both zone stop altitude and error level of specified zone = 2, Adjust only zone stop altitude of specified zone = 3, Adjust only error level of specified zone
12-14	I3	Altitude zone identifier (1-6)
15-17	I3	Not Used

¹ Must be used with **ALTERROR** card.

² A separate **ALTERRZN** card must be used for each adjusted error zone (up to 6).

³ Default zone stop altitudes and error levels:

<i>Zone</i>	<i>Altitude</i>	<i>Error Level</i>
1	90 (km)	1.00
2	120	0.90
3	200	0.92
4	250	0.70
5	600	0.40
6	1000	0.15

ALTERRZN cont'd

18-38	G21.14	Adjusted zone stop altitude (I1 = 1 or I1 = 2)⁴
39-59	G21.14	Adjusted zone error level (I1 = 1 or I1 = 3)⁵
60-80	G21.14	Not Used

⁴ Stop altitude cannot be greater than default stop altitude of next level. User should modify both zones to increase altitude beyond this limit.

⁵ Error level must be between 0 and 1.

ATMOSDEN

- Card format: (A8, 3I3, 3G21.14)
- Applicable programs: EPHEM, DC, FILTER
- Detailed format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	ATMOSDEN - keyword for reading atmospheric density table in GTDS or DODS format and specifying the density model
9-11	I3	Density table entry number (0, 1, 2, ..., 60); 0 indicates no density table
12-14	I3	Table in DODS format (punch any integer value) ⁶
15-17	I3	Atmospheric density model: = 1, Jacchia-Roberts = 2, Harris-Priester (default) = 3, Jacchia-64 (NORAD) = 4, Jacchia-70 (NORAD) = 5, MSIS-77 (NORAD) = 6-8, Reserved for RADARSAT = 9, MSISE-90
18-38	G21.14	Height (km)
39-59	G21.14	Minimum density at height (kg/km ³)
60-80	G21.14	Maximum density at height (kg/km ³)

This keyword has a dual purpose, to specify the density model and to supply a density table for the Harris-Priester density model. The Jacchia-Roberts density model is specified by placing a 1 in the third field of the ATMOSDEN keyword card. Since the

⁶ Density table input in DODS format is specified with any punch in this field. The formats are given in the GTDS User's Guide for the ATMOSDEN control card. The first card in the DODS deck specifies the number of density data cards. The maximum is 60.

ATMOSDEN Cont'd

Jacchia-Roberts (and other models except Harris-Priester) do not use an atmospheric density table, no additional data is needed when calling them.

When using the Harris-Priester atmospheric density model, multiple atmosden keyword cards must be supplied. Each card specifies a row of the Harris-Priester density table, which has a maximum of 60 rows. DODS units for minimum and maximum densities (gm/cm^3) are converted internally to kg/km^3 .

ATMOSRPT

- Card format: (A8, 3I3, 3G21.14)
- Applicable programs: FILERPT
- Detailed format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	ATMOSRPT - keyword to specify report of the Atmospheric Density Models File
9-11	I3	Type of file: = 1, Harris-Priester = 2, Jacchia-Roberts = 3, MSISE-90
12-14	I3	Summary report: = 0, no = 1, yes
15-17	I3	Specific/Detailed model reports: = 0, no = 1, detailed report ⁷ = n, ($1 \leq n \leq 10$) implies print model number ⁸ = -1, All 10 Harris-Priester models
18-80	3G21.14	Not Used

⁷ For Jacchia-Roberts and MSIS-90 models only.

⁸ Harris-Priester specific model number:

- 1 = 1964 Harris-Priester Atmosphere (Min-Max) 0-1000 km, F = 65
- 2 = 1964 Harris-Priester Atmosphere (Min-Max) 0-1000 km, F = 75
- 3 = 1964 Harris-Priester Atmosphere (Min-Max) 0-1000 km, F = 100
- 4 = 1964 Harris-Priester Atmosphere (Min-Max) 0-1000 km, F = 125
- 5 = 1964 Harris-Priester Atmosphere (Min-Max) 0-1000 km, F = 150
- 6 = 1964 Harris-Priester Atmosphere (Min-Max) 0-1000 km, F = 175
- 7 = 1964 Harris-Priester Atmosphere (Min-Max) 0-1000 km, F = 200
- 8 = 1964 Harris-Priester Atmosphere (Min-Max) 0-1000 km, F = 225
- 9 = 1964 Harris-Priester Atmosphere (Min-Max) 0-1000 km, F = 250
- 10 = 1964 Harris-Priester Atmosphere (Min-Max) 0-1000 km, F = 275

AVRDRAG
(OGOPT)

AVRDRAG

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	AVRDRAG - keyword to set second order averaging options in the VOP averaging (Draper Semianalytical) satellite theory
9-11	I3	Second order drag effects (IDRDR) = 0, Iszak's J2 height correction (if on) [default] = 1, J2-drag = 2, J2-drag, drag-drag = 3, J2-drag, drag-drag, numeric drag-J2 = 4, J2-drag, drag-drag, analytic drag-J2 = 5, Iszak J2 height-correction (if on), analytic drag-J2
12-14	I3	Number of frequencies for the J2 short periodics used in computing the mean element rates, JSPJ2 [default is four (4)]
15-17	I3	Number of frequencies for the drag short periodics used in computing the mean element rates, JSPDRG [default is four (4)]
18-80	3G21.14	Not Used

DRAGPAR2

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	DRAGPAR2 - keyword to set atmospheric drag options associated with the differential correction (DC) process for the high precision and averaged VOP orbit generators. Previously, these options were hardwired in subroutine ESTSET. KPAR also controls the lift coefficients solve-for options.
9-11	I3	Harris-Priester Atmosphere Model, KATMOS = 1, Standard GTDS model [default] = 2, Full Adaptive Model
12-14	I3	Atmospheric Drag/Lift Solve Option, KPAR <u>With the high precision orbit generator:</u> = 0, solve for GTDS parameter ρ_1 = 1, solve for GTDS parameter C_D = 6, solve for GTDS parameters C_D , & C_{L-N} = 7, solve for GTDS parameters C_D , & C_{L-W} = 8, solve for GTDS parameters C_D , C_{L-N} , & C_{L-w} <u>With the averaged orbit generator:</u> = 0, No = 1, solve for GTDS parameter C_D = 2, solve for GTDS parameter C_D and A1 in the Adaptive Harris-Priester Atmosphere Model

DRAGPAR2 Cont'd.

- = 3, solve for GTDS parameter C_D and A1 and A2 in the Adaptive Harris-Priester Atmosphere Model
- = 4, solve for GTDS parameter C_D and A1, A3, and A4 in the Adaptive Harris-Priester Atmosphere Model
- = 5, solve for GTDS parameter C_D and A1, A2, A3, and A4 in the Adaptive Harris-Priester Atmosphere Model

Note 1.- A DRAGPAR card also must be included to solve for any atmospheric drag parameters

Note 2. - In the new Adaptive Harris-Priester Atmosphere Model, the instantaneous coefficient of drag is

$$C_D = C_{Drag} [1 + A_1 t + A_2 t^2 + A_3 \sin(\omega_s t) + A_4 \cos(\omega_s t)]$$

where

- C_{Drag} = constant coefficient of drag
- t = time from epoch in days
- ω_s = rotation rate of the Sun in radians per day

15-17	I3	Not used
18-80	3G21.14	Not used

ESKFCNTL

- **Card format: (A8, 3I3, 3G21.14)**
- **Applicable programs: FILTER**
- **Detailed Format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	ESKFCNTL - KEYWORD to set the three flags that determine the Extended Semianalytical Kalman Filter (ESKF) and Semianalytical Kalman Filter (SKF) operation.
9-11	I3	Relinearization of the Semianalytical nominal trajectory for the SKF and ESKF (IUPD) =1, Relinearize at the end of an integration grid by adding the filter correction to the final grid state =2, Semianalytical linearization is global over the observation span
12-14	I3	State Transition Matrix Inverse Interpolator (INTINV) =1, Interpolator is used =2, Required Inverse is computed explicitly
15-17	I3	Selection of ESKF versus SKF (IESKF) =1, ESKF with position and velocity interpolation turned off (INTPOS = 2 in Common /HWIRE/) =2, SKF =3 ESKF with position and velocity interpolation turned on (INTPOS = 1 in Common /HWIRE/)
18-80	3G21.14	Not Used

ESKFOUT

- **Card format: (A8, 3I3, 3G21.14)**
- **Applicable programs: FILTER**
- **Detailed Format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	ESKFOUT - KEYWORD to select print options unique to the SKF and ESKF filter options.
9-11	I3	Print the nominal trajectory elements before and after adding in the filter correction (ISLVPT) =1, Yes =2, No
12-14	I3	Print the filter state and covariance in cartesian, keplerian, and mean equinoctial (for SKF/ESKF) elements as predicted at the current observation time (IWPRD) =1, Yes =2, No
15-17	I3	Print the corresponding state and covariances after the measurement update (IWUPD) =1, Yes =2, No
18-38	G21.14	The output Fortran Reference Number (FRN) for the filter history data (IPRFIL) =6, output to GTDS File \$006 (The filter history data is included in the GTDS output data file)

=43, output to GTDS File \$043 (The filter history data is given in a separate file. This file was set up to support export of the filter history data to separate plotting routines; this output maybe more complete)

39-80 2G21.14 Not used

ESKFPAR1

- **Card format: (A8, 3I3, 3G21.14)**
- **Applicable programs: FILTER**
- **Detailed Format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	ESKFPAR1 - KEYWORD to select detail print options for partial derivatives used the SKF and ESKF filter options.
9-11	I3	Print the partial derivatives of the observations with respect to the mean equinoctial elements (IDHDX) =1, Yes =2, No
12-14	I3	Print the partial derivatives of the osculating position and velocity with respect to the mean equinoctial elements (IPARTH) =1, Yes =2, No
15-17	I3	Print the state transition matrix (IKFPPR) =1, Yes =2, No
18-38	G21.14	Print file for observation, transformation, and transition partials =6, output to GTDS File \$006 =44, output to GTDS File \$044
39-80	2G21.14	Not used

ESKFPAR2

- Card format: (A8, 3I3, 3G21.14)
- Applicable programs: FILTER
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	ESKFPAR2 - KEYWORD to select detail print options for partial derivatives used the SKF and ESKF filter options.
9-11	I3	Print the partial derivatives of the observations with respect to the mean equinoctial elements (IDHDX) =1, Yes =2, No
12-14	I3	Print the partial derivatives of the osculating position and velocity with respect to the mean equinoctial elements (IPARTH) =1, Yes =2, No
15-17	I3	Print the state transition matrix (IKFPPR) =1, Yes =2, No
18-38	G21.14	Print file for observation, transformation, and transition partials =6, output to GTDS File \$006 =44, output to GTDS File \$044
39-80	2G21.14	Not used

GAUSERR1

- **Card format: (A8, 3I3, 3G21.14)**
- **Applicable programs: FILTER**
- **Detailed format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	GAUSERR1 - keyword used to define the parameters of the 1st Gaussian normal function which is part of the error estimator function.
9-11	I3	Use 1st Gaussian Normal in error function: = 0, Do not use (Default) = 1, Use
12-17	2I3	Not Used
18-38	G21.14	Scalar multiplier, k, of 1st Gaussian normal
39-59	G21.14	Mean, μ, of 1st Gaussian normal
60-80	G21.14	Standard deviation, σ, of 1st Gaussian normal

**GAUSERR2
(FILOPT)**

GAUSERR2

- **Card format: (A8, 3I3, 3G21.14)**
- **Applicable programs: FILTER**
- **Detailed format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	GAUSERR2 - keyword used to define the parameters of the 2nd Gaussian normal function which is part of the error estimator function.
9-11	I3	Use 2nd Gaussian Normal in error function: = 0, Do not use (Default) = 1, Use
12-17	2I3	Not Used
18-38	G21.14	Scalar multiplier, k, of 2nd Gaussian normal
39-59	G21.14	Mean, μ, of 2nd Gaussian normal
60-80	G21.14	Standard deviation, σ, of 2nd Gaussian normal

GAUSERR3

- **Card format: (A8, 3I3, 3G21.14)**
- **Applicable programs: FILTER**
- **Detailed format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	GAUSERR3 - keyword used to define the parameters of the 3rd Gaussian normal function which is part of the error estimator function.
9-11	I3	Use 3rd Gaussian Normal in error function: = 0, Do not use (Default) = 1, Use
12-17	2I3	Not Used
18-38	G21.14	Scalar multiplier, k, of 3rd Gaussian normal
39-59	G21.14	Mean, μ, of 3rd Gaussian normal
60-80	G21.14	Standard deviation, σ, of 3rd Gaussian normal

GAUSERR4

- **Card format:** (A8, 3I3, 3G21.14)
- **Applicable programs:** FILTER
- **Detailed format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	GAUSERR4 - keyword used to define the parameters of the 4th Gaussian normal function which is part of the error estimator function.
9-11	I3	Use 4th Gaussian Normal in error function: = 0, Do not use (Default) = 1, Use
12-17	2I3	Not Used
18-38	G21.14	Scalar multiplier, k, of 4th Gaussian normal
39-59	G21.14	Mean, μ, of 4th Gaussian normal
60-80	G21.14	Standard deviation, σ, of 4th Gaussian normal

GAUSSERR

- **Card format: (A8, 3I3, 3G21.14)**
- **Applicable programs: FILTER**
- **Detailed format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	GAUSSERR⁹ - keyword used to define the day of launch and scaling factors of the complete, two-variable error function .
9-11	I3	Last two digits of launch year
12-14	I3	Month of launch year
15-17	I3	Day of launch year
18-38	G21.14	Remainder of day of launch
39-59	G21.14	Scaling factor for position components
60-80	G21.14	Scaling factor for velocity components

⁹ **Must be used to turn on Gaussian Error Functional Model. Model can be used in addition to constant process noise model (SPNOISE).**

GAUSSPER

- **Card format: (A8, 3I3, 3G21.14)**
- **Applicable programs: FILTER**
- **Detailed format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	GAUSSPER - keyword used to define approximations of the orbital period during the satellite's mission phase, and just before decay. Also used to define the initial standard deviation of the Gaussian error function.
9-17	3I3	Not Used
18-38	G21.14	Initial standard deviation of complete error function
39-59	G21.14	Orbital period of satellite in mission phase
60-80	G21.14	Orbital period before decay

LIFTPAR

- Card format: (A8, 3I3, 3G21.14)
- Applicable programs: EPHEM, DC, FILTER
- Detailed format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	LIFTPAR ^{10, 11, 12, 13, 14} - keyword for reading atmospheric lift solve-for options
9-11	I3	Lift indicator switch (INDLIFT): = 0, Lift is neither a solve-for or consider parameter (default) = 1, include lift accelerations for both N and W directions = 2, include lift acceleration for N direction only = 3, include lift acceleration for W direction only
12-14	I3	Partial Derivative Calculation Switch = 1, calculate numerically (default) = 2, calculate analytically (currently not operable)
15-17	I3	Not Used
18-38	G21.14	A-priori value of C_{L_N} (LPARN).
39-59	G21.14	A-priori value of C_{L_w} (LPARW).
60-80	G21.14	Numerical integration step size (Δx) for partial derivative calculation.

¹⁰ Drag acceleration must be turned on with the DRAG card to use LIFTPAR.

¹¹ Must be used with the DRAGPAR and DRAGPAR2 cards.

¹² Currently designed to only support the MSISE-90 atmospheric model.

¹³ Can only be used with spherical configuration (KONFIG = 0).

¹⁴ DRAGPAR2 card's KPAR variable indicates solve-for options.

LIFTPAR2

- Card format: (A8, 3I3, 3G21.14)
- Applicable programs: EPHEM, DC, FILTER
- Detailed format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	LIFTPAR2 ¹⁵ - keyword for reading values for atmospheric lift variance and standard deviation, as well as print options.
9-11	I3	Lift data print file switch (LSWPRN) ¹⁶ : = 0, do not print file (default) = 1, print lift coefficients and accelerations in both N & W directions for each call of AERO
12-17	2I3	Not Used
18-38	G21.14	Variation of Lift-N coefficient.
39-59	G21.14	Variation of Lift W coefficient.
60-80	G21.14	Not Used

¹⁵ Must be used with LIFTPAR card.

¹⁶ Print file contains:

- Julian date
- Gregorian date
- Acceleration due to drag (magnitude and vector components)
- Acceleration due to lift in the N direction (magnitude and vector components)
- Acceleration due to lift in the W direction (magnitude and vector components)

**LKFCNTL
(FILOPT)**

LKFCNTL

- **Card format: (A8, 3I3, 3G21.14)**
- **Applicable programs: FILTER**
- **Detailed Format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	LKFCNTL - KEYWORD to control the Cowell Linear Kalman Filter relinearization. This card allows the Cowell LKF to be configured to emulate the Semianalytical Kalman Filter (SKF).
9-11	I3	Relinearize the Cowell Linear Kalman Filter at the interval specified by first real field on this card (ILKFUP) =1, Yes =2, No
12-17	2I3	Not Used
18-38	G21.14	Relinearization interval size (seconds) [DTLKF]
39-59	G21.14	Initial filter time (seconds) [TLLKF]
60-80	G21.14	Not Used

MSISOPT1

- **Card format:** (A8, 3I3, 3G21.14)
- **Applicable programs:** EPHEM, DC, FILTER
- **Detailed format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	MSISOPT1 - keyword for defining option array for use with the MSIS-90 atmospheric density model. Must be used in conjunction with the ATMOSDEN card for the MSISE-90 (9) model. MSISOPT1 controls the Mass and SW(1-5) variables.
9-11	I3	Mass number (only density for selected gas is calculated): = 0, Temperature calculated - no densities = n, (n = density of selected gas) = 48, All densities calculated (default)
12-14	I3	SW(1), F10.7 effect on mean switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
15-17	I3	SW(2), Time independent switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
18-38	G21.14	SW(3), Symmetrical annual switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
39-59	G21.14	SW(4), Symmetrical semiannual switch = 0, off = 1, on (default) = 2, main effects off but cross terms on

MSISOPT1 Cont'd

60-80	G21.14	SW(5), Asymmetrical annual switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
--------------	---------------	--

MSISOPT2

- **Card format:** (A8, 3I3, 3G21.14)
- **Applicable programs:** EPHEM, DC, FILTER
- **Detailed format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	MSISOPT2 - keyword for defining option array for use with the MSIS-90 atmospheric density model. Must be used in conjunction with the ATMOSDEN card for the MSIS-90 (9) model. MSISOPT1 controls the SW(6-11) variables.
9-11	I3	SW(6), Asymmetrical semiannual switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
12-14	I3	SW(7), Diurnal switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
15-17	I3	SW(8), Semidiurnal switch = 0, off = 1, on (default) = 2, main effects off but cross terms on

MSISOPT2 Cont'd

18-38	G21.14	SW(9), Daily Ap switch = -1, The Ap variable becomes a 7-element array¹⁷ (default) = 0, off = 1, on = 2, main effects off but cross terms on
39-59	G21.14	SW(10), All UT/LONG effects switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
60-80	G21.14	SW(11), Longitudinal switch = 0, off = 1, on (default) = 2, main effects off but cross terms on

¹⁷ **Array Containing:**

- (1) Daily Ap**
- (2) 3 hour Ap index for current time**
- (3) 3 hour Ap index for 3 hours before current time**
- (4) 3 hour Ap index for 6 hours before current time**
- (5) 3 hour Ap index for 9 hours before current time**
- (6) Average of eight 3 hour Ap indices from 12 to 33 hours prior to current time**
- (7) Average of eight 3 hour Ap indices from 36 to 59 hours prior to current time**

MSISOPT3

- Card format: (A8, 3I3, 3G21.14)
- Applicable programs: EPHEM, DC, FILTER
- Detailed format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	MSISOPT3 - keyword for defining option array for use with the MSIS-90 atmospheric density model. Must be used in conjunction with the ATMOSDEN card for the MSIS-90 (9) model. MSISOPT1 controls the SW(12-17) variables.
9-11	I3	SW(12), UT and Mixed UT/LONG switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
12-14	I3	SW(13), Mixed AP/UT/LONG switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
15-17	I3	SW(14), Terdiurnal switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
18-38	G21.14	SW(15), Departures from diffusive equilibrium switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
39-59	G21.14	SW(16), All TINF VAR switch = 0, off = 1, on (default) = 2, main effects off but cross terms on

MSISOPT3 Cont'd

60-80	G21.14	SW(17), All TLB VAR switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
--------------	---------------	---

MSISOPT4

- **Card format:** (A8, 3I3, 3G21.14)
- **Applicable programs:** EPHEM, DC, FILTER
- **Detailed format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	MSISOPT3 - keyword for defining option array for use with the MSIS-90 atmospheric density model. Must be used in conjunction with the ATMOSDEN card for the MSIS-90 (9) model. MSISOPT1 controls the SW(18-23) variables.
9-11	I3	SW(18), All TN1 VAR switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
12-14	I3	SW(19), All S VAR switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
15-17	I3	SW(20), All TN2 VAR switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
18-38	G21.14	SW(21), All NLB VAR switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
39-59	G21.14	SW(22), All TN3 VAR switch = 0, off = 1, on (default) = 2, main effects off but cross terms on

MSISOPT4 Cont'd

60-80	G21.14	SW(23), Turbo Scale Height VAR switch = 0, off = 1, on (default) = 2, main effects off but cross terms on
--------------	---------------	--

**RANGEGAT
(OGOPT)**

RANGEGAT

- **Card format: (A8, 3I3, 3G21.14)**
- **Applicable programs: DC, EPHEM, FILTER**
- **Detailed format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	RANGEGAT - keyword used to turn on or off the range gating feature of GTDS.
9-11	I3	Range gating switch: = 0, On (default) = 1, Off
12-17	2I3	Not Used
18-80	3G21.14	Not Used

SPDRAG

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SPDRAG - keyword to set the atmospheric drag options for the Fourier coefficients in the short periodic expansions of the VOP averaging (Draper Semianalytical) satellite theory.
9-11	I3	Quadrature control switch for atmospheric drag short periodics, IDRAG = 0, do not include the drag short periodics = 1, include the drag short periodics <u>The default is to include the drag short periodics if drag is included in the mean element eqs. of motion.</u>
12-14	I3	Quadrature order for drag short periodics, NDRAG = 1, 12-point quadrature = 2, 16-point quadrature = 3, 20-point quadrature = 4, 24-point quadrature = 5, 32-point quadrature = 6, 40-point quadrature = 7, 48-point quadrature (default)
15-17	I3	Short-periodic expansion longitude for drag, LDRAG = 1, mean longitude (default) = 2, eccentric longitude = 3, true longitude
18-38	G21.14	Maximum frequencies for drag, JDRAG The default value of JDRAG is six (6).
39-80	2G21.14	Not used

SPGRVFRC

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SPGRVFRC - keyword to set the gravitational force model options for the Fourier coefficients in the short periodic expansions of the VOP averaging (Draper Semianalytical) satellite theory
9-11	I3	Central body zonal harmonic short periodic option = 1, analytical coefficients (default) = 2, numerical coefficients = 3, off
12-14	I3	Central body m-daily tesseral harmonic short periodic option = 1, analytical coefficients (default) = 3, off
15-17	I3	Central body high frequency (linear combination term) tesseral short periodic option = 1, analytical coefficients (default) = 3, off
18-38	G21.14	Third body short periodic option = 1, analytical coefficients (default) = 2, numerical coefficients = 3, off
39-59	G21.14	Central body J2-squared short periodic option = 1, analytical coefficients (default) = 3, off

SPGRVFRC Cont'd

60-80	G21.14	Central body J2 secular/tesseral m-daily coupling short periodic option
		= 1, analytical coefficients (default)
		= 3, off

SPINTCOF

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SPINTCOF - keyword to set interpolator options for the Fourier coefficients in the short periodic expansions of the VOP averaging (Draper Semianalytical) satellite theory
9-11	I3	Interpolate for the Fourier coefficients = 1, yes (default) = 2, no
12-14	I3	Number of time points used to construct the Lagrangian interpolators = 1, 1 point formula = 2, 2 point formula = 3, 3 point formula (default) = 4, 4 point formula
15-17	I3	Not used
18-38	G21.14	Nominal interval between interpolator points, seconds (default=86400.0 secs) [not operational]
39-80	2G21.14	Not used

Note - presently the Draper Semianalytical Theory must use the Runge-Kutta integrator when the short periodics are turned on. In this implementation, the interval between the Fourier coefficient interpolator time points is the same as the RK stepsize.

SPINTPOS

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SPINTPOS - keyword to set interpolator options for position, velocity, and partial derivatives of position and velocity in the VOP averaging (Draper Semianalytical) satellite theory
9-11	I3	Interpolate for position, velocity, and partial derivatives = 1, yes = 2, no (default)
12-14	I3	Number of time points used in the Hermite interpolators = 2 = 3, (default)
15-17	I3	Not used
18-38	G21.14	Nominal interval between position and velocity interpolator points (seconds) (default = 120 seconds)
39-80	2G21.14	Not used

SPJ2MDLY

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SPJ2MDLY - keyword to set the central body J2 secular/tesseral m-daily coupling harmonic options for the Fourier coefficients in the short periodic expansions of the VOP averaging (Draper Semianalytical) satellite theory.
9-11	I3	Maximum degree of the central body J2 secular/tesseral m daily coupling field, NJ2MD. NJ2MD can vary from 2 to 21. NJ2MD is defaulted to use the maximum degree NMD specified for the tesseral m-daily field. See the description of the SPMDAILY card.
12-14	I3	Maximum order of the central body J2 secular/tesseral m-daily coupling field, MJ2MD MJ2MD can vary from 1 to NJ2MD. MJ2MD is defaulted to use the maximum order MMD specified for the tesseral m-daily field. See the description of the SPMDAILY card.
15-17	I3	Maximum power of e, LJ2MD The lower limit for this parameter is zero. The upper limit is two less than the specified degree (NJ2MD - 2). The upper limit gives the full solution for these terms. This parameter is defaulted to use NMD - 2. See the description of the SPMDAILY card.
18-38	G21.14	Use drag secular/tesseral m-daily coupling (IDRMD) = 1, yes (default) = 2, no
39-80	2G21.14	Not used

SPLUNARA

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SPLUNARA - keyword to set the <u>Lunar third-body</u> high frequency options for the Fourier coefficients in the short periodic expansions of the VOP averaging (Draper Semianalytical) satellite theory. [This card supports the single phase angle expansion option]
9-11	I3	Maximum power of a/r, NTH(1) This parameter specifies the maximum degree of the third-body field. NTH(1) can vary from 2 to 20. NTH(1) is defaulted to 8.
12-14	I3	Maximum frequency in eccentric longitude F, JMAXTH(1) The lower limit for JMAXTH(1) is one (1). The upper limit is NTH(1) + 1. JMAXTH(1) is defaulted to the upper limit.
15-17	I3	Maximum power of e, LTH(1) The lower limit for LTH(1) is zero. The upper limit is NTH(1) + JMAXTH(1). LTH(1) is defaulted to the upper limit.
18-38	G21.14	Method of computing time derivatives, ITDTH(1) = 1, analytical = 2, finite differences (default)
39-59	G21.14	Order of the highest time derivative, NTDTH(1) NTDTH(1) is defaulted to zero.
60-80	G21.14	Not used

SPMDAILY

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SPMDAILY - keyword to set the central body m-daily tesseral harmonic options for the Fourier coefficients in the short periodic expansions of the VOP averaging (Draper Semianalytical) satellite theory
9-11	I3	Maximum degree of the central body tesseral m-daily field, NMD NMD can vary from 2 to 21. NMD is defaulted to use the maximum degree specified on the MAXDEGEQ card or the default for that parameter.
12-14	I3	Maximum order of the central body tesseral m-daily field, MMD MMD can vary from 1 to NMD. MMD is defaulted to use the maximum order specified on the MAXORDEQ card or the default for that parameter.
15-17	I3	Maximum power of e, LMD The lower limit for LMD is zero. The upper limit is two less than the specified maximum degree (NMD - 2). The upper limit gives the full solution for these terms. LMD is defaulted to give the full solution.
18-80	3G21.14	Not used

SPNUMGRV

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SPNUMGRV - keyword to set the numerical gravity options for the Fourier coefficients in the short periodic expansions of the VOP averaging (Draper Semianalytical) satellite theory.
9-11	I3	Quadrature order for numerical gravity short periodics, NGRAV = 1, 12-point quadrature = 2, 16-point quadrature = 3, 20-point quadrature = 4, 24-point quadrature = 5, 32-point quadrature = 6, 40-point quadrature = 7, 48-point quadrature (default)
12-14	I3	Short-periodic expansion longitude for numerical gravity, LGRAV = 1, mean longitude (default) = 2, eccentric longitude = 3, true longitude
15-17	I3	Maximum frequencies for numerical gravity, JGRAV. The default value of JGRAV is six (6).
18-38	G21.14	Method of computing Fourier coefficient time derivatives for the numerical gravity perturbations, IDGRAV = 1, numerical quadrature = 2, finite differences (default)

- | | | |
|-------|--------|--|
| 39-59 | G21.14 | Order of highest time derivative for gravitational perturbations, NDGRAV. The default is zero (0). |
| 60-80 | G21.14 | Time step for numerical time derivatives, DTGRAV. The default is 3600 seconds. |

SPOUTPUT

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SPOUTPUT - keyword to set the output options for the short periodic expansions of the VOP averaging (Draper Semianalytical) satellite theory.
9-11	I3	Print coefficients of the position and velocity interpolator, KINTPV = 1, yes = 2, no (default)
12-14	I3	Print coefficients of the interpolator for short-periodic coefficients, KINTCF = 1, yes = 2, no (default)
15-17	I3	Print short-periodic variations, KSP = 1, yes = 2, no (default)
18-38	G21.14	Print Fourier coefficients of the short-periodic variations, KSPCF = 1, yes = 2, no (default) The short periodic coefficients are very useful in understanding the size and frequency content of the short periodics.
39-80	2G21.14	Not used

SPSHPER

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SPSHPER - key word to select a pre-stored short periodic option in the VOP averaging (Draper Semianalytical) satellite theory
9-11	I3	Pre-Stored Short Periodic Option = 1, Mean Elements Only, All short periodic models are turned off = 2, Low Altitude, Near Circular Orbit (Moderate Accuracy Option) J2 Short Periodics Tesseral M-Dailies = 3, Low Altitude, Near Circular Orbit (Improved Accuracy Option) Zonal Short Periodics Tesseral M-Dailies Tesseral Linear Combinations J2-Squared Terms J2 Secular/Tesseral M-Daily Coupling = 4, 24 hr. Geosynchronous Orbit (Moderate Accuracy) J2 Short Periodics Lunar-Solar Short Periodics

SPSHPER Cont'd.

= 5, 12 hr., high eccentricity Molniya Orbit (Moderate Accuracy)

J2 Short Periodics
Tesseral M-Dailies
Lunar-Solar Short Periodics

= 6, Allow processing of individual short periodic option cards in a second OGOPT subdeck

12-14 2I3 Not used

18-80 3G21.14 Not used

Note - If one of these pre-stored short periodic options (1 through 5) is selected, other Semianalytical Satellite Theory short periodic keyword cards in the same OGOPT subdeck which adjust the force models are ignored.

SPSHER OPTION VARIABLES

Mean Elements Only

IZONAL = 3
IMDALY = 3
ITESS = 3
ITHIRD = 3
IJ2J2 = 3
IJ2MD = 3
IDRAG = 0
ISOLAR = 0

SPSHPER Cont'd.

Low Altitude, Near Circular Orbit (Moderate Accuracy Option)

IZONAL	=	1
IMDALY	=	1
ITESS	=	3
ITHIRD	=	3
IJ2J2	=	3
IJ2MD	=	3
IDRAG	=	0
ISOLAR	=	0
NZN	=	2
LZN	=	1
JZN	=	5
NMD	=	Minimum of INDEG and 12
MMD	=	Minimum of INORD and 12
LMD	=	Minimum of NMD-2 and 4

Low Altitude, Near Circular Orbit (Improved Accuracy Option)

IZONAL	=	1
IMDALY	=	1
ITESS	=	1
ITHIRD	=	3
IJ2J2	=	1
IJ2MD	=	1
IDRAG	=	0
ISOLAR	=	0
LTS	=	4
LTSHAN	=	2
JMINTS	=	- NTS - LTS
JMAXTS	=	+ NTS + LTS
NZN	=	Minimum of INDEG and 12
LZN	=	Minimum of NZN-1 and 4
JZN	=	Minimum of (2 NZN+1) and NZN+4

SPSHPER Cont'd.

NMD	=	Minimum of INDEG and 12
MMD	=	Minimum of INORD and 12
LMD	=	Minimum of NMD-2 and 4
NJ2MD	=	Minimum of NMD and 4
MJ2MD	=	Minimum of MMD and 4
LJ2MD	=	NJ2MD-2
NTS	=	Minimum of INDEG and 8
MTS	=	Minimum of INORD and 8

24 hr Geosynchronous Orbit (Moderate Accuracy Option)

IZONAL	=	1
IMDALY	=	3
ITESS	=	3
ITHIRD	=	1
IJ2J2	=	3
IJ2MD	=	3
IDRAG	=	0
ISOLAR	=	0
NZN	=	2
LZN	=	1
JZN	=	5
NTH(1)	=	8
JMAXTH(1)	=	9
LTH(1)	=	4
ITDTH(1)	=	2
NTDTH(1)	=	0
NTH(2)	=	4
JMAXTH(2)	=	5
LTH(2)	=	4
ITDTH(2)	=	2
NTDTH(2)	=	0

SPSHPER Cont'd.

12 hr., high eccentricity Molniya Orbit (Moderate Accuracy Option)

IZONAL	=	1
IMDALY	=	1
ITESS	=	3
ITHIRD	=	1
IJ2J2	=	3
IJ2MD	=	3
IDRAG	=	0
ISOLAR	=	0
NTH(1)	=	8
JMAXTH(1)	=	9
LTH(1)	=	4
ITDTH(1)	=	2
NTDTH(1)	=	0
NTH(2)	=	4
JMAXTH(2)	=	5
LTH(2)	=	4
ITDTH(2)	=	2
NTDTH(2)	=	0
LZN	=	NZN-1
JZN	=	2 NZN+1
LMD	=	NMD-2
NZN	=	Minimum of INDEG and 4
NMD	=	Minimum of INDEG and 4
MMD	=	Minimum of INORD and 4

SPSOLARA

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SPSOLARA - keyword to set the <u>Solar third-body</u> high frequency options for the Fourier coefficients in the short periodic expansions of the VOP averaging (Draper Semianalytical) satellite theory. [This card supports the single phase angle expansion option]
9-11	I3	Maximum power of a/r , NTH(2) This parameter specifies the maximum degree of the third-body field. NTH(2) can vary from 2 to 20. NTH(2) is defaulted to 4.
12-14	I3	Maximum frequency in eccentric longitude F, JMAXTH(2) The lower limit for JMAXTH(2) is one (1). The upper limit is NTH(2) + 1. JMAXTH(2) is defaulted to the upper limit.
15-17	I3	Maximum power of e , LTH(2) The lower limit for LTH(2) is zero. The upper limit is NTH(2) + JMAXTH(2). LTH(2) is defaulted to the upper limit.
18-38	G21.14	Method of computing time derivatives, ITDTH(2) = 1, analytical = 2, finite differences (default)
39-59	G21.14	Order of the highest time derivative, NTDTH(2) NTDTH(2) is defaulted to zero.
60-80	G21.14	Not used

SPSRP

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SPSRP - keyword to set the solar radiation pressure options for the Fourier coefficients in the short periodic expansions of the VOP averaging (Draper Semianalytical) satellite theory.
9-11	I3	<p>Quadrature control switch for atmospheric drag short periodics, ISOLAR</p> <p>= 0, do not include the solar radiation pressure short periodics</p> <p>= 1, include the solar radiation pressure short periodics</p> <p><u>The default is to include the solar radiation pressure short periodics if solar radiation pressure is included in the mean element eqs. of motion.</u></p>
12-14	I3	<p>Quadrature order for solar radiation pressure short periodics, NSOLAR</p> <p>= 1, 12-point quadrature</p> <p>= 2, 16-point quadrature</p> <p>= 3, 20-point quadrature</p> <p>= 4, 24-point quadrature</p> <p>= 5, 32-point quadrature</p> <p>= 6, 40-point quadrature</p> <p>= 7, 48-point quadrature (default)</p>
15-17	I3	<p>Short-periodic expansion longitude for solar radiation pressure, LSOLAR</p> <p>= 1, mean longitude (default)</p> <p>= 2, eccentric longitude</p> <p>= 3, true longitude</p>

SPSRP Cont'd.

18-38 G21.14 Maximum frequencies for solar radiation pressure, JSOLAR. The default value of JSOLAR is six (6).

39-80 2G21.14 Not used

SPTESLCL

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SPTESLCL - keyword to set the central body high frequency tesseral harmonic options for the Fourier coefficients in the short periodic expansions of the VOP averaging (Draper Semianalytical) satellite theory.
9-11	I3	Maximum degree of the central body high frequency tesseral field, NTS NTS can vary from 2 to 21. NTS is defaulted to use the maximum degree specified on the MAXDEGEQ card or the default for that parameter.
12-14	I3	Maximum order of the central body high frequency tesseral field, MTS MTS can vary from 1 to NTS. MTS is defaulted to use the maximum order specified on the MAXORDEQ card or the default for that parameter.
15-17	I3	Maximum d'Alembert characteristic (maximum power of the eccentricity outside the Hansen coefficients), LTS LTS is defaulted to 4. LTS should be approximately twice LTSHAN (see below). The default value of LTS was designed for a near circular orbit.
18-38	G21.14	Maximum power of e^{**2} in the power series expansion for the Hansen coefficients, LTSHAN LTSHAN is defaulted to 2. The default value of LTSHAN was designed for a near circular orbit.

SPTESSLC Cont'd

Note for high e cases -

If the eccentricity of the satellite orbit is large, then the terms in the power series expansion for a given Hansen coefficient increase rapidly for a while, reach a maximum, and then begin to decrease, eventually becoming smaller than the Hansen coefficient itself. The first term in the decreasing part of the expansion which is smaller than the Hansen coefficient itself marks the onset of convergence, and the exponent of e^2 in this term can be called the "convergence index."

The value of LSTHAN must be greater than the convergence index of each Hansen coefficient in the high frequency tesseral short periodic model. If LSTHAN is smaller than the biggest convergence index, then the error in the high frequency tesseral short periodic variations may be bigger than the short periodics themselves.

39-59 G21.14 Minimum frequency in lambda (mean longitude) in the central body high frequency tesseral field, JMINTS

JMINTS is defaulted to - NTS - LTS. It is recommended that JMINTS and JMAXTS be equal in magnitude. See Note below.

60-80 G21.14 Maximum frequency in lambda (mean longitude) in the central body high frequency tesseral field, JMAXTS

JMAXTS is defaulted to + NTS + LTS. It is recommended that JMINTS and JMAXTS be equal in magnitude.

Note - The settings of JMINTS, JMAXTS, and MTS control the number of frequencies due to tesserals in the double angle expansion according to the rule

$$NDOUBL = (JMAXTS - JMINTS + 1) * MTS$$

There is currently a limit of 400 on NDOUBL in the IBM mainframe version of Draper R&D GTDS and a limit of 1600 on NDOUBL in the VAX and UNIX (Sun and SGI) versions of Draper R&D GTDS.

SPZONALS
(OGOPT)

SPZONALS

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SPZONALS - keyword to set the central body zonal harmonic options for the Fourier coefficients in the short periodic expansions of the VOP averaging (Draper Semianalytical) satellite theory
9-11	I3	Maximum power of r/a, (NZN) This parameter specifies the maximum degree of the central body zonal field. Therefore it can vary from 2 to 21. This parameter is defaulted to use the maximum degree specified on the MAXDEGEQ card or the default for that parameter.
12-14	I3	Maximum power of e, (LZN) The lower limit for this parameter is zero. The upper limit is one less than the specified degree (NZN - 1). The upper limit gives the full solution for these terms. This parameter is defaulted to use one less than the maximum degree specified on the MAXDEGEQ card or the default for that parameter.
15-17	I3	Maximum frequency in true longitude L, (JZN) The lower limit for this parameter is one. The upper limit is (2 NZN + 1). The upper limit gives the full solution for these terms. This parameter is defaulted to the upper limit implied by the maximum degree specified on the MAXDEGEQ card or the default for that parameter.
18-80	3G21.14	Not used

SSTAPGFL

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SSTAPGFL - keyword to set the perturbations included in the element rate partial derivatives for the VOP averaging (Draper Semianalytical) satellite theory. Previously, these options were hardwired in subroutine ESTSET.
9-11	I3	Include the central-body gravitational field in the element rate partials, ICBVAR = 0, no = 1, analytically = 2, finite differences, using the degree and order used in the averaged equations of motion = 3, finite differences, using degree 4 and order 0
12-14	I3	Include the J2-squared central body gravitational field effect in the element rate partials, J22VAR = 0, no = 1, finite differences
15-17	I3	Include third-body gravitational fields in the element rate partials, ITBVAR = 0, no = 1, finite differences
18-38	G21.14	Include atmospheric drag in the element rate partials, IDRVAR = 0, no = 1, finite differences

SSTAPGFL cont'd

- 39-59 G21.14 Include second-order atmospheric drag effects in the
element rate partials, IDRDRV
- = 0, no
 - = 1, J2-drag
 - = 2, J2-drag, drag-drag
 - = 3, J2-drag, drag-drag, numeric drag-J2
 - = 4, J2-drag, drag-drag, analytic drag-J2
 - = 5, Iszak J2 height-correction, analytic drag-J2
 - = 6, Iszak J2 height-correction
-
- 60-80 G21.14 ISRVAR
- = 0, no
 - = 1, finite differences
 - =2, numerical quadrature
 (not active)

SSTESTFL

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SSTESTFL - keyword to set the primary computational options for VOP averaging partial derivative theory. This theory is compatible with the VOP averaging (Draper Semianalytical) satellite theory. Previously, these options were hardwired in subroutine ESTSET.
9-11	I3	<p>Compute element rate partial derivatives analytically (IANAL)</p> <p>= 0, No</p> <p>= 1, A matrix only (Default)</p> <p>= 2, D matrix only</p> <p>= 3, Both A and D matrices</p>
12-14	I3	<p>Compute element rate partial derivatives with finite differences (IDIFF)</p> <p>= 0, No (Default)</p> <p>= 1, A matrix only</p> <p>= 2, D matrix only</p> <p>= 3, Both A and D matrices</p>
15-17	I3	<p>Compute element rate partial derivatives by numerical quadrature (IQDRT)</p> <p>= 0, No (Default)</p> <p>= 1, A matrix only (Not Active)</p> <p>= 2, D matrix only (Not Active)</p> <p>= 3, Both A and D matrices (Not Active)</p>

SSTESTFL Cont'd.

18-38	G21.14	Compute Short-Periodic partial derivatives = 0, No (Default) = 1, B1 matrix only = 2, B1 and B4 matrices = 3, B4 matrix only
39-80	2G21.14	Not Used

Notes:

1. The A matrix is comprised of the partial derivatives of element rates with respect to elements
2. The D matrix is comprised of the partial derivatives of element rates with respect to the dynamic parameters
3. The B1 matrix is comprised of the partial derivatives of the short periodics with respect to mean elements.
4. The B2 matrix is comprised of the partial derivatives of the mean elements with respect to the epoch mean elements [*this quantity is also known as the mean element state transition matrix*].
5. The B3 matrix is comprised of the partial derivatives of the mean elements with respect to the dynamic parameters.
6. The B4 matrix is comprised of the partial derivatives of the short periodics with respect to the dynamic parameters.

SSTESTOU

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SSTESTOU - keyword to set the output options for the VOP averaging partial derivative theory. This theory is compatible with the VOP averaging (Draper Semianalytical) satellite theory. Previously, these options were hardwired in subroutine ESTSET.
9-11	I3	Print the B1 matrix, KPRTB1 = 1, yes = 2, no
12-14	I3	Print the B2 matrix (mean el transition matrix), KPRTB2 = 1, yes = 2, no
15-17	I3	Print the B3 matrix, KPRTB3 = 1, yes = 2, no
18-38	G21.14	Print the B4 matrix, KPRTB4 = 1, yes = 2, no
39-80	2G21.14	Not used

Note - definitions of the B1, B2, B3, and B4 matrices are given with the SSTESTFL card.

SSTSPPGF

- Card format: (A8, 3I3, 3G21.14)
- Applicable Programs: (DC, EPHEM, FILTER)
- Detailed Format:

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	SSTSPPGF - keyword to set the perturbations to be included in the short-periodic partial derivatives for the VOP averaging (Draper Semianalytical) satellite theory. Previously, these options were hardwired in subroutine ESTSET.
9-11	I3	Include the central-body gravitational field in the short periodic partials, using the degree and order used in the averaged equations of motion, KGRAVP = 0, no = 1, analytically (J2 only) = 2, finite differences (full field)
12-14	I3	Include the central-body gravitational field in the short periodic partials, using a degree of "NZONP" (see below) and an order of zero, KZONP = 0, no = 1, analytically (J2 only) = 2, finite differences (full field)
15-17	I3	Include the third-body gravitational fields in the short periodic partials, KTHRP = 0, no = 2, finite differences

SSTSPPGF Cont'd

18-38	G21.14	Include atmospheric drag in the short periodic partials, KDRAGP = 0, no = 2, finite differences
39-59	G21.14	Include solar radiation pressure in the short periodic partials, KSOLRP = 0, no = 2, finite differences
60-80	G21.14	Maximum degree of the central-body gravitational field used in computing the short-periodic partials. This parameter is used only if KZONP = 2 (see above), NZONP = 1, tbs

**TAERROR
(FILOPT)**

TAERROR

- **Card format: (A8, 3I3, 3G21.14)**
- **Applicable programs: FILTER**
- **Detailed format:**

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1-8	A8	TAERROR - keyword used to define the true anomaly cyclical amplification function scaling factor and function start (to be used with the Gaussian error function).
9-17	3I3	Not Used
18-38	G21.14	True anomaly amplifier scaling factor
39-59	G21.14	True anomaly amplifier start date (days from launch)
60-80	G21.14	Not Used

C.2 Code Modifications and Additions

The purpose of the following section is to afford the user an in-depth outline of the code modifications and additions necessary in the evolution of NT-GTDS from version PR3 (standard PC version) to the current PR6 version. For each of the following sections, changes are stratified into routine modifications, new programs, and new variables. For Section C.2.4, two additional tables are provided, future modifications, and variable comparisons. The future modifications table is included to outline modifications that could make the SLM operable with more GTDS functions and propagators. The variable comparison table is presented to give the reader comparison variables for the SLM with the drag model. It is the author's hope that the interested reader will use these tables in an effort to build a standalone Filter which incorporates all functionality for the SLM model.

C.2.1 MSISE-90 Atmosphere Model Modifications

Table C.1: MSISE-90 Atmospheric Model NT-GTDS Existing Subroutine Modifications

<i>Routine</i>	<i>Modification/Addition</i>	<i>Lines</i>
AERO	<i>Calculates vector forces on spacecraft due to aerodynamic forces</i>	
	Modify Documentation	
	Define height check for MSIS model	
	Add call to MSISDRV	
	* Aerodynamic force plot file added, but moved to AEROLIFT	
	Add check for rho = 0 to avoid division by 0 for C _D position partial	30
FILESBD	<i>Defines reference numbers for standard GTDS input/output files.</i>	
	Modify Documentation	
	Comment out old, un-used NVNGRD identifier	
	Define NMSIS and set it to 76	8

OUTPAD	<i>Generates a report describing the permanent atmospheric density file and/or a complete permanent atmospheric density set.</i>	
	Modify documentation	
	Change NKEY dimension	
	Add if loop to check for MSISE-90 option, and if necessary, call OUTPADMS to print file	12
OUTSEC	<i>Generates sectioning dependent a-priori parameters report</i>	
	Modify documentation	
	Define HMSIS data array and initialize values	
	Add if loop to print MSISE-90 descriptor if needed (IND(86) = 9)	10
SETDAF	<i>Opens all GTDS files</i>	
	Modify documentation	
	Open file 76 for use as the JRMSIS atmospheric data input files	11
SETOG1	<i>Interprets all orbit generator optional keyword cards that come after DRAG in SETORB</i>	
	Modify Documentation	
	Define MSIS common block	
	Add processing loop for MSISOPT1, MSISOPT2, MSISOPT3, MSISOPT4 keyword cards to define MSMASS, INITMSIS, and MSSW array input variables	207
SETORB	<i>Reads and interprets orbit generator optional keyword cards</i>	
	Modify documentation	
	Add MSISOPT1, MSISOPT2, MSISOPT3, and MSISOPT4 cards to the SYM array	
	Increment NKEY counter to 98	20
SETPFR	<i>Reads input options for the Permanent File Report program and calls various subroutines to write the requested permanent file reports.</i>	
	Modify documentation	
	Change NKEY dimension	
	Add file type 3 options for ATMOSRPT	
	Set default setting for type 3 option	
	Set default options for report on input card	
	Add call to OUTPAD	
	Write any error messages, and new error message 245	
	Initialize values for type 3 option if not used	65
TOTAL		363

Table C.2: MSISE-90 Atmospheric Model New NT-GTDS Subroutines¹

<i>Routine</i>	<i>Description</i>	<i>Lines</i>
CSWBD	Internal MSISE-90 common block used to initialize switching options for the model.	31
<i>ELIPSD.CMN</i>	<i>Contains size and shape parameters for the central body ellipsoid.</i>	56
<i>ELIPSDBD</i>	<i>Common block used for the size and shape parameters for the central body ellipsoid, and initialization switches.</i>	74
<i>ELLGEO</i>	<i>Sets size and shape parameters for the reference ellipsoid of the geodetic coordinate system.</i>	146
ERRMSG2.DAT	Error message file which contains a new message (245) to indicate that the specified time is not supported by the current MSIS data file (FRN 76).	binary
F107APBD	Common block used to initialize input variables for the MSISE-90 model.	50
<i>GEOCRT</i>	<i>Transforms earth-fixed Cartesian coordinates to geodetic latitude, longitude, and height.</i>	304
MSISBD	Common block used to initialize options for the MSISE-90 model.	33
MSISDRV	Called by AERO to furnish atmospheric density using the MSISE-90 atmospheric model. The model computes two output arrays with various densities and temperatures. The only thing used by AERO, however is the atmospheric density.	334
MSIS_WF	Fills common block F107AP with a time IT, magnetic activity indices, and Ap values. Depending on switch (9) from the MSIS switching array in common block CSW, Ap is either the daily value or a 7 element array of hourly values.	226
OUTPADMS	Produces a Permanent File Report for the MSISE-90 model in conjunction with OUTPAD.FOR.	271
TOTAL		1525

Table C.3: MSISE-90 Atmospheric Model New NT-GTDS Variables for the CSW Common Block

<i>Variable</i>	<i>Description</i>	<i>Type</i>	<i>I/O</i>
SW	MSISE-90 main switching array (0 = Off, 1 = On, 2 = Off but coupling terms on)	Real*4	I
SWC	MSISE-90 coupling terms array	Real*4	I
ISW	MSISE-90 switch initialization variable	Integer*4	I

¹ Italicized routines were created by Leo Early and taken from the DSST Standalone code.

Table C.4: MSISE-90 Atmospheric Model New NT-GTDS Variables for the ELIPSD Common Block

<i>Variable</i>	<i>Description</i>	<i>Type</i>	<i>I/O</i>
SEMMAJ	Equatorial radius	Real*8	O
SEMMIN	Polar radius	Real*8	O
ECCEN	Eccentricity	Real*8	O
ESQR	Eccentricity ²	Real*8	O
RTESQR	Square root of (1 – Eccentricity ²)	Real*8	O
FLAT	Earth flattening coefficient	Real*8	O
FLTINV	Inverse flattening coefficient	Real*8	O
INIELL	Initialize central-body ellipsoid option switch	Logical	I
PRTELL	Iteration print option switch	Logical	I

Table C.5: MSISE-90 Atmospheric Model New NT-GTDS Variables for the F107AP Common Block

<i>Variable</i>	<i>Description</i>	<i>Type</i>	<i>I/O</i>
F107A	81 Day average of F _{10.7} flux (centered on day 41)	Real*4	O
F107	Daily flux for previous day	Real*4	I
AP	A _p output array if SW(9) = -1	Real*4	O
APD	Magnetic index (daily)	Real*4	I
APH	Temporary A _p hourly values	Real*4	I
FAV	Input temporary array for 81-day F _{10.7} average	Real*4	I
F	Input temporary array for F _{10.7}	Real*4	I
IT	Time of first day of TC data in common block	Integer*4	O
IDREC1	First data point on data file	Integer*4	O
IDRECE	Last day of extrapolated data	Integer*4	O
NREC	Number of records on input file	Integer*4	O
LGD	Last good point on data file	Integer*4	O
APDA	Daily A _p temporary array	Integer*4	O
APHA	Hourly A _p temporary array	Integer*4	O
FIRSTRD	First record read initialization variable	Logical*4	I

Table C.6: MSISE-90 Atmospheric Model New NT-GTDS Variables for the MSIS Common Block

<i>Variable</i>	<i>Description</i>	<i>Type</i>	<i>I/O</i>
MSSW	MSISE-90 input main switching array (0 = Off, 1 = On, 2 = Off but coupling terms on)	Real*4	I
MSMASS	MSISE-90 input mass parameter	Integer*4	I
INITMSIS	MSISE-90 initialization flag	Logical*4	i

C.2.2 Filter Run-Time Modifications

Table C.7: Filter Run-Time NT-GTDS Existing Subroutine Modifications

<i>Routine</i>	<i>Modification/Addition</i>	<i>Lines</i>
ELEME	<i>Converts an element set vector input as Cartesian, Keplerian orbital elements, or spherical coordinates to the remaining two systems.</i>	
	Modify Documentation	
	Add OBSREC common block and define TOBYMD and TOBHMS variables	
	Add SWITCH common block and define IND(48) variable	
	Add DCING common block and define ITER and ITCONV variables	
	Define CDRAG variable from FRC common block	
	Add write statement to output SMA, ECC, time, lift/drag coefficients, DC iteration number, and convergence status so user can monitor DC performance real-time	
	Add if statement for DC specific options, so user can monitor Filter performance real-time with SMA, ECC, and time parameters.	45
INTEGBD	<i>Defines and initializes integrator parameters.</i>	
	Modify Documentation	
	Change MAXIT(1) to increase/decrease maximum number of corrector iterations for CSTEP process	1
RESTAT	<i>Computes observation statistics which are to be printed in the DC report, updates the total number of observations accepted, and computes the weighted residuals</i>	
	Modify documentation	
	Add OWRMS negative square root check	
	Add OSIG negative square root check	
	Add negative/zero check for STASTT(IS,1,ISTA1)	
	Modify several zero checks to negative checks	
	Add temp variable check structure to process through a negative sum of squares value for STASTT(IS,3,ISTA1) numerical accuracy error	
	Add write statements to alert user of negative values	37

RPTRES	<i>Writes out the Kalman Filter observation residuals report.</i>	
	Modify Documentation	
	Add write statement header for residuals output file	
	Add write statements 3001, 3002, 5002, 5003 to print residuals to residual output file	28
SECHEK	<i>Short version of the GTDS sectioning routine used to determine when terminal conditions of an orbit generation have been met.</i>	
	Add OBSREC common block and define TOBYMD and TOBHMS variables	
	Reduce minimum height to 0 km	
	Add write statement to indicate time of impact	15
TOTAL		126

C.2.3 Filter Input Processor Modifications

Table C.8: Filter Input Processor NT-GTDS Existing Subroutine Modifications

<i>Routine</i>	<i>Modification/Addition</i>	<i>Lines</i>
AVRINT	<i>Sets the perturbation options used by the averaged orbit generator and writes informative messages.</i>	
	Modify documentation	
	Added a call to ESKFOUT at the end of the routine to report on the run-time status of the filter options during a filter run	8
SETDAF	<i>Opens all GTDS files</i>	
	Modify documentation	
	Open file 43 for use as a plotting data file of the Kalman Filter history data	
	Open file 44 for use as a print file for observation, transformation, and transition partial derivatives	25
SETFIL	<i>Processes the Kalman Filter option sub-deck</i>	
	Modify documentation	
	Add a call to SETESKF so that NT-GTDS will recognize new input processor keyword cards.	12

SKFSET	<i>Sets the a-priori values of switches used in the semianalytical Kalman Filter</i>	
	Modify documentation	
	Define SKF common block	
	Define ESKFCM common block for holding and processing input data	
	Add processing loop for ESKFCNTL keyword card to define IUPD, INTINV, and IESKF input variables	
	Add processing loop for LKFCNTL keyword card to define ILKFUP, DTLKF, and TLLKF input variables	
	Add processing loop for ESKFOUT keyword card to define ISLVPT, IWPRD, IWUPD, and IPRFIL input variables	
	Add processing loop for ESKFPAR1 keyword card to define IDHDX, IPARTH, IKFPPR, and IPRFL2 input variables	
	Add processing loop for ESKFPAR2 keyword card to define IPHT, IPST, IPHR, IPSR, and IPHTTV input variables	189
TOTAL		234

Table C.9: Filter Input Processor New NT-GTDS Subroutines

<i>Routine</i>	<i>Description</i>	<i>Lines</i>
ESKFOUT	Prints out the status at run-time of the SKF/ESKF options.	220
SETESKF	Interprets the option cards for the Semianalytic Kalman Filter.	303
SKFBD	This common block already exists in GTDS, but not in block data form (its own file). This file was created to simply use of the common block and identify what each variable represents. All Semianalytic Kalman Filter variables are in the SKF common block.	117
ESKFCMBD	This common block is used to flag when Kalman Filter option cards are used (and their respective values) so that SETESKF knows not to use the default values.	35
TOTAL		675

Table C.10: Filter Input Processor New NT-GTDS Variables for the SKF Common Block

<i>Variable</i>	<i>Description</i>	<i>Type</i>	<i>I/O</i>
PARTH	The transformation partials of position and velocity with respect to the solve vector	Real*8	O
PHIS	The State Transition Matrix (PHI)	Real*8	O
PSIS	The partials of the state wrt solve-parameters (PSI)	Real*8	O
VEPIN	Orbit element partials interpolation array	Real*8	O
VEVIND	Orbit element partials rates interpolation array	Real*8	O
WSAV	Temporary STM variable	Real*8	O

TLST	Time at last call of SPSKF (only for ESKF)	Real*8	O
TLPR	Time at last output request (for SKFPRT)	Real*8	O
TLLKF	Initial filter time	Real*8	I
DTLKF	Cowell LKF relinearization interval size	Real*8	I
TLPHI	Time at last calculation of PHI	Real*8	O
SKFTM	Unused spaces in SKF common block	Real*8	O
OMSAV	The mean equinoctial elements, position, velocity, and partials at the integration grid points	Real*8	O
IUPD	Relinearization of the semianalytical nominal trajectory for ESKF/SKF option	Integer*4	I
IOBSP	Correct solve-vector with observations switch	Integer*4	I
IOBPRT	End of filter transformation partials processing switch	Integer*4	I
IESKF	ESKF mode selection	Integer*4	I
ISLVPT	Print nominal trajectory elements before/after adding integer*2 the filter correction option	Integer*4	I
INTINV	Transition matrix inverse interpolation option	Integer*4	I
ILKFUP	Relinearize Cowell LKF at interval DTLKF option	Integer*4	I
IDHDX	Print partials of observations wrt mean equinoctial elements option	Integer*4	I
IPARTH	Print partials of osculating position and velocity wrt mean equinoctial elements option	Integer*4	I
IPHT	Print PHI option (grid start reference)	Integer*4	I
IPST	Print PSI option (grid start reference)	Integer*4	I
IPHR	Print PHI option (last request time reference)	Integer*4	I
IPSR	Print PSI option (last request time reference)	Integer*4	I
IPHTTV	Print PHI*inverse(PHI) to check accuracy option	Integer*4	I
IWUPD	Print solve-vector variances after an observation update option	Integer*4	I
IWPRD	Print solve-vector variances after time update option	Integer*4	I
ISKFOU	Compute new short periodic coefficients for SKF, OSC, PV, & KEP print options	Integer*4	I
IBL1, IBL2, IBL3, ISKFDM	Not currently used	Integer*4	I
IPRFIL	FRN number of filter history data file	Integer*4	I
IKFPPR	Print state transition matrix option	Integer*4	I
IEFFPR	Redundant calculation flag (ORBITV) – currently not used	Integer*4	I
IPRFL2	FRN number of print file for observations, transformations, and transformation partials	Integer*4	I

Table C.11: Filter Input Processor New NT-GTDS Variables for the ESKFCM Common Block

Variable	Description	Type	I/O
ESKFINT	Array which contains the input values from the KF option cards for various filter options	Integer	I/O
ITEM_READ	Logical array which flags that a value has been read in from the KF option input cards	Logical	I/O
ITEMS	Maximum input options parameter	Integer	I/O

C.2.4 Simplified Lift Model Modifications

Table C.12: Simplified Lift Model NT-GTDS Existing Subroutine Modifications

Routine	Modification/Addition	Lines
ACCEL	<i>Calculate perturbative accelerations acting on spacecraft</i>	
	Modify Documentation	
	Define LFTVAR common block for INDLIFT, ACLN, and ACLW	
	Add IF loops to AERO call for lift only options	
	Add ACLN and ACLW to ANCF	30
AERO	<i>Calculates vector forces on spacecraft due to aerodynamic forces</i>	
	Modify Documentation and variable definition structure	
	Define LFTVAR common block	
	Add zero division checks	
	Add Lift acceleration loop	
	Add call to AEROLIFT to determine lift accelerations and partials	15
AEROPR	<i>Initialize spacecraft configuration for AERO</i>	
	Modify Documentation	
	Define LFTVAR common block	
	Calculate SCLNM and SCLWM	
	Calculate SAELN and SAELW	31

COREST	<i>Add filter corrections to estimated parameters</i>	
	Modify Documentation	
	Define LFTPAR common block for LPARN, LPARW, SAELN, and SAELW	
	Define local temporary variables RHOLN, RHOLW, CLNOLD, CLWOLD	
	Add KPAR 6-8 option loop for CSUBL update and define RHOLN/RHOLW	
	Add 3025 and 3050 write statements for CSUBLN and CSUBLW	
	Update SAELN and SAELW parameters	53
DRAGV	<i>Copies partials from common blocks to calling sequence for use in VARFRC</i>	
	Modify documentation	
	Modify calling sequence	
	Modify variable names	
	Define LFTPAR and ESTFLG common blocks	
	Add Lift-N and Lift-W acceleration effects	
	Define N and W dynamic parameter partials	
	Add lift partials to transfer loops	71
GQFUN	<i>Computation of orbital element rates</i>	
	Modify documentation	
	Define LFTPAR common block for ACLN/ACLW variables	
	Define ESTFLG common block for KPAR	
	Increase ACCPRT by ACLN/ACLW as dictated by KPAR 6-8 options before assigning to ACC	23
GVCVL	<i>Alter title array for solve-for parameters</i>	
	Define CLIFTN and CLIFTW data labels	
	Add lift option statements to define VCVTTL with KPAR variable	32
INTIND	<i>Set location indices and find NEQ</i>	
	Modify Documentation	
	Define LFTPAR common block	
	Add KPAR index increment loop for INDY	
	Add loop to equate INDXL and INDYL if no state partials required	19
INTOGF	<i>Integration initialization from ORBIT file</i>	
	Modify Documentation	
	Define ESTFLG common block for KPAR	
	Add If statement for NEQ increment for KPAR 6-7 options	22

INTOGN	<i>Set parameters for orbit generator program & INDX/INDY type arrays</i>	
	Modify Documentation	
	Add KPAR options to NPART increment loop	17
KFUPDT	<i>Updates filter covariance and estimated parameters</i>	
	Modify documentation	
	Define LFTPAR common block for LPARN and LPARW	
	Define ESTFLG common block for KPAR	
	Add CSUBL print option block, increment JJ/KK as necessary according to KPAR	41
OUTOG1	<i>Initial conditions report</i>	
	Modify Documentation	
	Add call to write statements 2048, 2049 to include LPAR and LPARW	29
OUTPAR	<i>Output partials</i>	
	Modify documentation	
	Add KPAR 6-8 options to NPARTS calculation	
	Add HDR2 array and define CLN/CLW data statements	
	Add KPAR 6-8 options to drag print loop (L modifications too)	23
OUTSEC	<i>Generates sectioning dependent a-priori parameters report</i>	
	Modify documentation	
	Define LIFTVAR common block and variable INDLIFT	
	Add If qualified write statement 2020 for lift	24
OUTSLV	<i>Apriori conditions for solve-for, print options</i>	
	Modify Documentation	
	Define LFTPAR common block	
	Add KPAR 6-8 options to drag output loop and calculate sigmas	
	Add Lift output statements	41
PSET	<i>Reset dynamic parameters adjusted by DC</i>	
	Modify Documentation	
	Define LFTPAR common block for LPARN, LPARW, SAELN, and SAELW	
	Define local temporary variables RHOLN and RHOLW	
	Add options 1356 – 1359 on KPAR Goto statement to account for 6-8 options, and define RHOLN/RHOLW for SAELN/SAELW update	42
RPTEST	<i>Generate report of estimated parameters</i>	
	Modify Documentation	
	Define LFTPAR common block for LPARN and LPARW	
	Add KPAR 6-8 loop in the drag print options loop (9030)	18

SETAPC	<i>Alter a priori filter covariance calculation</i>	
	Modify documentation	
	Define LFTPAR common block	
	Increment NDY and N variables to account for new solve-fors	
	Add new values to APCOV for KPAR 6-8 options	25
SETDAF	<i>Opens all GTDS files</i>	
	Modify documentation	
	Open file 50 for use as an aerodynamic vector plot file	
	Open file 99 for use as a debugger output file	18
SETOG1	<i>Interprets all orbit generator optional keyword cards that come after DRAG in SETORB</i>	
	Modify Documentation	
	Define LFTVAR and FILES common block	
	Modify DRAGPAR2 keyword read options	
	Add LIFTPAR keyword read options	
	Add LIFTPAR2 keyword read options	129
SETORB	<i>Reads and interprets orbit generator optional keyword cards</i>	
	Modify Documentation	
	Increase size of keyword array SYM and NKEY descriptor, then add LIFTPAR/LIFTPAR2 cards	16
SOLTAB	<i>Sets up dynamic solve parameter tables and sets necessary switches</i>	
	Modify Documentation	
	Define LFTPAR common block for LPARN, LPARW, VARCLN, VARCLW	
	Add KPAR options 6-8 after CDRAG calculation	
	Add KPAR Goto loop with lines 3000, 3100, 3200 to account for 3 options and solve for LPARN and LPARW.	43
VARFRC	<i>Compute partials of total acceleration vector</i>	
	Modify documentation	
	Define ESTFLG common block for KPAR	
	Add local variables: PLNR, PLNV, PLWR, PLWV, PLN1, PLW1	
	Initialize all lift partials arrays	
	Modify call to DRAGV to accommodate lift changes and alter definition of PX to include lift partials	
	Add loops to define ACCPAR partials for lift in accordance with KPAR 6-8 options	
	Add lift partials to the time-regularized option scaling factors section	42
TOTAL		804

Table C.13: Simplified Lift Model New NT-GTDS Subroutines

<i>Routine</i>	<i>Description</i>	<i>Lines</i>
AEROLIFT	Produces the vector force on a spacecraft due to the aerodynamic lift forces in conjunction with AERO, and computes necessary partials for lift solve-for options	309
LFTPARBD	Common block for lift variables	70
LFTPAR.CMN	Include statement block	28
LIFTACC	Calculates the acceleration due to lift in the specified direction as a function of position - to be used with numerical calculation of position partial derivative	170
VRELPART	Calculates the partial derivatives of the N and W unit vectors with respect to the relative velocity vector	256
TOTAL		833

Table C.14: Simplified Lift Model New NT-GTDS Variables for the LFTPAR Common Block

<i>Variable</i>	<i>Description</i>	<i>Type</i>	<i>I/O</i>
LPARN	Apriori value of CSUBL for the N direction	Real*8	I
LPARW	Apriori value of CSUBL for the W direction	Real*8	I
LDELTA	Numerical integration step size for position partial derivative calculation	Real*8	I
VARCLN	Standard deviation of Lift N variation	Real*8	I
VARCLW	Standard deviation of Lift W variation	Real*8	I
ACLN	Acceleration in the N direction	Real*8	O
ACLW	Acceleration in the W direction	Real*8	O
SAELN	Spacecraft lift equation parameter - N direction	Real*8	O
SAELW	Spacecraft lift equation parameter - W direction	Real*8	O
LNDRADV(3,3)	Partials of lift wrt V in the N direction	Real*8	O
LWDRADV(3,3)	Partials of lift wrt V in the W direction	Real*8	O
LNDRADR(3,3)	Partials of lift wrt R in the N direction	Real*8	O
LWDRADR(3,3)	Partials of lift wrt R in the W direction	Real*8	O
LNDRADI(3)	Partial of ACLN wrt TVARCLN	Real*8	O
LWDRADI(3)	Partial of ACLW wrt TVARCLW	Real*8	O
INDLIFT	Lift acceleration include switch	Integer*4	I
LSWPRN	Lift parameter plot file switch	Integer*4	I
LNUMSW	Numeric/Analytic position partial switch	Integer*4	I

Table C.15: Simplified Lift Model Future NT-GTDS Subroutine Modifications Listing²

<i>Routine</i>	<i>Description</i>
<i>ANHARM</i>	<i>Numerical averaging</i>
<i>AVRINT</i>	<i>Output lift model options to print file</i>
<i>VARSP/VRSPFD</i>	<i>B1/B4 differencing</i>
<i>VOQUAD</i>	<i>Quadrature order (ANAVINBD)</i>
<i>VARDIF/AVEPAR</i>	<i>Compute A & D for averaged variational eqns. (SPDRAG, SSTAPGFL, SSTEMFL)</i>
<i>AVRAGE</i>	<i>Averaged element rates</i>
<i>SETAVR</i>	<i>Sets averaged options</i>
<i>SETSPG</i>	<i>Sets short periodic options (haven't listed any of the multitude of SP routines)</i>
<i>GMAT</i>	<i>Partials of G matrix wrt solve-fors</i>
<i>CSTEPX</i>	<i>Advance time derivatives of VOP elements</i>
<i>AVEPAR</i>	<i>Include lift in element rate partials (second order effects too)</i>
<i>ORBSKF</i>	<i>Semianalytic filter calculations</i>
<i>SPJ2MD</i>	<i>Drag/M-daily coupling</i>
<i>VARANAL</i>	<i>Compute A & D matrices</i>
<i>ANLHDR</i>	<i>Header output for SST variables to Orbit File</i>
<i>SKFMAT</i>	<i>Executive program for calculation of G matrix</i>
<i>SPCOTO</i>	<i>Determines total short periodic coefficients</i>
<i>SPGOUT</i>	<i>Prints SST options at run-time</i>
<i>SKFUPDT</i>	<i>Update covariance for SST Filter</i>
<i>OUTPAR</i>	<i>Output adaptive parameters</i>
<i>GETHDR</i>	<i>Reads header record from ORBIT file</i>
	<i>CNVOM, COMETA, ELERD, INTOGF, OBOUT2, ORBSW, OUTEA3, READWF</i>
<i>OGBUG</i>	<i>Intermediate output for common blocks (IND(11), CSUBD)</i>
<i>ELSWF</i>	<i>Retrieves element sets from a permanent elements file or "24-hour hold" file and places into common block (CSUBD, RHO1).</i>
<i>EPHEM</i>	<i>Writes a satellite ephemeris data sets in the DODS Ephem format – ORB1 (CSUBD, RHO1).</i>
<i>EXCCON</i>	<i>Set switches for considers/solve-fors in generalized consider mode</i>
	<i>Future improvements should include this in order to account for consider options</i>
<i>MANEUV</i>	<i>Updates parameters and partials (ie. SAE) after an impulsive maneuver (CSUBD – note INTOGN must also have SDRAG changes).</i>
<i>ORBINT</i>	<i>Generate satellite ephemeris on ORBIT file (CSUBD).</i>
<i>OUTSIM</i>	<i>Outputs file of simulated observations (CSUBD, RHO1)</i>

² Italicized routines indicate routines necessary to make the SLM work with DSST.

OUTWSD	Creates a working file from the satellite central directory (CSUBD)
WPELWF	Writes the elements computed in a DC run to the permanent elements file (CSUBD).
ERRGET	Lists the partials available on a GTDS ORBIT file as compared to those requested by the user (IND(41)).
CONEXP	To expand and contract certain arrays for the generalized consider mode for DC (VARCD) – EXCCON.
ACPDOT	Period partial derivative contributions (IND(11)), Used in ELEM with IPDOT = 1, for OUTDC8.
FLIP	Interchanges elements and parameters when dealing with satellite to satellite data (note – INTOGN must also have SDRAG changes).
SETEA	EAP – Processes card inputs of EAP related parameters editing criteria and observation data (VARCD).
SCDWF	EAP – Reads data from EAP working file (CSUBD)
OUTEA3	EAP – report mapping results for consider and solve parameters (IDRGSC, IND(41)).
PHIX	EAP – Transform PXTXO into required array phi for each satellite (IDRGSC, IND(41)).
OUTSC	EAP – List the solve and consider parameters, their units, and their a priori standard deviations (IDRGSC, IND(41)).

Table C.16: Simplified Lift Model New NT-GTDS LFTPAR/ Drag Variable Comparison

<i>Lift Variable</i>	<i>Drag Counterpart</i>	<i>Description</i>
LPARN	CSUBDZ - /FRC/ (154)	Coefficient of Lift/Drag
LPARW		
ACLN	ACCD - /PASS/ (40)	Acceleration due to drag/lift
ACLW		
INDLIFT	IND(11) - /SWITCH/ (11)	Lift switch
SAELN	SAE - /THRUST/ (61)	Spacecraft lift/drag equation parameter
SAELW		
LNDRADV(3,3)	DDRADV(3,3) - /FRC/ (46)	Partials wrt V
LWDRADV(3,3)		
LNDRADR(3,3)	DDRADR(3,3) /FRC/ (14)	Partials wrt R
LWDRADR(3,3)		
VARCLN	VARCD - /DCINT/ (199)	Standard deviation of lift/drag variation
VARCLW		
LNDRAD1(3)	DDRAD1(3) - /FRC/ (172)	Partial of Acceleration wrt time variation
LWDRAD1(3)		

C.2.5 Error Function Modifications

Table C.17: Error Function NT-GTDS Existing Subroutine Modifications

<i>Routine</i>	<i>Modification/Addition</i>	<i>Lines</i>
ANOISE	<i>Computes the analytic state process noise for the Kalman Filter.</i>	
	Modify documentation	
	Define ERRFN common block	
	Define FILES common block and write statements if Q matrix is desired	
	Add call to ERRFUNCT	
	Add QERRFN process noise matrix update to the Q matrix	42
SETFIL	<i>Processes the Kalman Filter option sub-deck</i>	
	Modify documentation	
	Define TEMPJUL and TEMPSEC temporary variables	
	Increase size of SYM array	
	Define ERRFN common block	
	Set equivalence for FILES output file NOUT for use with output statements	
	Add all new keyword card names: ALTERROR, ALTERRZN, GAUSSERR, GAUSERR1, GAUSERR2, GAUSERR3, GAUSERR4, GAUSSPER, and TAERROR	
	Increase NBRKEY to 19	
	Initialize ALTFLG variable	
	Add new lines to READ branch statement for the new cards	
	Define all ERRFN input variables using data from keyword cards in specified entry paths	
	Define output write statements for both ADEF and time dependent functions to output error function statistics to output file	
	If ADEF is being used, calculate ALTHW, ALTINTHW, ZSTART, and ZSTOP arrays and output using ADEF write statement	
	If time dependent being used, calculate time statistics and output using time dependent write statement	380
TOTAL		422

Table C.18: Error Function New NT-GTDS Subroutines

<i>Routine</i>	<i>Description</i>	<i>Lines</i>
ALTEF	Returns a value for the altitude dependent error function to be used with the filter program for process noise when variable EFSW is true.	127
ERRFNBD	Block data which initializes variables for the error function estimators of the Q matrix for the KF, EKF, SKF, and ESKF programs.	107
ERRFUNCT	Define the process noise matrix update array using the functional process noise representations.	239
GNEF4	Returns a value for the four Gaussian normal error function to be used with the filter program when variable EFSW is true.	99
TOTAL		572

Table C.19: Error Function New NT-GTDS Variables for the ERRFN Common Block

<i>Variable</i>	<i>Description</i>	<i>Type</i>	<i>I/O</i>
LYEAR	Year of satellite launch	Real*8	I
LMON	Month of satellite launch	Real*8	I
LDAY	Day of satellite launch	Real*8	I
JLAUNCH	Julian date of satellite launch	Real*8	O
GEFSD	Initial standard deviation of error function	Real*8	I
KXYZ	Scaling factor for position components	Real*8	I
KXYZDOT	Scaling factor for velocity components	Real*8	I
PERLNCH	Period of satellite orbit during mission phase	Real*8	I
PERDECAY	Period of satellite orbit just prior to decay	Real*8	I
DREMNR	Remainder of day of launch	Real*8	I
KGN	Gaussian scalar multiplier array	Real*8	I
MUGN	Gaussian mean array	Real*8	I
SDGN	Gaussian standard deviation array	Real*8	I
TAK	True anomaly function scaling factor	Real*8	I
TASTART	True anomaly function start	Real*8	I
ALTZONE	Altitude zone termination altitude array	Real*8	I
ALTZERR	Altitude zone error level array	Real*8	I
ALTHW	Altitude zone half width array	Real*8	O
ALTINTHW	Altitude zone interpolation half width array	Real*8	O
QERRFN	Functional process noise matrix update array	Real*8	O
ZSTART	Interpolation interval start array	Real*8	O
ZSTOP	Interpolation interval stop array	Real*8	O
GNSW	Gaussian normal switch function array (1 = use, 0 = do not use)	Integer*4	I

EFSW	Error function switch variable (True = use, False = do not use)	Logical*4	I
ALTFLG	ADEF or Gaussian error function flag (True = ADEF, False = Gaussian)	Logical*4	I

C.2.6 PR4 Operational Modifications**

Table C.20: PR4 Operational NT-GTDS Existing Subroutine Modifications

Routine	Modification/Addition	Lines
BROLYD	<i>Brouwer-Lyddane orbit generator routine</i>	
	Modify Documentation	
	Removed numerous superfluous write statements to the output file FRN 6	26
JACROB	<i>Called by AERO to furnish atmospheric density using Jacchia-Roberts 1971 atmospheric model.</i>	
	Modify Documentation	
	Added numerical accuracy error check in the calculation of the sun hour angle (SHA)	12
RESTAT	<i>Computes observation statistics which are to be printed in the DC report, updates the total number of observations accepted, and computes the weighted residuals</i>	
	Modify documentation	
	Include write statement and error check for negative root of standard deviation due to numerical accuracy error (Test case 18)	20
SETOG1	<i>Interprets all orbit generator optional keyword cards that come after DRAG in SETORB</i>	
	Modify Documentation	
	Cefola added PPT2_TLC keyword card for Tesseral Linear Combination model options and necessary input variables (17 JUL 96)	27
SETORB	<i>Reads and interprets orbit generator optional keyword cards</i>	
	Cefola modified SYM array and NKEY to add PPT2_TLC keyword card (17 JUL 96)	2
TOTAL		87

** The reader should note that other (non-listed) modifications were done to routines in the PPT2_SOU directory by Cefola.

C.2.7 Range-Gating Option Modifications

Table C.21: Range-Gating Option NT-GTDS Existing Subroutine Modifications

<i>Routine</i>	<i>Modification/Addition</i>	<i>Lines</i>
CONSTBD	<i>Defines and initializes variables in the CONST common block</i>	
	Modify Documentation	
	Add variable RANGTSW to common block in space 50	
	Modify spares variable to single space and redefine SPR	
	Initialize RANGTSW to 0	8
RANGAT	<i>Finds the range ambiguity correction for SRE, USB, ATSR, and GRARR trackers.</i>	
	Modify documentation	
	Add CONST common block and define variable RANGTSW	
	Add return loop to execute if range – gating option is off	15
SETOG1	<i>Interprets all orbit generator optional keyword cards that come after DRAG in SETORB</i>	
	Modify Documentation	
	Add CONST common block and define variable RANGTSW	
	Define MSIS common block	
	Add processing loop for RANGEGAT keyword card to define the RANGTSW input variable	15
SETORB	<i>Reads and interprets orbit generator optional keyword cards</i>	
	Modify documentation	
	Add RANGEGAT card to the SYM array	
	Increment NKEY counter	4
TOTAL		42

Table C.22: Range-Gating Option New NT-GTDS Variables for the CONST Common Block

<i>Variable</i>	<i>Description</i>	<i>Type</i>	<i>I/O</i>
RANGTSW	Range – gating option switch (1 = skip, 0 = calculate)	Integer*4	I

C.3 GTDS Modification Summary

In summary, there were 2,078 lines of NT-GTDS code modified in the following 39 routines:

Table C.23: Listing of All NT-GTDS Subroutines Modified

ACCEL	INTEGBD	RESTAT
AERO	INTIND	RPTEST
AEROPR	INTOGF	RPTRES
ANOISE	INTOGN	SECHEK
AVRINT	JACROB	SETAPC
BROLYD	KFUPDT	SETDAF
CONSTBD	OUTOG1	SETFIL
COREST	OUTPAD	SETOG1
DRAGV	OUTPAR	SETORB
ELEME	OUTSEC	SETPFR
FILESBD	OUTSLV	SKFSET
GQFUN	PSET	SOLTAB
GVCVL	RANGAT	VARFRC

In addition, 3,605 lines of code in the following 24 new routines were added to GTDS. The reader should note that the MSISE-90 routines, listed and described in Table 6.1 were also added as a result of this investigation.

Table C.24: Listing of All New NT-GTDS Subroutines

AEROLIFT	ERRMSG2.DAT	LFTPAR.CMN
ALTEF	ESKFCMBD	MSISBD
CSWBD	ESKFOUT	MSISDRV
ELIPSD.CMN	F107APBD	MSIS_WF
ELIPSDBD	GEOCRT	OUTPADMS
ELLGEO	GNEF4	SETESKF
ERRFNBD	LIFTACC	SKFBD
ERRFUNCT	LFTPARBD	VRELPART

Appendix D

Selected GTDS Control Cards

Chapter 4 describes all the necessary files to make execute a NT-GTDS run, including Makefiles, data assignment files, control cards, and batch execution files. With the exception of the GTDS control cards, these files are fairly straightforward and, as such, do not require further explanation. The control cards, however, are a very complicated combination of input parameters. In order to assist the reader in replicating the results of this thesis, Appendix D is a listing of representative control cards from the various types of NT-GTDS runs executed in this investigation. The reader should note that the station location and accuracy data included in the control cards is run and time dependent, and an analysis should be done using the tools from Appendix A to determine which stations should be used. All control cards used in this investigation are kept in various directories on the D:\ drive of \$E53-BOB. The reader is encouraged to reference the *GTDS User's Guide* ([82], [129]) and Appendix C to identify what the various keyword option values used represent.

D.1 Differential Correction Batch and Ephemeris Generation Runs

The Differential Correction Batch Runs described in Sections 5.3 and 7.2 were used for very different purposes and are both presented below. In Section 5.3, a series of EPHEM, DC, and COMPARE runs between generated truth Cowell data and DSST data were used to identify the J_2^2 truncation error. The control cards for the truth and J_2 minimized runs, as well as a sample COMPARE run, are presented below. Section 7.2 was the first attempt at processing real data from catastrophic decay cases at

approximately nine months before decay. Example control cards are given for both the Cowell and DSST propagators within this region.

D.1.1 J_2^2 Error Identification Runs

The first step in the J_2^2 error identification was to generate a Cowell truth orbit and a corresponding ORB1 file to compare with the full model DSST EPHEM ORB1 file generated with elements from a DSST DC fit over the Cowell data (Figure D.1). The EPHEM, DC, and COMPARE programs are all called in Figure D.1 to do a complete comparison of the truth model and full DSST model. Next, the process was repeated with the addition of the CNM keyword card given in Figure D.2 to minimize J_2 .

CONTROL	EPHEM				MOLNIYA	12066
EPOCH			950101.0	110328.656576		
ELEMENT1	1 2 1		26468.65555714934	0.7420325967186030	62.57257988924757	
ELEMENT2			115.3960820961710	261.7856598626032	16.43842432368172	
OUTPUT	1 2 1		950501.0	113947.000000	432000.0	
ORBTYPE	1 1 1		250.0			
OGOPT						
POTFIELD	1 4					
MAXDEGEQ	1		2.			
MAXORDEQ	1		0.			
DRAG	1		1.0			
ATMOSDEN		9				
MSISOPT2			-1.0			
SPDRAG	0					
SOLRAD	1		1.0			
SPSRP	0					
SCPARAM			1.0D-6	100.D0		
SPGRVFR	1 1 3		1.0	1.0	3.0	
SPZONALS	4 3 8					
SPMDAILY	4 4 2					
STATEPAR	3					
STATETAB	1 2 3		4.0	5.0	6.0	
DRAGPAR	1					
DRAGPAR2	1 1					
SSTESTFL	1 2 0		0.0			
SSTAGFL	1 0 0		1.0	0.0	1.0	
OUTOPT	1		950101113947.000	950501113947.000	3600.0	
END						
FIN						
CONTROL	DC				MOLNIYA	12066
EPOCH			950101.0	110328.656576		
ELEMENT1	1 6 1		26468.65555714934	0.7420325967186030	62.57257988924757	
ELEMENT2			115.3960820961710	261.7856598626032	16.43842432368172	
OBSINPUT	15		950101113947.000	950301113947.000		
ORBTYPE	5 1 1		43200.0	1.0		

```

DCOPT
EDIT      2      3.0
PRINTOUT  1      4      1.
CONVERG   30     1      1.D-3
END
OGOPT
POTFIELD  1      4
MAXDEGEQ  1      2.
MAXORDEQ  1      0.
DRAG      1      1.0
ATMOSDEN          9
MSISOPT2          -1.0
SPDRAG    0
SOLRAD    1      1.0
SPSRP     0
SCPARAM          1.0D-6      100.DO
SPGRVFRC  1      1      3      1.0      1.0      3.0
SPZONALS  4      3      8
SPMDAILY  4      4      2
STATEPAR  3
STATETAB  1      2      3      4.0      5.0      6.0
DRAGPAR   1
DRAGPAR2  1      1
SSTESTFL  1      2      0      0.0
SSTAPGFL  1      0      0      1.0      0.0      1.0
END
FIN
CONTROL   EPHEM          OUTPUT          MOLNIYA      12066
OUTPUT    1      2      1      950301.0      113947.000      432000.0
ORBTYPE   5      1      1      43200.0      1.0
OGOPT
POTFIELD  1      4
MAXDEGEQ  1      2.
MAXORDEQ  1      0.
DRAG      1      1.0
ATMOSDEN          9
MSISOPT2          -1.0
SPDRAG    0
SOLRAD    1      1.0
SPSRP     0
SCPARAM          1.0D-6      100.DO
SPGRVFRC  1      1      3      1.0      1.0      3.0
SPZONALS  4      3      8
SPMDAILY  4      4      2
STATEPAR  3
STATETAB  1      2      3      4.0      5.0      6.0
DRAGPAR   1
DRAGPAR2  1      1
SSTESTFL  1      2      0      0.0
SSTAPGFL  1      0      0      1.0      0.0      1.0
OUTOPT    21     2      1      950101113947.000      950501113947.000      3600.0
END
FIN
CONTROL   COMPARE
COMPOPT
CMPEPHEM  1102102      950101113947.000      950301113947.000      1080.00
CMPLOT    3
HISTPLOT  2102102      950101113947.000      950301113947.000      64800.0
END
FIN

```

CONTROL	COMPARE			
COMPOPT				
CMPEPHEM	1102102	950301113947.000	950501113947.000	1080.00
CMPLOT	3			
HISTPLOT	2102102	950301113947.000	950501113947.000	64800.0
END				
FIN				

Figure D.1: J_2^2 Truth Model EPHEM-DC-EPHEM-COMPARE Control Card

CNM	3	2	0	-.00108263D-05	1
-----	---	---	---	----------------	---

Figure D.2: Minimized J_2 Keyword Card

D.1.2 Real Data Runs

Using a one month fit span, DC batch fits were accomplished for NSSC-13390 with both Cowell (Figure D.3) and DSST (Figure D.4) propagators. The reader should note that later in the decay, noise values for the sensors were increased by factors of ten in order to allow for an acceptable fit. Eventually, by about the six month point, fits could no longer be obtained.

CONTROL	DC			NSSC	13390
EPOCH		940531.0	184339.342144		
ELEMENT1	1 2 1	25410.5537264	0.7403623	62.2855	
ELEMENT2		217.2979	250.96733	21.4273	
OBSINPUT	5	940601000000.00	940701000000.00		
ORBTYP	1 1 1	150.0			
DMOPT					
/CAGC	100211 13	1510.200	334902.0478	2532022.9045	
/CBGC	100212 13	1510.200	334902.0500	2532024.6167	
/MAGC	100231 13	3058.600	204228.8382	2034431.4261	
/MBGC	100232 13	3058.600	204228.8418	2034432.5421	
/ALTQ	100334 3	62.700	92413.0781	1672856.9910	
/GOOQ	100382 3	741.880	305841.8440	2592649.8120	
/GOOF	100383 3	741.880	305841.8440	2592649.8120	
/ROBF	100385 3	55.880	323452.5360	2762550.8440	
/EGLQ	100399 3	0.380	303420.7790	2734706.5526	
/NAVQ	100745 3	305.300	333314.3388	2611413.5272	
/MAUC	100951 13	3059.500	204230.6810	2034431.8649	
END					


```

DCOPT
EDIT          2          3.0
TRACKELV    3 13          5.0          5.0
/CAGC      006007        18.0          18.0
/CBGC      006007        18.0          18.0
/MAGC      006007        18.0          18.0
/MBGC      006007        18.0          18.0
/ALTQ      001004005      8.0          28.8          28.8
/GOOQ      001004005     40.0          36.0          36.0
/GOOF      001004005     40.0          36.0          36.0
/ROBF      001004005     40.0          36.0          36.0
/EGLQ      001004005     30.0          45.0          45.0
/EGLQ      009           500.0
/NAVQ      001004005    1979.0          64.8          122.4
/MAUC      006007         9.0          9.0
PRINTOUT    1          4
CONVERG     26          1          1.0
ELLMODEL    1          6378.135          298.26
ELLMODEL    2          6378.137          298.257223563
/CAGC      200001
/CBGC      200001
/MAGC      200001
/MBGC      200001
/ALTQ      200002
/GOOQ      200002
/GOOF      200002
/ROBF      200002
/EGLQ      200001
/NAVQ      200002
/MAUC      200002
END
OGOPT
DRAG        1          1.0
SCPARAM          1.D-6          100.D0
ATMOSDEN          9
MSISOPT2          -1.0
SOLRAD       1          1.0
POTFIELD    1  4
MAXDEGEQ    1          8.0
MAXORDEQ    1          8.0
STATEPAR     2
STATETAB    1  2  3  4.0          5.0          6.0
DRAGPAR      1
DRAGPAR2     1  1
END
FIN
CONTROL     EPHEM          OUTPUT          NSSC          12066
OUTPUT      1  2  1  940701.0          000000.00          86400.0
ORBTYPE     1  1  1  150.0
OGOPT
DRAG        1          1.0
ATMOSDEN          9
MSISOPT2          -1.0
SOLRAD       1          1.0
SCPARAM          1.0D-6          100.D0
DRAGPAR      0
OUTOPT      1  2  1  940601000000.00          940701000000.00          3600.0
END
FIN

```

Figure D.3: Real Data DC-EPHEM Batch Fit Control Card - Cowell Propagator

CONTROL	DC				NSSC	13390
EPOCH			940531.0	184339.342144		
ELEMENT1	1 2 1		25410.5537264	0.7403623	62.2855	
ELEMENT2			217.2979	250.96733	21.4273	
OBSINPUT	5		940601000000.00	940701000000.00		
ORBTYP	5 1 1		43200.0	1.0		
DMOPT						
/CAGC	100211 13		1510.200	334902.0478	2532022.9045	
/CBGC	100212 13		1510.200	334902.0500	2532024.6167	
/MAGC	100231 13		3058.600	204228.8382	2034431.4261	
/MBGC	100232 13		3058.600	204228.8418	2034432.5421	
/ALTQ	100334 3		62.700	92413.0781	1672856.9910	
/GOOQ	100382 3		741.880	305841.8440	2592649.8120	
/GOOF	100383 3		741.880	305841.8440	2592649.8120	
/ROBF	100385 3		55.880	323452.5360	2762550.8440	
/EGLQ	100399 3		0.380	303420.7790	2734706.5526	
/NAVQ	100745 3		305.300	333314.3388	2611413.5272	
/MAUC	100951 13		3059.500	204230.6810	2034431.8649	
END						
DCOPT						
EDIT	2		3.0			
TRACKELV	3 13		5.0	5.0		
/CAGC	006007		180.0	180.0		
/CBGC	006007		180.0	180.0		
/MAGC	006007		180.0	180.0		
/MBGC	006007		180.0	180.0		
/ALTQ	001004005		80.0	280.8	280.8	
/GOOQ	001004005		400.0	360.0	360.0	
/GOOF	001004005		400.0	360.0	360.0	
/ROBF	001004005		400.0	360.0	360.0	
/EGLQ	001004005		300.0	450.0	450.0	
/EGLQ	009		5000.0			
/NAVQ	001004005		19790.0	640.8	1220.4	
/MAUC	006007		90.0	90.0		
PRINTOUT	1 4					
CONVERG	30 1				1.0	
ELLMODEL	1		6378.135	298.26		
ELLMODEL	2		6378.137	298.257223563		
/CAGC	200001					
/CBGC	200001					
/MAGC	200001					
/MBGC	200001					
/ALTQ	200002					
/GOOQ	200002					
/GOOF	200002					
/ROBF	200002					
/EGLQ	200001					
/NAVQ	200002					
/MAUC	200002					
END						

```

OGOPT
DRAG      1      1.0
SCPARAM          1.D-6      100.D0
ATMOSDEN      9
MSISOPT2     -1.0
SPDRAG      0
SPSRP      0
SOLRAD      1      1.0
SPGRVFRC    1  1  3  1.0      1.0      3.0
SPZONALS    4  3  8
SPMDAILY    4  4  2
POTFIELD    1  4
MAXDEGEQ    1      8.0
MAXORDEQ    1      8.0
STATEPAR    3
STATETAB    1  2  3  4.0      5.0      6.0
DRAGPAR     1
DRAGPAR2    1  1
SSTESTFL    1  2  0  1.0
SSTAPGFL    1  0  0  1.0      0.0      1.0
END
FIN
CONTROL  EPHEM          OUTPUT          NSSC      12066
OUTPUT   1  2  1  940701.0      000000.00      86400.0
ORBTYPE  5  1  1  43200.0      1.0
OGOPT
DRAG      1      1.0
ATMOSDEN      9
MSISOPT2     -1.0
SPDRAG      0
SOLRAD      1      1.0
SPSRP      0
SCPARAM          1.0D-6      100.D0
SPGRVFRC    1  1  3  1.0      1.0      3.0
SPZONALS    4  3  8
SPMDAILY    4  4  2
DRAGPAR     0
OUTOPT     1  2  1  940601000000.00      940701000000.00      3600.0
END
FIN

```

Figure D.4: Real Data DC-EPHEM Batch Fit Control Card - DSST Propagator

D.2 Kalman Filter Runs

The Kalman Filter was used extensively during this investigation. Besides the obvious time and station differences between runs, the three major differences were the use of the Time Dependent Error Function (TDEF), Altitude Dependent Error Function (ADEF), and Simplified Lift Model (SLM). Since all of these runs are very similar, only two full cards will be presented in addition to the differing functionality blocks. Figure

D.5 is a DSST Extended Filter run using the TDEF, and Figure D.6 shows a Cowell Linearized Filter run. Figure D.7 lists a sample constant noise matrix used with a February run with NSSC-13390, where SPNOISE is included in the FILOPT sub-deck. Figure D.8 is a listing of the necessary keyword cards to modify and use the ADEF. Finally, D.9 is an EPHEM run to generate a truth ORB1 file that incorporates lift accelerations, and D.10 lists the necessary keyword cards to solve for both lift coefficients, where all cards presented must be in the OGOPT sub-deck.

CONTROL	FILTER				NSSC	13390
EPOCH			941001.00		0.0000000	
ELEMENT1	1 2 1		25343.897544947049		0.7407693665899586	62.13100647032569
ELEMENT2			194.12685289639759		252.84532753344249	163.89896530387509
OBSINPUT	5		941001010000.00		941129000000.00	
ORBTYPE	5 1 1		43200.0		1.0	
DMOPT						
/CAGC	100211 13		1510.200		334902.0478	2532022.9045
/CBGC	100212 13		1510.200		334902.0500	2532024.6167
/CCGC	100213 13		1510.200		334901.2558	2532024.4061
/MAGC	100231 13		3058.600		204228.8382	2034431.4261
/MBGC	100232 13		3058.600		204228.8418	2034432.5421
/MCGC	100233 13		3058.700		204230.5690	2034432.6400
/DBGC	100242 13		-61.200		-72441.9533	722708.3149
/ALTQ	100334 3		62.700		92413.0781	1672856.9910
/GOOQ	100382 3		741.880		305841.8440	2592649.8120
/GOOF	100383 3		741.880		305841.8440	2592649.8120
/ROBF	100385 3		55.880		323452.5360	2762550.8440
/EGLQ	100399 3		0.380		303420.7790	2734706.5526
/NAVQ	100745 3		305.300		333314.3388	2611413.5272
/MAUC	100951 13		3059.500		204230.6810	2034431.8649
END						
DCOPT						
EDIT		2	3.0			
TRACKELV	3 13		0.0		0.0	
/CAGC	006007		180.0		180.0	
/CBGC	006007		180.0		180.0	
/CCGC	006007		27.0		27.0	
/MAGC	006007		180.0		180.0	
/MBGC	006007		180.0		180.0	
/MCGC	006007		27.0		27.0	
/DBGC	006007		18.0		18.0	
/ALTQ	001004005		80.0		280.8	280.8
/GOOQ	001004005		400.0		360.0	360.0
/GOOF	001004005		400.0		360.0	360.0
/ROBF	001004005		400.0		360.0	360.0
/EGLQ	001004005		300.0		450.0	450.0
/EGLQ	009		5000.0			
/NAVQ	001004005		19790.0		640.8	1220.4
/MAUC	006007		90.0		90.0	
PRINTOUT	1 4		22.0			
CONVERG	30 1					1.0
ELLMODEL	1		6378.135		298.26	
ELLMODEL	2		6378.137		298.257223563	

```

/CAGC 200001
/CBGC 200001
/CCGC 200001
/MAGC 200001
/MBGC 200001
/MCGC 200001
/DBGC 200001
/ALTQ 200002
/GOOQ 200002
/GOOF 200002
/ROBF 200002
/EGLQ 200001
/NAVQ 200002
/MAUC 200002
END
OGOPT
COVARNC 11 12 13 0.2789141E-03 0.1146060E-06 0.1909816E-05
COVARNC 14 15 16 0.1529662E-05 0.3672238E-06 -0.3003441E-05
COVARNC 22 23 24 0.7851803E-09 0.9623243E-09 -0.8924471E-09
COVARNC 25 26 33 0.2361157E-08 -0.1281194E-08 0.4289842E-07
COVARNC 34 35 36 0.3915589E-07 -0.2481267E-07 -0.6409061E-07
COVARNC 44 45 46 0.4688122E-07 -0.2163702E-07 -0.5719133E-07
COVARNC 55 56 66 0.7690499E-07 0.3569489E-07 0.9660173E-07
ATMOSDEN 9
MSISOPT2 -1.0
DRAG 1 1.0
DRAGPAR 1
DRAGPAR2 1 1
DRAGPAR 3 0 1.0
SCPARAM 5.8765D-6 1200.0
SPDRAG 0
SPSRP 0
SOLRAD 1 1.0
SPGRVFRC 1 1 3 1.0 1.0 3.0
SPZONALS 4 3 8
SPMDAILY 4 4 2
POTFIELD 1 4
MAXDEGEQ 1 8.0
MAXORDEQ 1 8.0
STATEPAR 3
STATETAB 1 2 3 4.0 5.0 6.0
SSTESTFL 1 2 0 1.0
SSTAPGFL 1 0 0 1.0 0.0 1.0
END
FILOPT
FILTER 2 1 1.0 259200.0
INPUT 1
DYNOISE 1.0E-8
GAUSSERR 82 7 22 0.0 1.0 0.1
GAUSSPER 10.4954 11.6547 1.5485
GAUSERR1 1 375.0 4300.0 350.0
GAUSERR2 1 160.0 4630.0 90.0
GAUSERR3 1 3200.0 4706.0 45.0
GAUSERR4 1 79.0 4621.0 9.0
TAERROR 2.64 3200.0
ESKFCNTL 1 2 1
END
FIN

```

Figure D.5: Extended Kalman Filter Run Control Card – DSST Propagator

CONTROL	FILTER		NSSC 13390		
EPOCH			950308.00	190000.00	
ELEMENT1	1	2	1	15103.26813693932	0.5723616112581422
ELEMENT2				161.3063778788564	255.4861613203682
OBSINPUT	5			950308190001.00	950308200000.00
ORBTYP	2	1	1	10.0	
DMOPT					
/ALTQ	100334	3		62.700	92413.0781
/ASCQ	100354	3		56.100	-75414.1869
/ANTQ	100363	3		5.000	170840.1240
/MILQ	100369	3		123.100	423702.6530
/ROBQ	100384	3		55.880	323452.5360
/ROBF	100385	3		55.880	323452.5360
/PPEF	100387	3		50.580	414508.8200
/PPWQ	100388	3		82.780	390809.8520
/PPWF	100389	3		82.780	390809.8520
/EGLF	100398	3		0.380	303420.7790
/EGLQ	100399	3		0.380	303420.7790
/SNDQ	100741	3		119.600	323438.6962
/ELBQ	100742	3		1411.100	332645.4697
/SILQ	100743	3		4.300	330848.3065
/FTSQ	100744	3		24.660	315836.5268
/RDRQ	100746	3		51.900	331950.3864
/HAWQ	100747	3		71.300	321715.1012
/MAUC	100951	13		3059.500	204230.6810
END					
DCOPT					
EDIT		2		30.0	
TRACKELV	3	13		0.0	0.0
/ALTQ	001004005			8.0	28.8
/ASCQ	001004005			77.0	90.0
/ASCQ	009			120.0	
/ANTQ	001004005			27.0	25.2
/ANTQ	009			100.0	
/MILQ	001004005			2.5	14.8
/MILQ	009			0.5	
/ROBQ	001004005			40.0	36.0
/ROBF	001004005			40.0	36.0
/PPEF	001004005			40.0	36.0
/PPWQ	001004005			40.0	36.0
/PPWF	001004005			40.0	36.0
/EGLF	001004005			100.0	180.0
/EGLF	009			500.0	
/EGLQ	001004005			30.0	45.0
/EGLQ	009			500.0	
/SNDQ	001004005			1979.0	64.8
/ELBQ	001004005			1979.0	64.8
/SILQ	001004005			1979.0	64.8
/FTSQ	001004005			1979.0	64.8
/RDRQ	001004005			1979.0	64.8
/HAWQ	001004005			1979.0	64.8
/MAUC	006007			9.0	9.0
PRINTOUT	1		4		
CONVERG	40		1		1.0
ELLMODEL	1			6378.135	298.26
ELLMODEL	2			6378.137	298.257223563

```

/ALTQ 200002
/ASCQ 200002
/ANTQ 200002
/MILQ 200001
/ROBQ 200002
/ROBF 200002
/PPEF 200002
/PPWQ 200002
/PPWF 200002
/EGLF 200001
/EGLQ 200001
/SNDQ 200002
/ELBQ 200002
/SILQ 200002
/FTSQ 200001
/RDRQ 200002
/HAWQ 200002
/MAUC 200002
END
OGOPT
COVARNC 11 12 13 1.0 0.0 0.0
COVARNC 14 15 16 0.0 0.0 0.0
COVARNC 22 23 24 1.0 0.0 0.0
COVARNC 25 26 33 0.0 0.0 1.0
COVARNC 34 35 36 0.0 0.0 0.0
COVARNC 44 45 46 1.0E-6 0.0 0.0
COVARNC 55 56 66 1.0E-6 0.0 1.0E-6
ATMOSDEN 9
MSISOPT2 -1.0
DRAG 1 1.0
SOLRAD 1 1.0
SCPARAM 5.8765D-6 1200.0
POTFIELD 1 4
MAXDEGEQ 1 8.0
MAXORDEQ 1 8.0
STATEPAR 2
STATETAB 1 2 3 4.0 5.0 6.0
END
FILOPT
FILTER 2 1 1.0
INPUT 1
DYNOISE 1.0E-8
GAUSSERR 82 7 22 0.0 0.001 0.0
GAUSSPER 10.4954 11.6547 1.5485
GAUSERR1 1 375.0 4300.0 350.0
GAUSERR2 1 160.0 4630.0 90.0
GAUSERR3 1 3200.0 4706.0 45.0
GAUSERR4 1 79.0 4621.0 9.0
TAERROR 2.64 3200.0
ESKFCNTL 1 2 1
END
FIN

```

Figure D.6: Linearized Kalman Filter Run Control Card – Cowell Propagator

DYNOISE			1.0E-8		
SPNOISE	11	12	13	2.0E-8	0.0
SPNOISE	14	15	16	0.0	0.0
SPNOISE	22	23	24	2.0E-16	0.0
SPNOISE	25	26	33	0.0	0.0
SPNOISE	34	35	36	0.0	2.0E-16
SPNOISE	44	45	46	2.0E-17	0.0
SPNOISE	55	56	66	2.0E-17	0.0
					8.0E-16

Figure D.7: Sample SPNOISE Constant Noise Matrix Definition

ALTERROR	1		0.001	0.001
ALTERRZN	1	1	100.0	0.97
ALTERRZN	2	2	110.0	
ALTERRZN	3	3		0.85
ALTERRZN	1	4	400.0	0.12
ALTERRZN	1	5	800.0	0.05
ALTERRZN	1	6	3000.0	0.00253

Figure D.8: Sample Altitude Dependent Error Function Definition

CONTROL	EPHEM			NSSC	13390
EPOCH			950318.00	100356.1600012	
ELEMENT1	1	2	1	8943.140571980275	0.2773462717071357
ELEMENT2				150.14647130797340	256.67790849037198
OUTPUT	1	2	1	950325.0	001900.00
ORBTYP	2	1	1	10.0	7200.0
OGOPT					
ATMOSDEN			9		
MSISOPT2			-1.0		
DRAG	1		1.0		
DRAGPAR	3	0	0.85		
LIFTPAR	1		36.0	-0.5	
LIFTPAR2	1		0.05	0.04	
SOLRAD	1		1.0		
SCPARAM			5.8765D-6	1200.0	
POTFIELD	1	4			
MAXDEGEQ	1		8.0		
MAXORDEQ	1		8.0		
STATEPAR	2				
STATETAB	1	2	3	4.0	5.0
OUTOPT	1			950318100400.000	950325001900.000
					120.0
END					
FIN					

Figure D.9: Simplified Lift Model Ephemeris Generation Run Control Card

DRAG	1		1.0		
DRAGPAR	1				
DRAGPAR2	1	8			
DRAGPAR	3	0	0.85		
LIFTPAR	1	1	36.0	-0.5	1.0
LIFTPAR2	1		0.05	0.04	

Figure D.10: Simplified Lift Model Solve-For Required Keyword Cards

D.3 Permanent File Report Runs

In order to develop the data file Table in Chapter 4, Table 4.2, it was necessary to run permanent file reports on some of the data files. In addition, the permanent file reports were very useful in validating the MSISE-90 data files developed as shown in Figure 6.4. Given their small size, the following control cards are presented in their entirety. The reader should note that only the SLP file report (Figure D.11 and D.12, used for GTDS Files \$014 and \$078 respectively) and atmospheric file reports (Figure D.13 and D.14, used for GTDS files \$075 and \$076 respectively) are presented below. The PFR program can actually do reports for all types of data files using the following keyword cards:

- CONSTRPT (astrodynamics constants file)
- ELSRPT (GTDS permanent elements file)
- ELS24RPT (GTDS 24-hour hold elements file)
- EPOTRPT (Earth potential fields file)
- GEODRPT (tracking stations geodetics file)
- INTCRPT (integration coefficients file)
- LPOTRPT (lunar potential field file)
- MANURPT (impulsive maneuvers file)
- SECTRPT (flight sectioning model file)

```
CONTROL  FILERPT
PFROPT
SLPRPT
END
FIN
```

Figure D.11: Permanent File Report Control Card – SLP File \$014

```
CONTROL  FILERPT
PFROPT
SLPRPT  0
END
FIN
```

Figure D.12: Permanent File Report Control Card – SLP File \$078

```
CONTROL  FILERPT
PFROPT
ATMOSRPT 2  1  1
END
FIN
```

Figure D.13: Permanent File Report Control Card – Jacchia-Roberts Atmosphere File \$075

```
CONTROL  FILERPT
PFROPT
ATMOSRPT 3  1  1
END
FIN
```

Figure D.14: Permanent File Report Control Card – MSISE-90 Atmosphere File \$076

Appendix E

Macysma Program Additions and Modifications

In order to search for closed-form, second order, equations of motion in J_2 for use with DSST, a symbolic algebra language called Macysma was employed to help with the intense algebraic manipulations. Chapter 5 explains the problem in detail, as well as a description of the Macysma program blocks used to attack the problem. These program blocks, with the exception of the Zeis utilities which are listed in reference [32], are listed below. As opposed to the majority of code, utilities, and modifications presented in this thesis which might require substantial support such as a Fortran compiler, and NT-GTDS code, the blocks represent simple programs which can be run by any user with Macysma. The code is presented with no explanation, please refer to Chapter 5 for any additional documentation. For the reader with access to \$E53-BOB, all Macysma files can be found in the BOB\D:\EVO-HEO\MACSYMA\J22\PROGRAMS directory.

E.1 BPART2(I)

```
BPART2 (I) :=BLOCK(

    /* Version of 28 March 1997 - MACSYMA BLOCK          */
    /* THIS BLOCK COMPUTES THE PARTIAL DERIVATIVES OF   */
    /* A/R WITH RESPECT TO THE EQUINOCTIAL ELEMENTS     */
    /* (A, H, K, P, Q, LAMBDA)                          */
    /*CALLING SEQUENCE                                  */
    BPART2 (I)
    /*LOCAL VARIABLES                                   */
    I      DIFFERENTIATION VARIABLE (EQUINOCTIAL)
    /*AUXILIARY VARIABLE                                */
    X      INVERSE OF THE SQUARE ROOT OF (UNITY MINUS
```

```

                                THE SQUARE OF THE ECCENTRICITY)
L      TRUE LONGITUDE
B      DUMMY VARIABLE FOR A/R
BETA   1 / (1+X)
F      ECCENTRIC LONGITUDE      */

/*METHOD
      EXPLICIT FORMULAS.      */

/*PROGRAMMER
      J. FISCHER, USAF/MIT (1997)      */

/*RESTRICTIONS
      I BE AN INTEGER BETWEEN 1 AND 6 TO REPRESENT THE
      EQUINOCTIAL ELEMENTS (A,H,K,P,Q,LAMBDA)      */

/* PARTIAL WITH RESPECT TO A      */

IF I=1 THEN ( RETURN(0)),

/* PARTIAL WITH RESPECT TO H      */

IF I=2 THEN ( RETURN(- \b^2*((COS(\f)*(-X)*BETA-BETA^2*X*H*
      (\f-LAMBDA))* (K*SIN(\l)-H*COS(\l))+BETA*COS(\l)*
      (\f-LAMBDA)-SIN(\l))))),

/* PARTIAL WITH RESPECT TO K      */

IF I=3 THEN ( RETURN(- \b^2*((-X*COS(\f)-BETA*SIN(\f)-BETA^2*X*
      K*(\f-LAMBDA))* (K*SIN(\l)-H*COS(\l))-BETA*SIN(\l)*
      (\f-LAMBDA)-COS(\l))))),

/* PARTIAL WITH RESPECT TO P      */

IF I=4 THEN ( RETURN(0)),

/* PARTIAL WITH RESPECT TO Q      */

IF I=5 THEN ( RETURN(0)),

/* PARTIAL WITH RESPECT TO LAMBDA      */

IF I=6 THEN ( RETURN(-X* \b^2*(K*SIN(\l)-H*COS(\l))))$

```

E.2 FFUNCT(I)

```
FFUNCT(I) :=BLOCK([J],

    /*Version of 28 March 1997
      MACSYMA BLOCK */

    /*THIS BLOCK COMPUTES THE FUNCTIONS F USED IN CALCULATING
      THE SECOND ORDER CONTRIBUTIONS OF J2 */

    /*CALLING SEQUENCE
      FFUNCT(I) */

    /*BLOCKS CALLED
      UPARTIAL[I]
      POISSON2(I,J,MAXE) */

    /*LOCAL VARIABLES
      I      FUNCTION NUMBER
      J      SUMMATION INDEX */

    /*AUXILIARY VARIABLE
      C      SQUARE OF THE TANGENT OF HALF OF THE
             INCLINATION
      L      TRUE LONGITUDE
      B      DUMMY VARIABLE FOR A/R */

    /*METHOD
      SUMMATION OF EXPLICIT FORMULAS. */

    /*PROGRAMMER
      J. FISCHER, USAF/MIT (1997) */

    /*RESTRICTIONS
      I BE AN INTEGER BETWEEN 1 AND 6 TO REPRESENT THE
      EQUINOCTIAL ELEMENTS (A,H,K,P,Q,LAMBDA) */

    /* SUM OVER ALL ELEMENTS FOR GIVEN INDEX */

    RETURN(-SUM(POISSON2(I,J,2)*UPARTIAL(J),J,1,6)))$
```

E.3 FFUNCT2(I)

```
FFUNCT2(I) :=BLOCK([D1,D2,D3,D4,D5,PANDQ,LAMHK,LAMU,KANDH],

    /*Version of 28 March 1997
      MACSYMA BLOCK */

    /*THIS BLOCK COMPUTES THE FUNCTIONS F USED IN CALCULATING
      THE SECOND ORDER CONTRIBUTIONS OF J2 WITH D FUNCTION */
```

```

/*CALLING SEQUENCE
      FFUNCT2 (I)
*/

/*BLOCKS CALLED
      UDERIV (I)
      POISSON2 (I, J, MAXE)
*/

/*LOCAL VARIABLES
      I      FUNCTION NUMBER
      D1, D2,
      D3, D4,
      D5, D6  VARIABLES DEFINED BELOW FOR COMPUTATIONAL
              EFFICIENCY
      PANDQ  P TIMES PARTIAL OF U WRT P PLUS Q TIMES
              PARTIAL OF U WRT Q
      HANDK  SAME AS PANDQ ONLY WITH H AND K
      LAMHK  SAME AS PANDQ ONLY WITH H AND K PLUS PARTIAL
              OF U WRT LAMBDA
      LAMU   PARTIAL OF U WRT LAMBDA
*/

/*AUXILIARY VARIABLE
      X      INVERSE OF THE SQUARE ROOT OF (UNITY MINUS
              THE SQUARE OF THE ECCENTRICITY)
      BETA   1 / (1+X)
      F      ECCENTRIC LONGITUDE
      C      SQUARE OF THE TANGENT OF HALF OF THE
              INCLINATION
      L      TRUE LONGITUDE
      B      DUMMY VARIABLE FOR A/R
*/

/*METHOD
      SUMMATION OF EXPLICIT FORMULAS USING PREDEFINED
      VARIABLES.
*/

/*PROGRAMMER
      J. FISCHER, USAF/MIT (1997)
*/

/*RESTRICTIONS
      I BE AN INTEGER BETWEEN 1 AND 6 TO REPRESENT THE
      EQUINOCTIAL ELEMENTS (A, H, K, P, Q, LAMBDA)
*/

/* DEFINE D VARIABLES
*/

D1:2/(A*N),
D2:1/(A^2*N*(X+1)),
D3:((C+1)*X)/(2*A^2*N),
D4:1/(A^2*N*X),
D5:((C+1)^2*X)/(4*A^2*N),

/* DEFINE COMMONLY USED VALUES
*/

PANDQ:P*UDERIV2(4)+Q*UDERIV2(5),
HANDK:H*UDERIV2(2)+K*UDERIV2(3),
LAMU:UDERIV2(6),
LAMHK:LAMU-H*UDERIV2(3)+K*UDERIV2(2),

```

```

/* DEFINE F FUNCTIONS */

IF I=1 THEN (RETURN(-D1*LAMU)),

IF I=2 THEN (RETURN(D2*H*LAMU-D3*K*PANDQ-D4*UDERIV2(3))),

IF I=3 THEN (RETURN(D2*K*LAMU+D3*H*PANDQ+D4*UDERIV2(2))),

IF I=4 THEN (RETURN(D3*P*LAMHK-D5*UDERIV2(5))),

IF I=5 THEN (RETURN(D3*Q*LAMHK+D5*UDERIV2(4))),

IF I=6 THEN (RETURN(-D3*PANDQ-D2*HANDK+D1*UDERIV2(1)))$

```

E.4 FLOAD

```

load("c:\\macsyms\\BPART2.mac")$
load("c:\\macsyms\\LPART2.mac")$
load("c:\\macsyms\\UPART2.mac")$
load("c:\\macsyms\\UDERIV2.mac")$
load("c:\\macsyms\\FFUNCT2.mac")$

```

E.5 F[L, M, P](I)

```

F[L,M,P](I):=BLOCK([FF],

/*Version of 16 February 1997
MACSYMA BLOCK */

/*REFERENCE
AIAA PAPER 76-839
ZEIS, E., "A COMPUTERIZED ALGEBRAIC UTILITY FOR THE
CONSTRUCTION OF NONSINGULAR SATELLITE
THEORIES", MIT SM THESIS, 1978 */

/*
THIS TEST BLOCK COMPUTES THE INCLINATION FUNCTIONS
F[L,M,P](I) DEFINED BY KAULA */

/*CALLING SEQUENCE
F[L,M,P](I) */

/*BLOCKS CALLED

```

```

VFUNCT1
SFUNCT1 */

/*LOCAL VARIABLES
FF INTERMEDIATE VALUE OF INCLINATION
FUNCTION */

/*AUXILIARY VARIABLE
C SQUARE OF THE TANGENT OF HALF OF THE
INCLINATION */

/*METHOD
EXPLICIT FORMULAS BASED ON VFUNCT1 AND SFUNCT1 */

/*PROGRAMMER
E. ZEIS, MIT (1978)
P. CEFOLA, DRAPER/MIT (1997) */

/*RESTRICTIONS
TBS */

FF:VFUNCT1[L,L-2*P,M]*SFUNCT1[L,M,L-2*P](0,Q),

/* THE AUXILIARY VARIABLE C AND THE EQUINOCTIAL VARIABLE
Q ARE REPLACED BY THEIR EXPRESSIONS IN TERMS OF I */

FF:FACTOR(SUBST([C=(1-COS(I))/(1+COS(I)),Q=(1-
COS(I))/SIN(I)],FF)),

IF INTEGERP((L-M)/2)=FALSE THEN FF:%I*FF,

/* APPLICATION OF TRIGNOMETRIC RULE DEFINED PREVIOUSLY
AND EXPANSION OF THE RESULT */

FF:EXPAND(LETSIMP(FF)),

/* COS(I) IS SUBSTITUTED FOR SQRT(1-SIN(I)^2) IN THE
FINAL RESULT */

RETURN(SUBST(COS(I),SQRT(1-SIN(I)^2),FF))$

```

E.6 G[L, P, Q, MAXE](E)

```

G[L,P,Q,MAXE](E):=BLOCK(

/*Version of 21 February 1997
MACSYMA BLOCK */

/*REFERENCE
AIAA PAPER 76-839
ZEIS, E., "A COMPUTERIZED ALGEBRAIC UTILITY FOR THE
CONSTRUCTION OF NONSINGULAR SATELLITE
THEORIES", MIT SM THESIS, 1978 */

```



```

/*
  THIS TEST BLOCK COMPUTES THE ECCENTRICITY FUNCTIONS
  DEFINED BY KAULA
*/

/*CALLING SEQUENCE
      G[L,P,Q,MAXE] (E)
*/

/*BLOCKS CALLED
      HANSEN1
      HANSEN2
*/

/*METHOD
      EXPLICIT FORMULAS
*/

/*PROGRAMMER
      E. ZEIS, MIT (1978)
      P. CEFOLA, DRAPER/MIT (1997)
*/

/*RESTRICTIONS
      TBS
*/

IF MAXE>=0 THEN RETURN(EXPAND(HANSEN1[L-2*P+Q,-L-1,L-2*P,MAXE](0,E))),
IF (L-2*P+Q)=0 THEN RETURN(EXPAND(SUBST(1/SQRT(1-E^2),X,
      HANSEN2[-L-1,L-2*P,-1](0,E) )))

ELSE RETURN(ERROR)$

```

E.7 LPART2(I)

```

LPART2(I):=BLOCK(

  /* Version of 28 March 1997 - MACSYMA BLOCK
  */

  /* THIS BLOCK COMPUTES THE PARTIAL DERIVATIVES OF THE
  TRUE LONGITUDE WITH RESPECT TO THE EQUINOCTIAL ELEMENTS
  (A,H,K,P,Q,LAMBDA)
  */

  /*CALLING SEQUENCE
      LPART2(I)
  */

  /*LOCAL VARIABLES
      I      DIFFERENTIATION VARIABLE (EQUINOCTIAL)
  */

  /*AUXILIARY VARIABLE
      X      INVERSE OF THE SQUARE ROOT OF (UNITY MINUS
              THE SQUARE OF THE ECCENTRICITY)
      L      TRUE LONGITUDE
      B      DUMMY VARIABLE FOR A/R
      BETA   1 / (1+X)
  */

```

```

          F          ECCENTRIC LONGITUDE          */

/*METHOD
          EXPLICIT FORMULAS.          */

/*PROGRAMMER
          J. FISCHER, USAF/MIT (1997)          */

/*RESTRICTIONS
          I BE AN INTEGER BETWEEN 1 AND 6 TO REPRESENT THE
          EQUINOCTIAL ELEMENTS (A,H,K,P,Q,LAMBDA)          */

/* PARTIAL WITH RESPECT TO A          */

IF I=1 THEN ( RETURN(0) ),

/* PARTIAL WITH RESPECT TO H          */

IF I=2 THEN ( RETURN(\b*(-COS(\f)*((X^(-1)*BETA*\b)+BETA)+(X*H*
          BETA^2-X^(-1)*H*BETA^2*\b-BETA*SIN(\l))*
          (\f-LAMBDA)-COS(\l))),

/* PARTIAL WITH RESPECT TO K          */

IF I=3 THEN ( RETURN(\b*(SIN(\f)*((X^(-1)*BETA*\b)+BETA)+(-X*K*
          BETA^2+X^(-1)*K*BETA^2*\b+BETA*COS(\l))*
          (\f-LAMBDA)+SIN(\l))),

/* PARTIAL WITH RESPECT TO P          */

IF I=4 THEN ( RETURN(0) ),

/* PARTIAL WITH RESPECT TO Q          */

IF I=5 THEN ( RETURN(0) ),

/* PARTIAL WITH RESPECT TO LAMBDA          */

IF I=6 THEN ( RETURN((1/X)* \b^2)))$

```

E.8 POISSON2(I,J,MAXE)

```

POISSON2(I,J,MAXE):=BLOCK([XP,HP,KP],

/*Version of 22 February 1997
          MACSYMA BLOCK          */

/*REFERENCE
          CSC/TR-77/6010

```

AIAA PAPER 76-839
 ZEIS, E., "A COMPUTERIZED ALGEBRAIC UTILITY FOR THE
 CONSTRUCTION OF NONSINGULAR SATELLITE
 THEORIES", MIT SM THESIS, 1978 */

/*
 THIS BLOCK COMPUTES THE POISSON BRACKETS (I,J) OF THE
 EQUINOCTIAL ELEMENTS WITH AN EXPANSION TO ORDER MAXE
 IN TERMS OF THE ECCENTRICITY */

/*CALLING SEQUENCE
 POISSON2(I,J,MAXE) */

/*BLOCKS CALLED
 POISSON2(I,J,MAXE) */

/*LOCAL VARIABLES
 XP DUMMY NAME FOR THE AUXILIARY VARIABLE X
 HP DUMMY NAME FOR THE EQUINOCTIAL ELEMENT H
 KP DUMMY NAME FOR THE EQUINOCTIAL ELEMENT K */

/*AUXILIARY VARIABLE
 C SQUARE OF THE TANGENT OF HALF OF THE
 INCLINATION
 X INVERSE OF THE SQUARE ROOT OF (UNITY MINUS
 THE SQUARE OF THE ECCENTRICITY)
 N MEAN MOTION */

/*METHOD
 EXPLICIT FORMULAS.
 THE ANTI-SYMMETRY OF THE POISSON BRACKETS IS
 UTILIZED */

/*PROGRAMMER
 E. ZEIS, MIT (1978)
 P. CEFOLA, DRAPER/MIT (1997) */

/*MODIFIED
 J. FISCHER, USAF/MIT (1997) */

/*RESTRICTIONS
 MAXE MUST BE AN INTEGER GREATER THAN OR EQUAL TO
 ZERO.
 I AND J MUST BE INTEGERS BETWEEN 1 AND 6 TO
 REPRESENT THE EQUINOCTIAL ELEMENTS (A,H,K,P,Q,
 LAMBDA) */

/*THE AUXILIARY VARIABLE $X=1/\sqrt{1-H^2-K^2}$ IS
 REPLACED BY 1 IF $MAXE \leq 1$ AND IS KEPT UNDER ITS CLOSED
 FORM IF $MAXE > 1$ */

IF MAXE=0 THEN (XP:1,HP:0,KP:0)
 ELSE (HP:H,
 KP:K,

```

                IF MAXE=1 THEN XP:1
                    ELSE XP:X),

/* DEFINITION OF THE POISSON BRACKETS OF THE FORM (A,J) */
IF I=1 THEN ( IF J=6 THEN RETURN(-2/(N*A))
                ELSE RETURN(0)),

/* DEFINITION OF THE POISSON BRACKETS OF THE FORM (H,J) */
IF I=2 THEN ( IF J=3 THEN RETURN(-1/(N*A^2*XP))
                ELSE IF J=4 THEN RETURN(-KP*P*XP*(1+C)/(2*N*A^2))
                ELSE IF J=5 THEN RETURN(-KP*Q*XP*(1+C)/(2*N*A^2))
                ELSE IF J=6 THEN RETURN(HP/(N*A^2*(XP+1)))
                ELSE RETURN(0) ),

/* IN THE FOLLOWING PART WE WILL TAKE ADVANTAGE OF THE FACT
   THAT (I,J) = -(J,I)
                                                                    */

/* DEFINITION OF THE POISSON BRACKETS OF THE FORM (K,J) */
IF I=3 THEN ( IF J=4 THEN RETURN(HP*P*XP*(1+C)/(2*N*A^2))
                ELSE IF J=5 THEN RETURN(HP*Q*XP*(1+C)/(2*N*A^2))
                ELSE IF J=6 THEN RETURN(KP/(N*A^2*(XP+1)))
                ELSE IF J=2 THEN RETURN(-POISSON2(J,I,MAXE))
                ELSE RETURN(0) ),

/* DEFINITION OF THE POISSON BRACKETS OF THE FORM (P,J) */
IF I=4 THEN ( IF J=5 THEN RETURN(-XP*(1+C)^2/(4*N*A^2))
                ELSE IF J=6 THEN RETURN(P*XP*(1+C)/(2*N*A^2))
                ELSE IF J=2
                    OR J=3 THEN RETURN(-POISSON2(J,I,MAXE))
                ELSE RETURN(0) ),

/* DEFINITION OF THE POISSON BRACKETS OF THE FORM (Q,J) */
IF I=5 THEN ( IF J=6 THEN RETURN(Q*XP*(1+C)/(2*N*A^2))
                ELSE IF J=2
                    OR J=3
                    OR J=4 THEN RETURN(-POISSON2(J,I,MAXE))
                ELSE RETURN(0) ),

/* DEFINITION OF THE POISSON BRACKETS OF THE FORM (L,J) */
IF I=6 THEN ( IF J=6 THEN RETURN( 0 )
                ELSE RETURN(-POISSON2(J,I,MAXE))),

/* DEFINE DERIVATIVE OF N
                                                                    */
GRADEF(N,A,(-3/2)*SQRT(MU/A^5)))$

```

E.9 UDERIV2(I)

```
UDERIV2(I):=BLOCK(  
  
    /* Version of 28 March 1997 - MACSYMA BLOCK */  
  
    /* THIS BLOCK COMPUTES THE PARTIAL DERIVATIVES OF THE  
    GRAVITATIONAL POTENTIAL FOR THE ZONAL HARMONIC J2 WITH  
    RESPECT TO THE EQUINOCTIAL ELEMENTS  
    (A,H,K,P,Q,LAMBDA) */  
  
    /*CALLING SEQUENCE  
    UPARTIAL(I) */  
  
    /*BLOCKS CALLED  
    UPART2(I)  
    LPART2(I)  
    BPART2(I) */  
  
    /*LOCAL VARIABLES  
    I DIFFERENTIATION VARIABLE (EQUINOCTIAL) */  
  
    /*AUXILIARY VARIABLE  
    X INVERSE OF THE SQUARE ROOT OF (UNITY MINUS  
    THE SQUARE OF THE ECCENTRICITY)  
    BETA 1 / (1+X)  
    F ECCENTRIC LONGITUDE  
    C SQUARE OF THE TANGENT OF HALF OF THE  
    INCLINATION  
    L TRUE LONGITUDE  
    B DUMMY VARIABLE FOR A/R */  
  
    /*METHOD  
    EXPLICIT FORMULA. */  
  
    /*PROGRAMMER  
    J. FISCHER, USAF/MIT (1997) */  
  
    /*RESTRICTIONS  
    I BE AN INTEGER BETWEEN 1 AND 6 TO REPRESENT THE  
    EQUINOCTIAL ELEMENTS (A,H,K,P,Q,LAMBDA) */  
  
    RETURN(UPART2(I)+UPART2(7)*LPART2(I)+UPART2(8)*BPART2(I))$
```

E.10 UPART2(I)

```
UPART2(I):=BLOCK(  
  
    /* Version of 28 March 1997 - MACSYMA BLOCK */  
  
    /* THIS BLOCK COMPUTES THE PARTIAL DERIVATIVES OF THE  
    GRAVITATIONAL POTENTIAL FOR THE ZONAL HARMONIC J2 WITH
```

```

RESPECT TO THE EQUINOCTIAL ELEMENTS
(A,H,K,P,Q,LAMBDA) */

/*CALLING SEQUENCE
UPARTIAL(I) */

/*LOCAL VARIABLES
I DIFFERENTIATION VARIABLE (EQUINOCTIAL) */

/*AUXILIARY VARIABLE
C SQUARE OF THE TANGENT OF HALF OF THE
INCLINATION
L TRUE LONGITUDE
B DUMMY VARIABLE FOR A/R */

/*METHOD
EXPLICIT FORMULAS.
*/

/*PROGRAMMER
J. FISCHER, USAF/MIT (1997) */

/*RESTRICTIONS
I BE AN INTEGER BETWEEN 1 AND 8 TO REPRESENT THE
EQUINOCTIAL ELEMENTS (A,H,K,P,Q,LAMBDA) AND THE
PARTIALS WITH RESPECT TO TRUE LONGITUDE (L) AND
(A/R) */

/* PARTIAL WITH RESPECT TO A */

IF I=1 THEN ( RETURN(((3 * \j[2] * (\r[e])^2 * mu * \b^3 *
((3 * ((p^2 - q^2) * cos(2 * \l) - 2 * p * q *
sin(2 * \l)))/(c + 1)^2)) - ((c^2 - 4 * c + 1)/
(2 * (c + 1)^2)))/(a^4))),

/* PARTIAL WITH RESPECT TO H */

IF I=2 THEN ( RETURN(0)),

/* PARTIAL WITH RESPECT TO K */

IF I=3 THEN ( RETURN(0)),

/* PARTIAL WITH RESPECT TO P */

IF I=4 THEN ( RETURN(- ((\j[2] * (\r[e])^2 * mu * \b^3 *
( - ((12 * p * ((p^2 - q^2) * cos(2 * \l) - 2 *
p * q * sin(2 * \l)))/(c + 1)^3)) + ((3 * (2 *
p * cos(2 * \l) - 2 * q * sin(2 * \l)))/
((c + 1)^2)) - ((4 * c * p - 8 * p)/
(2 * (c + 1)^2)) + ((2 * (c^2 - 4 * c + 1) * p)/
((c + 1)^3)))/(a^3))),

```

```

/* PARTIAL WITH RESPECT TO Q */
IF I=5 THEN ( RETURN( - ((\j[2] * (\r[e])^2 * mu * \b^3 *
( - ((12 * q * ((p^2 - q^2) * cos(2 * \l) - 2 *
p * q * sin(2 * \l)))/((c + 1)^3)) + ((3 * ( - 2 *
p * sin(2 * \l) - 2 * q * cos(2 * \l)))/
((c + 1)^2)) - ((4 * c * q - 8 * q)/
(2 * (c + 1)^2)) + ((2 * (c^2 - 4 * c + 1) * q)/
((c + 1)^3)))/((a^3))),

/* PARTIAL WITH RESPECT TO LAMBDA */
IF I=6 THEN ( RETURN(0)),

/* PARTIAL WITH RESPECT TO L */
IF I=7 THEN ( RETURN( - ((3 * \j[2] * (\r[e])^2 * mu * \b^3 *
( - 2 * (p^2 - q^2) * sin(2 * \l) - 4 * p * q *
cos(2 * \l)))/((a^3 * (c + 1)^2))))),

/* PARTIAL WITH RESPECT TO A/R */
IF I=8 THEN ( RETURN( - ((3 * \j[2] * (\r[e])^2 * mu * \b^2 *
(((3 * ((p^2 - q^2) * cos(2 * \l) - 2 * p * q *
sin(2 * \l)))/((c + 1)^2)) - ((c^2 - 4 * c + 1)/
(2 * (c + 1)^2)))))/((a^3))))))$

```

E.11 UPARTIAL(I)

```

UPARTIAL(I):=BLOCK(

/* Version of 28 March 1997 - MACSYMA BLOCK */

/* THIS BLOCK COMPUTES THE PARTIAL DERIVATIVES OF THE
GRAVITATIONAL POTENTIAL FOR THE ZONAL HARMONIC J2 WITH
RESPECT TO THE EQUINOCTIAL ELEMENTS
(A, H, K, P, Q, LAMBDA) */

/*CALLING SEQUENCE
UPARTIAL(I) */

/*LOCAL VARIABLES
I DIFFERENTIATION VARIABLE (EQUINOCTIAL) */

/*AUXILIARY VARIABLE
C SQUARE OF THE TANGENT OF HALF OF THE
INCLINATION
L TRUE LONGITUDE
B DUMMY VARIABLE FOR A/R */

/*METHOD
EXPLICIT FORMULAS. */

```

```

/*PROGRAMMER
      J. FISCHER, USAF/MIT (1997)
*/

/*RESTRICTIONS
      I BE AN INTEGER BETWEEN 1 AND 6 TO REPRESENT THE
      EQUINOCTIAL ELEMENTS (A,H,K,P,Q,LAMBDA)
*/

/* PARTIAL WITH RESPECT TO A
*/
IF I=1 THEN ( RETURN(\u^(DA))),

/* PARTIAL WITH RESPECT TO H
*/
IF I=2 THEN ( RETURN(\u^(DH))),

/* PARTIAL WITH RESPECT TO K
*/
IF I=3 THEN ( RETURN(\u^(DK))),

/* PARTIAL WITH RESPECT TO P
*/
IF I=4 THEN ( RETURN(\u^(DP))),

/* PARTIAL WITH RESPECT TO Q
*/
IF I=5 THEN ( RETURN(\u^(DQ))),

/* PARTIAL WITH RESPECT TO LAMBDA
*/
IF I=6 THEN ( RETURN(\u^(D(LAMBDA))))$

```

E.12 UZONCF[n](a, p, q, B, L)

```

UZONCF[N](A,P,Q,B,L):=BLOCK([USTAR],

/*Version of 4 March 1997
      MACSYMA BLOCK
*/

/*REFERENCE
      AIAA PAPER 76-839
      ZEIS, E., "A COMPUTERIZED ALGEBRAIC UTILITY FOR THE
      CONSTRUCTION OF NONSINGULAR SATELLITE
      THEORIES", MIT SM THESIS, 1978
*/

/*
      THIS BLOCK COMPUTES THE FUNCTIONS U WHICH REPRESENTS
      N,0
      THE CLOSED FORM REPRESENTATION OF THE GRAVITATIONAL
      POTENTIAL EXPANDED IN TERMS OF ZONAL HARMONICS
*/

```



```

/*CALLING SEQUENCE
      UZONCF[N] (A, P, Q, B, L)
*/

/*BLOCKS CALLED
      VFUNCT1[N, R, M]
      SFUNCT1[N, M, S] (P, Q)
*/

/*LOCAL VARIABLES
      USTAR REAL AND IMAGINARY PARTS OF POTENTIAL
      R[E] RADIUS OF THE EARTH
      J[N] ZONAL HARMONIC DEGREE N
      L TRUE LONGITUDE
*/

/*AUXILIARY VARIABLE
      B THE RATIO OF A/R, SEMI-MAJOR AXIS TO
        POSITION
*/

/*METHOD
      EXPLICIT FORMULAS BASED ON VFUNCT1 AND SFUNCT1
*/

/*PROGRAMMER
      J. FISCHER, DRAPER/USAF (1997)
*/

/*RESTRICTIONS
      N MUST BE INTEGER
      N MUST BE GREATER THAN 0
*/

/* CHECK VALUE OF N
*/
IF N<0 THEN RETURN(ERROR) ,

/* FIND USTAR
*/
USTAR: ((-MU/A) * ((R[E]/A)^2) * (B^3) * (\j[N])) *SUM(
      VFUNCT1[N, K, 0] *SFUNCT1[N, 0, K] (P, Q) *EXP(%I*K*L) , K, -N, N) ,

/* DEFINE THE DERIVATIVES OF B
*/
GRADEF(B, A, A) ,

/* RETURN REAL PORTION OF USTAR
*/
RETURN(REALPART(USTAR)) $

```

E.13 UZONCF2(a, p, q, B, L)

```

UZONCF2 (A, P, Q, B, L) :=BLOCK([USTAR] ,

/*Version of 4 March 1997
      MACSYMA BLOCK
*/

/*REFERENCE
      AIAA PAPER 76-839

```

ZEIS, E., "A COMPUTERIZED ALGEBRAIC UTILITY FOR THE
CONSTRUCTION OF NONSINGULAR SATELLITE
THEORIES", MIT SM THESIS, 1978 */

```

/*
  THIS BLOCK COMPUTES THE FUNCTIONS U WHICH REPRESENTS
                                     2,0
  THE CLOSED FORM REPRESENTATION OF THE GRAVITATIONAL
  POTENTIAL EXPANDED IN TERMS OF ZONAL HARMONICS */

/*CALLING SEQUENCE
      UZONCF[N] (A, P, Q, B, L) */

/*BLOCKS CALLED
      VFUNCT1[N, R, M]
      SFUNCT1[N, M, S] (P, Q) */

/*LOCAL VARIABLES
      USTAR REAL AND IMAGINARY PARTS OF POTENTIAL
      R[E] RADIUS OF THE EARTH
      J2 J2 ZONAL HARMONIC DEGREE
      L TRUE LONGITUDE */

/*AUXILIARY VARIABLE
      B THE RATIO OF A/R, SEMI-MAJOR AXIS TO
        POSITION */

/*METHOD
      EXPLICIT FORMULAS BASED ON VFUNCT1 AND SFUNCT1 */

/*PROGRAMMER
      J. FISCHER, DRAPER/USAF (1997) */

/* FIND USTAR */

USTAR: ((-MU/A) * (R[E]/A)^2) * (B^3) * J[2] * SUM(
  VFUNCT1[2, K, 0] * SFUNCT1[2, 0, K] (P, Q) * EXP(%I*K*L), K, -2, 2),

/* RETURN REAL PORTION OF USTAR */

RETURN(REALPART(USTAR))$

```

E.14 ZEIS

```

load("c:\\macsyma\\legz1.mac")$
load("c:\\macsyma\\vfunct1.mac")$
load("c:\\macsyma\\jacob1.mac")$
load("c:\\macsyma\\sfunct1.mac")$
load("c:\\macsyma\\ifunct1.mac")$
load("c:\\macsyma\\newcomb1.mac")$
load("c:\\macsyma\\hansen1.mac")$
load("c:\\macsyma\\hansen2.mac")$

```

```
load("c:\\macsyms\\gfunct1.mac")$
load("c:\\macsyms\\poisson.mac")$
load("c:\\macsyms\\poisson2.mac")$
load("c:\\macsyms\\ustar.mac")$
load("c:\\macsyms\\slz.mac")$
load("c:\\macsyms\\vopeq.mac")$
load("c:\\macsyms\\slzvop.mac")$
load("c:\\macsyms\\UZONCF.mac")$
load("c:\\macsyms\\UPARTIAL.mac")$
load("c:\\macsyms\\FFUNCT.mac")$
load("c:\\macsyms\\s1.mac")$
load("c:\\macsyms\\s2.mac")$
load("c:\\macsyms\\tesdel.mac")$
load("c:\\macsyms\\zsimp.mac")$
```

[This page intentionally left blank.]

References

- [1] Chobotov, V. *Orbital Debris Hazard Assessment and Mitigation Methods*. U.S. Space Command Statistics. Presentation given at the Aerospace Forum on Space Debris, Collision Avoidance, and Reentry Hazards. November 4-5, 1997. Proceedings available from William Ailor, Aerospace Corporation.
- [2] Vallado, David. *Fundamentals of Astrodynamics and Applications*. McGraw-Hill Space Technology Series, New York. 1997.
- [3] Gauss, Carl Friedrich. *Theory of the Motion of the Heavenly Bodies Moving About the Sun in Conic Sections*. A translation of *Theoria Motus* (1857) by Charles H. Davis. New York, Dover Publications, Inc. 1963.
- [4] *Goddard Trajectory Determination System (GTDS) Mathematical Theory*. NASA's Operational GTDS Mathematical Specification. Revision 1. Edited by Computer Sciences Corporation and NASA Goddard Space Flight Center. Contract NAS 5-31500. Task 213. July 1989.
- [5] Sabol, Christopher. *Application of Sun-Synchronous, Critically Inclined Orbits to Global Personal Communications Systems*. Master of Science Thesis. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. CSDL-T-938. January 1987.

- [6] Sabol, C. A., P. J. Cefola, and R.W. Metzinger, "Application of Sun-Synchronous, Critically Inclined Orbits to Global Personal Communications Systems", AAS/AIAA Space Flight Mechanics Conference, Albuquerque, NM: February, 1995, AAS 95-222. This paper was published in the Proceedings of the Conference, Volume 89, Part 2 of the Advances in the Astronautical Sciences: pp. 1459-1483.
- [7] Sabol, C.A., J. Draim, and P. J. Cefola, "Refinement of a Sun-Synchronous, Critically Inclined orbit for the ELLIPSO™ Personal Communications System", AAS/AIAA Astrodynamics Conference, Halifax, NS: August, 1995, AAS 95-340 This paper was published in the Proceedings of the Conference, Volume 90, Part 1 of the Advances in the Astronautical Sciences. It has also been published in the *Journal of the Astronautical Sciences* Volume 44, No. 4, pp. 467-489: October-December, 1996.
- [8] Deprit, A., S. Coffey, and E. Deprit, "Painting Phases to put Frozen Orbits in Context", AAS 91-427, presented at the AAS/AIAA Astrodynamics Conference, Durango, CO: August, 1991.
- [9] Larson, Wiley J. *et al.* *Space Mission Analysis and Design*. Torrance, California: Microcosm, Inc. 1992.
- [10] King-Hele, D. G., "Analysis of the Orbit of 1970-114F in its Last 20 Days", *Planetary Space Sciences*, Vol. 24, 1976, pp. 1-16.
- [11] Walters, Louis G. "Appendix 5: Spiral Decay." *Space Surveillance, Asteroids and Comets, and Space Debris, Volume I: Space Surveillance*. United States Air Force Scientific Advisory Board, SAB-TR-96-04. June, 1997.

- [12] Eisele, Anne. "Cause of Mars-96 Failure Is Still A Mystery." *Space News*, <http://www.spacenews.com/mars96.html>. 20 January, 1997.
- [13] Patera, Russel P., and William H. Ailor. "The Realities of Reentry Disposal," AAS 98-174. Presented at the AAS/AIAA Space Flight Mechanics Conference, Monterey, CA. February 9-11, 1998.
- [14] Tapley, Byron D. *et al.* "Precision Orbit Determination for TOPEX/POSEIDON". *Journal of Geophysical Research*. Vol. 99, No. C12, 24383-24404, December 15, 1994.
- [15] Johnson, Nicholas. *The Soviet Year in Space: 1988*. Teledyne Brown Engineering, Colorado Springs. January, 1989.
- [16] Wedel, E. *Langenscheidt's Pocket Russian Dictionary*. Langenscheidt KG. Berlin, Germany, 1969.
- [17] Sellers, Jerry J. *et al.* *Understanding Space: An Introduction to Astronautics*. New York: McGraw-Hill, Inc. 1994, p. 263.
- [18] King-Hele, D. G., "The Orbital Lifetimes of Molniya Satellites", *Journal of the British Interplanetary Society*, Vol. 28, 1975, pp. 783-796.
- [19] Fieger, Martin E. *An Evaluation of Semianalytical Satellite Theory Against Long Arcs of Real Data For Highly Eccentric Orbits*. Master of Science Thesis. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. CSDL-T-938. January 1987.

- [20] Fieger, M., R. Proulx, and P. J. Cefola, "An Evaluation of Semi-analytical Satellite Theory Against Long Arcs of Real Data for Highly Eccentric Orbits", AAS 87-444, presented at the AAS/AIAA Astrodynamics Conference, Kalispell, MT: August, 1987.
- [21] Brumberg, Eugene, "Analytical Perturbation Technique for Highly Eccentric Orbits Based on Elliptic Function Theory", *Dynamics and Astrometry of Natural and Artificial Celestial Bodies*, 1993, pp. 167-174.
- [22] Brumberg, Eugene, "Analytical Linear Perturbation Theory for Highly Eccentric Satellite Orbits", *Celestial Mechanics and Dynamical Astronomy*, Vol. 61, 1995, pp. 369-387.
- [23] Lidov, M., and Solov'ev A. "Atmospheric Perturbations On the Motion of a Satellite In a High-Eccentricity Orbit." Translated *from Kosmicheskie Issledovaniya*, Vol. 16, No. 6, pp. 806-821, November-December, 1978. Plenum Publishing Corporation, 1979.
- [24] Yurasov, V. "Universal Semianalytic Satellite Motion Propagation Method." Paper presented at the U.S.- Russian Space Surveillance Workshop in Poznan, Poland. July 4-6, 1996.
- [25] Andrewshchenko, V., et. al. "Reentry Time Determination Analysis for Cosmos-398 and FSW-1-5." Paper presented at the U.S.- Russian Space Surveillance Workshop in Poznan, Poland. July 4-6, 1996.
- [26] Johnson, Nicholas. *The Soviet Year in Space: 1985*. Teledyne Brown Engineering, Colorado Springs. January, 1986.

- [27] *The Satellite Encyclopedia*. <http://www.TELE-satellite.com/tse>. 13 March, 1998.
- [28] Alderete, S. Personal Correspondence. Kirtland AFB, PL/WSA. 25 September, 1997.
- [29] *Ninfiger Productions: Soviet Space Photo*.
<http://www.ninfiger.org/~sven/models/sovietsp/mlnya03.html>. 13 March, 1998.
- [30] Johnson, Nicholas, et al. *History of Soviet/Russian Satellite Fragmentations – Joint U.S.-Russian Investigation*. Printed by Kaman Sciences Corporation. October, 1995. Work done under NASA Contract NAS 9-19215.
- [31] *Mark Wade's Encyclopedia Astronautica*.
<http://solar.rtd.utk.edu:81/~mwade/spaceflt.html>. 13 March, 1998.
- [32] Zeis, Eric G. *A Computerized Algebraic Utility for the Construction of Nonsingular Satellite Theories*. Master of Science Thesis. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. September 1978.
- [33] *Webster's Ninth New Collegiate Dictionary*. Merriam-Webster Inc., Publishers. Springfield, MA, 1989.
- [34] Bate, R., D. Mueller, and J. White. *Fundamentals of Astrodynamics*. New York: Dover Publications, Inc. 1971.

- [35] Deprit, Andre, et al. *The Main Problem of Satellite Theory for Small Eccentricities*. Boeing Scientific Research Laboratories Document D1-82-0888. August, 1969. This paper was also published in *Celestial Mechanics: An International Journal of Space Dynamics*, Volume 2, No. 2, p. 166: November, 1970.
- [36] Roy, A. E. *Orbital Motion*. New York: Adam Hilger. 1991.
- [37] Moore, W., and P. Beaudet. "The Testing of Fixed-Step Numerical Integration Processes for the Cowell Method of Special Perturbations," Proceedings of the Conference on the Numerical Solution of Ordinary Differential Equations, Springer Lecture Notes in Mathematics, Vol. 362: 1973.
- [38] Velez, C., and P. Cefola, et al. "Calculation of Precision Satellite Orbits with Non-Singular Elements," Proceedings of the Conference on the Numerical Solution of Ordinary Differential Equations. Springer Lecture Notes in Mathematics, Vol. 362: 1973.
- [39] Battin, Richard H. *An Introduction to the Mathematics and Methods of Astrodynamics*. AIAA Education Series. New York: 1987.
- [40] Carter, Scott S. *Precision Orbit Determination From GPS Receiver Navigation Solutions*. Master of Science Thesis. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. CSDL-T-1260. June 1996.
- [41] Shah, Naresh H. *Automated Station-Keeping For Satellite Constellations*. Master of Science Thesis. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. CSDL-T-1288. June 1997.

- [42] McClain, Wayne D., *A Semianalytic Artificial Satellite Theory, Volume 1: Application of the Generalized Method of Averaging to the Artificial Satellite Problem*, 1992, available from Wayne McLain, Charles Stark Draper Laboratory, 555 Technology Square, Cambridge, MA 02139
- [43] R. Broucke, and P. Cefola, "On the Equinoctial Orbital Elements," *Celestial Mechanics*, Vol. 5, 1972, pp. 303-310.
- [44] Cefola, P. and R. Broucke, "On the Formulation of the Gravitational Potential in terms of the Equinoctial Variables," AIAA 75-9, AIAA Aerospace Sciences Meeting, Pasadena CA, January 1975.
- [45] Broucke, R., "On the Matrizant of the Two-Body Problem," *Astonomy and Astrophysics*, Vol. 6, pp. 173-182.
- [46] Danielson, D. A., et al. *Semianalytic Satellite Theory (SST): Mathematical Algorithms*. Naval Postgraduate School. Report Number NPS-MA-94-001. January 1994.
- [47] Poincaré, H. *Les Méthods Nouvelles de la Mécanique Celeste*. Three Volumes. Paris: Gauthier-Villars, 1899 (epuisée). English translation: *New Methods of Celestial Mechanics*, NASA TTF-450, 1967.
- [48] Bogoliubov, N., and Y. Mitropol'ski. *Asymptotic Mehtods in the Theory of Nonlinear Oscillations*. Gordon & Breach, New York: 1961.
- [49] Mitropol'ski, Y.. *Problems of the Asymptotic Theory of Non-stationary Vibrations*. English translation by the Israel Program for Scientific Translations, (Jerusalem, 1965). Daniel Davey and Company, Inc. New York: 1965.

- [50] Zill, D., and M. Cullen. *Advanced Engineering Mathematics*. PWS-Kent Publishing Company. Boston: 1992.
- [51] Anderson, J. *Introduction to Flight*. McGraw-Hill Inc., New York: 1989.
- [52] Vinh, N., A. Busemann, and R. Culp. *Hypersonic and Planetary Entry Flight Mechanics*. The University of Michigan Press, Ann Arbor: 1980.
- [53] Neuenfeldt, B. and W. Henderson. *A Survey of Uncontrolled Satellite Reentry and Impact Prediction*. Master of Science Thesis. Naval Postgraduate School, Monterey, California. September, 1993.
- [54] Barker, W., T. Eller, and L. Herder. "A New Approach in Treating the Ballistic Coefficient in the Differential Correction Fitting Program." AAS 89-374. (Presented at the AAS/AIAA Astrodynamics Conference, Stowe, VT. August 7-10, 1989). *Astrodynamics 1989*. Volume 71, Part I: Advances in the Astronautical Sciences. Univelt, Incorporated, San Diego. pp. 345-364.
- [55] Nazarenko, A. "Justification and Description of the Algorithm of Construction of the Atmosphere Density Variation Model From Orbital Data of Space Surveillance Systems (SSS), as well as the Algorithm of Forecasting of Variations." Stages 1-3. This report was executed pursuant to Purchase Order Number DL-C-493354 between C.S. Draper Laboratory and NUCLON. Completed 30 June, 1997.
- [56] Hallman, W. *Reentry Breakup Activities at Aerospace*. The Aerospace Corporation. Presentation given at the Aerospace Forum on Space Debris, Collision Avoidance, and Reentry Hazards. November 4-5, 1997.
- [57] *Satellite Toolkit, Version 4.0.3*. Professional Version 4.0 Documentation. 30 March, 1998.

- [58] Anselmo, L., C. Pardini, and A. Rossi. *Reentry Predictions for Cosmos 398, FSW-1 5, and TSS-1R*. CNUCE/CNR, Via S. Maria 36, 56126 Pisa, Italy. 6 May, 1997.
- [59] King-Hele, D. G. "Methods for Predicting Satellite Orbital Lifetimes," *Royal Aircraft Establishment*, TR 77111: 1977.
- [60] Klinkrad, H. "Salyut-7/Kosmos-1686 Reentry Prediction Activities at ESOC," (ESA-SP-345). *The Reentry of Salyut-7/Kosmos-1686*, Proceedings of the International Workshop ESOC, Darmstadt (D): 9 April, 1991. pp. 17-34.
- [61] Barker, R. H., Jr. *Estimation of Decayed Satellite Reentry Trajectories*. Ph.D. Dissertation, Air Force Institute of Technology. Wright Patterson AFB, OH: 1982.
- [62] Lefebvre, S.V.. *Analysis of Tracking and Impact Predictions*. Master's Thesis, Air Force Institute of Technology. Wright Patterson AFB, OH: December, 1991.
- [63] Strang, G. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press. Wellesley: 1986.
- [64] Drake, A. *Fundamentals of Applied Probability Theory*. McGraw-Hill, Inc. New York: 1967.
- [65] Taylor, Stephen P. *Semianalytical Satellite Theory and Sequential Estimation*. Master of Science Thesis. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. CSDL-T-757. September 1981.
- [66] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," Paper 59-IRD-11 presented at ASME Instruments and Regulators Conference, March 29-April 2, 1959 (also *Transactions of ASME, Series D, Journal of Basic Engineering*, Vol. 82, March 1960, pp. 35-45).

- [67] Kalman, R.E. and R.S. Bucy, "New Results in Linear Filtering and Prediction Theory," *Transactions of ASME, Series D, Journal of Basic Engineering*, Vol. 83, March 1961, pp. 95-108.
- [68] Lefferts, E.J., F.L. Markley, and M.D. Shuster. "Kalman Filtering for Spacecraft Attitude Estimation." AAS Paper 82-0070R, AIAA 20th Aerospace Sciences Meeting, Orlando, FL, Jan. 11-14, 1982.
- [69] du Plessis, R. *Poor Man's Explanation of Kalman Filtering or How I Stopped Worrying and Learned To Love Matrix Inversion*. Autonetics Division, Rockwell International: June, 1967.
- [70] Fitzgerald, R.J. "Divergence of the Kalman Filter." *IEEE Transactions on Automatic Control*. Vol. 16, No. 6, December, 1971: pp. 736-747.
- [71] *The Research and Development Goddard Trajectory Determination System (R&D GTDS) Filter Program Software Specifications and User's Guide*. Contract NAS 5-24300, Task Assignment 977. January, 1980.
- [72] Dowd. *Adaptive Estimation of Atmospheric Drag on Near Earth Satellites*, IASOM TR 77-4, University of Texas at Austin, December 1977.
- [73] Torroglosa, V. *Filtering Theory Applied to Orbit Determination*. Goddard Space Flight Center, Report X-582-73-379: December, 1973.
- [74] Wright, J. "Sequential Orbit Determination with Auto-Correlated Gravity Modeling Errors," *J. Guidance and Control*, Volume 4, Number 3, May 1981.
- [75] Hujsak, R. "Gravity Process Noise for Constellations of Satellites," AAS 94-175: 1994.

- [76] Myers. *Filtering Theory Methods and Applications to the Orbit Determination Problem for Near-Earth Satellites*, Ph.D. Dissertation, University of Texas at Austin, 1974.
- [77] Ingram, D. S., and B. D. Tapley. "Lunar Orbit Determination in the Presence of Unmodeled Accelerations," *Celestial Mechanics*. Vol. 9: 1974. pp. 191-211.
- [78] Tapley, B. D., and D. S. Ingram. "Orbit Determination in the Presence of Unmodeled Accelerations," *IEEE Transactions on Automatic Control*. Vol. 18, No. 4: August, 1973. pp. 369-373.
- [79] Tapley, B. D., and B. E. Schutz. *Estimation of Unmodeled Forces on a Lunar Satellite*. Department of Aerospace Engineering and Engineering Mechanics, University of Texas. Austin, Texas: 1974.
- [80] Jazwinski, A. H. "Adaptive Sequential Estimation with Applications," Fifth IFAC Symposium on Automatic Control in Space, Genoa, Italy: June, 1973. Work done under NASA contracts NAS 8-26973 and NAS 5-22011.
- [81] *Functional Requirements for the Lunar/Planetary Orbit Determination Subsystem of the Goddard Trajectory Determination System*. Goddard Space Flight Center: 1970.
- [82] *Goddard Trajectory Determination System (GTDS) User's Guide*. NASA's Operational GTDS User's Guide. Updated by Rick Metzinger, CSDL: 1995. Copy available through Paul Cefola at (617) 258-1787.

- [83] Neelon, J., P. Cefola, and R. Proulx. "Current Development of the Draper Semianalytical Satellite Theory Standalone Orbit Propagator Package." AAS 97-731. Presented at the AAS/AIAA Space Flight Mechanics Conference, Sun Valley, ID: August 4-7, 1997.
- [84] Carter, D., R. Metzinger, R. Proulx, and P. Cefola. "The Transition of GTDS to the UNIX Workstation Environment." Presentation given at the NASA GSFC Flight Mechanics/Estimation Theory Symposium, Greenbelt, MD: May 16-18, 1995.
- [85] Metzinger, Richard W. *Validation of the Workstation Version of R&D GTDS*. Charles Stark Draper Laboratory: February 24, 1993. Copy available through Dr. Paul Cefola. (617) 258-1787.
- [86] Fonte, Daniel J. *Implementing a 50x50 Gravity Field Model in an Orbit Determination System*. Master of Science Thesis. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. CSDL-T-1169. June 1993.
- [87] Wallace, Scott T. *Parallel Orbit Propagation and the Analysis of Satellite Constellations*. Master of Science Thesis. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. CSDL-T-1245. June 1995.
- [88] Kantsiper, Brian. *A Systematic Approach to Station-Keeping of Constellations of Satellites*. Doctor of Philosophy Thesis. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. CSDL-T-1298. March, 1998.
- [89] *Lahey Make Reference: Revision A*. Opus Software. Incline Village, NV: January, 1992.

- [90] Fonte, Daniel J., Jr. and Chris Sabol. *Optimal DSST Input Decks for Various Orbit Types: Final Report*. Phillips Laboratory, Space and Missiles Technology Directorate. PL-TR-95-1072: June 1995.
- [91] Cefola, Paul. "A Recursive Formulation for the Tesseral Disturbing Function in Equinoctial Variables." AAS Paper 76-839, AAS/AIAA Astrodynamics Specialist Conference. San Diego, California: August 18-20, 1976.
- [92] Cefola, Paul, and Wayne McClain. "A Recursive Formulation of the Short-Periodic Perturbations in Equinoctial Variables." AAS Paper 78-1383, AAS/AIAA Astrodynamics Specialist Conference. Palo Alto, California: August 7-9, 1978.
- [93] McClain, W., and M. Slutsky. "A Theory for the Short Periodic Motion of Artificial Satellites." AIAA preprint 80-1658, presented at the AIAA/AAS Astrodynamics Specialist Conference. Danvers, MA: August, 1980.
- [94] Slutsky, M., *Zonal Harmonic Short-Periodic Model*, Draper Laboratory, Division 15Z, IOC, PL-016-81-MS, November 30, 1981.
- [95] Green, Andrew J. *Orbit Determination and Prediction Processes For Low Altitude Satellites*. Doctor of Philosophy Thesis. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. CSDL-T-703. December 1979.
- [96] Kaniecki, Jean-Patrick R. *Short Periodic Variations in the First-Order Semianalytical Satellite Theory*. Master of Science Thesis. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. August, 1979.

- [97] Metris, G. *et. al.* “Long Period Variations of the Motion of a Satellite Due To Non-Resonant Tesserel Harmonics of a Gravity Potential.” *Celestial Mechanics and Dynamical Astronomy* 57: 1993. pp. 175-188.
- [98] Metris, G. “Mean Vlaues of Particular Functions in the Elliptic Motion.” *Celestial Mechanics and Dynamical Astronomy*, Vol. 52: 1001. pp. 1991.
- [99] Metris, G. and P. Exertier. “Semi-analytical Theory of the Mean Orbital Motion.” *Astronomy and Astrophysics*, Vol. 294: 1995. pp. 278-286.
- [100] Metris, G. “Théorie du Mouvement du Satellite Artificiel,” *Astronomie Fondamentale, Mécanique Céleste et Géodésie*: 1995.
- [101] Ahmed, Hussein-Salma. *A Semi-Analytic Theory for Satellite Orbit Prediction*. Doctor of Philosophy Thesis. Department of Aeronautics and Astronautics, The University of Texas at Austin: May, 1985.
- [102] Blitzer, Leon. *Handbook Of Orbital Perturbations*. University of Arizona: 1970.
- [103] Kozai, Y., “Analytical Orbital Theories for Satellites.” *Proceedings of the International Symposium on the Use of Artificial Satellites for Geodesy and Geodynamics*, Athens: May 16-21, 1973. pp. 237-262.
- [104] Karr, G. R. *A Study of Effects of the Gas-Surface Interaction on Spinning Convex Bodies With Application to Satellite Experiments*. Doctoral Thesis, Department of Aeronautical and Astronautical Engineering, University of Illinois., 1969.
- [105] Karr, G. R., R. E. Smith. “Influence of Satellite Aerodynamics on Atmospheric Density Determination.” *International Conference on Aerospace and Aeronautical Meteorology*, Washington, D.C.: May 22-26, 1972.

- [106] Chastain, S.H., C.C. Chao. "An Evaluation of the Jacchia 1971 and MSISE 1990 Atmospheric Models Using NASA ODERACS2 Data", AAS 97-685. Presented at the AAS/AIAA Space Flight Mechanics Conference, Sun Valley, ID: August 4-7, 1997.
- [107] Picone, M., and A. Hedin. Naval Research Laboratory. Personal correspondence. August 1996 – April 1998.
- [108] NASA's National Space Science Data Center (NSSDC) Home Page. Space Physics: Space Physics Models and Model Based Tools: Atmospheric Models: MSISE Model. <http://nssdc.gsfc.nasa.gov/space/model/atmos/atmos_index.html> .
- [109] NOAA NGDC Solar Terrestrial Physics Division Datasets FTP Site. Contact at (303) 497-6132, ftp@ngdc.noaa.gov.
<ftp://ftp.ngdc.noaa.gov/STP/GEOMAGNETIC_DATA/INDICES/KP_AP/>,
<ftp://ftp.ngdc.noaa.gov/STP/SOLAR_DATA/SOLAR_RADIO/FLUX/>
- [110] Schatten, K. National Science Foundation. Personal correspondence. June, 1997.
- [111] Hedin, A. E., "MSIS-86 Thermospheric Model", *Journal of Geophysical Research*, Vol. 92, No. A5, 1 May 1987, pp. 4649-4662.
- [112] Regan, F. J., S. M. Anandakrishnan. *Dynamics of Atmospheric Re-Entry*. American Institute of Aeronautics and Astronautics: New York, NY; 1993.
- [113] Karr, G. R. *Environmental Dynamics at Orbital Altitudes*. Alabama Univ., Huntsville School of Graduate Studies and Research. Report: NASA-CR-2765; M-193, December, 1976.
- [114] Gorenbukh, P.I., "A Correlation for Drag Coefficients of Bodies in Hypersonic Flow of Rarefied Gases," *Fluid Mechanics - Soviet Research*, Vol. 15, No. 4, July-Aug. 1986, pp. 56-64; translated from *Uchenyye Zapiski TsAGI*, 17, 2, pp. 99-105.

- [115] Karr, G.K., "Dual Falling Sphere Determination of Density and Transition Flow Parameter." Paper submitted to AIAA 12th Aerospace Sciences Meeting by G.R. Karr; Washington, D.C., 30 Jan. - 1 Feb., 1974.
- [116] Challinor, R. A. "Neutral-air Winds in the Ionospheric F-Region for an Asymmetric Global Pressure System", *Journal of Atmospheric Science*: 1969. pp. 1097-1106.
- [117] *Standard Agena Space Vehicle: Model SS01-B*. Lockheed Missiles & Space CompanyL Space Systems Division, LMSC-A397890A: 15 October, 1967.
- [118] *NASA Super Orbital Re-entry Test Program Planning Study: Volume 2 – Stimulation at Lifting Re-entry Trajectories With Delayed Separation*. Lockheed Missiles & Space CompanyL Space Systems Division, LMSC-325532-Vol. 2: 31 October, 1961. Work carried out under NASA Contract NAS 1-1180.
- [119] Hujsak, D. Logicon. Personal correspondence. August 1997 – April 1998.
- [120] Cefola, P. GTDS Range Gating for Observations Input via the Observation Card Data File. Draper Intralab Memorandum, E50-96-135: 25 March, 1996.
- [121] Jazwinski, A. *Stochastic Processes and Filtering Theory*. Academic Press. New York, NY: 1970.
- [122] Smith, Jim. *Incorporation of Earth Albedo Model into Draper Lab GTDS Orbit Determination System*. U.S. Air Force Academy, Cadet Summer Research Program Final Report: June, 1996.
- [123] Nazarenko, A., P. Cefola, and V. Yurasov "Estimating Atmosphere Density Variations to Improve LEO Orbit Prediction Accuracy." AAS 98-190. Presented at the AAS/AIAA Space Flight Mechanics Conference, Monterey, CA. February 9-11, 1998.

- [124] Lafontaine, Jean de. *Orbital Dynamics in a Stochastic Atmosphere and a Nonspherical Gravity Field*. Doctor of Philosophy Thesis. Institute for Aerospace Studies, University of Toronto. 1986.
- [125] Ellis, T. M. *A Structured Approach to Fortran 77 Programming*. Addison-Wesley Publishing Company. Reading, MA: 1982.
- [126] Etter, D. M. *Structured Fortran 77 For Scientists and Engineers: Fourth Edition*. The Benjamin/Cummings Publishing Company, Inc. Redwood City, CA: 1993.
- [127] Cefola, P. R&D GTDS Semianalytical Satellite Theory Input Processor. Draper Intralab Memorandum. ESD-92-582, SGI GTDS-92-001, Revision 1: 8 February, 1993.
- [128] Cefola, P., and D. Fonte. "Extension of the Naval Space Command Satellite Theory PPT2 to Include a General Tesseral M-Daily Model." AIAA 96-3606. Presented at the AIAA/AAS Astrodynamics Specialist Conference, San Diego, CA. July 29-31, 1996.
- [129] *Goddard Trajectory Determination System (GTDS) User's Guide*. NASA's Operational GTDS User's Guide. Revision 2, Update 8. Edited by Computer Sciences Corporation and NASA Goddard Space Flight Center. Contract NAS 5-31500. Task 52 617. September 1994.
- [130] Amelina, T., et al. "Comparison of Atmosphere Density Models" Paper presented at the U.S.- Russian Space Surveillance Workshop in Poznan, Poland. July 4-6, 1996.
- [131] Hedin, A. E., "Extension of the MSIS Thermosphere Model Into the Middle and Lower Atmosphere", *Journal of Geophysical Research*, Vol. 96, No. A2, 1 Feb 1991, pp. 1159-1172.

[132] Justus, C., et al. *The NASA/MSFC Global Reference Atmospheric Model – MOD 3 (With Spherical Harmonic Wind Model)*. NASA Contractor Report 3256. NASA Contract NAS 8-32897: March, 1980.

THESIS PROCESSING SLIP

FIXED FIELD: ill _____ name _____

index _____ biblio _____

► COPIES Archives Aero Dewey Eng Hum
Lindgren Music Rotch Science

TITLE VARIES ► _____

NAME VARIES ► _____

IMPRINT (COPYRIGHT) _____

► COLLATION 506 p

► ADD. DEGREE: _____ ► DEPT.: _____

SUPERVISORS: _____

NOTES:

cat'r.

date:

► DEPT: Aero

page.
<u>J155</u>

► YEAR: 1998 ► DEGREE: M.S.

► NAME: FISCHER, ~~Jack~~ Jack David