Reduced-Order Atmospheric Density Modeling for LEO Satellite Orbital Reentry Prediction

by

Nicolette LeAnn Clark

B.S., United States Air Force Academy (2021)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored by: Nicolette LeAnn Clark
Department of Aeronautics and Astronautics
May 18, 2023

Certified by: Richard Linares
Associate Professor
Thesis Supervisor

Accepted by: Jonathan P. How
R. C. Maclaurin Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

THIS PAGE INTENTIONALLY LEFT BLANK

Reduced-Order Atmospheric Density Modeling for LEO Satellite Orbital Reentry Prediction

by

Nicolette LeAnn Clark

Submitted to the Department of Aeronautics and Astronautics
on May 18, 2023, in partial fulfillment of the
requirements for the degree of
MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

Abstract

Atmospheric density modeling and uncertainty quantification for fast and accurate orbit propagation are vital to drag force estimation for satellite reentry prediction and conjunction assessment in today's increasingly cluttered Low Earth Orbit (LEO) environment. Current density models can be computationally expensive, and often contain large errors, which make density modeling a leading cause of uncertainty in drag estimation. Reduced-order atmospheric density Models (ROMs) have shown potential to provide good predictive performance at a significantly lower computational cost, by propagating a low-dimensional representation of the atmospheric density state instead of the entire density state.

In this thesis, ROMs were implemented in a high-fidelity orbital propagator and tested on the problem of reentry modeling for LEO objects. First, uncertainty quantification was performed on a test case for three significant sources of uncertainty in reentry modeling to compare the impact of uncertainty from initial state, ballistic coefficient, and space weather indices on residual lifetime estimation. These results highlighted features of interest in ROM behavior relative to empirical models.

Second, ROMs were used to predict reentry of three LEO object test cases. ROMs were found to provide residual lifetime estimation performance comparable to current empirical models such as JB2008 and NRLMSISE-00, with a run time reduction of up to 70% compared to the empirical models. ROMs were especially effective for longer predictions starting two or more days prior to LEO object reentry, where in some cases ROMs outperformed both empirical models while saving hours of run time. These findings validate the utility of ROMs for orbit propagation applications such as reentry prediction.

Thesis Supervisor: Richard Linares
Title: Associate Professor

Plain Language Summary

Over the last several decades, the number of satellites in Low-Earth Orbit (LEO) has increased rapidly. With the growing population of LEO satellites comes an urgent requirement for accurate orbit propagation to predict future satellite "states" (positions and velocities). Orbit propagation gives satellite operators time to react to potentially hazardous situations before they occur. Examples of such hazardous scenarios include debris-generating collisions such as the 2009 Iridium-Cosmos collision, and atmospheric reentry of large spacecraft which may impact the ground and cause casualties or damage, as in the May 2020 reentry of a Chinese rocket body which damaged buildings in the Ivory Coast. Orbit propagator algorithms function by modeling the forces experienced by a object in orbit, including gravity, drag, and solar radiation pressure, in order to predict its motion.

Orbit propagation for reentry prediction is used to forecast the time and/or location of a satellite's reentry into the lower atmosphere. Reentry prediction may be applied to assess the damage/casualty risk from pieces of spacecraft which survive the reentry process to impact the ground instead of burning up during reentry. Additionally, reentry analysis may be used to investigate premature deorbit events such as the February 2022 loss of 38 Starlink satellites, when unexpectedly high atmospheric densities induced by a geomagnetic storm caused the Starlink satellites to reenter the atmosphere within weeks after launch.

In the LEO environment between 100 and 1000 km above Earth's surface, air drag is one of the largest non-gravitational forces on a satellite. To accurately determine satellites' future positions, propagators must be able to model drag. However, drag force calculations require knowledge of air density at LEO altitudes, where the atmosphere is thin but influential to satellite orbits. A variety of atmospheric models predict densities, but many exhibit large errors. Uncertainty in atmospheric density is a limiting factor in the accuracy of LEO orbit propagation. Solar and geomagnetic activity (space weather) increase uncertainty in a satellite's future state by causing rapid changes in density, which may not be well-predicted by atmospheric models. Another challenge of orbit propagation is its computational expense, due largely to the computational demands of atmospheric density models.

In this thesis, a new type of atmospheric density model is tested on reentry prediction. Reduced-Order density Models (ROMs) show potential for accurate density modeling at a low computational cost, but they have not yet been widely applied in orbit propagation. The goal of this thesis is to provide preliminary validation of ROM performance in orbit propagation by demonstrating their utility on reentry prediction for three spacecraft test cases. ROM behaviors and features of interest in reentry modeling are explored in an uncertainty quantification analysis, prior to extensive testing of the reentry prediction performance of four ROMs and two empirical density models. ROMs are found capable of offering comparable reentry prediction accuracy to commonly-used density models, at a fraction of the computational expense. The results of this thesis validate the potential utility of ROMs in fast and accurate orbit propagation for reentry prediction and other applications.

Acknowledgments

First, I wish to thank Dr. Richard Linares for his help and support as my advisor throughout my program of study and the writing of this thesis. Dr. Linares, you gave me the perfect balance of providing direction where I needed it, while also encouraging me to branch out and explore my research interests. Thank you so much for your guidance during the last two years!

Secondly, to my labmates in ARCLab, thank you for your many contributions to my work in the form of conversations, critique, and collaboration. In particular, thank you to Dr. Peng Mun Siew and to Julia Briden for your contributions to my thesis research in the process of our work on a paper for the Advanced Maui Optical and Space Surveillance Technologies. Both of you provided amazing support, encouragement, assistance, and feedback on my work both during and after the conference paper. Thank you so much for the time and energy you put into helping me. I'd also like to thank Will Parker for feedback and discussion which contributed to this thesis, especially in the area of potential operational applications.

I would also like to thank the Lincoln Laboratory Military Fellows program and Lincoln Lab Group 93 for sponsoring and supporting me through this degree program. I extend special thanks to my Lincoln Laboratory mentor, Alan Smith.

I acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC, database, and consultation resources that have contributed to the research results reported within this thesis.

I extend thanks to David Gondelach for sharing BC estimation code used in this thesis.

The JB2008 solar and geomagnetic indices are provided for scientific use courtesy of Space Environment Technologies. Earth orientation parameters and additional space weather indices are courtesy of CelesTrak.

I also wish to express my unending gratitude for my loving and supportive family. Every step of the way, you have done everything possible to encourage me and help me succeed. I'm especially grateful for my brother Chris, whose footsteps I followed into the Air Force and to MIT. Chris, I'm so honored to have been your sister for 23 years. I'll always look up to you.

Finally, I praise my Lord and Creator, the Maker of the heavens and the earth. I thank you with a humble heart for Your love and salvation through Your Son, Jesus Christ.

The views expressed in this thesis are the views of its author, and do not reflect the official policy or position of MIT Lincoln Laboratory, the U.S. Air Force, the U.S. Department of Defense, or the U.S. Government.

For Chris.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

1 Introduction **32**

1.1 Motivation . . . 32

1.2 Reduced-Order Density Modeling and Reentry Prediction . . . 36

1.3 Goal, Overview, and Scope of Thesis . . . 40

2 Literature Review **44**

2.1 Reentry prediction techniques and accuracy . . . 44

2.2 Sources of error in reentry prediction . . . 51

2.2.1 TLE use and error . . . 54

2.2.2 Drag Modeling Error . . . 58

2.3 Ballistic Coefficient . . . 61

2.4 Atmospheric Density Modeling . . . 66

2.4.1 Empirical atmospheric density models . . . 69

2.4.2 Physics-based atmospheric density models . . . 73

2.4.3 Reduced-order atmospheric density models . . . 76

2.5 Space Weather Impact on Satellite Reentry . . . 80

2.6 Density Model Comparisons for Reentry . . . 85

2.7 Thesis Description . . . 89

3 Methods **92**

3.1 Hardware, Language, and Software Packages . . . 92

3.2 Orbit Propagation . . . 94

3.2.1 Description of the orbital propagator and code structure . . . 94

3.2.2 Orbit Propagation Theory 97

3.3 Test Cases and Object Parameters 102

3.3.1 Test Cases and Basic Parameters 102

3.3.2 Initial States 105

3.3.3 Ballistic Coefficient Estimation and Modeling 106

3.4 Atmospheric Density Models 109

3.4.1 Empirical Models 109

3.4.2 Reduced-Order Model Theory 109

3.4.3 DESTO and DESQU Toolboxes 118

3.4.4 Reduced-Order Model Implementation 120

3.5 Uncertainty Quantification and Monte Carlo 121

3.5.1 TLE Uncertainty 122

3.5.2 BC Uncertainty 123

3.5.3 Space Weather Uncertainty 124

3.6 Simulations and Expected Results 127

4 Uncertainty Quantification 132

4.1 Initial-State Uncertainty 132

4.2 Ballistic Coefficient Uncertainty 137

4.3 Space Weather Uncertainty 142

4.3.1 Noise in ROM state Initialization 143

4.3.2 No Noise in ROM state Initialization 146

4.4 Summary of Uncertainty Quantification 149

5 Reentry Simulations 152

5.1 Upper Atmosphere Research Satellite (UARS) 152

5.1.1 Saunders-Gondelach Mean BC 154

5.1.2 Pardini-Anselmo Mean BC 157

5.1.3 Pardini-Anselmo time-varying BC 160

5.1.4 UARS Overall Results 162

5.1.5 A Note on Reentry Locations 168

5.2 Fobos-Grunt 170
5.2.1 Saunders-Gondelach Mean BC with NRLM00 172
5.2.2 Pardini-Anselmo Mean BC 175
5.2.3 Pardini-Anselmo Time-Varying BC 178
5.2.4 Saunders-Gondelach Mean BC with JB2008 180
5.2.5 Overall Results 183
5.3 Progress-M 27M 188
5.3.1 Saunders-Gondelach Mean BC Results 190
5.3.2 Pardini-Anselmo Mean BC 193
5.3.3 Pardini-Anselmo Time-Varying BC 196
5.3.4 Overall Results 200
5.3.5 A Final Note on Reentry Locations 204
5.4 Summary 206

6 Conclusions 212

6.1 Key Findings and Importance 212
6.2 Strengths and Weaknesses of the Research 216
6.3 Directions for Future Research 218

A Appendix A 222

A.1 UARS Run Time Plots 222
A.2 Fobos-Grunt Run Time Plots 224
A.3 Progress-M 27M Run Time Plots 227

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

1-1 Number of tracked objects in Low Earth Orbit (apogee ≤2000 km) per calendar year. Data is from Spacetrack Satellite Situation Report. . . 33

1-2 Illustration of spacecraft reentry and ground impact footprint [19]. Created with Microsoft PowerPoint. 34

1-3 Illustration of the order-reduction process for a ROM. Figure courtesy of Julia Briden, adapted from Fig. 3 in [35]. 38

1-4 Initial application of ROMs to reentry simulation for an analysis of one of the satellites lost in the February 2022 Starlink deorbit event. Altitudes shown are geocentric. Figure originally published in [10]. . . 39

1-5 Diagram of thesis organization. 41

2-1 Reentry time prediction study by Pardini and Anselmo (2013) for the ROSAT reentry. Red indicates prediction, and blue indicates uncertainty window. Figure is from [5]. 48

2-2 Histogram of mean in-track TLE error reported by Kelso (2007) at propagation time = 0 for 22 satellites in the GPS constellation. Data is from [41]. 55

2-3 Diagram of atmospheric layers [23]. Created with Microsoft PowerPoint. 67

2-4 Neutral mass densities predicted by NRLMSISE-00 at 400 km altitude at 12:00 UTC on 10 May 2023. 71

2-5 Neutral mass densities predicted by WAM-IPE at 400 km altitude at 12:00 UTC on 13 May 2023. Data from NOAA SWPC. 75

2-6 First 10 spatial modes identified by POD in NRLMSISE-00 snapshots. Figure reproduced from [59], courtesy of Dr. Richard Linares. 77

2-7 Time-dependent coefficients associated with first 10 spatial modes identified by POD in NRLMSISE-00 snapshots. Figure reproduced from [59], courtesy of Dr. Richard Linares. 78

2-8 Illustration of the Earth's magnetosphere (light blue lines) [57]. Created with Microsoft PowerPoint. 82

2-9 Base-10 logarithm of densities computed by JB2008 empirical density model at a fixed latitude of 30 degrees North, at a time of low space weather activity on 12 October 2019 (left) and during a geomagnetic storm on 4 February 2022 (right) Labeled points indicate approximate orbital altitudes of various satellites of interest. 83

2-10 Percent increase in densities calculated by JB2008 during a geomagnetic storm, as compared to densities during low SW activity. 84

3-1 Simplified code structure diagram. 96

3-2 Architecture of the NN autoencoder and decoder. Reproduced from [10], courtesy of Dr. Peng Mun Siew. 113

3-3 Illustration of TLE uncertainty distribution in 3 dimensions for a satellite at point [2.5, 2.5, 2.5]. 123

3-4 Example of a set of noisy SW values for the UARS test case. Unique sets of noisy SW values were used for each MC iteration. True values are plotted for reference. 126

4-1 UARS reentry trajectory ranges predicted by three density models during $N = 150$ Monte Carlo simulation with radial, along-track, and out-of-plane error added to initial position. 133

4-2 Distribution of change in reentry time predicted by three density models during $N = 150$ Monte Carlo simulation with radial, along-track, and out-of-plane error added to TLE. Change in reentry time given relative to unperturbed reentry time predicted by each respective model, as a percent of true residual lifetime. 134

4-3 Reentry trajectories predicted by three density models with Saunders-Gondelach mean BC and ±25% BCs. 137

4-4 Residual lifetime predicted by three density models with Saunders-Gondelach mean BC and ±25% BCs. 138

4-5 Reentry time prediction study by Pardini and Anselmo (2013) for the UARS reentry. Red indicates prediction, and blue indicates uncertainty window. Figure is from [5]. 139

4-6 Residual lifetime percent error predicted by three density models with Saunders-Gondelach mean BC and ±25% BCs. 140

4-7 UARS reentry trajectory ranges predicted by three density models (with noise in ROM initialization) during Monte Carlo simulation with Gaussian zero-centered noise added to F10.7, S10.7, M10.7, and Y10.7. 144

4-8 Distribution of change in reentry time predicted by three density models (with noise in ROM initialization) during $N = 150$ Monte Carlo simulation with Gaussian zero-centered noise added to F10.7, S10.7, M10.7, and Y10.7. Change in reentry time given relative to unperturbed reentry time predicted by each respective model, as a percent of true residual lifetime. 145

4-9 UARS reentry trajectory ranges predicted by three density models (no noise in ROM initialization) during Monte Carlo simulation with Gaussian zero-centered noise added to F10.7, S10.7, M10.7, and Y10.7. . . 147

4-10 Distribution of change in reentry time predicted by three density models (no noise in ROM initialization) during $N = 150$ Monte Carlo simulation with Gaussian zero-centered noise added to F10.7, S10.7, M10.7, and Y10.7. Change in reentry time given relative to unperturbed reentry time predicted by each respective model, as a percent of true residual lifetime. 148

5-1 UARS ballistic coefficient models. 153

5-2 UARS reentry trajectories predicted with the Saunders-Gondelach mean BC using 6 different atmospheric density models and 6 different simulation durations. 154

5-3 Percent error (%) of each model over time using the Saunders-Gondelach mean BC. 155

5-4 Absolute error (hrs) of each model over time using the Saunders-Gondelach mean BC. 156

5-5 UARS reentry trajectories predicted with the Pardini-Anselmo mean BC using 6 different atmospheric density models and 6 different simulation durations. 157

5-6 Percent error (%) of each model over time using the Pardini-Anselmo mean BC. 158

5-7 Absolute error (hrs) of each model over time using the Pardini-Anselmo mean BC. 158

5-8 UARS reentry trajectories predicted with the Pardini-Anselmo time-varying BC using 6 different atmospheric density models and 6 different simulation durations. 160

5-9 Percent error (%) of each model over time using the Pardini-Anselmo time-varying BC. 161

5-10 Absolute error (hrs) of each model over time using the Pardini-Anselmo time-varying BC. 161

5-11 Average residual lifetime prediction MAPE across all simulation durations, for all density models and all BC models, for the UARS test case. 163

5-12 UARS residual lifetime prediction MAPE for all density and BC models for short prediction windows of one nominal day or less. 164

5-13 UARS residual lifetime prediction MAPE for all density and BC models for longer prediction windows of 2-7 nominal days. 165

5-14 Average HPOP run time by density model for UARS test case. 167

5-15 Ground tracks and 100-km reentry locations (starred) predicted for UARS. 169

5-16 Fobos-Grunt ballistic coefficient models. 170

5-17 Fobos-Grunt reentry trajectories predicted with the Saunders-Gondelach mean BC using 6 different atmospheric density models and 6 different simulation durations. 173

5-18 Percent error (%) of each model over time using the Saunders-Gondelach mean BC. 174

5-19 Absolute error (hrs) of each model over time using the Saunders-Gondelach mean BC. 174

5-20 Fobos-Grunt reentry trajectories predicted with the Pardini-Anselmo mean BC using 6 different atmospheric density models and 6 different simulation durations. 175

5-21 Percent error (%) of each model over time using the Pardini-Anselmo mean BC. 176

5-22 Absolute error (hrs) of each model over time using the Pardini-Anselmo mean BC. 177

5-23 Fobos-Grunt reentry trajectories predicted with the Pardini-Anselmo time-varying BC using 6 different atmospheric density models and 6 different simulation durations. 178

5-24 Percent error (%) of each model over time using the Pardini-Anselmo time-varying BC. 179

5-25 Absolute error (hrs) of each model over time using the Pardini-Anselmo time-varying BC. 180

5-26 Fobos-Grunt reentry trajectories predicted with the Saunders-Gondelach mean BC fitted with the JB2008 density model, using 6 different atmospheric density models and 6 different simulation durations. 181

5-27 Percent error (%) of each model over time using the Saunders-Gondelach mean BC derived with the JB2008 density model. 182

5-28 Absolute error (hrs) of each model over time using the Saunders-Gondelach mean BC derived with the JB2008 density model. 182

5-29 Average residual lifetime prediction MAPE across all simulation durations, for all density models and all BC models, for the Fobos-Grunt test case. 183

5-30 Fobos-Grunt residual lifetime prediction MAPE for all density and BC models for short prediction windows of one nominal day or less. . . . 185

5-31 Fobos-Grunt residual lifetime prediction MAPE for all density and BC models for longer prediction windows of 2-7 nominal days. 185

5-32 Average HPOP run time by density model for Fobos-Grunt test case. 187

5-33 PM27M Ballistic coefficient estimates. 189

5-34 PM27M reentry trajectories predicted with the Saunders-Gondelach mean BC using 6 different atmospheric density models and 6 different simulation durations. 191

5-35 Percent error (%) of each model over time using the Saunders-Gondelach mean BC. 192

5-36 Absolute error (hrs) of each model over time using the Saunders-Gondelach mean BC. 192

5-37 PM27M reentry trajectories predicted with the Pardini-Anselmo mean BC using 6 different atmospheric density models and 6 different simulation durations. 194

5-38 Percent error (%) of each model over time using the Pardini-Anselmo mean BC. 195

5-39 Absolute error (hrs) of each model over time using the Pardini-Anselmo mean BC. 195

5-40 PM27M reentry trajectories predicted with the Pardini-Anselmo time-varying BC using 6 different atmospheric density models and 6 different simulation durations. 197

5-41 Percent error (%) of each model over time using the Pardini-Anselmo time-varying BC. 198

5-42 Absolute error (hrs) of each model over time using the Pardini-Anselmo time-varying BC. 198

5-43 Mean absolute percent error (MAPE) across all prediction window lengths, for all models and all BCs. 200

5-44 PM27M residual lifetime prediction MAPE for all density and BC models for short prediction windows of one nominal day or less. 201

5-45 PM27M residual lifetime prediction MAPE for all density and BC models for longer prediction windows of 2-7 nominal days. 202

5-46 Average HPOP run time by density model for PM27M test case. . . . 204

5-47 Ground tracks and 100-km reentry locations (starred) predicted for PM27M with Pardini-Anselmo time-varying BC. 205

5-48 Percent error distributions by density model, for all BCs, all prediction windows, and all test cases. 208

6-1 Run time per hour of simulation for all density models, as a percent of slowest model's run time per simulation hour. Averaged across all BCs, prediction windows, and test cases. 213

6-2 Recommended directions for future work, based on the goal of interest. 219

A-1 Run time performance of all density models on UARS test case for all simulation durations for the Saunders-Gondelach mean BC simulation set. 222

A-2 Run time performance of all density models on UARS test case for all simulation durations for the Pardini-Anselmo mean BC simulation set. 223

A-3 Run time performance of all density models on UARS test case for all simulation durations for the Pardini-Anselmo time-varying BC simulation set. 223

A-4 Run time performance of all density models on UARS test case for all simulation durations, averaged across all BC models. 224

A-5 Run time performance of all density models on Fobos-Grunt test case for all simulation durations for the Saunders-Gondelach mean BC simulation set. 224

A-6 Run time performance of all density models on Fobos-Grunt test case for all simulation durations for the Pardini-Anselmo mean BC simulation set. 225

A-7 Run time performance of all density models on Fobos-Grunt test case for all simulation durations for the Pardini-Anselmo time-varying BC simulation set. 225

A-8 Run time performance of all density models on Fobos-Grunt test case for all simulation durations for the simulation set using Saunders-Gondelach mean BC derived with the JB2008 density model. 226

A-9 Run time performance of all density models on Fobos-Grunt test case for all simulation durations, averaged across all BC models. 226

A-10 Run time performance of all density models on Progress-M 27M test case for all simulation durations for the Saunders-Gondelach mean BC simulation set. 227

A-11 Run time performance of all density models on Progress-M 27M test case for all simulation durations for the Pardini-Anselmo mean BC simulation set. 227

A-12 Run time performance of all density models on Progress-M 27M test case for all simulation durations for the Pardini-Anselmo time-varying BC simulation set. 228

A-13 Run time performance of all density models on Progress-M 27M test case for all simulation durations, averaged across all BC models. . . . 228

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

3.1 Complete list of Julia packages used. 93

3.2 Comparison of HPOP and AIDA propagator configurations. 95

3.3 Spacecraft test case physical parameters. 102

3.4 TLE filter settings used in Saunders-Gondelach BC estimation [31, 49]. Values and descriptions derived from code by David Gondelach and Aleksander Lidtke (2015). 106

3.5 Modeling thermospheric density using a ROM. 117

3.6 PODROM training data parameters, from [33]. 119

3.7 ROM space weather inputs, reconstructed from [33]. 119

3.8 Uncertainty quantification simulations. 128

3.9 Standard simulations. 128

5.1 Density models ranked by MAPE performance on UARS test case with Saunders-Gondelach mean BC. 166

5.2 Density models ranked by MAPE performance on Fobos-Grunt test case with Pardini-Anselmo mean BC. 186

5.3 Density models ranked by MAPE performance on PM27M test case with Pardini-Anselmo time-varying BC. 203

6.1 Pros and cons of ROM density models for orbit propagation. 215

THIS PAGE INTENTIONALLY LEFT BLANK

Nomenclature

Abbreviations

2BE	Two-Body Equation
2BP	Two-Body Problem
6DOF	Six-degree-of-freedom
ADR	Active Debris Removal
AIDA	Accurate Integrator for Debris Analysis
AU	Astronomical Unit
BC	Ballistic Coefficient
CHAMP	Challenging Minisatellite Payload
CIRA	COSPAR International Reference Atmosphere
CME	Coronal Mass Ejection
CNES	National Centre for Space Studies
CNN	Convolutional Neural Network
COSPAR	Committee On Space Research
CPU	Central processing unit
DAS	Debris Assessment Software
DCA	Dynamic Calibration Atmosphere
DESQU	Dynamic Data-Driven Thermospheric Density Estimation with Quantified Uncertainties
DESTO	Density Estimation Toolbox
DMDc	Dynamic Mode Decomposition with Control
DRAMA	Debris Risk Assessment and Mitigation Analysis

Dst	Disturbance storm time index
DSTDTC	Temperature change computed from Dst index
DTM	Drag Temperature Model
EAS	Early Ammonia Servicer
ECI	Earth-Centered Inertial
EGM2008	Earth Gravitational Model 2008
EGM96	Earth Gravitational Model 1996
ESA	European Space Agency
EUV	Extreme Ultraviolet
GB	Gigabyte
GEO	Geosynchronous Orbit
GITM	Global Ionosphere-Thermosphere Model
GOCE	Gravity field and steady-state Ocean Circulation Explorer
GPS	Global Positioning System
GRACE	Gravity Recovery and Climate Experiment
HASDM	High Accuracy Satellite Drag Model
HEO	Highly Elliptical Orbit
HPOP	High-Precision Orbit Propagator
HS-DMDc	Hermitian Space-Dynamic Mode Decomposition with Control
IAASS	International Association for the Advancement of Space Safety
IADC	Inter-Agency Space Debris Coordination Committee
ID	Identification
IMPACT	Integrated Modeling of Perturbations in Atmospheres for Conjunction Tracking
IPE	Ionosphere Plasmasphere Electrodynamics
IRI	International Reference Ionosphere
ISS	International Space Station

JB2008	Jacchia-Bowman 2008
JB8ROM	JB2008-based PODROM
JPL	NASA Jet Propulsion Laboratory
JR	Jacchia-Roberts
JSpOC	Joint Space Operations Center
KB	Kilobyte
km	Kilometers
LEO	Low Earth Orbit
LVLH	Local-vertical, local-horizontal
MAPE	Mean Absolute Percent Error
MB	Megabyte
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
ML	Machine Learning
ML ROM	Machine Learning Reduced Order Model
MSIS	Mass Spectrometer and Incoherent Scatter radar
NASA	National Aeronautics and Space Administration
NCAR	National Center for Atmospheric Research
NN	Neural Network
NOAA	National Oceanic and Atmospheric Administration
NORAD	North American Aerospace Defense Command
NRLM00	Alternative abbreviation for NRLMSISE-00
NRLMSISE-00	Naval Research Laboratory Mass Spectrometer Incoherent Scatter radar Extended 2000
NRLROM	NRLMSISE-00 based PODROM
ODE	Ordinary Differential Equation
ORSAT	Object Re-entry Survival Analysis Tool

PA	Pardini-Anselmo ballistic coefficient estimation technique
PM27M	Progress-M 27M
POD	Proper Orthogonal Decomposition
PODROM	Proper Orthogonal Decomposition-Reduced Order Model
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RMS	Root Mean Square
RNN	Recurrent Neural Network
ROM	Reduced-Order Model
ROSAT	Roentgen Satellite
RSO	Resident Space Object
SARA	Survival And Risk Analysis
SATRAP	Satellite Reentry Analysis Program
SCARAB	Spacecraft Atmospheric Re-entry and Aerothermal Breakup
SEP	Solar Energetic Particles
SESAM	Spacecraft Entry Survival Analysis Module
SG	Saunders-Gondelach ballistic coefficient estimation technique
SGP4	Simplified General Perturbations 4
SRP	Solar Radiation Pressure
SRPC	Solar Radiation Pressure Coefficient
STELA	Semi-analytic Tool for End of Life Analysis
SVD	Singular Value Decomposition
SW	Space Weather
SWPC	Space Weather Prediction Center
TIE-GCM	Thermosphere-Ionosphere-Electrodynamic General Circulation Model
TIEROM	TIE-GCM based POD ROM
TLE	Two-Line Element

UARS	Upper Atmosphere Research Satellite
UQ	Uncertainty Quantification
USSPACECOM	United States Space Command
UTC	Coordinated Universal Time
WAM	Whole Atmosphere Model
WAM-IPE	Whole Atmosphere Model-Ionosphere Plasmasphere Electrodynamics
WFS	WAM-IPE Forecast System

Symbols

$\bar{\square}$	Mean
$\dot{\square}$	First derivative
$\ddot{\square}$	Second derivative
$\tilde{\square}$	Variance
$\overrightarrow{\square}$	Vector quantity
$\langle\square\rangle$	Inner product operation
$\sqrt{\square}$	Square root
a	Acceleration
$\mathbf{A}$	Discrete-time DMDc dynamic matrix
a_{drag}	Acceleration due to aerodynamic drag
$\mathbf{A}_c$	Continuous-time DMDc dynamic matrix
A_{eff}	Effective cross-sectional frontal area of satellite
F	Force
$A_{\odot}$	Satellite area normal to sun
$\mathbf{B}$	Discrete-time DMDc input matrix
$\mathbf{B}_c$	Continuous-time DMDc input matrix
BC	Ballistic coefficient (quantity $\frac{c_D A_{eff}}{m}$)

B^*	Two-line Element B-star drag term
c_D	Dimensionless drag coefficient
c_R	Satellite reflectivity
$c(t)$	Time-dependent coefficients of POD
d	Dimension of density snapshot matrix
ϵ_r	POD reconstruction truncation error
$f(t,y)$	Force model
g	In Eqn. 3.15: Autoencoder convolutional layer output
G	Universal gravitational constant
h	Altitude above ellipsoid (Earth's surface)
H	Scale height (table lookup quantity)
h_0	Reference altitude (table lookup quantity)
JD_{start}	Julian date at desired simulation start time
$JD_{reentry}$	Julian date of true reentry time
JD_{TLE0}	Julian date of initial TLE (true simulation start time)
L_{sim}	Desired/nominal simulation length
$\mathcal{L}$	Mean square error loss function (See Eqn. 3.13)
M_3	Third-body mass
m	Mass of satellite
m_{Earth}	Earth's mass
μ	Earth gravitational parameter
n	Number of density snapshots
N	Number of MC simulation runs
ν	Fraction of solar disk visible from satellite position
r_3	Vector from third-body mass to satellite
r	In orbit propagation: Satellite position
r	In Reduced-order Modeling: Number of modes

$r_{\odot}$	Vector from sun to satellite
p_{srp}	Solar pressure at 1 AU
$\Phi(\mathbf{s})$	Orthonormal basis functions of POD
ρ	Atmospheric density
ρ_0	In Eqn. 2.3: Atmospheric density reference value (see Eqn. 2.4)
ρ_0	In Eqn. 2.5: Reference density (table lookup quantity)
$\mathbf{s}$	Spatial position
σ	Standard deviation
t	Time
T	Time between density snapshots
t_0	Initial/start time
t_f	Final time
t_r	Predicted reentry time
$t_{r,true}$	True reentry time
$\mathbf{U}$	Full SVD U-matrix (left singular vectors)
$\mathbf{u}_k$	Vector of space weather inputs at epoch k
$\mathbf{U}_r$	Truncated SVD U-matrix (first r columns)
v	Satellite velocity vector
v_{rel}	Relative velocity between satellite and local atmosphere
x	Autoencoder convolutional layer input
$\mathbf{x}(\mathbf{s}, t)$	Atmospheric density distribution
$\bar{\mathbf{x}}(\mathbf{s})$	Mean density distribution
$\mathbf{x}'(\mathbf{s}, t)$	Encoded and reconstructed density field
$\tilde{\mathbf{x}}(\mathbf{s}, t)$	Density variance distribution
$\mathbf{X}$	Matrix of density snapshots
$\mathbf{X}'$	Encoded and reconstructed matrix of density snapshots
y	State vector

y_0	Initial state vector
Υ	Matrix of space weather vectors
$\mathbf{z}$	ROM state
$\mathbf{z}_0$	Initial ROM state
$\mathbf{z}_k$	ROM state at epoch k
$\mathbf{Z}_1,\ \mathbf{Z}_2$	Time-shifted density snapshot matrices
$\mathscr{Z}(\mathbf{s}, t)$	Statistical z-score of base-10 logarithm of density snapshot
$\mathscr{Z}'(\mathbf{s}, t)$	Encoded and reconstructed statistical z-score of base-10 logarithm of density snapshot

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

1.1 Motivation

In 1957, a man-made satellite orbited the Earth for the first time [2]. In 1978, after barely two decades of human activity in space, the growth of the artificial satellite population into the thousands prompted National Aeronautics and Space Administration (NASA) scientist Donald Kessler to sound an alarm by describing a hypothetical condition of cascading collisions between orbiting Resident Space Objects (RSOs), since nicknamed the "Kessler Syndrome," in which debris-generating collisions create clouds of spacecraft fragments, which spawn additional collisions until a belt of artificial debris forms around the Earth [42]. Thirty years after Kessler's premonitory warning, the 2009 collision of the Cosmos 2251 and Iridium 33 satellites produced upwards of 1,600 debris fragments, many of which have the potential to destroy or damage other RSOs, including manned spacecraft such as the International Space Station (ISS) [108]. Worse still, accidental and intentional [99] fragmentation events, defunct satellites, rocket bodies, and other objects contribute to a growing population of long-lived space debris, some of which will remain in orbit for years to decades [108, 1]. Already, active satellites constitute only a small fraction of the total population of RSOs in Earth orbit [3]. Although increasing attention towards debris mitigation strategies has prompted research and experimentation on the feasibility of active debris removal (ADR) [8, 24, 16], widespread usage of ADR is currently nonexistent.

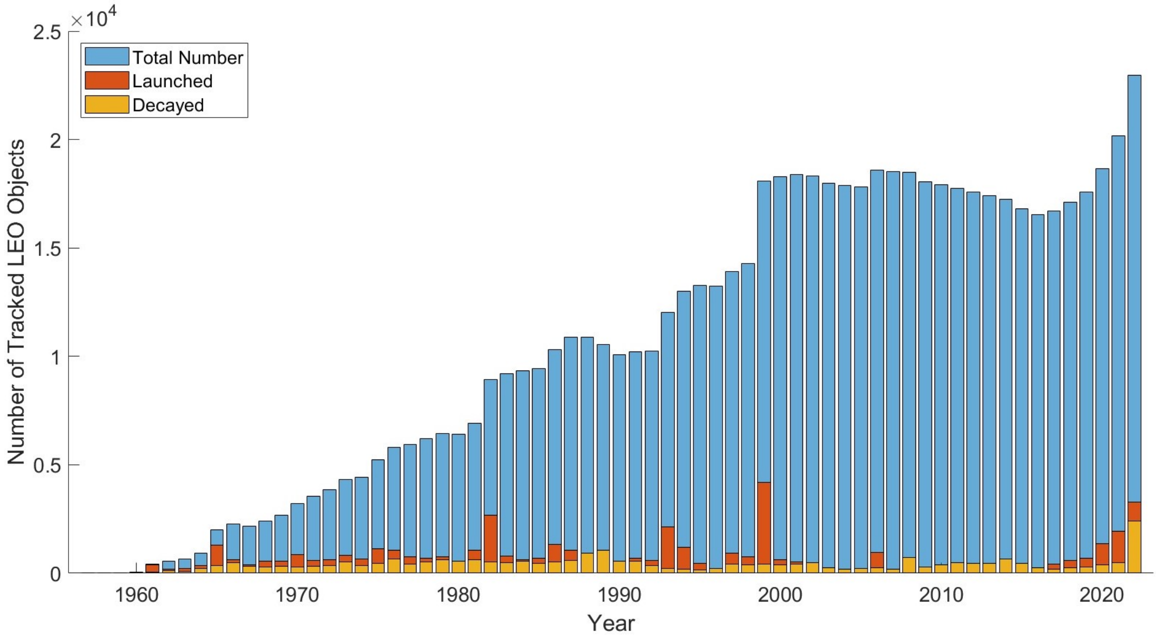


Figure 1-1: Number of tracked objects in Low Earth Orbit (apogee ≤2000 km) per calendar year. Data is from Spacetrack Satellite Situation Report.[1]

Meanwhile, space activity continues to multiply. In 2019, the population of more than 2,000 active satellites was set to quintuple over the next decade [93], a prediction which seems likely to be realized. As of 2022, SpaceX's Starlink constellation alone exceeded 2,000 active satellites, with plans to build a megaconstellation of up to 42,000 satellites total [84]. And unlike early satellites, whose primary missions were often scientifically or politically motivated, today's satellites are deeply integrated into global technological systems with applications ranging from navigation and communication to emergency response and disaster management. In other words, the growth of space traffic is set to exponentially increase, while the implications of a catastrophic chain of collisions grow increasingly dire.

[1] https://www.space-track.org/#ssr

Preservation of the LEO space environment for current and future use is a top priority, given its enormous and growing importance for a vast range of technological applications. But managing space traffic to prevent collisions and near-approaches among the estimated 29,000 LEO objects greater than 10 cm in size, not to mention the 170+ million smaller objects [4], is a challenging logistical, political, economic, and technical problem. And the challenges of a crowded space environment do not end with conjunction assessment. Greater LEO activity leads to a rising number of objects reentering the atmosphere each year, which creates another kind of risk: the ground impact of deorbiting spacecraft.

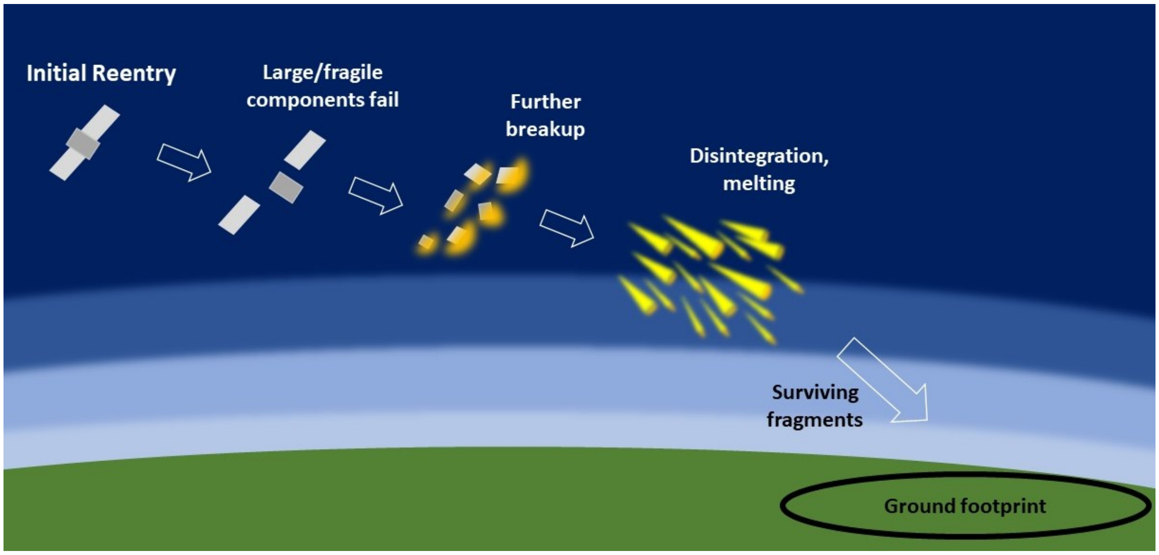


Figure 1-2: Illustration of spacecraft reentry and ground impact footprint [19]. Created with Microsoft PowerPoint.

Although most deorbiting objects will burn up or melt due to heating during the reentry process, some debris may survive the reentry process and fall to Earth, potentially causing casualties or property damage. This is especially concerning if the object is large, contains high amounts of toxic materials, or contains durable materials likely to survive reentry, such as titanium [75]. Unfortunately, the spent rocket bodies left in uncontrolled orbit after most launches are just such an example of large objects capable of surviving reentries [12]. In fact, in May of 2020, pieces of a Chinese rocket body survived atmospheric reentry and damaged buildings in two villages in the Ivory Coast [12]. Although falling space debris is not yet a widespread hazard, Byers et. al. estimated a 10% chance of casualties over the next decade [12]. Besides ground impact, space debris could also pose a threat to aviation safety [19]. For both controlled and uncontrolled objects, reentry prediction is essential for assessing the hazards that surviving materials may pose to people and property on the ground and in the air. Additionally, for controlled objects, reentry predictions can be used to facilitate safe disposal over unpopulated areas of the Earth. With the disposal of the ISS on the horizon, NASA recently underscored the importance of reentry prediction by announcing its intent to spend up to $1 billion on ensuring the safe and controlled deorbit of the ISS [25].

Among the technical challenges inherent in space traffic management is the need for fast, accurate orbit propagation for conjunction assessment and reentry prediction. In LEO, uncertainty in future satellite states requires a margin for error in conjunction assessment and reentry prediction. On orbit, maintaining these margins for error between satellites will become increasingly difficult as the LEO environment becomes more crowded, so reducing the uncertainty is a priority. For reentry modeling, this uncertainty may decrease the ability to determine whether people or property are at risk from space debris ground impact. One of the leading causes of uncertainty in orbit propagation is drag modeling inaccuracy due to errors in atmospheric density modeling [106]. Despite decades of efforts towards fast, accurate atmospheric density models, the models commonly used in practice today are known to contain significant uncertainties. Improving the accuracy of density models and quantifying associated uncertainty in LEO orbit propagation is a vital area of research for the continued safe and productive use of LEO.

1.2 Reduced-Order Density Modeling and Reentry Prediction

Accurate modeling of atmospheric density is vital to orbit propagation in Low Earth Orbit (LEO), since air drag is directly proportional to atmospheric (air) density, and is the largest non-gravitational force acting on LEO satellites [104]. The primary effect of drag on an LEO satellite orbit is to circularize and shrink the orbit, which results in a gradual reduction of orbital altitude over time until the satellite reenters the denser lower layers of the Earth's atmosphere and burns up, disintegrates, and/or impacts the ground. Because of the dominance and impact of the drag force in LEO, propagating a LEO satellite's orbit for reentry prediction or conjunction analysis is heavily reliant on the atmospheric density modeling capability.

Atmospheric density models in use today typically fall into one of two classes: empirical or physics-based models [20]. Empirical density models involve large historical data sets of measurements, including actual density values and/or quantities related to density, such as temperature [82]. Density-related quantities are incorporated using the relevant physical equations to derive densities from these data sets. Empirical density models may also impose physical constraints [82]. However, the key feature of empirical models is that they estimate densities primarily by performing interpolation and/or extrapolation upon their underlying data sets to estimate a density for the particular input conditions given. Input conditions include, at a minimum, date, time, and location (typically in latitude, longitude, and altitude), but usually also include other inputs such as solar and geomagnetic conditions. Empirical models are often used in operational applications since they are relatively fast to run. However, even empirical models can become computationally expensive for high-fidelity simulations, which may require density computations every few seconds over a propagation window of hours to days. Additionally, empirical models do not incorporate the underlying dynamics of the atmosphere.

Physics-based models are, in a sense, the opposite of empirical models. While they may rely on empirical models or data sets to establish boundary conditions, physics-based models operate by solving the systems of equations that describe fluid flow [20]. Since these equations must be solved at hundreds or thousands of grid points in latitude, longitude, and altitude to construct density fields for each time of interest, physics-based models can be extremely computationally expensive to run, although they may have some benefits in density forecasting ability [20]. Because of their prohibitively high computational cost, physics-based models are not widely used for orbit propagation.

The hypothetical "golden standard" of atmospheric density models is a model capable of combining the benefits of empirical and physics-based models by providing a fast run time equal to or faster than today's empirical models while leveraging the underlying dynamics of the thermosphere to improve forecasting ability. Recent progress towards such a hybrid model has occurred with the development of Reduced-Order atmospheric density Models (ROMs), first developed in [59, 62] and expanded in [60, 63, 33, 32, 30, 61, 34, 103]. Reduced-order models encode a spatial grid of density points into a lower-dimensional "reduced-order space," which can represent an entire global distribution of density values in a very compressed form. Propagation (forecasting) of the density state is conducted in the reduced-order space, and the final reduced-order state is then converted back into the full-density space. Density state propagation in the low-dimensional reduced-order state space is very efficient, making ROMs computationally inexpensive.

ROMs can be trained on density datasets generated by any density model, including physics-based models. Provided the data set is sufficiently extensive, the trained ROM density model remains valid outside of the time period of the original training data. In other words, ROMs can be used to capture the key dynamics of a density

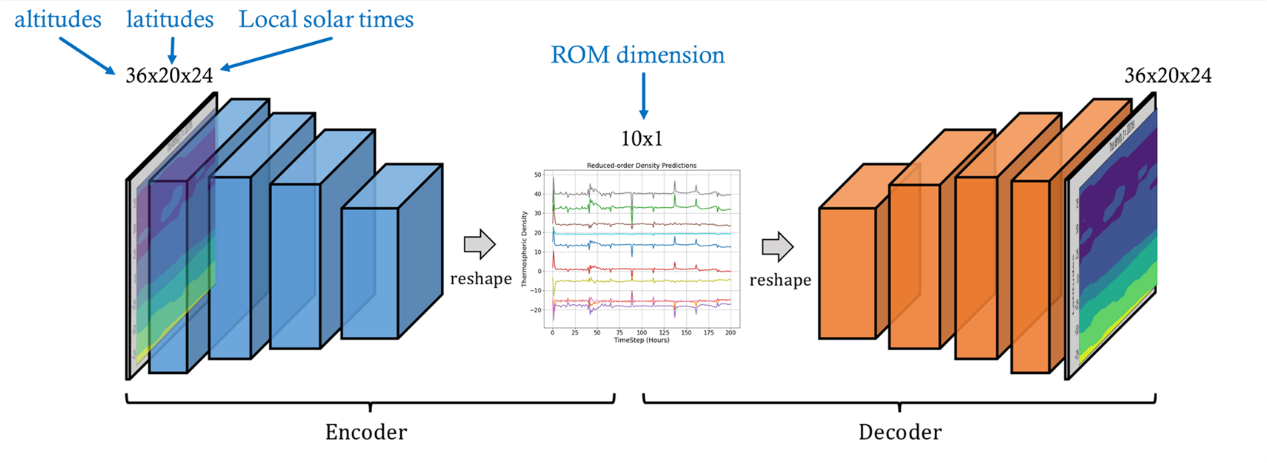


Figure 1-3: Illustration of the order-reduction process for a ROM. Figure courtesy of Julia Briden, adapted from Fig. 3 in [35].

data set and approximately replicate those dynamics at a greatly reduced computational cost. This offers the potential for improved density forecasts at a fraction of the computational costs of physics-based density models. Added benefits of ROMs include easy data assimilation from a variety of data sources for model calibration [60, 33, 61, 34] and uncertainty quantification of ROM densities [32, 30].

Despite extensive validation performed throughout the development of the ROM technique, discussed in [59, 62, 103] and other papers, ROMs have not yet been widely applied to common operational density model use cases, such as orbit propagation. Initial application of ROMs in orbit propagation for conjunction assessment was conducted in [30, 32], and ROMs were used for reentry simulation by the author of this thesis and others in [10]. This thesis expands significantly on the reentry modeling work in [10]. Application of ROMs to common density model use cases represents an opportunity for further validation of this class of density models, and a chance to comparatively evaluate their performance in typical applications relative to other density models.

Besides its importance in ground impact risk assessment, reentry prediction is an appealing area for ROM application for multiple reasons. First of all, it is a

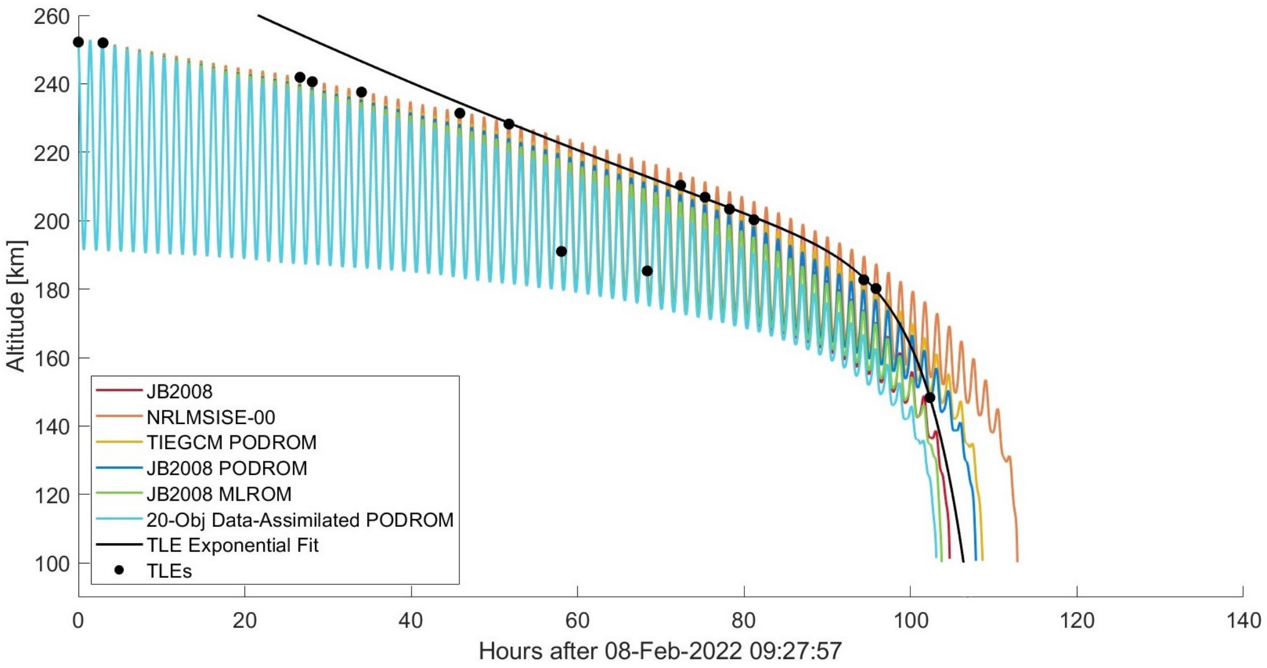


Figure 1-4: Initial application of ROMs to reentry simulation for an analysis of one of the satellites lost in the February 2022 Starlink deorbit event. Altitudes shown are geocentric. Figure originally published in [10].

challenging problem under the best of conditions, with residual lifetime prediction errors in reentry campaigns often exceeding 10% [76] and sometimes surpassing 20% [46]. Secondly, as Solar Cycle 25 ramps up [96], the prospect of increasing frequency and intensity of elevated solar/geomagnetic activity means that reentry predictions will rarely be made "under the best of conditions." Space weather in the form of solar and geomagnetic activity can have dramatic impacts on LEO satellites' orbits, even to the point of inducing severely premature reentry, as seen in the SpaceX Starlink loss event of February 2022, when 38 Starlink satellites were lost within weeks of launch due to geomagnetic storm conditions [10, 21]. This event highlighted the need for research into space weather impacts on orbital reentry. Applying ROMs to the problem of reentry, therefore, offers the chance to fill this gap, while also comparing ROM responses to space weather to those observed in traditional empirical density models.

1.3 Goal, Overview, and Scope of Thesis

The objective of this thesis is to validate the performance of reduced-order atmospheric density models on the exemplary orbit propagation problem of reentry simulation and to conduct quantitative analysis of uncertainty sources in reentry prediction in order to explore density model behavior and identify areas for improvement of orbit propagation results using reduced-order density models. This thesis builds upon initial work conducted in [10] by selecting reentry test cases with improved data availability and known true reentry times, expanding the scope of density model comparisons by including multiple ballistic coefficients (BC) models and examining the evolution of density model performance over time, and adding uncertainty quantification to the reentry study.

Towards this end, uncertainty quantification (UQ) is first accomplished for the three most significant uncertainty sources in the reentry prediction process, apart from inherent density model uncertainty: ballistic coefficient, initial state, and space weather uncertainty. Ballistic coefficient impacts are examined by varying the BC above and below its nominal value to determine its contribution to the reentry time uncertainty. The contribution of initial-state uncertainty is examined with Monte Carlo (MC) techniques. Finally, MC methods are applied to space weather uncertainty in two separate ways, which explore 1) the impact of space weather uncertainty on reentry predictions for two ROMs and an empirical density model, and 2) the impact of space weather uncertainty on ROM initialization. The purpose of the UQ analysis is to reveal the challenges of separating density model error from other sources of error and to outline important characteristics of ROM behavior.

Second, reentry prediction is conducted on a sample set of spacecraft reentering the atmosphere from LEO using four ROMs and two empirical atmospheric density models. Density models are compared on the basis of accuracy and run time. The results produced by various ballistic coefficient (BC) modeling strategies and reentry prediction window durations are simultaneously investigated. This section comprises the bulk of the ROM validation work, by rigorously comparing the performance of four ROMs to the JB2008 and NRLMSISE-00 density models in terms of absolute error, relative error, and run time. Differences in density model performance are observed with multiple BC models and prediction windows for every test case. The organization of this thesis is described below.

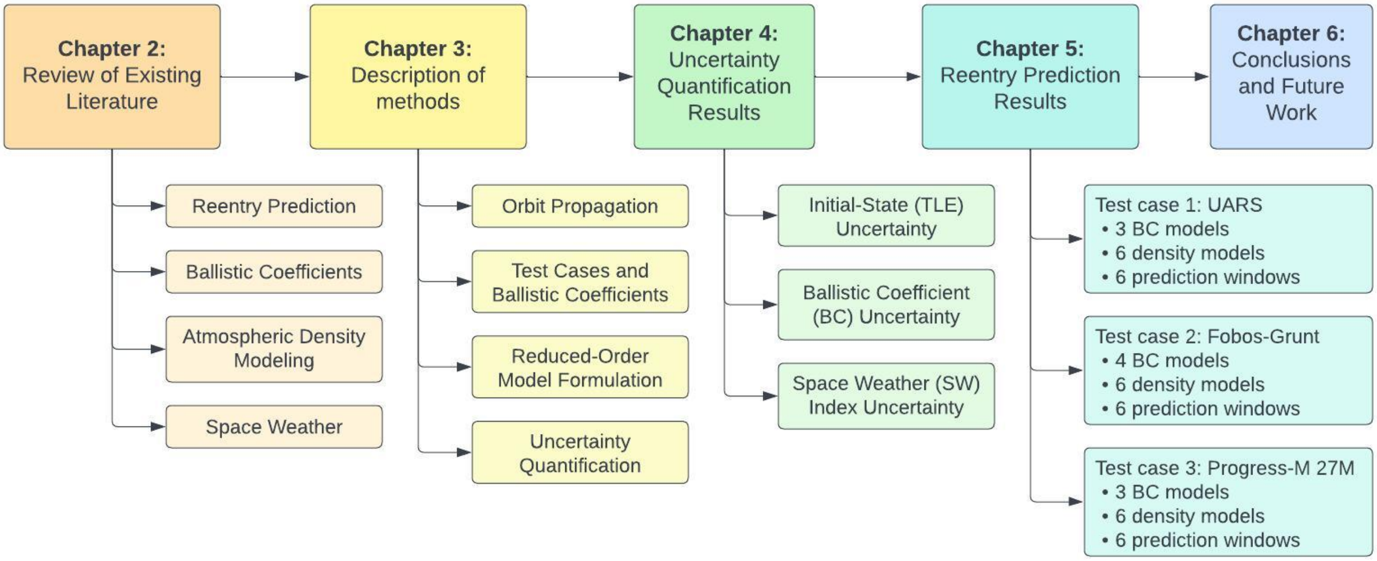

Chapter 2: Review of Existing Literature
Reentry Prediction
Ballistic Coefficients
Atmospheric Density Modeling
Space Weather
Chapter 3: Description of methods
Orbit Propagation
Test Cases and Ballistic Coefficients
Reduced-Order Model Formulation
Uncertainty Quantification
Chapter 4: Uncertainty Quantification Results
Initial-State (TLE) Uncertainty
Ballistic Coefficient (BC) Uncertainty
Space Weather (SW) Index Uncertainty
Chapter 5: Reentry Prediction Results
Test case 1: UARS • 3 BC models • 6 density models • 6 prediction windows
Test case 2: Fobos-Grunt • 4 BC models • 6 density models • 6 prediction windows
Test case 3: Progress-M 27M • 3 BC models • 6 density models • 6 prediction windows
Chapter 6: Conclusions and Future Work

Figure 1-5: Diagram of thesis organization.

Chapter 2 offers a rigorous literature review on the topics of reentry prediction and its associated uncertainty sources, ballistic coefficient and atmospheric density modeling, and space weather impacts. Criteria for evaluating reentry prediction accuracy are justified in Section 2.1. The comparative performance of density models for reentry prediction or similar applications is highlighted in Section 2.7.

Chapter 3 outlines the methods and techniques applied in this thesis. This chapter describes the basic theory of orbit propagation using the special perturbations method and presents the formulation, training, and implementation of reduced-order atmospheric density models. Additionally presented are details on code structure and availability, LEO object test cases and parameters, ballistic coefficient models, and uncertainty quantification methodology. At the end of this chapter, a set of reentry simulations using various density models, BC models, prediction windows, and uncertainty sources is selected for analysis.

Chapter 4 conducts uncertainty quantification to determine the relative contributions of initial state, ballistic coefficient, and space weather to uncertainty in predicted reentry time. As part of the uncertainty quantification, the impact of initialization error on ROMs is investigated by introducing space weather error into the ROM initialization process. Chapter 4 lays foundations for understanding ROM and reentry simulation behavior, and helps interpret the ROM validation work in Chapter 5.

Chapter 5 presents the performance of each density model and BC model on the set of three LEO object test cases. Observations and comparisons of density/BC model performance are offered on the basis of run time and accuracy. Results are discussed comparatively, with reference to the analyses performed in Chapter 4 and to the performance criteria established in Section 2.1.

Finally, Chapter 6 summarizes the findings presented in Chapters 4 and 5 and offers takeaways on ROM performance and behavior and uncertainty quantification in reentry prediction. Opportunities for improvement and future research are presented.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Literature Review

2.1 Reentry prediction techniques and accuracy

Predicting the time and location of an orbiting object's reentry into the atmosphere is a challenging task, given the significant uncertainties in the satellite's characteristics and environment [38]. Precise reentry predictions rely on accurate knowledge of parameters such as the satellite's ballistic coefficient (affected by satellite mass, composition, shape, and rotational state, as well as by the atmosphere's local chemical composition), and the local density of the atmosphere. Atmospheric density prediction depends on solar and geomagnetic activity data, which also suffer from uncertainty in predicted values. Complicating the problem is the extreme sensitivity of the satellite's reentry location to small changes in the reentry time. Orbital velocities are extremely high in LEO, with a full orbit around the Earth lasting only about 90 minutes, so small errors in reentry time yield significant errors in reentry location [38]. Factors such as computational resources available may present additional constraints on the achievable accuracy of reentry prediction techniques.

Organizations with an interest in reentry prediction include space agencies such as NASA, the European Space Agency (ESA), and the French National Centre for Space

Studies (CNES); international bodies such as the Inter-Agency Space Debris Coordination Committee (IADC) and the International Association for the Advancement of Space Safety (IAASS); military organizations such as the U.S. Space Command (USSPACECOM); and satellite owners and operators seeking to plan for the safe and responsible disposal of their spacecraft. Depending on the particular purpose of the reentry prediction, varying levels of accuracy may be desired. For example, a prelaunch assessment of how long it will take for a spacecraft to deorbit after completing its mission may seek only a rough estimate with an acceptable uncertainty of several months or even years. A research paper studying the effects of geomagnetic activity on the premature deorbit of satellites may seek results with uncertainty on the order of a few hours, while reentry predictions for ground casualty risk assessment may demand higher accuracy and better uncertainty quantification (UQ).

Depending on the resources available and the level of accuracy needed for the particular application, reentry prediction techniques of choice range from high-precision propagation using six-degree-of-freedom (6DOF) modeling of satellite translational and rotational state [26], to Monte Carlo techniques simulating an ensemble of possible uncertainty conditions to predict a maximum-likelihood reentry point [38], to purely data-based predictions with a neural network absent any underlying consideration of orbital dynamics [39]. Many reentry analysis tools also include survivability analysis [50], which is useful in evaluating the level of casualty and damage risk from debris fragments surviving reentry. Reentry footprint predictions, as in [43], focus on finding ground impact locations of debris fragments for risk assessment.

Common tools for reentry analysis generally fall into one of two categories: object-oriented and spacecraft-oriented techniques [50]. Examples of object-oriented methods include NASA's Debris Assessment Software (DAS), developed in 1998 by Lockheed; NASA's Object reentry Survival Analysis Tool (ORSAT), developed by NASA in 1993; and the Spacecraft Entry Survival Analysis Module (SESAM) in the ESA's

Debris Risk Assessment and Mitigation Analysis (DRAMA) [50]. Currently, the ESA's Spacecraft Atmospheric reentry and Aerothermal Breakup (SCARAB), developed for ESA by a German company in 1995, is the only spacecraft-oriented tool [13, 50]. Object-oriented techniques involve a simplified model of the spacecraft's components, and assume that these individual components do not begin to disintegrate until after the spacecraft reaches its breakup altitude [50]. This second assumption can lead to an exaggerated assessment of ground casualty and damage risk [50], which may be useful when seeking an upper bound on ground risk. Spacecraft-oriented techniques attempt to model the full spacecraft dynamics using a model of all spacecraft components and accounting for complex effects, such as shielding of some components behind others, in order to predict the altitude and dynamics of breakup [13, 50].

The DAS tool provides a fast, simplified breakup model for preliminary assessments, which tends to notably overestimate risk due to its binary predictive capabilities: DAS predicts either survival or demise for each component, based on the predicted heat absorption [50]. It does not account for effects such as fragmentation of components, or partial melting [50]. ORSAT does model partial melting but is still limited in its accuracy due to non-consideration of lift forces on reentering objects [50]. SESAM is a slightly modified version of the ORSAT techniques, which is used to predict the velocity, angle, and location of impacting fragments, which can, in turn, be used for risk assessment [50]. SCARAB uses a complete geometrical spacecraft model and 6DOF propagation to predict the satellite's translational and rotational state, while simultaneously analyzing heating and stress on components to predict melting and fragmentation [13, 50].

Comparisons of ORSAT, DAS, SESAM, and SCARAB were conducted by Lips and Fritsche (2005) [50]. They found reasonably good agreement between the four tools concerning aerodynamic and aerothermodynamic effects [50]. The most significant difference found was in predictions of material properties; DAS and ORSAT predicted higher emissivity and heat capacity than did the other tools, which made DAS and ORSAT more likely to predict the survival of satellite components [50]. More recent tools such as the CNES Semi-analytic Tool for End of Life Analysis software (STELA) and the ESA's Survival And Risk Analysis (SARA) module of DRAMA may offer improvements over older tools, especially in predicting residual lifetime and ground risk [13].

The focus of this thesis is residual lifetime prediction (a prerequisite to precise risk assessment) using a set of atmospheric density models. Ground risk and reentry footprint prediction are not treated herein, though they are also essential for the safe disposal of RSOs. While residual lifetime prediction cannot and should not be wholly separated from its role in ground risk assessment, we turn now to recent work in reentry time prediction techniques and accuracy. Generally speaking, residual lifetime prediction can be accomplished by propagating a spacecraft's initial position forward in time until its altitude drops below a predefined "deorbit altitude." As long as the deorbit altitude is above 80-90 km, reentry can be accurately modeled with orbital dynamics [77]. While sophisticated reentry analysis tools such as ORSAT and SCARAB include such techniques, these full-analysis tools are not strictly necessary if survivability, ground risk, and reentry footprint are not considered. Simpler reentry simulations can be accomplished using orbital propagators ranging from SGP4 to high-fidelity 6DOF propagators, such as the one Geul et. al. (2018) used to analyze the Gravity field and steady-state Ocean Circulation Explorer (GOCE) reentry [26]. Even when the problem of reentry analysis is restricted to only residual lifetime prediction, the ability to consistently and accurately predict orbital reentry remains a work in progress, due to large uncertainties in atmospheric density, ballistic coefficient, and other factors involved in these predictions. Uncertainty sources in reentry prediction are discussed in depth beginning in Section 2.2. Reentry time predictions may be especially challenging for highly elliptical orbits [68], though filtering of measurements used for initial state determination may improve results [67, 68].

In 2012 and 2013, Pardini and Anselmo examined the accuracy of real-time reentry predictions for three large spacecraft made in support of an IADC reentry campaign and the Italian Civil Protection Authorities [75, 5]. These spacecraft (NASA's Upper Atmosphere Research Satellite (UARS), German Aerospace Center's Roentgen Satellite (ROSAT), and Roscosmos' Fobos-Grunt) were of particular interest due to their large size and estimated risk of ground casualties surpassing the 1/10,000 standard used as an "alert threshold" by a number of agencies, including the National Aeronautics and Space Administration (NASA) [75]. Average residual lifetime errors for ROSAT and Fobos-Grunt were found to be 3% and 4%, respectively, while the average residual lifetime error for UARS was 15%; a large terminal drop in the UARS' ballistic coefficient may have been partially responsible for the greater errors in its predicted lifetime [75].

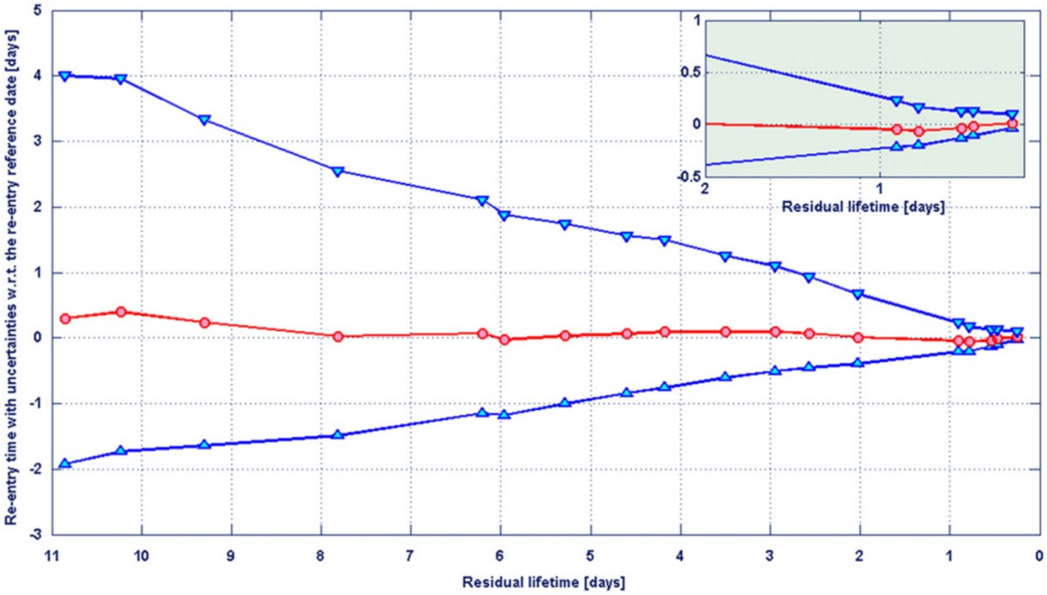


Figure 2-1: Reentry time prediction study by Pardini and Anselmo (2013) for the ROSAT reentry.[1] Red indicates prediction, and blue indicates uncertainty window. Figure is from [5].

[1] Figure reproduced in accordance with the agreement between MIT and publisher Elsevier. For more information, see https://libraries.mit.edu/scholarly/publishing/using-published-figures/.

Another 2012 study of UARS reentry time prediction developed a Markov Chain Monte Carlo (MCMC) technique for quantifying uncertainty both in reentry time and location [38]. Factors considered as sources of uncertainty were the NRLMSISE-00 density model, the predicted values of F10.7 used as an input to the density model, and the ballistic coefficient. 150 iterations of the MCMC algorithm yielded an asymmetric distribution of reentry times centered at 04:16 on 24 September 2011, only 16 minutes after the observed reentry time, with a 1-sigma value of ∼37 minutes [38]. Horsley observed that the distribution of predicted reentry times was skewed right, with later reentry times having a longer tail in the predicted reentry time distribution [38]. While recent works by Trisolini and Colombo have outlined alternative methods of reentry uncertainty quantification [101, 102], Monte Carlo techniques remain a valid approach when the computational cost is not a significant barrier.

In 2013, the ESA performed reentry predictions using a combination of MSIS-90e and CIRA-72 to estimate atmospheric density, along with a series of three drag coefficient estimation techniques to provide drag coefficients with the uncertainty of ±10% or less [46]. Klinkrad found that ESA reentry time predictions were accurate to within ±20% in about 95% of cases; however, this finding was based on IADC reentry campaigns, for which more high-quality data is available than for reentries without associated IADC campaigns [46]. Therefore, it is unclear whether Klinkrad's finding can accurately characterize the quality of ESA reentry predictions for cases without an associated IADC reentry campaign. In the same year, Pardini and Anselmo (2013) found mean relative reentry epoch prediction errors of $>$10% (either for the whole campaign, for the last 48 hours, or both) in 5 out of 15 examined test cases, although only one case had the error above 20% [76].

A 2016 study of reentry predictions for Progress-M 27M, a failed Russian cargo vessel, found a mean residual lifetime prediction error of 12% over the 8-day period prior to its reentry. The peak error of around 20% falling 3 days before the reentry was likely a result of a geomagnetic storm the next day, which led to higher densities than the 3-day predictions accounted for. The lowest error of 5% occurred just over one day prior to reentry [77].

One alternative method of reentry prediction bears mentioning. In a recent paper, Jung et. al. explored the use of a recurrent neural network (RNN) to predict reentry trajectories [39]. The novelty of this approach lies in the fact that the RNN method was entirely independent of the system dynamics; that is, no direct modeling of the forces experienced by reentering satellites was performed. The RNN was trained on a database consisting of epoch and altitude information of objects with known reentry times [39]. Reentry times were then predicted using epoch and altitude information for test cases (excluded from the training set) [39]. Provided that the RNN was trained on objects with a similar BC to those it was tested on, the average relative error of its prediction for 4 test cases was around 7%. This was on par with the 6-9% average relative error observed in IADC reentry campaigns for the same 4 objects [39]. However, when the RNN was trained on objects with different ballistic coefficients than those it was tested on, its average relative error increased to almost 36% [39].

This method shows promise for providing quick, data-driven reentry predictions, but it is likely to suffer in cases where system dynamics differ significantly from training data. For example, the RNN approach might break down for reentries impacted by geomagnetic storms, unless it was specifically trained on a data set of similar cases. The RNN method represents a realistic alternative for reentry prediction, as long as a training set composed of many reentries under similar conditions is available. However, dynamics-based approaches are likely to be more effective when training data under similar conditions is limited.

Based on the accuracy evaluations discussed above, an approximate benchmark for reentry time prediction accuracy can be given. A residual lifetime prediction error of ±10% is a reasonable goal for prediction performance given current methods, while single-digit error (especially below 5%) can be considered excellent performance. In order to compete with existing reentry prediction methods, prediction error should be under ±20% for the vast majority of test cases. These metrics for residual lifetime prediction performance can be used to evaluate how well a particular method performs relative to existing counterparts.

2.2 Sources of error in reentry prediction

The general form of a reentry prediction problem using propagation techniques is that of an initial-value ordinary differential equation (ODE) problem, that is, a problem of the form:

Given:

$f(t, y) = \dot{y}$ and $y(t_0) = y_0$

Find:

$y(t)$ for $t \in [t_0, t_f]$

$$where\ y = \begin{pmatrix} r_x \\ r_y \\ r_z \\ \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \end{pmatrix} \ and\ \dot{y} = \begin{pmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \\ \ddot{r}_x \\ \ddot{r}_y \\ \ddot{r}_z \end{pmatrix}$$

$$with\ \dot{r} = \frac{dr}{dt}\ and\ \ddot{r} = \frac{d^2r}{dt^2}$$

Solving this problem requires two primary pieces of information: an initial state vector y_0 describing the position and velocity of the satellite at initial time t_0, and a function $f(t, y)$ describing the time rate of change (derivative) of the components of the satellite state vector. The first three components of the state vector derivative are the derivatives of the position components, that is, the velocity components. These are obtained directly from the last three elements of the state vector. The last three components of the state vector derivative are the satellite's acceleration. The acceleration components can be obtained using Newton's Second Law:

$$\sum F = m\overrightarrow{a} \tag{2.1}$$

By describing and summing the forces acting on the satellite in the x, y, and z directions and dividing out the satellite's mass, we can describe the satellite's acceleration a in each direction. Because it calculates acceleration from the sum of the forces acting on a satellite, the function $f(t, y)$ in orbit propagation is sometimes called the "force model." The force model includes the gravitational force of the Earth, as well as orbital perturbations including the forces of atmospheric drag, solar radiation pressure, and third-body gravitational effects.

Orbital propagation is subject to error in both the initial state y_0 and in the force model $f(t, y)$. High-accuracy initial states may be obtained from Global Positioning System (GPS) data; however, not all satellites carry GPS trackers, and the availability of this data (especially in real time) is often limited. Frequently, the only source of orbital data available for satellites is publicly-available Two-Line Elements (TLEs) produced by the North American Aerospace Defense Command (NORAD). Unfortunately, TLE accuracy leaves much to be desired. TLEs are known to contain errors up to 10 km [41], and can suffer from other problems such as misattribution of TLEs to the wrong RSOs [92]. TLE error and its impacts are discussed in the following section. Besides initial state error, the force model also introduces errors into the orbital propagation process. Error in the force model comes from such sources as the absence of forces that are intentionally unmodeled because of their negligible magnitude; inaccuracies in the satellite parameters (mass, area, drag / solar radiation coefficients) used in calculating solar radiation pressure and atmospheric drag; and errors in the atmospheric density model used in calculating the drag force. Small errors are also introduced by the numerical techniques (integrators) used for solving the differential equation problem of orbital propagation.

Some of these errors are relatively minor, but significant errors exist in the areas of 1) insufficient tracking data (leading to low accuracy in the initial state); 2) ballistic coefficient determination, due to uncertainty in satellite configuration, attitude, and drag coefficient; and 3) atmospheric density model uncertainty, both from the inherent uncertainty of the model and from uncertainty in predicted space weather inputs [76]. The majority of these uncertainties impact the estimation of the aerodynamic drag force, which is collectively the largest source of uncertainty for LEO satellite orbits [56]. In reentry predictions for the GOCE satellite using a high-fidelity propagator, Geul et. al. (2018) found that two drag-related uncertainties (spacecraft attitude and density model uncertainty) led to the largest uncertainties in reentry time [26]. Atmospheric density uncertainty was found to be the largest single source of uncertainty in the GOCE reentry prediction uncertainty analysis [26].

A complete description and quantification of every error source present in orbital propagation is unwarranted for the purposes of this thesis, since the dominating uncertainties come from only a few sources. Instead, this thesis focuses on the largest sources of uncertainty, especially on drag force modeling. In the following sections, three of the significant error sources relevant to reentry modeling are discussed: TLE error, ballistic coefficient estimation, and atmospheric density modeling. The latter two uncertainty sources are discussed both individually and collectively, since both fall under the broader topic of drag modeling uncertainty.

2.2.1 TLE use and error

NORAD releases TLE sets for a large catalog of objects orbiting the Earth. TLEs contain the satellite's identifying information and mean orbital elements [70], along with the date and time of the TLE. TLEs can be obtained from CelesTrak[2] and from the 18th Space Defense Squadron[3] [105]. A full description of the TLE format can be found on CelesTrak's documentation page.[4] NORAD TLEs are generated with the Simplified General Perturbations 4 (SGP4) propagator [37, 105], and direct propagation of TLEs should be accomplished using this propagator.[5]

TLEs represent a vital source of data for space researchers and satellite operators, since no other readily available data source provides such comprehensive data about most of the RSOs in Earth orbit [105]. Other sources of tracking data may be available only to the operators of a particular satellite (for example, positioning data from an onboard GPS unit) or to those with access to specialized equipment (such as radar tracking data from observatories). In the absence of other sources of data, TLEs are widely used for applications ranging from conjunction assessment [105] to ballistic coefficient estimation [95] to computing density model corrections [110] and more. While the large repository of publicly available data provided by TLEs is a distinct advantage for research and operational applications, there are downsides: TLEs often contain significant inaccuracies, and moreover do not contain covariance information [41, 70]. This means not only that TLEs can be erroneous, but also that it is difficult to quantify how large the errors may be [70]. Efforts have been made to quantify TLE uncertainty [70, 27], but inconsistent TLE accuracy continues to plague users.

Kelso (2007) assessed the accuracy of TLEs and TLE propagation by using SGP4 to propagate TLEs both 15 days forward and 15 days backward in time from an initial

[2] https://celestrak.org/

[3] https://www.space-track.org/

[4] https://celestrak.org/NORAD/documentation/tle-fmt.php

[5] See SpaceTrack documentation at https://www.space-track.org/documentation#sgp4

TLE and comparing the results to precise GPS ephemerides for satellites in the GPS constellation [41]. Kelso examined and plotted[6] radial, in-track, and cross-track errors across the 30-day window for each of the 22 satellites examined. The mean TLE error at the epoch time is found from Kelso's plots by examining the mean error at x = 0. Figure 2-2 summarizes in-track mean TLE error at the epoch time from all 22 plots generated by Kelso (rounding error magnitude shown on plots down to the nearest integer value).

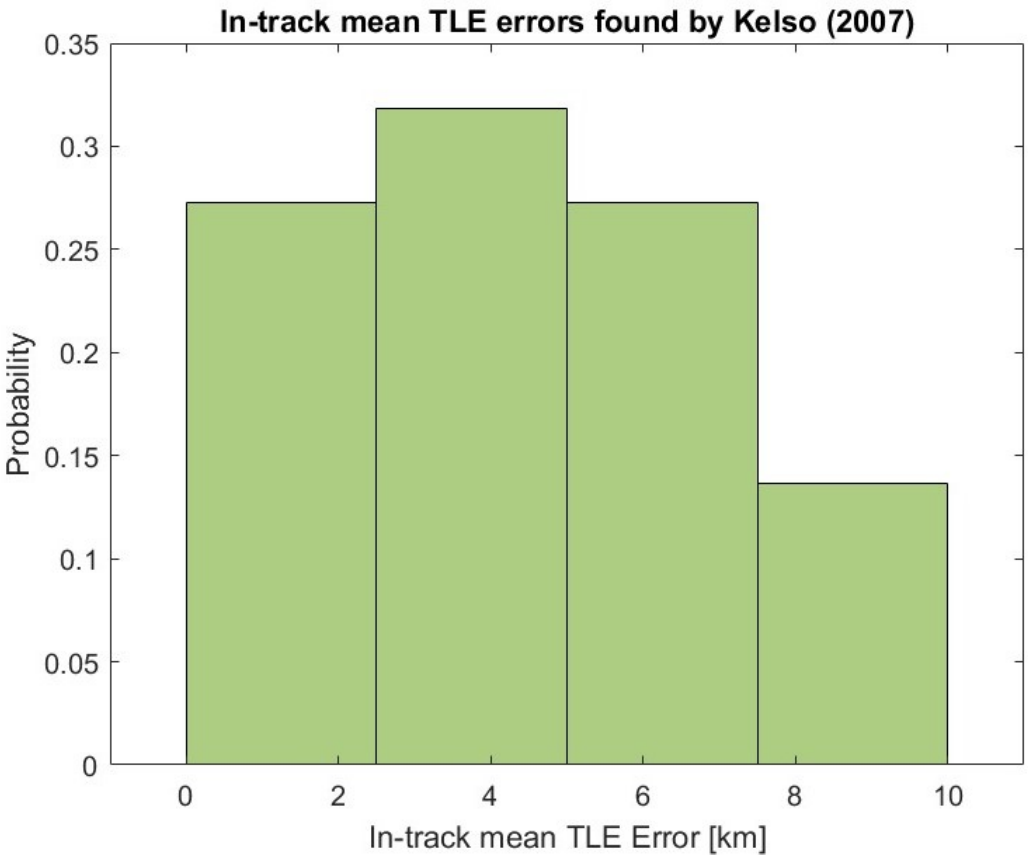


Figure 2-2: Histogram of mean in-track TLE error reported by Kelso (2007) at propagation time = 0 for 22 satellites in the GPS constellation. Data is from [41].

Kelso recorded mean errors of up to 10 km in the in-track direction at the epoch time, and errors of 50 km or more in the in-track direction after 15 days of forward propagation [41]. Less bias was seen in the radial and across-track errors, which

[6] http://celestrak.org/publications/AAS/07-127/

generally centered around zero, but the variance in radial error was found to be up to about 2 km at the TLE epoch and increased with forward propagation. Based on Kelso's plots,[7] cross-track error variance was also up to 2 km at the TLE epoch, but was in most cases less than radial error. Kelso noted that errors were not symmetric in forward versus backward propagation, with larger errors generally resulting from forward propagation [41]. In fact, backward propagation regularly yielded lower errors than those at the TLE epoch time [41].

Similar TLE accuracy results were found for small satellites. In an analysis of TLE propagation accuracy for three small satellites in low Earth orbit (LEO), Kahr et. al. (2013) found that TLE propagation errors reached up to 50 km in the in-track direction after two weeks of forward propagation, while radial and cross-track errors remained much lower, around 1 km [40]. Like Kelso, Kahr et. al. found that errors were asymmetric with respect to the direction of propagation, and forward propagation yielded larger errors than backward propagation [40]. However, TLE in-track error was much lower at the epoch time (2-3 km in the in-track direction), except when the satellites maneuvered (TLEs do not account for maneuvers) [40].

In a study of 10 cubesats deployed from the ISS, Riesing found much larger errors after a shorter forward propagation period, though it is possible that these large errors are unique to ISS-released cubesats, which could be at higher risk for misidentification (where the TLE is incorrectly labeled as belonging to the wrong satellite) [92]. It is also possible that the higher TLE propagation errors in this study may be influenced by atmospheric drag. Although SGP4 is the preferred propagator for TLEs, Osweiler found that SGP4 performance was notably worse for orbits low enough to experience drag [70]. Since drag is a major influence in the LEO regime, poor handling of drag by SGP4 could have contributed to propagation errors in Riesing's study.

Fortunately, the problem of TLE error may not always be as grim as these findings

[7] http://celestrak.org/publications/AAS/07-127/

suggest. TLE propagation error using SGP4 is of less concern for the purposes of this thesis than is LEO TLE error at epoch time, that is, at the time of the TLE.

TLE uncertainty at epoch time varies based on orbital characteristics, such as eccentricity, inclination, and orbital regime [22]. Flohrer et. al. (2008) found that TLE along-track uncertainty was lower for objects in circular or near-circular orbits [22]. In addition, for objects in higher-eccentricity orbits, a higher inclination led to lower along-track uncertainty, though medium-inclined orbits had the largest out-of-plane uncertainty [22]. The orbital regime also played a significant role, with the lowest uncertainties found in the MEO regime, followed by the LEO regime [22]. Higher orbits, like the orbits of the GPS satellites in Kelso's study, typically had higher uncertainties, as did highly-elliptical orbits [22]. While larger errors did occur in some cases, the uncertainty in TLE position averaged across the entire TLE catalog in the LEO regime was found to be about 0.1 km in the across-track and out-of-plane directions, and about 0.5 km in the along-track direction [22].

Improvements in TLE generation procedures around 2013 led to higher TLE quality since that time [27]. Geul et. al. developed the Weighted Differences (WD) procedure for estimating TLE uncertainty in [27] and applied this technique to an error quantification of TLEs for the GOCE satellite, concluding that post-2013 TLEs had approximately half the uncertainty of pre-2013 TLEs [26]. However, TLE inaccuracies remain. Besides the errors discussed above, TLEs are sometimes mislabeled as belonging to other objects [49, 92], and are a relatively sparse source of data, with only around two TLEs released per day, per object [26]. TLE filtering for outliers and large time gaps is helpful in some applications, but this results in even sparser data [31].

The work in this thesis utilizes initial state vectors derived from TLEs, due to lack of access to alternate, higher-accuracy data sources. TLE-derived initial states are used for propagation with a modified version of the High-Precision Orbit Propagator

(HPOP). While SGP4 is the propagator optimally compatible with TLE data, it is not well-suited to the inclusion of multiple atmospheric density models for comparison. In addition, it tends to perform worse in the presence of drag [70]. To minimize the error introduced by using TLEs with an alternate propagator, the TLEs used in this thesis are first converted with the SGP4 propagator into a state vector in an Earth-Centered Inertial (ECI) coordinate frame. Then, the initial ECI vector is propagated with HPOP.

The author recognizes that this procedure is not without drawbacks, since the propagator used is not SGP4. However, this analysis is conducted using TLEs in view of the fact that frequently (including in this case), TLEs are the only data source available. The impact of TLE uncertainty on reentry prediction results is explored using Monte Carlo simulation. For further details, see Section 3.5.

2.2.2 Drag Modeling Error

The drag force, that is, the force produced on a satellite by the transfer of momentum from molecules in the atmosphere to the satellite, is the second-largest perturbing force on a near-Earth satellite's orbit, after the gravitational effects produced by Earth's oblateness [106, 104] (p.551). It is also the largest source of uncertainty in determining satellite orbits below about 600-700 km in altitude [56, 57, 44]. Atmospheric drag is calculated with the following equation [104] (p.551):

$$\overrightarrow{a}_{drag} = -\frac{1}{2}\frac{c_D A_{eff}}{m}\rho v_{rel}^2 \frac{\overrightarrow{v}_{rel}}{|v_{rel}|} \tag{2.2}$$

where $\overrightarrow{a}_{drag}$ is the acceleration due to aerodynamic drag, c_d is the dimensionless drag coefficient, A_{eff} is effective cross-sectional frontal area of the satellite, m is the satellite mass, ρ is the local atmospheric density, and v_{rel} is the relative velocity between the satellite and the local atmosphere. A comprehensive review of uncertainty sources in atmospheric drag modeling can be found in Vallado and Finkleman (2010) [106]. Vallado and Finkleman identify 32 potential sources of uncertainty in drag modeling, ranging from uncertainty in the density model (both inherent and in the space weather inputs) to uncertainty in winds, satellite characteristics, and orbit determination/propagation methods [106].

Because it is the largest source of uncertainty for low LEO satellite orbits, drag prediction improvements are critical for satellite conjunction assessment [83] and for reentry predictions, since drag is the primary force that induces reentry. While drag effects can be a limiting factor for low-flying satellites with limited fuel available for orbit maintenance maneuvers, drag can also be leveraged to reduce the proliferation of space debris, for example by equipping cubesats with a drag sail to induce prompt deorbit following mission completion [87, 36]. Whether the goal is to mitigate, maximize, or simply to plan for drag effects, accurate drag prediction is as important as it is challenging.

A dominant source of uncertainty in drag modeling is an error in the neutral atmospheric density, ρ, used to calculate the drag force [26]. Atmospheric density at orbital altitudes varies over ten or more orders of magnitude, and is a complex function of altitude, latitude, local solar time, solar and geomagnetic activity (space weather), and other factors, which makes it challenging to predict. Some of these factors, such as solar and geomagnetic activity, also need to be predicted in order to forecast future atmospheric densities, which adds another layer of potential error. Geomagnetic activity can cause major fluctuations in atmospheric density, leading to faster orbital decay and greater errors in drag predictions. The impact of space weather on satellite reentries is discussed in Section 2.5.

Another significant source of error in atmospheric drag estimation is the ballistic coefficient (BC), a parameter that combines several factors describing how the satellite interacts with the atmosphere. Like atmospheric density, BC has a major impact on drag calculations. For example, reentry simulations of the GOCE satellite found that drag predictions that assumed that the satellite was tumbling were roughly 3 times higher than those assuming a stable rotational position [26], due to the increased BC caused by the tumbling. In a study using only TLE-derived satellite data, Saunders et. al. (2009) also found that BC estimation was “the prime factor in accurately predicting reentry dates” [94]. Because of the techniques used for ballistic coefficient estimation, it is not always possible to separate BC uncertainty from density model uncertainty. BC estimation techniques are discussed in Section 2.3.

Despite the complexity of drag prediction in practice, it is conceptually simple. Drag prediction involves 1) estimating the density of the medium surrounding an object, and 2) describing how that medium interacts with the object [106]. Conveniently, two key error sources in drag prediction fit directly into these categories: atmospheric density modeling, and determination of a satellite’s ballistic coefficient. In the study on reentry predictions of the GOCE satellite, Geul et. al. found that “decay time distribution accuracy. . . is dominated primarily by systemic errors in atmospheric density and attitude control, whereas its precision is dominated by the atmospheric uncertainty” [26]. Attitude control influences drag (and thus reentry time) due to its impact on the ballistic coefficient. In other words, BC error and density error were the primary reasons for incorrect reentry time predictions, while uncertainties in atmospheric density were the largest contributors to the width of the spread of reentry predictions.

Discussion of atmospheric drag error and uncertainty in this thesis is therefore restricted to initial-state TLE, BC, and atmospheric density, due to their large contribution to the total error/uncertainty in drag and reentry prediction. Uncertainty in solar and geomagnetic proxies as inputs to density models is covered as a sub-topic of atmospheric density uncertainty. The three areas of ballistic coefficient, density, and space weather indices are discussed in detail in Sections 2.3, 2.4, and 2.5, respectively.

2.3 Ballistic Coefficient

As previously discussed, atmospheric drag is calculated with Eqn. 2.2 [104] (p.551):

$$\overrightarrow{a}_{drag} = -\frac{1}{2}\frac{c_D A_{eff}}{m}\rho v_{rel}^2 \frac{\overrightarrow{v}_{rel}}{|v_{rel}|}$$

The term $\frac{c_D A_{eff}}{m}$ is collectively known as the ballistic coefficient (BC). It describes how significantly the satellite is impacted by atmospheric drag, based on the physical characteristics of the satellite [104] (p. 552). (Note that some texts define BC as the reciprocal of $\frac{c_D A_{eff}}{m}$ [104] (p.552). In this work, BC is consistently defined as $\frac{c_D A_{eff}}{m}$.) In this definition, m is the mass of the satellite and is the only component of the BC that can often be well-defined based on knowledge of the satellite. c_D is the dimensionless drag coefficient, which encapsulates gas-surface interactions and is influenced by both satellite shape and atmospheric chemical composition [79, 104] (p.551). The minimum physical value for c_D is 2 [95]. The drag coefficient is often approximated as 2.2, but even for uniform spheres at low altitudes (325 km and below), this value can vary by a small amount of about ±5% [36]. At higher altitudes, drag coefficients tend to increase [79]. Theoretical techniques can be used to approximate c_D for satellites, with an estimated uncertainty of ±15% [71]. Finally, A_{eff} is the effective cross-sectional frontal area of the satellite. This is the area of the satellite perpendicular to the relative velocity vector between the satellite and the surrounding atmosphere. As intuition suggests, objects with higher BCs are more impacted by drag than objects with low BCs: an object with a small mass and large area (e.g., a piece of paper) will be "blown around" more than an object with a high mass and small frontal area (such as a glass marble).

The effective frontal area of a satellite can be even more difficult to determine than the drag coefficient. With the exception of uniformly spherical satellites, the effective frontal area varies based on satellite attitude (rotational orientation around the satellite's center of mass). However, satellite rotational attitude is often not precisely known, and can change rapidly, especially for a tumbling satellite. Many satellites tumble during reentry, which makes frontal area determination even more difficult during the reentry process. Rotational attitude can be modeled, for example with a 6DOF propagator such as the one in [26]. However, propagation techniques are limited by the need for an accurate initial rotational state. Additionally, propagation must include torques from the satellite's attitude control system, if present. These also may be difficult to determine, especially if the system is no longer fully functional as the satellite approaches reentry at the end of its lifetime. Even where 6DOF propagation is possible, it may not be practical due to its complexity and computational cost.

Instead of attempting to independently determine c_D, A_{eff}, and m at each point in a satellite's orbit, estimation techniques often model the ballistic coefficient as a single quantity. Because of its impact on the drag force, which is one of the primary orbital perturbations in LEO, estimation of the BC is a vital topic for orbital propagation. In a 2009 assessment of an orbital propagator intended for use in estimating thermospheric density from orbital data, Saunders et. al. found that poor estimation of ballistic coefficients was a significant limiting factor on the propagator's ability to predict reentry [94].

The naive technique for estimating a satellite's ballistic coefficient is to calculate it from the B-star (B^*) drag term contained in publicly-available TLEs. In theory, the ballistic coefficient can be calculated from B^* as follows [37]:

$$B = \frac{2B^*}{\rho_0} \tag{2.3}$$

where B^* is the TLE B-star drag term, B is the ballistic coefficient, and ρ_0 is an atmospheric density reference value defined by [37]:

$$\rho_0 = (2.461 \times 10^{-5}) \times (6378.135)[\frac{kg}{m^2 \cdot Earth\ Radii}] = 0.1570[\frac{kg}{m^2 \cdot Earth\ Radii}] \tag{2.4}$$

Unfortunately, while the nominal definition of B^* allows a simple calculation of the ballistic coefficient, the reality is more complex. According to Vallado (p. 106), B^* is "an arbitrary free parameter in differential correction," meaning that it may be "completely unrelated to drag effects in the presence of... maneuvers" and other orbital perturbations [104]. Because the B^* term in TLEs can even be negative [95, 104], the ballistic coefficients calculated with this technique can be nonphysical. If used to calculate drag acceleration, a nonphysical negative ballistic coefficient value will yield a drag acceleration value that appears to add energy to the satellite's orbit (speeding it up, rather than slowing it down).

Even in cases where the B^* term is consistent with possible physical values, it is not necessarily accurate. Saunders et. al. (2009) found that reentry predictions using ballistic coefficients calculated from the B^* term overestimated the remaining satellite lifetime in all four test cases, indicating that the B^*-derived ballistic coefficients consistently underestimated the true BC [94]. These problems with the TLE drag term make the B^* method generally unreliable for determining accurate ballistic coefficients.

Saunders et. al. (2012) therefore explored an alternative method of deriving ballistic coefficients empirically from TLE data by comparing the change in semimajor axis observed from TLEs with drag-induced change in the semimajor axis predicted by the AETHER orbit propagator, then updating the BC guess using the Secant method and re-running the AETHER propagation until the predicted and observed changes in semimajor axis converged [95]. To minimize error from TLE inaccuracy, Saunders et. al. repeated this process using TLE pairs across the historical record of TLEs for each object, then chose the most commonly predicted value (or mode) as the final ballistic coefficient estimate [95]. Note that this technique can be used either to determine the ballistic coefficient directly, or to determine drag coefficient c_D in cases where satellite frontal area A_{eff} is known and stable, such as for uniform spherical satellites [71]. Saunders' method closely resembles the estimation procedure used by Pardini and Anselmo in [71, 72, 73, 79].

One characteristic of this method is that it tends to absorb atmospheric density error into the BC estimate, since it uses BC as a fit parameter [95]. Pardini and Anselmo leveraged this tendency to reveal characteristics of several density models' performance; however, when trying to determine a physically accurate BC, as in Saunders et. al. (2012), the absorption of density model error into the ballistic coefficient is problematic [71, 72, 73, 79, 95]. In fact, some of Saunders' drag coefficient estimates returned nonphysical values, which were attributed to errors in the NRLMSISE-00 density model used [95]. Saunders et. al. therefore cautioned that, when using BCs obtained with this technique for orbit propagation, the BCs should be paired with the same density model used to generate them [95]. Additionally, this technique was found to be highly influenced by the initial TLE used for the BC estimation, meaning that propagation using these BC estimates should ideally use the same initial state as the BC estimate [31].

Gondelach et. al. (2016) used a modified version of this ballistic coefficient estimation technique to simultaneously estimate BC and solar radiation pressure coefficient (SRPC) from filtered TLEs for reentry prediction of spent rocket bodies with highly-eccentric orbits [31, 29]. The authors found that TLE outliers in perigee radius resulted in BC outliers, meaning that filtering TLEs by perigee radius (or equivalently, by semimajor axis and eccentricity) was important to accurate BC estimation using the modified version of Saunders' technique [29]. Gondelach et. al. concluded that TLE error was a major contributor to reentry prediction and BC estimation accuracy, and that filtering TLEs was important in obtaining accurate results [29]. High-quality TLE-derived BC estimates rely on filtering of TLEs for errors and outliers, such as removing TLEs with a negative (non-physical) B^* term and those that are outliers in inclination, eccentricity, and perigee radius [31, 29, 49, 48]. The importance of filtering procedures was underscored in [49], where Lidtke et. al. found that removing TLE outliers was "vital in order to reduce the prediction errors" for reentry predictions of spent rocket bodies [49]. TLE-based BC estimation may be more effective for circular than for eccentric orbits, since TLE errors exhibit a bias between perigee and apogee [95].

As an alternative to estimating BC from TLE data, a similar technique has been applied to estimate BC from radar tracking data [15, 14]. Like TLE-based estimation, radar-based estimation allows BC to absorb errors from multiple sources. If observations can be made at a sufficient frequency, even relatively sparse radar measurements may yield average BC estimates comparable to those from onboard GPS monitoring [14]. As a development on BC estimation using both radar and TLEs, the authors of [109] propose the use of mean ballistic coefficient over a single orbit for low-eccentricity orbits, as a means of accounting for short-term BC variations due to altitude change. Finally, another recent effort used simulated photometric measurements of non-resolved objects to derive drag coefficients by determining the attitude of rocket bodies with known geometry [64]. This technique holds promise for improving drag estimates for non-resolved objects, which could subsequently be used to calibrate atmospheric models [64].

For the purposes of this thesis, ballistic coefficient estimation is accomplished with two methods. The first is Gondelach's technique of estimating BC by iteratively comparing observed semimajor axis change from filtered TLEs to predicted semimajor axis change using the Accurate Integrator for Debris Analysis (AIDA) propagator [65]. This BC estimation method was accomplished with code provided by David

Gondelach from [31, 65]. The second method involved using BCs estimated by Pardini and Anselmo with the CDFIT tool in [75, 77]. For further discussion of BC estimation and modeling in this thesis, see Section 3.3.3.

2.4 Atmospheric Density Modeling

As briefly discussed in the previous section, one of the complicating factors of ballistic coefficient estimation is uncertainty/error in atmospheric density. Returning again to Equation 2.2, we find that drag acceleration is linearly dependent on both atmospheric density ρ and BC:

$$\overrightarrow{a}_{drag} = -\frac{1}{2}\frac{c_D A_{eff}}{m}\rho v_{rel}^2 \frac{\overrightarrow{v}_{rel}}{|v_{rel}|}$$

Even in cases where drag-induced acceleration can be precisely determined (for example, by subtracting precisely-modeled non-drag perturbations from onboard satellite accelerometer readings), the dependency of drag-induced acceleration on BC and ρ makes it difficult to separate the two. Doing so requires precise knowledge of at least one of the two factors, which is rarely available. For propagation and reentry modeling, the ideal case is a physically accurate BC paired with an accurate atmospheric density model. But due to BC error, it can be difficult to determine the accuracy of a density model. For drag coefficient or BC estimation using the semimajor axis technique described in Section 2.3, an atmospheric density model such as NRLMSISE-00 is used in the propagation process [95]. However, as described by Saunders et. al. (2012), this leads to the absorption of density model error into the BC [95].

To begin a discussion of atmospheric density modeling, let us look at a naive approach to the problem.

The Earth's atmosphere is composed of layers, beginning with the troposphere at low altitudes of 10 km or less, and extending up to the thermosphere at LEO orbital altitudes (from around 100 km to between 500-1000 km) [23]. Although atmospheric density decreases quickly with altitude, most of the thermosphere is still dense enough to have a significant impact on satellite orbits.

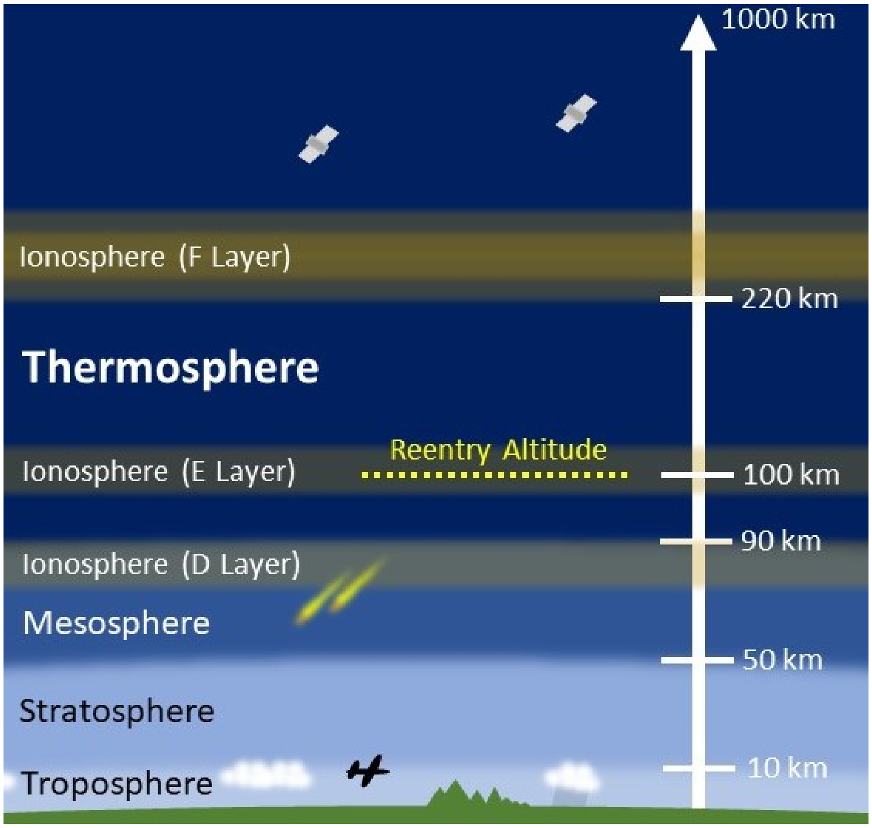


Figure 2-3: Diagram of atmospheric layers [23]. Created with Microsoft PowerPoint.

Density models can be either static (depending only on altitude) or time-varying (depending on time and time-based factors) [104] (p.553). Both types of models rely on underlying physics principles, including the ideal-gas law and the hydrostatic equation, but static models are significantly simpler than their time-varying counterparts [104] (p.553). The simplest density model of all is the static exponentially decaying density model, described by Markley and Crassidis in [58] (p.406) and Vallado [104] (p.565). This model describes density as a simple function of altitude and assumes that density at a given altitude is invariant in longitude, latitude, and time [104] (p.565). The exponential model is described by the following equation [58] (p.406):

$$\rho(h) = \rho_0 \cdot \exp(-\frac{h - h_0}{H}) \tag{2.5}$$

In this equation, ρ_0 and h_0 are reference density and reference altitude, respectively, and H is the scale height. These three quantities are determined from tabulated values available in Table 8-4 in Vallado [104] (p.567) and Table 11.1 in Markley and Crassidis [58] (p.407). Finally, h is the actual altitude above the ellipsoid (above Earth's surface) [104] (p.566). While the exponential model can provide a reasonable and computationally efficient approximation for low-fidelity simulations [58] (p.406), and this model or variations of it are used in some cases such as [18], the static exponentially decaying density model is not suitable for precision applications, because it neglects significant factors including space weather and latitude-, longitude-, and time-dependent dynamics. Other static table-based density models include the 1962 and 1976 Standard Atmosphere models [104] (p.568).

Early attempts at more sophisticated density models include the semi-theoretical Committee on Space Research (COSPAR) International Reference Atmosphere (CIRA) models (including CIRA-65, CIRA-72, and CIRA-90), and the partially static Harris-Priester model, which utilizes interpolation between density tables referenced to different points in the solar cycle and also includes averaged values for the major known variations impacting the thermosphere [104] (pp.568-569). The Jacchia and Jacchia-Roberts models, including J70, J71, and JR-71, are early empirically-based models involving diffusion equation integration [104] (pp.569-571), [58] (p.408). Another early empirical model was the fast-to-evaluate Russian GOST density model, based on data collected from Russian Cosmos satellites [104] (pp.571-572), [58] (pp.408-409). Other early empirical models include the DTM models and the MSIS family of models derived from the DTM [104] (p.571). Note that data-based empirical models are not necessarily based on measured thermospheric density data, but may be derived from related quantities such as temperatures and incoherent radar scatter data

[104] (p.571).

Today, commonly-used models can be classified as empirical or physical/physics-based [20]. Empirical models, like the MSIS family, rely on large data sets and are typically fast to evaluate. Though they lack a framework for the underlying dynamics of the thermosphere, empirical models often include physical constraints [20]. Physics-based models, on the other hand, solve the governing fluid equations to represent atmospheric densities [20] on a grid of points in time and space. Physics-based models may be informed by empirical models, for example by using boundary conditions derived empirically [20]. The knowledge of underlying thermospheric dynamics present in physics-based density models may provide an advantage in density forecasting, as compared to empirical models, but it comes at a high computational cost [59]. Recently, the development of reduced-order density models has offered the possibility of improved forecasting ability at a low computational cost [59]. Reduced-order models are used extensively in this thesis, and are described in 2.4.3. The reduced-order model formulation, training, and implementation are detailed in Sections 3.4.2, 3.4.3, and 3.4.4.

2.4.1 Empirical atmospheric density models

Emmert (2015) provides a description of the three leading empirical density models: Naval Research Laboratory Mass Spectrometer Incoherent Scatter radar Extended (NRLMSISE-00), Drag Temperature Model (DTM-2013), and Jacchia-Bowman (JB2008) [20]. All three of these models are commonly used for satellite drag calculations [51]. The primary differences between these models include their underlying data sets, and the mathematical formulations they use to describe the altitude-dependent change in thermospheric temperature [20]. While these empirical models are fairly accurate at describing regular, large-scale variations on neutral mass density, they may not capture small-scale dynamics with the accuracy desired for drag predictions [51].

The NRLMSISE-00 and JB2008 empirical models are used in this thesis, while the DTM-2013 is not. For a description of DTM-2013, see [20].

The NRLMSISE-00 (or NRLMSIS-00) builds off of the MSIS family of density models (MSIS-86, MSIS-90) by including a wider range of data sets [82]. Like the previous MSIS models, the NRLMSISE-00 uses a set of parametric equations fitted to its underlying data sets to interpolate/extrapolate atmospheric conditions at the time and location of interest [82]. Unlike the earlier MSIS models, however, the NRLMSISE-00 was expanded to include additional data sets of orbital drag and decay data, accelerometer readings, and additional incoherent scatter radar datasets over a longer period of time than any of the former MSIS models, to help account for gradual long-term changes in the thermosphere [82]. These additional datasets made the NRLMSISE-00 competitive with the older Jacchia models, and Picone (2002) found that the NRLMSISE-00 slightly outperformed the MSIS-90 and Jacchia-70 [82]. However, Marcos et. al. (2006) found that the NRLMSISE-00 tends to underestimate densities during some geomagnetic storms by 10-15% [55]. In addition, in a comparison of NRLMSISE-00 predictions against density data obtained from satellite tracking, the standard deviation of the data-to-model ratio for NRLMSISE-00 increased with altitude between 200 and 600 km, topping off at over 0.3 (a 30% standard deviation of model error as compared to satellite tracking densities!) [55].

Various techniques exist for improving NRLMSISE-00 density estimates for drag modeling and reentry prediction applications. For example, Yurasov et. al. (2008) improved reentry time predictions made with NRLMSISE-00 by using TLE-based density corrections [110]. This improved reentry prediction accuracy by 60% or more for spherical objects, and by approximately 32% for non-spherical objects due to their time-varying ballistic coefficients [110]. However, Yurasov's NRLMSISE-00 corrections are limited by the sparsity of TLE data at very low altitudes for computing corrections at these altitudes, and by the inability to calculate density corrections in advance, making real-time predictions with this technique impracticable [110]. Shi et. al. (2015) used TLE data for NRLMSISE-00 calibration at solar maximum, resulting in a lower bias and standard deviation, and RMS improvements of more than 10% at some altitudes [97]. Another development on the NRLMSISE-00 model was the recent release of the new NRLMSISE 2.0, an updated version of the model [107]. However, Virgili et. al. (2021) found that the NRLMSISE 2.0 "does not seem to provide any significant improvement" for reentry predictions, relative to other density models (including the NRLMSISE-00) [107].

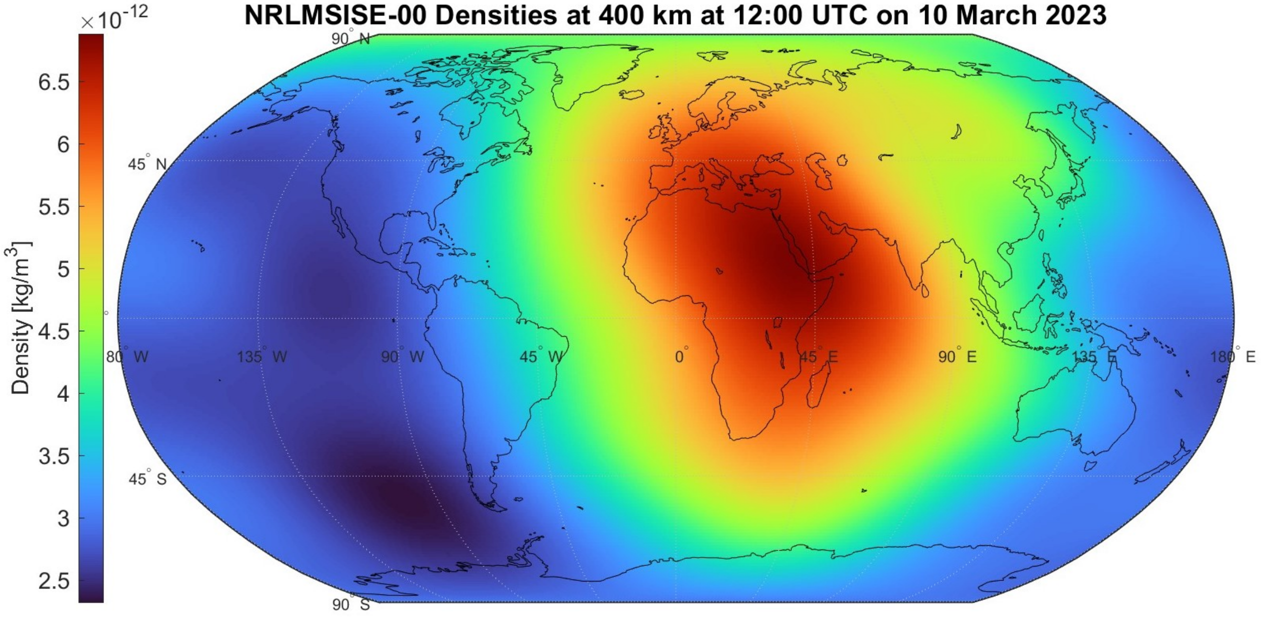


Figure 2-4: Neutral mass densities predicted by NRLMSISE-00 at 400 km altitude at 12:00 UTC on 10 May 2023.

Another empirical model in common use is the JB2008, described in depth by Bowman et. al. [9]. Like NRLMSISE-00, JB2008 is publicly available in several programming languages. JB2008 density modeling is built on four key data sources: orbit-derived densities from the U.S. Air Force (1997-2007), density data from the Air Force's High Accuracy Satellite Drag Model (HASDM) (2001-2005), accelerometer densities from the Challenging Minisatellite Payload (CHAMP) (2001-2005), and accelerometer densities from the Gravity Recovery and Climate Experiment (GRACE)

(2002-2005) [9]. The JB2008 model uses four indices (F10.7, S10.7, M10.7, and Y10.7) as proxies for solar heating, in addition to the 81-day average values of these indices [9]. Disturbance storm time (Dst) index is used to improve JB2008 modeling of geomagnetic storms [9]. Bowman et. al. found that using Dst instead of the a_p magnetic index reduced error during major geomagnetic storms by a significant margin (∼20%) as compared to NRLMSISE-00 [9].

JB2008 also offers modest improvements over NRLMSISE-00 and Jacchia-70 during non-storm times, and overall provides standard deviations around 10% at 400 km in altitude [9]. Marcos et. al. (2010) also found that the standard deviation of the ratio between JB2008 densities and CHAMP density data was slightly lower than for NRLMSISE-00 and J70 [57], meaning that the spread of JB2008 predictions around the measured density values was smaller. However, depending on solar conditions and altitude, JB2008 does not always outperform its counterparts. By comparing drag coefficients fitted using different density models, Pardini and Anselmo (2012) found that at sunspot maximum, JB2008 consistently overestimated average density values by 8-9% at altitudes of ∼275-625 km [78]. In contrast, NRLMSISE-00 overestimated average densities by a similar amount (9%) around 275 km, but this value decreased with increasing altitude, and at 625 km NRLMSISE-00 density bias dropped to nearly zero [78]. This might give the NRLMSISE-00 an advantage in predicting average drag effects at higher altitudes, despite its greater variance at higher altitudes found by Marcos et. al. (2006) [55]. Bruinsma et. al. (2017) found that JB2008 out-performed NRLMSISE-00 for lower altitudes (170-275 km) in a comparison of empirical models to GOCE accelerometer densities, but noted that JB2008 performance suffered at times during low solar activity conditions [11]. Additionally, Arenillas et. al. (2020) found that the NRLMSISE-00 was less computationally expensive to run than JB2008 in a large-scale simulation [6]. As these comparisons show, determining a "best" empirical density model among those readily available is a non-trivial task, and may depend on the altitudes of interest, the solar conditions during the time period of interest, and the computational resources available.

The U.S. Air Force's HASDM model, on which the JB2008 is partially based, provides significantly better density modeling than JB2008, NRLMSISE, and J70 during a range of geomagnetic conditions. HASDM achieved standard deviations consistently below 10%, or about half the standard deviations of the next-best performer, JB2008, in a 2008 comparison by Bowman et. al. [9]. HASDM uses the Dynamic Calibration Atmosphere (DCA) model to provide high-resolution density estimates (and predictions up to 3 days in the future) based on observations of drag effects on a set of "calibration satellites" [98]. HASDM provides significant accuracy benefits over other models [98]. Unfortunately, HASDM is not publicly available, so it cannot be widely used for real-time or predictive density modeling. Archived HASDM density data sets for the years from 2000 to 2019 is available from the SET HASDM Density Database.[8]

2.4.2 Physics-based atmospheric density models

Unlike empirical models, physics-based models are typically not readily available for operational application due to their prohibitively high computational cost. Physics-based models solve fluid equations in three dimensions on grids of varying temporal and spatial resolution [20]. While physics-based models may have some advantage over empirical models for density forecasting [59], this advantage may be limited by the low availability of thermospheric data for model assimilation, and by the dependence of physics-based models on forecasts of solar conditions, which may contain error [20]. Most physics-based models include some assumptions (such as hydrostatic equilibrium) which may not be fully accurate [20, 44]. Additionally, data assimilation into physics-based models is more challenging than for empirical models [57].

[8] https://spacewx.com/hasdm/

Nonetheless, physics-based models are valuable tools for understanding the physical processes of the thermosphere, and are able to simulate conditions not covered by historical data sets, which empirical models cannot do [20]. Physics-based models include the Global Ionosphere-Thermosphere Model (GITM), the National Center for Atmospheric Research (NCAR) Thermosphere-Ionosphere-Electrodynamic General Circulation Model (TIE-GCM), and the National Oceanic and Atmospheric Administration's (NOAA) Whole Atmosphere Model-Ionosphere Plasmasphere Electrodynamics (WAM-IPE) Forecast System (WFS).

The NCAR/TIE-GCM was developed in the early 1990s, based on predecessors dating back to the early 1980s [90]. In addition to densities of the atmosphere's constituent species, the TIE-GCM also models temperature, winds, ion densities, and electron densities on a global grid, using pressure surfaces as its vertical coordinate [90, 86]. TIE-GCM includes electrodynamic interactions and feedback between the thermosphere and ionosphere [57], and has been valuable in ionosphere-thermosphere studies [85]. TIE-GCM testing has shown good general agreement with empirical models [90], and has been validated against data sets including satellite drag data and incoherent scatter radar data [85].

Developed in the mid-2000s, the GITM model diverges from other physics-based models by removing the hydrostatic equilibrium assumption, which allows the study of conditions during which this assumption does not hold [20]. Additionally, the GITM model provides a great deal of flexibility with adjustable resolution, acceptance of several models for electric fields and particle precipitation, multiple options for initial-state specification, and operation in both 1-D and 3-D [91]. GITM was validated against MSIS temperature profiles, International Reference Ionosphere (IRI) electron densities, and ground- and satellite-based measurements of neutral winds [91].

NOAA's WAM-IPE incorporates forcing from solar/geomagnetic activity, as well as from lower atmosphere dynamics [69]. The Whole Atmosphere Model (WAM) component describes temperature, wind, and atmospheric composition, while the coupled Ionosphere Plasmasphere Electrodynamics (IPE) describes the ionosphere [69]. Two-day neutral density forecasts using the WAM-IPE are available to the public from NOAA's Space Weather Prediction Center (SWPC).[9]

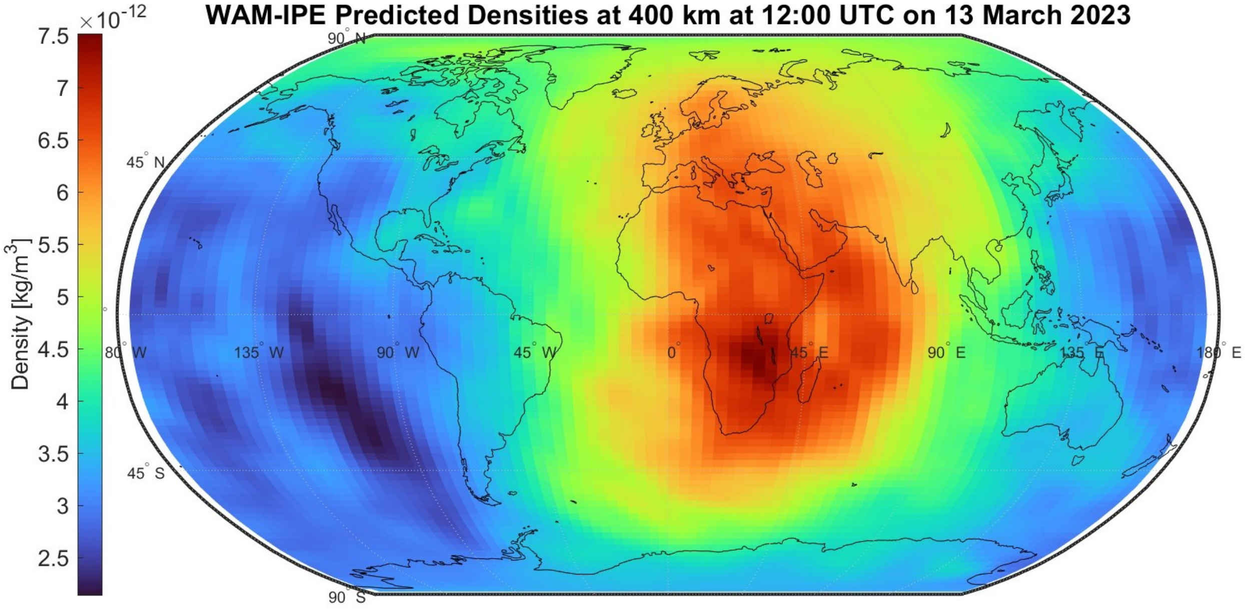


Figure 2-5: Neutral mass densities predicted by WAM-IPE at 400 km altitude at 12:00 UTC on 13 May 2023. Data from NOAA SWPC.[10]

While physics-based models are not the leading choice for satellite drag calculations due to their computational expense, greater requirements for inputs and boundary conditions, and difficulty in assimilating data, their descriptions of thermospheric dynamics show promise for improved density forecasting capabilities for drag modeling. In fact, Marcos et. al. (2010) went so far as to call operational application of physics-based density forecasting "The ultimate goal of satellite drag research" [57]. Physics-based models are already in use for some operational applications, such as the use of GITM in conjunction with analysis for the IMPACT project [44], and the use of WAM-IPE by the Space Weather Forecast Office for alerts/warnings [69].

[9] https://www.swpc.noaa.gov/products/wam-ipe

[10] https://www.swpc.noaa.gov/products/wam-ipe

With further study and development, the role of physics-based models in satellite drag predictions will continue to grow.

2.4.3 Reduced-order atmospheric density models

A newly-emerged branch of atmospheric density modeling is reduced-order modeling, which seeks to combine the low computational cost and easy data assimilation of empirical models with the predictive advantages of physics-based models. The key theme of this thesis is the application of a set of Reduced-Order atmospheric density Models (ROMs) to the problem of satellite reentry prediction. ROMs are therefore discussed more extensively in Section 3.4. A brief conceptual overview is presented here.

The basic idea of reduced-order modeling is to represent a complex, high-dimensional system in a low-dimensional way by identifying underlying basis modes/functions. In 2017, Mehta and Linares used proper orthogonal composition (POD) to decompose thermospheric density snapshots into a series of spatially-dependent basis functions, along with an associated set of time-dependent coefficients [59]. Although the use of a finite number of spatial modes to represent the original data snapshot does introduce truncation error (meaning that the original density data snapshots cannot be perfectly recovered from the reduced-order form), a relatively low number of modes is needed to well capture the system dynamics. Mehta and Linares (2017) demonstrated that using only the first 10 spatial modes identified with POD captured more than 99% of the system's energy [59]. These modes were shown to capture the dominant underlying dynamics of the thermosphere, including diurnal and semidiurnal solar tides. The time-dependent coefficients oscillated with periods reflecting the time scale of those tides (e.g., modes representing diurnal variations had coefficients oscillating with a period of 24 hours) [59].

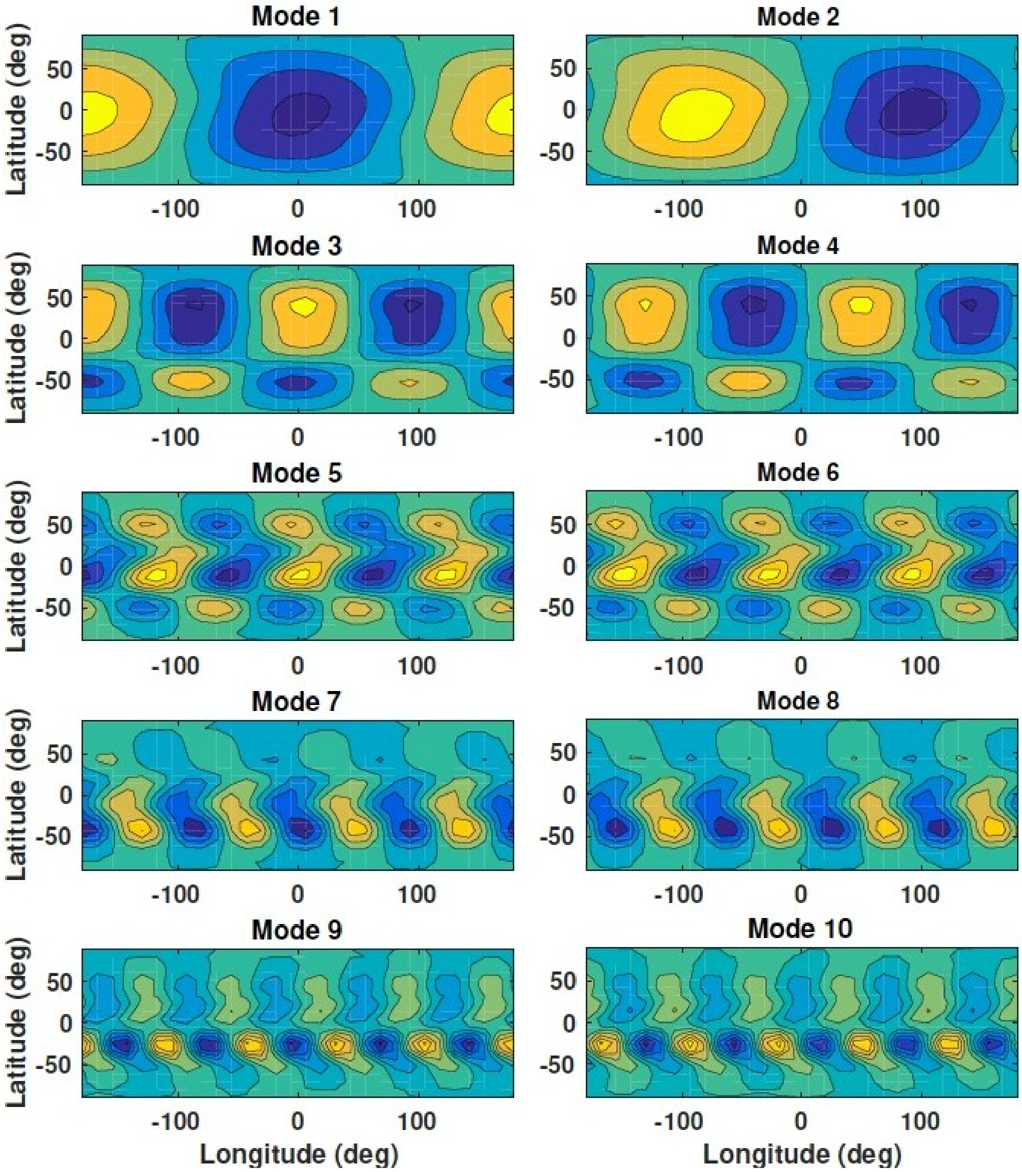


Figure 2-6: First 10 spatial modes identified by POD in NRLMSISE-00 snapshots. Figure reproduced from [59], courtesy of Dr. Richard Linares.

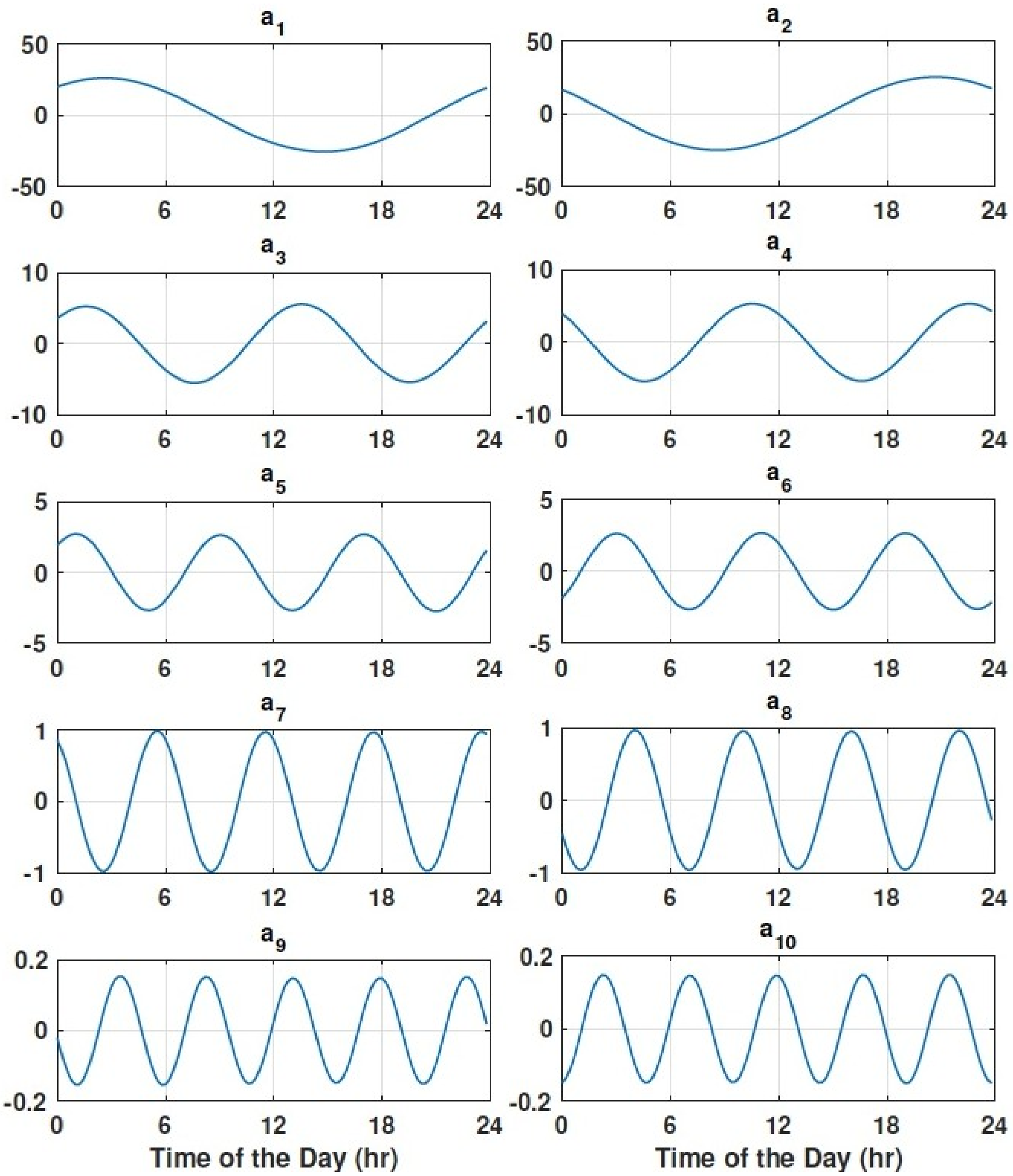


Figure 2-7: Time-dependent coefficients associated with first 10 spatial modes identified by POD in NRLMSISE-00 snapshots. Figure reproduced from [59], courtesy of Dr. Richard Linares.

When used to reconstruct global density snapshots generated by the NRLMSISE-00 model, the POD technique resulted in a mean error of ∼1% relative to the original NRLMSISE-00 snapshots, using 30 spatial modes [59]. It is important to note that all types of input conditions must be included in the snapshots used to construct the POD model to ensure the model's accuracy for any inputs [59]. In 2018, Mehta et. al. applied the POD order-reduction technique to density model snapshots generated with the TIE-GCM physics-based model [62]. In this paper, Mehta et. al. also extended the ROM model by developing a Hermitian Space-Dynamic Mode Decomposition with Control (HS-DMDc) algorithm which they used to propagate the time-dependent ROM coefficients. HS-DMDc accomplishes this by describing the unforced system dynamics with the "dynamic matrix" and including the effects of external inputs (such as space weather) with the "input matrix". This technique allows the ROM to propagate in time its low-dimensional representation of the thermospheric state, giving it predictive capabilities [62]. The TIE-GCM POD ROM can therefore act as a computationally fast substitute for the full TIE-GCM model, capturing most of the TIE-GCM model's dynamics at a small fraction of the cost. Mehta et. al. (2018) found that the TIE-GCM ROM was able to remain within ∼5% of simulated densities over a 24-hour forecast period [62].

Mehta, Linares, and others continued to develop the capabilities of ROMs by calibrating the TIE-GCM ROM via assimilation of accelerometer densities from the CHAMP and GOCE missions with a Kalman filter [60]; extending the ROM technique to modeling atmospheric composition and temperature in addition to neutral density [63]; developing ROMs based on other models, including JB2008 and NRLMSISE-00 [33]; quantifying uncertainty in ROM densities for conjunction assessment/collision probability estimation and orbit predictions [32, 30]; and expanding data assimilation capabilities that allowed for ROM calibration using TLEs, GPS measurements, and radar tracking data [33, 61, 34].

Another vector of development for ROMs has been the use of machine learning (ML) for the encoding and predicting functions of reduced-order modeling. The advantage of the ML approach over POD and DMDc is its ability to capture nonlinear dynamics, which POD/DMDc do not [103]. A deep or convolutional autoencoder can be used to encode density data to a lower dimension, after which prediction is accomplished either using DMDc or using a deep feedforward neural network (NN)

[103]. Turner et. al. (2020) found that the NN predictions outperformed the DMDc predictions, for JB2008 and NRLMSISE-00 density data encoded both with POD and with an autoencoder [103]. ROM techniques can be categorized by 1) the density model on which they are based (NRLMSISE-00, TIE-GCM, JB2008, etc.), 2) the encoding method (POD, autoencoder), and 3) the prediction method (DMDc, NN).

Preliminary application of ROMs to the problem of reentry prediction was conducted in [10]. This thesis expands on that work, using a set of test cases for which true reentry times are available. The reduced-order models used for reentry prediction in this thesis include POD-DMDc ROMs (henceforth "PODROMs") based on NRLMSISE-00, TIE-GCM, and JB2008, and a convolutional neural network (CNN) autoencoder-DMDc ROM (henceforth "ML ROM") based on JB2008. ROMs with NN-based prediction are not included in this thesis. For further details on POD- and ML ROMs, see Section 3.4.

2.5 Space Weather Impact on Satellite Reentry

One of the complicating factors of atmospheric density prediction is the significant impact of space weather on density at orbital altitudes. "Space weather" refers to both solar activity and the solar-driven response of the Earth's magnetosphere, known as geomagnetic activity. Space weather starts with the Sun, which has the largest magnetic field of any body in the Solar System [7]. The plasma composition of the Sun rotates at different rates depending on solar latitude, causing twists in the magnetic field [7]. Energy builds up in these twisted magnetic flux ropes until their magnetic field lines connect with a set of opposing field lines in the process of magnetic reconnection [7]. Magnetic reconnection releases enormous amounts of energy and can cause outflows of coronal mass (Coronal Mass Ejections or CMEs), radiation bursts (solar flares), and high-energy particle releases (Solar Energetic Particles or SEPs)

[7]. Solar activity increases and decreases on an approximately 11-year solar cycle (sometimes called the sunspot cycle) [7]. The number of sunspots on the surface of the Sun is a reliable indicator of the level of solar activity, since these darkened areas of the sun's surface result from the magnetic field in flux ropes inhibiting heat transfer [7]. In addition to the solar energy released during magnetic reconnection, the Sun also outputs a constant flow of conductive plasma and magnetic flux called the "solar wind" [7].

Solar activity is a major driver of thermospheric mass density, since the dominant source of thermospheric heating is extreme ultraviolet (EUV) radiation from the sun [57]. According to Marcos et. al. (2010), solar EUV accounts for 75-80% of thermospheric energy input [57]. In addition, solar activity drives another major thermospheric forcing mechanism: higher solar activity leads to increased frequency and intensity of geomagnetic activity. The solar wind and other releases of solar energy interact with the Earth through the Earth's magnetosphere [7]. One of the main processes is the buildup of energy in the magnetotail (the elongated region of Earth's magnetic field flowing out into space on the side of Earth away from the Sun), followed by magnetic reconnection in the tail, during which field lines in the magnetotail "snap" back together, releasing energy into the inner regions of the Earth's magnetosphere, primarily at high latitudes [7]. Larger geomagnetic events, called geomagnetic storms, are caused by CMEs and other solar events [7].

Geomagnetic activity drives electric fields in the magnetosphere, which produce heat as they dissipate [20]. Combined with the increased flux of energetic particles during higher geomagnetic activity, this heating results in increased neutral mass density [20]. In fact, geomagnetic activity via Joule heating and energetic particle flux is the second-strongest external forcing mechanism acting on the thermosphere, after solar irradiance [86]. Marcos et. al. (2010) estimate that geomagnetic storms are responsible for approximately 20% of thermospheric heating [57], even though high

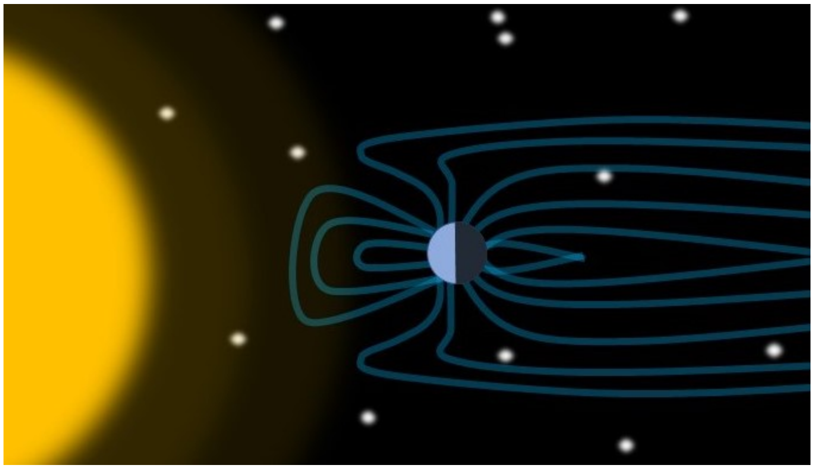


Figure 2-8: Illustration of the Earth's magnetosphere (light blue lines) [57]. Created with Microsoft PowerPoint.

geomagnetic activity is only present about 10% of the time [56]. On very large time and space scales, geomagnetic activity-driven density increases are relatively simple to predict [20]. However, accurate drag predictions require far greater granularity in time and space than can be predicted with general trends.

Atmospheric density models typically take space weather information as inputs in the form of solar and geomagnetic indices, which serve as proxies to represent solar and geomagnetic factors driving density [20]. Commonly used proxy indices include F10.7, the solar radio flux at 10.7 cm, which is used to represent solar extreme ultraviolet irradiance; the geomagnetic disturbance storm time index (Dst), which helps quantify the intensity of geomagnetic storms; and the a_p and k_p geomagnetic indices, which are globally averaged metrics of geomagnetic activity as measured at ground test stations [78, 20, 7]. Some density models, such as JB2008, also use more recently-developed solar indices, including S10.7, M10.7, and Y10.7 [78]. These indices are described in detail in [100].

The use of proxies in density models does introduce error, since proxies are not perfect representations of the driving factors they attempt to describe [20]. This type of error tends to cause underestimation of density response [20]. Since most density models take proxies as inputs, forecasting density into the future requires the prediction of solar/geomagnetic proxies. These predictions are subject to error, which can lead to poor estimates of atmospheric density (and thus of drag).

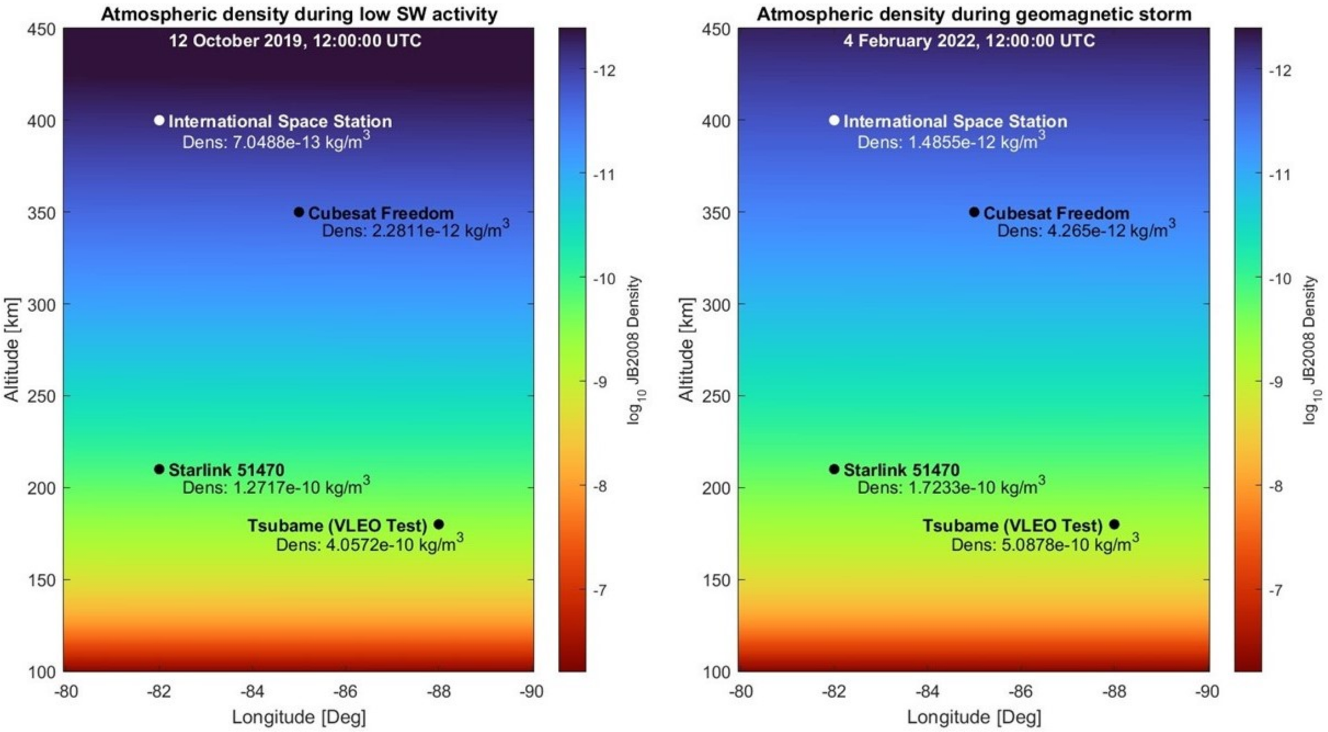


Figure 2-9: Base-10 logarithm of densities computed by JB2008 empirical density model at a fixed latitude of 30 degrees North, at a time of low space weather activity on 12 October 2019 (left) and during a geomagnetic storm on 4 February 2022 (right) Labeled points indicate approximate orbital altitudes of various satellites of interest.

Geomagnetic storms can produce dramatic changes in atmospheric density, resulting in rapid increases in drag experienced by satellites. As an example of the density changes produced during geomagnetic storms, the JB2008 atmospheric density model [53] is used to create Figure 2-9, which shows density on a base-10 logarithmic scale as a function of altitude during a time of low space weather activity (left) and during a geomagnetic storm (right). Labeled points at the approximate altitudes of various satellites of interest show the densities calculated by JB2008 at those altitudes under each condition.

An equivalent representation is the percent change in atmospheric density at each altitude between low space weather (SW) activity as compared to the geomagnetic storm. Figure 2-10 is a plot of the percent increase in density during the geomagnetic storm as compared to the low-SW densities. Each satellite point is labeled with the SW-induced percent increase at that altitude.

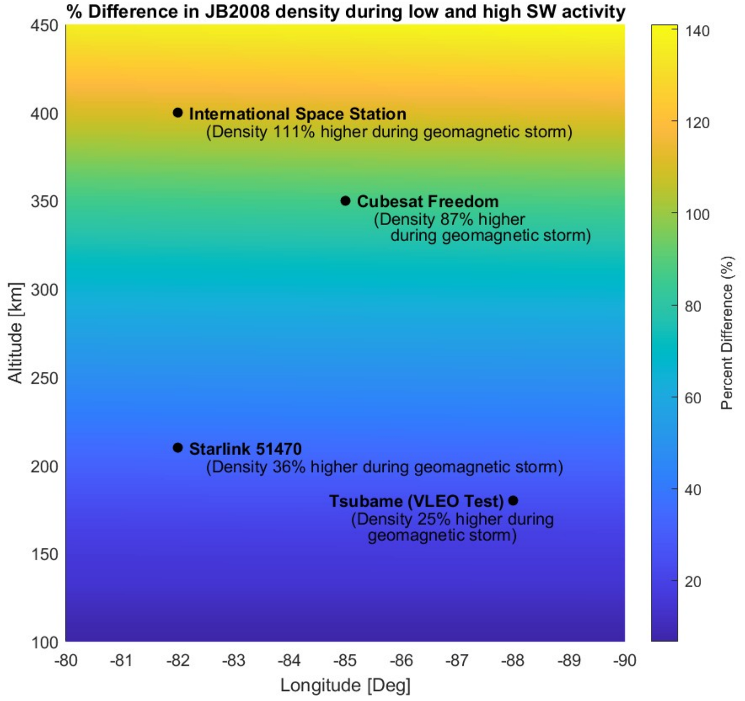


Figure 2-10: Percent increase in densities calculated by JB2008 during a geomagnetic storm, as compared to densities during low SW activity.

As Figure 2-10 shows, thermospheric density changes (such as those caused by space weather activity) become more pronounced at higher altitudes [78], but are still dramatic even at very low altitudes below 300 km. For example, according to the JB2008 model, the Starlink 51470 satellite featured in this plot at its initial perigee altitude of 210 km [21] could expect a 36% increase in density (and consequently in drag force) during a geomagnetic storm such as the two minor storms [88, 21] of 3-5 February, 2022. At its apogee altitude of 350 km [21], the density increase would be higher – almost 90% over non-storm values, according to the JB2008 model!

Such a sudden increase in drag can have especially dire consequences for very low

LEO satellites such as Starlink 51470, which may have little margin for error due to the already-high drag forces on low LEO objects. An additional complication is that the NRLMSISE-00, which is commonly used for operational applications (and is in fact used by the Starlink team), may significantly underestimate the impact of geomagnetic storms on neutral mass density [21]. In a comparison of NRLMSISE-00 densities to those predicted by the physics-based WAM-IPE model, Fang et. al. (2022) found that NRLMSISE-00 showed density enhancements of up to 20% due to the storm. WAM-IPE, on the other hand, demonstrated density increases of 50% or more in most areas [21]. During the geomagnetic storm of 3-5 February discussed by Fang et. al. and represented in Figs. 2-9 and 2-10, the storm-induced drag increase caused a total of 38 Starlink satellites, including Starlink 51470, to prematurely deorbit shortly after launch [88, 21]. With an estimated total cost of $15 million for a Starlink launch [47], the premature reentry of roughly 80% of the 49 satellites on this launch represents a loss of about $10-12 million. Space weather can be expensive!

While the prediction of space weather and space weather proxies is out of the scope of this thesis, it is important to realize that the challenges of space weather prediction can form a barrier to predicting atmospheric density. In recognition of this impact, Monte Carlo simulations are presented in this thesis to demonstrate the sensitivity of reentry predictions, using several density models, to noise in space weather inputs. For further details, see the Section 3.5.3.

2.6 Density Model Comparisons for Reentry

One early work on atmospheric density modeling and uncertainty quantification for reentry prediction was Klinkrad's 1996 benchmarking of Jacchia-71, TD-88, early MSIS versions, and other models against the MSISe-90 density model [45]. The other models were found to have a root mean square (RMS) error of about ±20% as compared to the MSISe-90, while the MSISe-90 was determined to estimate densities to about ±10% during mean solar conditions [45]. Combined with other reentry-associated errors, including spacecraft attitude uncertainty, initial state error, and space weather forecasting error, Klinkrad determined that drag force model uncertainties lead to reentry prediction uncertainties of ±20% of the spacecraft's estimated remaining lifetime [45].

Since Klinkrad's early work, further attention has turned to quantifying density model performance for orbit decay and reentry prediction, with additional regard to density model performance during varying solar/geomagnetic conditions and at varying altitudes. As Pardini and Anselmo found in a series of studies spanning over a decade, no single-density model is necessarily the best in all cases, even when comparisons are restricted to the particular application of reentry prediction [71, 73, 74, 79].

Beginning in the early-2000s, Pardini and Anselmo approached the question of optimal density modeling for reentry applications in a series of studies that leveraged the tendency of drag coefficients and/or BCs to absorb density model error, when they are estimated using a semimajor axis change technique similar to the one refined by Saunders et. al. [95]. The authors analyzed the accuracy of various atmospheric models, developed on distinct data sets, for predicting orbital decay and reentry. These models included MSIS-86, MSISE-90, JR-71, TD-88, and later JB-2006, JB2008, GOST-2004, and NRLMSISE-00 [71, 73, 74, 79, 78].

In a 2001 study, Pardini and Anselmo calculated drag coefficients for a set of nine spherical satellites at varying altitudes and under varying solar activity conditions, using JR-71, TD-88, MSIS-86, and MSIS-90 [71]. In this case, the uniform spherical configuration of the satellites used made a precise determination of frontal area A_{eff} possible, allowing for separation of the drag coefficient c_D from the ballistic coefficient. Drag coefficients were calculated by fitting the semimajor axis decay predicted by NASA Jet Propulsion Laboratory (JPL) Satellite Reentry Analysis Program (SATRAP) software to the actual semimajor axis decay observed in NORAD TLEs [71]. The drag coefficients were fit using each of the density models over time periods capturing a variety of altitude-solar activity combinations. Overall density model behavior during these periods was inferred by comparing the estimated drag coefficients to each other and to a theoretical drag coefficient calculated based on solar activity, altitude, atmospheric composition, and satellite surface composition, with an estimated accuracy of ±15% [71]. Higher drag coefficient estimates indicated that density predictions were low, requiring a higher drag coefficient in order to match SATRAP's orbital decay predictions to the satellite's observed behavior. This study found that for low solar activity and altitudes below 400 km, the MSIS models performed best. However, JR-71 was better for higher altitudes and higher solar activity. Nonetheless, Pardini and Anselmo noted that each of the models showed significant error in some altitude-solar activity regimes [71]. In a similar 2006 study using 11 spherical satellites and the JR-71 and MSIS-90 density models, the authors concluded that "no model considered in this study was able to correctly represent the air density at each altitude and environmental condition" [79].

In 2002, Pardini and Anselmo extended their 2001 density model comparison to reentry prediction for satellites and rocket bodies during high solar activity, using TD-88, JR-71, and MSIS-86 [72]. First, ballistic parameters (defined as $\frac{1}{2}$ BC) were estimated using a procedure similar to the drag coefficient estimation procedure described above. Predicted semimajor axis decay was fit to TLE-observed semimajor axis decay, for each density model and time period, using a least squares approach with ballistic parameter as the fit parameter [72]. From there, SATRAP was used to estimate the remaining lifetime of each satellite starting one month, one week, and one day prior to the nominal reentry times. These simulations were run using "a posteriori" knowledge of true solar and geomagnetic parameters to isolate reentry prediction uncertainty from solar/geomagnetic parameter uncertainty [72]. RMS errors in residual lifetime for the JR-71 and MSIS-86 models were 10-15% one month prior to reentry, and dropped to 8-10% for the cases run one week and one day prior to reentry, while TD-88 predictions remained >15% RMS error for all three cases [72]. A 2004 paper by the same authors conducted an almost identical analysis using a Russian Volstok upper stage as a test case, since this reentry was associated with an IADC reentry test campaign and therefore provided an opportunity for comparison against observed reentry time, rather than estimated nominal reentry times [73]. This study yielded consistent results to its predecessor, with JR-71 and MSIS-86 showing generally good accuracy and agreement, while the TD-88 showed "more irregular behavior and quite large errors" [73].

In a 2008 study on the Cosmos 1025 and EAS reentry campaigns, Pardini and Anselmo compared predictions of satellite residual lifetime during low solar activity using JR-71, MSISE-90, and NRLMSISE-00, with the addition of MSIS-86 for the Cosmos reentry and JB2006 for the Early Ammonia Servicer (EAS) reentry [74]. In the Cosmos case they found that JR-71 was the best predictor of residual lifetime, but only by a small margin, while the JB2006 model showed the lowest mean residual lifetime prediction error in the EAS case [74]. In a later 2012 study of density model performance at sunspot minimum, Pardini and Anselmo compared JR-71, MSISE-90, NRLMSISE-00, GOST-2004, JB2006, and JB2008 behavior using drag coefficients computed based on semimajor axis change as described previously [78]. Below 500 km, all models somewhat overestimated density, with the smallest overestimation biases found in JB2008, NRLMSISE-00, and GOST-2004 [78].

The performance of NRLMSISE-00 for reentry predictions may be improved for some cases, especially for spherical satellites, using density corrections based on observations in the form of TLE data [110]. However, this technique is not applicable in real-time, since the NRLMSISE-00 corrections can only be computed after-the-fact

[110]. The recently-released NRLMSISE 2.0, an update of the NRLMSISE-00, has not been found to show notable reentry prediction improvements [107].

To summarize the work of Pardini and Anselmo, the use of a series of density models for calculating drag parameters and residual satellite lifetime at varying altitudes and solar activity levels indicated no clear density model "winner." While several of the studies found that JR-71 and the MSIS family of models often outperformed other density models, results consistently indicated that no single model exhibited the best performance throughout all altitudes and solar conditions [71, 73, 74, 79]. It is notable, however, that in some cases individual model performance did appear correlated to altitude and solar activity. For example, JR-71 displayed relatively stronger performance at high altitudes and elevated solar activity levels [71]. Ideally, we would like a density model that demonstrates equally strong performance during high and low solar activity, and at a wide range of altitudes.

In this thesis, density model comparisons will be conducted for reentry predictions on a set of test cases. Predictions will be conducted using a posteriori knowledge of the system, including TLEs and true space weather inputs, which are not available for real-time reentry prediction. This allows density model error to be intentionally isolated from other uncertainty sources, such as space weather indices and ballistic coefficient evolution. Uncertainty quantification is conducted separately for TLE initial-state, BC, and space weather uncertainty sources.

2.7 Thesis Description

This thesis expands upon preliminary reentry analysis using reduced-order atmospheric density models, which was conducted in [10]. Three satellite test cases with known reentry times are propagated over time periods ranging from 7 days to 12 hours prior to reentry using the MATLAB HPOP propagator [52] paired with a series of atmospheric density models including two commonly-used empirical models (NRLMSISE-00 and JB2008) as well as three PODROMs (based on NRLMSISE-00, TIE-GCM, and JB2008) and one ML ROM (based on JB2008). Reentry predictions are restricted to predictions of the reentry epoch, and reentry footprints are not treated.

Initial satellite states are generated by using SGP4 to derive an ECI state from a TLE. Satellite ballistic coefficient models are based on BCs estimated using Gondelach's modified version of Saunders' technique [31, 29, 95], where TLEs are filtered prior to their use in estimating BCs by the method of observed versus predicted semimajor axis change. To isolate inherent density model performance from other sources of uncertainty where possible, true space weather inputs and TLEs are used for reentry prediction simulations.

To better evaluate the true uncertainties in the reentry epoch predictions provided, UQ is conducted using Monte Carlo simulation for TLE initial-state and space weather input uncertainty sources. Initial-state uncertainty values are obtained from [22], and space weather uncertainty values are derived from [100]. The effect of ballistic coefficient uncertainty is examined by varying the BC by ±25%. Other sources of uncertainty are not treated in this thesis.

The unique contribution of this thesis is the extensive application of reduced-order models to orbital reentry prediction of test cases with known reentry times, and the comparison of their performance on these test cases against the commonly-used empirical density models NRLMSISE-00 and JB2008. In addition, UQ for initial-state TLE, BC, and space weather explores the impact of uncertainty sources other than inherent density model uncertainty on reentry predictions and density model behavior.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Methods

This chapter discusses the methods and techniques applied in this thesis. Special attention is devoted to orbital propagation theory in Section 3.2.2 and to reduced-order density modeling theory, which is outlined in full in Section 3.4.2. UQ methods are here described last in Section 3.5. Since UQ methods involve strategic injection of error into portions of the reentry prediction algorithm, understanding the UQ methods requires prerequisite knowledge of the propagation and ROM techniques discussed in Sections 3.2 through 3.4.

3.1 Hardware, Language, and Software Packages

The BC estimation process, some TLE processing, and plotting of results were conducted on an HP Elitebook laptop with an Intel® Core™ i7-8650U (8th generation) CPU and 16.0 GB of random access memory (RAM). All simulations and other processing work were run on the MIT Supercloud supercomputer [89]. Some ML ROM simulations were run on the Supercloud using 8 cores, and all other simulations were run with 1 core.

The programming languages used, in order of increasing use, were Python, MATLAB, and Julia. The Julia Programming Language was used for the main simulation code and most subsidiary functions. Information about the Julia Programming Language can be found on the Julia web page[1]. A few data processing functions were left in MATLAB and called using the MATLAB.jl[2] and MAT.jl[3] packages, but all HPOP functions called inside of the propagation routine were translated into Julia to avoid the computational overhead of repeatedly opening and closing MATLAB engine sessions.

BC estimation and some TLE processing were conducted using MATLAB R2017b. All plots were generated using MATLAB 2022a. The ML ROM model used is from the Dynamic Data-Driven Thermospheric Density Estimation with Quantified Uncertainties (DESQU) Toolbox, which is available in Python[4]. The Python version of the ML ROM was called into the Julia-based propagator using the PyCall.jl package.

A summary of the Julia packages used in this project is shown in Table 3.1.

Table 3.1: Complete list of Julia packages used.

AstroTime.jl
Dates.jl
DelimitedFiles.jl
DifferentialEquations.jl
Distributions.jl
HTTP.jl
Interpolations.jl
JLD2.jl
LazyGrids.jl
LinearAlgebra.jl
MAT.jl
MATLAB.jl
NPZ.jl
Plots.jl
PyCall.jl
Random.jl
SatelliteDynamics.jl
SatelliteToolbox.jl
SPICE.jl
Statistics.jl
URIs.jl

[1] https://julialang.org/

[2] https://github.com/JuliaInterop/MATLAB.jl

[3] https://github.com/JuliaIO/MAT.jl

[4] https://github.com/SWQU/DESQU

3.2 Orbit Propagation

3.2.1 Description of the orbital propagator and code structure

The orbital propagator used in this thesis was a modified version of the High-Precision Orbital Propagator (HPOP) v2.2.1 written in MATLAB by Meysam Mahooti [52]. The latest version of the HPOP may be found on Mathworks File Exchange[5]. This algorithm uses the special perturbations approach to propagate an ECI satellite state vector over a specified period of time, using an initial state and satellite physical parameters supplied by the user. The following modifications were made to the original HPOP v2.2.1.

1. The HPOP propagator was translated into the Julia Programming Language.

2. The gravity model used was changed to the Earth Gravitational Model 2008 (EGM2008), with degree 10 and order 10. This change was made to align the HPOP propagator as closely as possible to the BC estimation propagator.

3. A deorbit condition was added to halt propagation when the object's geocentric altitude dropped below 100 km.

4. The integrator was changed to the Julia language's VCABM propagator from the package DifferentialEquations.jl. The radau integrator used in the original HPOP is available in Julia, but in a more primitive form which did not allow for the deorbit condition (which is necessary to prevent the propagator from attempting to propagate the satellite state beyond reentry). Additionally, according to documentation for the DifferentialEquations package,[6] VCABM is

[5] https://www.mathworks.com/matlabcentral/fileexchange/55167-high-precision-orbit-propagator

[6] https://docs.sciml.ai/DiffEqDocs/stable/solvers/ode_solve/

the most equivalent integrator to MATLAB's ode113, which was substituted for the original radau integrator in previous ROM reentry prediction work [10]. ode113 was also the integrator used during propagation for the BC estimation process. Both absolute and relative tolerance of VCABM was set to 1e-8.

5. Additional atmospheric density model options were added to the HPOP, including several PODROM options and a ML ROM option.

The HPOP configuration is flexible, with options to select the degree and order of the gravity model, which third-body effects to include, and whether to use solid Earth and/or ocean tides. Individual perturbations can be turned on or off in HPOP, including drag, solar radiation pressure, and relativity. The perturbations used in the HPOP propagation were matched as closely as possible to the perturbations used in the AIDA propagator during the BC estimation process, which is discussed in Section 3.3.3. The following table shows a comparison of the propagator configurations used in the reentry prediction and BC estimation processes.

Table 3.2: Comparison of HPOP and AIDA propagator configurations.

Parameter	**HPOP (Reentry)**	**AIDA (BC Estimation)**
Language	Julia	MATLAB R2017b
Integrator	VCABM	ode113
Gravitational Harmonics Model	EGM2008	EGM2008
Gravitational Harmonics Degree	10	10
Gravitational Harmonics Order	10	10
Solid Earth Tides Used	No	No
Ocean Tides Used	No	No
Third Body Effects	Sun, Moon	Sun, Moon
SRP Used	Yes	Yes
SRP Shadow Model	Geometric	Biconical
Relativity Used	No	No
Atmospheric Drag Model	Rotating Atmosphere	Rotating Atmosphere
Atmospheric Density Model	Various	NRLMSISE-00

The HPOP orbital propagator comprised the bulk of the reentry prediction process, with supporting scripts used to prepare HPOP inputs, call density models, and process/plot HPOP output. Figure 3-1 shows a simplified structure of the code setup used for reentry prediction. Monte Carlo simulations used the same basic setup, except that the added function of randomized uncertainty value generation, and the propagation process, were repeated in a loop for a set number of iterations.

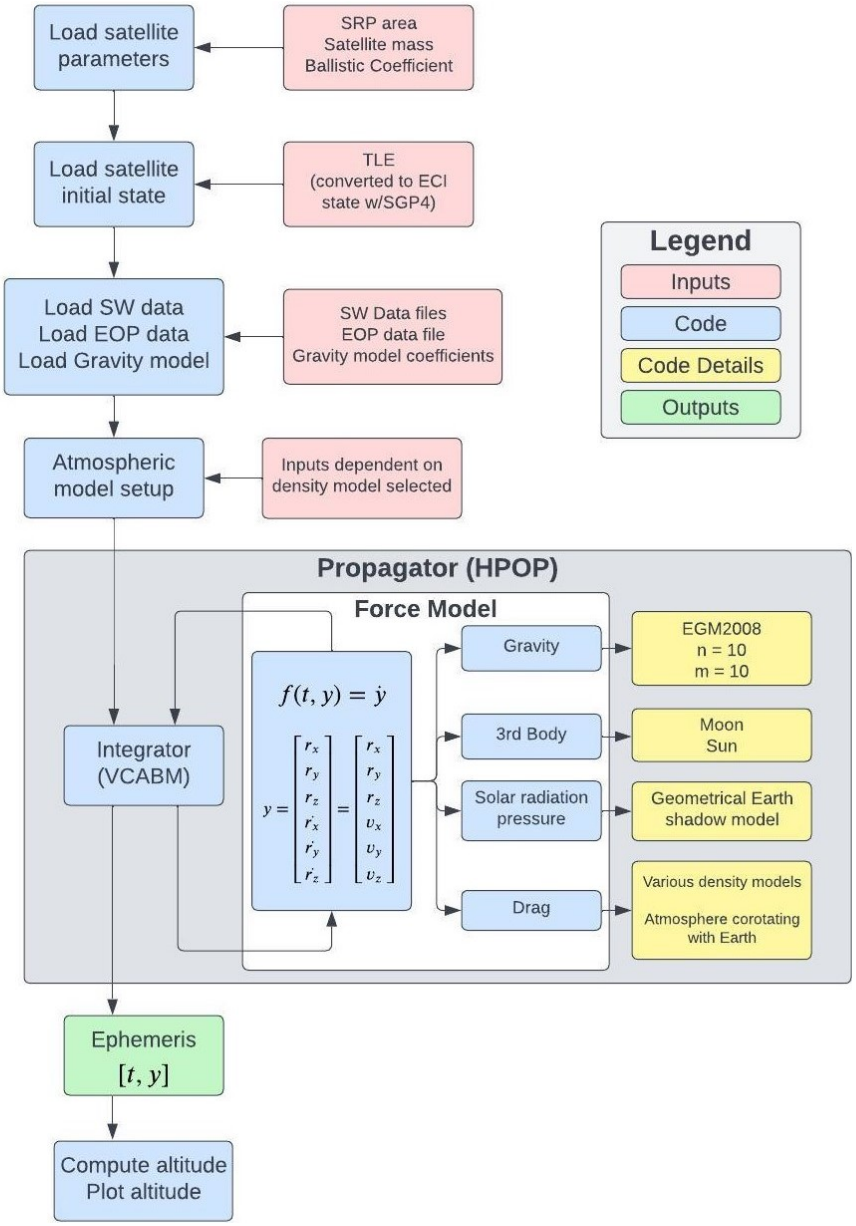

Load satellite parameters
SRP area
Satellite mass
Ballistic Coefficient
Load satellite initial state
TLE
(converted to ECI state w/SGP4)
Legend
Inputs
Code
Code Details
Outputs
Load SW data
Load EOP data
Load Gravity model
SW Data files
EOP data file
Gravity model coefficients
Atmospheric model setup
Inputs dependent on density model selected
Propagator (HPOP)
Force Model
$f(t, y) = \dot{y}$
$y = \begin{bmatrix} r_x \\ r_y \\ r_z \\ \dot{r_x} \\ \dot{r_y} \\ \dot{r_z} \end{bmatrix} = \begin{bmatrix} r_x \\ r_y \\ r_z \\ v_x \\ v_y \\ v_z \end{bmatrix}$
Integrator (VCABM)
Gravity
EGM2008
n = 10
m = 10
3rd Body
Moon
Sun
Solar radiation pressure
Geometrical Earth shadow model
Drag
Various density models
Atmosphere corotating with Earth
Ephemeris
$[t, y]$
Compute altitude
Plot altitude

Figure 3-1: Simplified code structure diagram.

3.2.2 Orbit Propagation Theory

Orbit propagation using the special perturbations technique models the state vector y of an RSO by solving an initial value ODE problem described as follows.

Given:

$f(t, y) = \dot{y}$ and $y(t_0) = y_0$

Find:

$y(t)$ for $t \in [t_0, t_f]$

$$where\ y = \begin{pmatrix} r_x \\ r_y \\ r_z \\ \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \end{pmatrix} \ and\ \dot{y} = \begin{pmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \\ \ddot{r}_x \\ \ddot{r}_y \\ \ddot{r}_z \end{pmatrix}$$

$$with\ \dot{r} = \frac{dr}{dt}\ and\ \ddot{r} = \frac{d^2r}{dt^2}$$

The ODE is solved by numerically integrating the satellite force model. First, a force model is constructed to describe the forces acting on the RSO, from which its instantaneous acceleration can be computed using Newton's Second Law. An integrator (also called an ODE solver) is then used to propagate the RSO state vector over a period of time. For the specific application of reentry modeling, propagation is continued until the satellite's altitude drops below a predetermined altitude, often 80-100 km.

The force model for an Earth-orbiting satellite consists of two primary types of forces: 1) the force of gravity exerted by the Earth as described by the Two-Body Equation (2BE), and 2) orbital perturbations. Earth's gravitational force is the dominant force on an Earth-orbiting RSO. Earth's gravity is approximated with the 2BE (for a complete description, see [104] pp. 20-33). Orbital perturbations (forces that "perturb" the satellite from the idealized orbit described by the 2BE) include gravitational effects as well, by stripping away the 2BE assumptions and including gravitational effects caused by the Earth's oblateness (non-spherical shape). Orbital perturbations also include other forces, such as drag, solar radiation pressure, third-body effects from the sun, moon, and planets, relativistic effects, and more.

Numerically integrating the satellite's force model is computationally expensive, especially when the prediction window is many hours to days and the desired precision is relatively high. One way to reduce the computational cost of the simulation is to truncate the number of perturbation forces included in the force model, at the cost of slightly lower precision. Depending on the precision needed in the orbital propagation, the force model may include many perturbations or only a few dominant ones.

The satellite's orbital altitude determines in part which perturbations must be included, and which can be disregarded without introducing significant error. For example, in LEO (at ~1000km or less) atmospheric drag is one of the dominant orbital perturbations [106] [104] (p.551). At geosynchronous (GEO) orbits altitudes of about 35,780 km, drag is insignificant, while other perturbations, such as third-body effects, become more important [104] (p.31). Only LEO orbits are treated in this thesis, so atmospheric drag is always relevant. When drag is considered, another way of minimizing computational cost is to choose an efficient, fast-evaluating air density model, such as a reduced-order model.

Idealized Gravity

By far the strongest force acting on a LEO satellite is Earth's gravity. Acceleration due to Earth's gravity is described by the 2BE [104] (p.23):

$$\ddot{\vec{r}} = -\frac{Gm_{Earth}}{r^2}\frac{\vec{r}}{r} = -\frac{\mu}{r^2}\frac{\vec{r}}{r} \tag{3.1}$$

where G is the gravitational constant, m_{Earth} is the mass of the Earth, $\overrightarrow{r}$ is the position vector of the satellite, and μ is Earth's gravitational parameter (the quantity Gm_{Earth}). Technically, this equation describes the relative motion between the Earth and the satellite (where the satellite's mass is ignored, since it is negligible in comparison to the Earth's mass) [104] (p.23). However, choosing a coordinate frame with its origin at the Earth's center allows the equation above to describe the satellite's motion with respect to the origin of the coordinate frame.

The Two-Body Equation introduces several assumptions: First, that the satellite's mass is negligible; second, that the coordinate frame chosen is inertial (which is not quite true, since the Earth orbits the sun, but is a good enough approximation); and third, that the Earth and the satellite are perfect spheres of uniform density, which can be treated as point masses [104] (p.23). This third assumption is inaccurate, since the Earth is neither perfectly spherical nor uniformly dense. Usually, the satellite is also non-spherical and non-uniform in density. However, if we are satisfied with describing the translational motion of the satellite's center of mass (rather than also considering its rotational state), then for the purposes of the two-body equation, we can assume that the satellite is a uniform sphere.

Orbital Perturbations

Non-spherical Earth gravitational effects do have a significant impact on satellite orbits, so these effects are included in the orbital perturbations described by the force model. The EGM2008 with degree and order 10 was used to estimate non-spherical Earth gravitational effects in this work. The EGM2008 is described in [80].

The second orbital perturbation type included in the force model is the third-body effects. While the Earth's gravity is the dominant gravitational force on an Earth-orbiting satellite, the satellite does experience small gravitational accelerations from other nearby bodies, such as the sun, moon, and planets. Each third-body gravitational force is approximated using a point mass approximation, which assumes that the third body producing the gravitational force is a point mass (or equivalently, a uniform spherical mass). The satellite's mass is neglected. Third-body gravitational effects can be described with the following equation from Vallado [104] (p.574):

$$\ddot{\vec{r}} = -GM_3 \left(\frac{\vec{r} - \vec{r_3}}{|\vec{r} - \vec{r_3}|^3} + \frac{\vec{r_3}}{|\vec{r_3}|^3} \right) \tag{3.2}$$

where M_3 is the mass of the third body, $\vec{r}$ is the position vector of the satellite, and $\vec{r_3}$ is the position vector of the third body. The term $\vec{r} - \vec{r_3}$ gives the vector from the third-body center of mass to the satellite center of mass. This equation can be applied to any third-body mass, given the relevant third-body mass and position in the applicable reference frame. The point-mass equation is used to calculate third-body gravitational effects for the sun and the moon. Third-body effects from planets are not included in the propagator setup for this thesis.

Solar radiation pressure (SRP) is the force due to the momentum of solar radiation incident on the satellite surface. SRP is stronger at higher altitudes and is always directed away from the sun [104] (p.578). At the very low altitudes considered in this thesis, SRP is small, but it is included for completeness. Consideration of SRP must account for the absence of this force when the satellite is in Earth's shadow (in eclipse), during which it is not exposed to incoming solar radiation. When the satellite is not in eclipse, the SRP force upon it at a distance of 1 AU from the sun, assuming a 0-degree angle of incidence, is given by the following equation [104] (p.580):

$$\ddot{\overrightarrow{r_{srp}}} = p_{srp} c_R \frac{A_\odot}{m} \frac{\vec{r_\odot}}{|\vec{r_\odot}|} \tag{3.3}$$

where p_{srp} is the solar pressure at 1 AU, c_R is the satellite reflectivity, $A_\odot$ is the satellite area facing the sun, $\vec{r_\odot}$ is the vector from the sun to the satellite, and m is the satellite mass. A satellite's reflectivity c_R is the proportion of solar radiation momentum absorbed by the satellite, ranging from 0 (a transparent satellite) to 2 (perfectly reflective satellite) [104] (p.580).

In HPOP, Eqn. 3.3 is modified to account for Earth shadow transition effects and the precise distance between the satellite and the sun (rather than approximating the distance as 1 AU at all times). A geometrical shadow model is used, with an eclipse factor that adjusts SRP by a factor ν. This factor represents the fraction of the solar disk visible from the spacecraft's position, so ν is 1 when the satellite is in the sun, 0 when it is in shadow, and between 0 and 1 during transitions between the sun and eclipse.

Solar radiation falls off as the surface area of a sphere, that is, proportionally to the inverse of the squared distance between the sun and the satellite. This value is converted into AU, giving a factor of $(\frac{[meters\ per\ AU]}{r[m]})^2$. Including this modification and the eclipse factor ν, the equation used to calculate SRP acceleration in HPOP is:

$$\overrightarrow{\ddot{r}_{srp}} = \nu p_{srp} c_R \frac{A_\odot}{m} \left[\text{meters/AU}\right]^2 \frac{\overrightarrow{r_\odot}}{|\overrightarrow{r_\odot}|^3} \tag{3.4}$$

where meters/AU is the conversion factor from meters to AU, $1.496 \times 10^{11}\text{m} = 1\ AU$.

The final orbital perturbation modeled in this thesis is atmospheric drag, which is the non-conservative force caused by the transfer of momentum from air particles to the satellite. This force was discussed in depth in Chapter 2. The equation for atmospheric drag is Equation 2.2 [104] (p.551):

$$\overrightarrow{a}_{drag} = -\frac{1}{2}\frac{c_D A_{eff}}{m}\rho v_{rel}^2 \frac{\overrightarrow{v}_{rel}}{|v_{rel}|}$$

Atmospheric drag is one of the largest perturbations for satellites in low Earth orbit, and it is the force primarily responsible for satellite reentry. Accurate drag predictions rely on a good estimate of the satellite's ballistic coefficient, and an accurate atmospheric density model. High-altitude winds are not included when calculating

v_{rel} in the HPOP. The atmosphere is assumed to co-rotate with the Earth.

3.3 Test Cases and Object Parameters

3.3.1 Test Cases and Basic Parameters

A total of three test cases are selected for analysis. These cases are NASA's Upper Atmosphere Research Satellite (UARS), Roscosmos' Fobos-Grunt probe (also spelled Phobos-Grunt), and the Roscosmos Progress-M 27M cargo vessel. These spacecraft are good candidates for test cases due to the IADC reentry campaigns conducted for all three objects, and the thorough reentry prediction analyses of these cases in [5, 75, 38, 77]. Table 3.3 summarizes the test case physical parameters. Individual test cases and parameters are discussed further in the following sections.

Table 3.3: Spacecraft test case physical parameters.

Test Case	**UARS**	**Fobos-Grunt**	**Progress-M 27M**
NORAD ID	21701	37872	40619
Mass [kg]	5668	13525	7289
A_{eff} **(Drag) [m^2]**	21.16	23.76	7.398
A_{eff} **(SRP) [m^2]**	21.16	23.76	7.398
c_d	2.2	2.2	2.2
c_R	1.0	1.0	1.0
Reentry Date	24-Sep-2011	15-Jan-2012	08-May-2015
Reentry Time (UTC)	04:00 ± 1 min	17:46 ± 1 min	02:20 ± 1 min

Upper Atmosphere Research Satellite (UARS)

The UARS satellite's 14-year mission to study the chemical composition of the atmosphere commenced in 1991 with its release from the shuttle Discovery [75]. Its mission orbit was circular at an altitude of 580 km, which was lowered via several maneuvers after mission completion in 2005, leaving the fuel tanks empty [75]. Maneuvers were completed in late 2005, after which the UARS satellite's orbit decayed under the influence of drag until its reentry on 24 September 2011 at 04:00 UTC ±1 minute [38].

Pardini and Anselmo identify the UARS satellite's dry mass and dimensions as 5668 kg and 4.6 m diameter with a length of 9.7 m [75]. From these dimensions, a drag area estimate of $(4.6 \text{ m})^2$ is chosen to calculate an initial ballistic coefficient starting guess. The ballistic coefficient estimation process (see Methods Section 3.3.3) is iterative, so the initial guess has minimal impact on the final BC estimate. Due to a lack of information on the spacecraft attitude, the same area estimate is used for the effective SRP area. No information on drag coefficient c_d or reflectivity c_R is available, so default values of 2.2 and 1.0, respectively, are chosen. Like the effective drag area, the estimated c_d is used only for calculating an initial BC guess for the iterative estimation process. Inaccuracies in the SRP area and reflectivity have minimal effect on the reentry predictions, since in the LEO regime, the drag perturbation is overwhelmingly larger than SRP.

Fobos-Grunt

The Fobos-Grunt spacecraft was a failed Mars moon probe launched on 9 November 2011 into an initial 208 × 344 km orbit [75]. Unfortunately, Roscosmos quickly lost control of the probe, possibly due to corruption of its computer processors, and the probe never departed its Earth orbit as planned [75]. The probe continued a series of small maneuvers, with solar panels deployed, until 22 November, after which its orbit decayed under the influence of drag and other natural perturbations until its reentry on 15 January 2012 at 17:46 UTC ±1 minute, according to the Joint Space Operations Center (JSpOC) [75]. Decay altitude was recorded as 80 km [75]; however, a decay altitude of 100 km was used in this thesis, since several of the density models used are valid only down to 100 km in altitude. Because orbital decay is very rapid at such low altitudes during the final phase of reentry, the error introduced by the reentry altitude difference is not here considered.

An IADC reentry campaign was conducted for the Fobos-Grunt probe due to its very large mass of 13,525 kg and the large quantity of highly toxic propellants it carried. Pardini and Anselmo give the probe diameter as 3.76×3.76×6.32 m with stowed solar panels, and 3.76×7.97×6.32 m with the panels deployed [75]. An effective area estimate of (3.76 m × 6.32 m) was used for both the initial BC estimate and the SRP area estimate, but again, the exact estimate of this area is not considered significant, since the initial BC guess has minimal effect on the final output of the iterative estimation process, and SRP is small. As for the UARS satellite, c_d and c_R are chosen as 2.2 and 1.0, respectively.

Progress-M 27M

The Progress-M 27M was a failed Roscosmos cargo mission to the ISS, which launched in April 2015 but quickly lost communications [77]. Its initial orbit of 189 × 269 km was very low, and as in the case of Fobos-Grunt probe, Progress-M 27M had a large mass and quantity of toxic propellant [77]. An IADC campaign was launched due to these factors, with the expectation that Progress-M 27M would deorbit within 2 weeks [77]. JSpOC recorded the reentry of Progress-M 27M at 02:20 UTC ±1 minute on 8 May 2015, at an altitude of 80 km [77].

Pardini and Anselmo record the cargo vessel's dimensions as 2.72 m in diameter and 7.23 m in length, with a mass of 7289 kg [77]. In this thesis, the assumed effective area for SRP and drag was $(2.72 \text{ m})^2$, the drag coefficient was 2.2, and the reflectivity was 1.0.

3.3.2 Initial States

Initial states for each reentry test case were derived from TLEs. As discussed in Section 2.2.1, TLEs are known to be subject to error and even mislabeling with the wrong object ID. Unfortunately, no other initial-state data sources were available for the test-case objects. For each test object, TLEs prior to reentry were processed using TLE handling code from BC estimation setup code by Gondelach et. al. (2017) [31], in addition to Beck and Vallado's MATLAB version of the SGP4 propagator. BC estimation code by Gondelach et. al. is not publicly available, but the SGP4 propagator code may be downloaded from Github.[7] For each test case simulation, the following initial state processing steps were used.

1. Read TLEs into a data structure for SGP4 input, including calculating the Julian date for each TLE.

2. Using the Julian dates, identify the TLE closest in time to the Julian date of the desired start time $JD_{start} = JD_{reentry} - L_{sim}$, where L_{sim} is the desired length of the simulation in days. $JD_{reentry}$ is the true reentry time, such that the simulation starts as close as possible to L_{sim} days prior to the true reentry, depending on the times of TLEs that are available.

3. Convert the initial-state TLE identified in step 2 to an ECI state vector $[\overrightarrow{r}, \overrightarrow{v}] = [r_x, r_y, r_z, v_x, v_y, v_z]$ using the SGP4 propagator.

Due to non-uniform distribution of TLEs in time, the durations of the simulations across all five test cases varied somewhat from the nominal simulation durations. Results are plotted against time using true simulation lengths, not nominal lengths, but the nominal-length notation is retained for convenience.

[7] https://github.com/shupp/sgp4

3.3.3 Ballistic Coefficient Estimation and Modeling

Several different ballistic coefficient estimates were used for the various test cases. The first method was the method developed by Saunders et. al. (2012) and Gondelach et. al. (2017) based on comparing semimajor axis change observed from TLEs with predicted semimajor axis change from an orbital propagator [31, 29, 95]. In this technique, henceforth the “Saunders-Gondelach" (SG) method, TLEs are first filtered to remove excessive time gaps, outliers, inconsistent or corrected TLEs, and TLEs with a nonphysical drag term, as described in [31, 49, 10]. Then, the change in the semimajor axis between a pair of TLEs is calculated, and compared to the change in the semimajor axis predicted by the high-fidelity Accurate Integrator for Debris Analysis (AIDA) propagator [65] with an initial BC estimate calculated from the estimated mass and frontal area of the object, with an assumed drag coefficient of 2.2. The following filter parameters were used for TLE filtering for all test cases. Note that a TLE sequence is "a set of TLEs that does not contain any events, which might change object’s physical properties" [49]. For details on TLE filtering, see [49].

Table 3.4: TLE filter settings used in Saunders-Gondelach BC estimation [31, 49]. Values and descriptions derived from code by David Gondelach and Aleksander Lidtke (2015).

Parameter	**Threshold Value**	**Description**
Minimum time gap	½ orbital period	Minimum allowed time between TLEs. Earlier TLE thrown out if violated.
Time gap percentile	0.9	Percentile of TLE time gaps in which to search for start of distribution tail.
Time gap threshold	3	Within above percentile, number of median absolute deviations of TLE time gap to use as distribution tail threshold.
Window length	3	Number of TLEs in sliding window.
Relative tolerance	0.6	Threshold relative difference between actual and expected change between TLEs. TLE w/higher relative difference is an outlier.
Absolute tolerance	5e-3	Absolute difference between TLE and regressing polynomial must be this many times higher than regression value at last TLE in window. TLE w/higher difference is an outlier.
Sequence outlier threshold	3	Number of mean motion absolute deviations above median change between sequences, above which sequence is deemed an outlier.
Minimum sequence length	3	Minimum number of TLEs in a sequence.

The difference between the observed and predicted semimajor axis change is assumed to be due only to inaccuracy in the ballistic coefficient estimate. The BC estimate is updated with the Secant method, and the predicted semimajor axis change is re-computed using the new estimate. This process is repeated iteratively until either 1) the observed and predicted semimajor axis changes converge to within 1 cm, or 2) the maximum number of iterations (10) is reached. The specifications of the AIDA propagator are described in Table 3.2 in Methods Section 3.2.1. The satellite parameters used in AIDA propagation are the same as those used in the HPOP propagator, which are described in Table 3.3 in Methods Section 3.3.1. BC estimates using this method were generated using code supplied by David Gondelach. This code is not currently publicly available, but it is described in detail in [31, 29].

Using the SG estimates, the mean ballistic coefficient over the simulation duration was taken as the BC value for each simulation. For example, for a 5-day reentry simulation, the BC value used was the mean value of the BCs estimated from TLEs in the final 5 days of the object's lifetime. Because they rely on TLEs after the start time of the simulation, the BC values generated with this technique are "a posteriori" values, which would not be available in real-time for an operational scenario. The purpose of using a posteriori values was to attempt to isolate atmospheric density uncertainty as much as possible from BC uncertainty. However, the Saunders-Gondelach technique is known to absorb error from the density model which it uses for BC fitting [95], which in this case was NRLMSISE-00. Ideally, each reentry simulation would use BC values generated using the same atmospheric density model, but AIDA is not available with integrated ROM density model options, so NRLMSISE-00 was used for most density estimates. One BC model for the Fobos-Grunt test case was estimated using JB2008 as the AIDA density model for BC estimation, using a modification of the BC code made by Julia Briden. In all other cases, NRLMSISE-00 was used.

Any non-physical (negative) BC values, of which there were very few, were discarded. Mean BC values were used in an attempt to smooth out time-dependent errors in the NRLMSISE-00 model absorbed into the BC estimates, but this was not always very effective. Mean BCs generated using the SG technique were used for all test cases.

The second source of ballistic coefficient estimates was data published by Pardini and Anselmo (2012 and 2016) from studies of the IADC reentry campaigns for UARS, Progress-M 27M, and Fobos-Grunt [75, 77]. These BC values and their corresponding times (in days before reentry) were obtained from Figures 11 and 12 in [75] for UARS, Figures 15 and 16 in [75] for Fobos-Grunt, and Figure 4 in [77] for Progress-M 27M. These ballistic coefficient estimates were generated using the NRLMSISE-based CDFIT method (henceforth abbreviated the “Pardini-Anselmo" (PA) technique). The CDFIT tool estimates ballistic coefficients by fitting the semimajor axis change observed from TLEs in the least squares sense [75], using the EGM96 gravitational harmonics model with degree and order 16, moon/sun third body effects, solar radiation pressure, and aerodynamic drag using the NRLMSISE-00 model [78]. The time arc over which these BC estimates were computed is not stated, but it is assumed that each BC estimate was calculated over the time period since the calculation of the previous estimate.

Pardini and Anselmo’s BC estimates were used in two ways: first, the mean value of the estimates for the simulation duration was used, calculated in the same way as the SG mean BCs; and second, the time-varying BC curve was used, with linear interpolation in time between neighboring BC points. The results of simulations with all three BC types are presented.

3.4 Atmospheric Density Models

3.4.1 Empirical Models

The two empirical models used in this project are the JB2008 and NRLMSISE-00 model, both of which are commonly used for research and operational applications. These are available in the Julia language in the package SatelliteToolbox.jl.[8] Documentation for these density models can be found on Github[9]. These models are discussed in Section 2.4.1. For further information on the JB2008 and NRLMSISE-00 models, see [9] and [82], respectively.

3.4.2 Reduced-Order Model Theory

As described in Section 2.4.3, Reduced-Order Models describe global atmospheric density distributions in a low-dimensional way. The order-reduction step of reduced-order modeling is accomplished by the projection of the density system onto a set of spatial modes, which are weighted by time-dependent coefficients [59]. Order reduction uses either POD or an autoencoder. The second step of reduced-order modeling is propagating the reduced state forward in time to predict future densities. The predictive step uses either DMDc or a neural network. Only DMDc-based prediction is used in this thesis.

Proper Orthogonal Decomposition

The goal of conducting POD is to represent in a compressed form global density distributions, which are sets of neutral mass density values at locations defined by a discretized grid in latitude, longitude, and altitude, at specific times. A “snapshot” is a single distribution giving density variance across the entire spatial grid, at a single

[8] https://github.com/JuliaSpace/SatelliteToolbox.jl

[9] https://juliaspace.github.io/SatelliteToolbox.jl/dev/man/earth/atmospheric_models/

point in time. The grid of density variance can be obtained directly from the grid of density values, via pointwise subtraction of the mean density value over time at each position on the spatial grid. When conducted on a series of snapshots, POD identifies the dominant spatial modes which capture the greatest amount of energy relative to the mean density value [59].

Let $\mathbf{s}$ be a spatial position on the discretized grid of latitude, longitude, and altitude positions, and let t represent time. The atmospheric density distribution $\mathbf{x}(\mathbf{s}, t)$ can be separated into a spatially-dependent distribution of density means over time ($\bar{\mathbf{x}}$), and a distribution of space- and time-dependent variances ($\tilde{\mathbf{x}}$) [33]:

$$\mathbf{x}(\mathbf{s}, t) = \bar{\mathbf{x}}(\mathbf{s}) + \tilde{\mathbf{x}}(\mathbf{s}, t) \tag{3.5}$$

Since $\bar{\mathbf{x}}(\mathbf{s})$ does not vary in time, it need only be calculated once from the set of density snapshots used to construct the POD model. To prepare the density data for POD, the base-10 logarithm is taken of the density distributions, then the mean values are calculated as [59]:

$$\bar{\mathbf{x}}(\mathbf{s}) = \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}(\mathbf{s}, t_k) \tag{3.6}$$

where n is the number of snapshots. $\bar{\mathbf{x}}$ is subtracted from each snapshot, and POD is conducted directly on the time-varying variance component of the density [33]:

$$\tilde{\mathbf{x}}(\mathbf{s}, t) = \log_{10}(\mathbf{x}(\mathbf{s}, t)) - \log_{10}(\bar{\mathbf{x}}(\mathbf{s})) \tag{3.7}$$

POD represents the distribution $\tilde{\mathbf{x}}(\mathbf{s}, t)$ as a set of orthonormal basis functions $\Phi(\mathbf{s})$ weighted by the uncorrelated time-dependent coefficients $c(t)$ [59]:

$$\tilde{\mathbf{x}}(\mathbf{s}) \approx \sum_{i=1}^{r} c_i(t) \Phi_i(\mathbf{s}) \tag{3.8}$$

where r is the number of modes used in the reconstruction, and r is restricted to being less than or equal to the number of snapshots n on which POD is conducted [59]. The POD reconstruction minimizes truncation error ϵ_r in the least squares sense [59].

$$\epsilon_r(\mathbf{s}, t_k) = \sum_{k=1}^{n} \frac{\langle(\tilde{\mathbf{x}}(\mathbf{s}, t_k) - \sum_{i=1}^{r} c_i(t_k)\Phi_i(\mathbf{s})), (\tilde{\mathbf{x}}(\mathbf{s}, t_k) - \sum_{i=1}^{r} c_i(t_k)\Phi_i(\mathbf{s}))\rangle}{\langle(\tilde{\mathbf{x}}(\mathbf{s}, t_k), (\tilde{\mathbf{x}}(\mathbf{s}, t_k)\rangle} \tag{3.9}$$

In the above equation, $\langle\square\rangle$ represents the inner product.

The spatial modes are calculated with a Singular Value Decomposition (SVD) of the matrix $\mathbf{X}$ of density snapshots. The snapshots are rearranged from the grid dimensions into vectors. The SVD is computed as [33, 10]:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \tilde{\mathbf{x}}_\mathbf{1} & \tilde{\mathbf{x}}_\mathbf{2} & ... & \tilde{\mathbf{x}}_\mathbf{n} \\ | & | & & | \end{bmatrix} = \mathbf{U}\Sigma\mathbf{V}^\top \tag{3.10}$$

The columns of the matrix $\mathbf{U}$ are the left singular vectors, which provide the orthonormal spatial modes $\Phi(\mathbf{s})$ [33, 10]. The reduced-order state $\mathbf{z}$ (the "ROM state") of time-dependent coefficients is calculated with a similarity transform using $\mathbf{U}_r$, the matrix of the first r columns of $\mathbf{U}$ (the first r POD modes) [33, 10]:

$$\mathbf{z} = \mathbf{U}_r^{-1}\tilde{\mathbf{x}} = \mathbf{U}_r^\top\tilde{\mathbf{x}} \tag{3.11}$$

Since the r spatial modes contained in $\mathbf{U}_r$ are invariant in time, they are calculated only once for the PODROM. Once $\mathbf{U}_r$ is calculated, the ROM state $\mathbf{z}$ is sufficient to completely reconstruct the entire density field [33]:

$$\mathbf{x}'(\mathbf{s}, t) = \mathbf{U}_r(\mathbf{s})\mathbf{z}(t) + \bar{\mathbf{x}}(\mathbf{s}) \approx \mathbf{x}(\mathbf{s}, t) \tag{3.12}$$

Further details on this process are available in [59] and [33].

Convolutional Autoencoder

Another method of order reduction is the use of an undercomplete autoencoder neural network, which encodes data into a dimension smaller than that of the original input data [10]. The encoder portion of the autoencoder reduces the data dimension, while the decoder portion reconstructs the original data. The encoder/decoder system is trained such that the mean square error loss function is minimized [103, 10]:

$$\mathcal{L}(\mathbf{X},\mathbf{X}') = \frac{||\mathbf{X}\text{-}\mathbf{X}'||^2}{d} \tag{3.13}$$

Where $\mathbf{X}$ is the original input data, $\mathbf{X}'$ is the encoded/decoded reconstruction, and d is the dimension of $\mathbf{X}$. The key strength of this technique is its ability to encode nonlinear features in the underlying data, whereas the linear POD method captures only linear features [103].

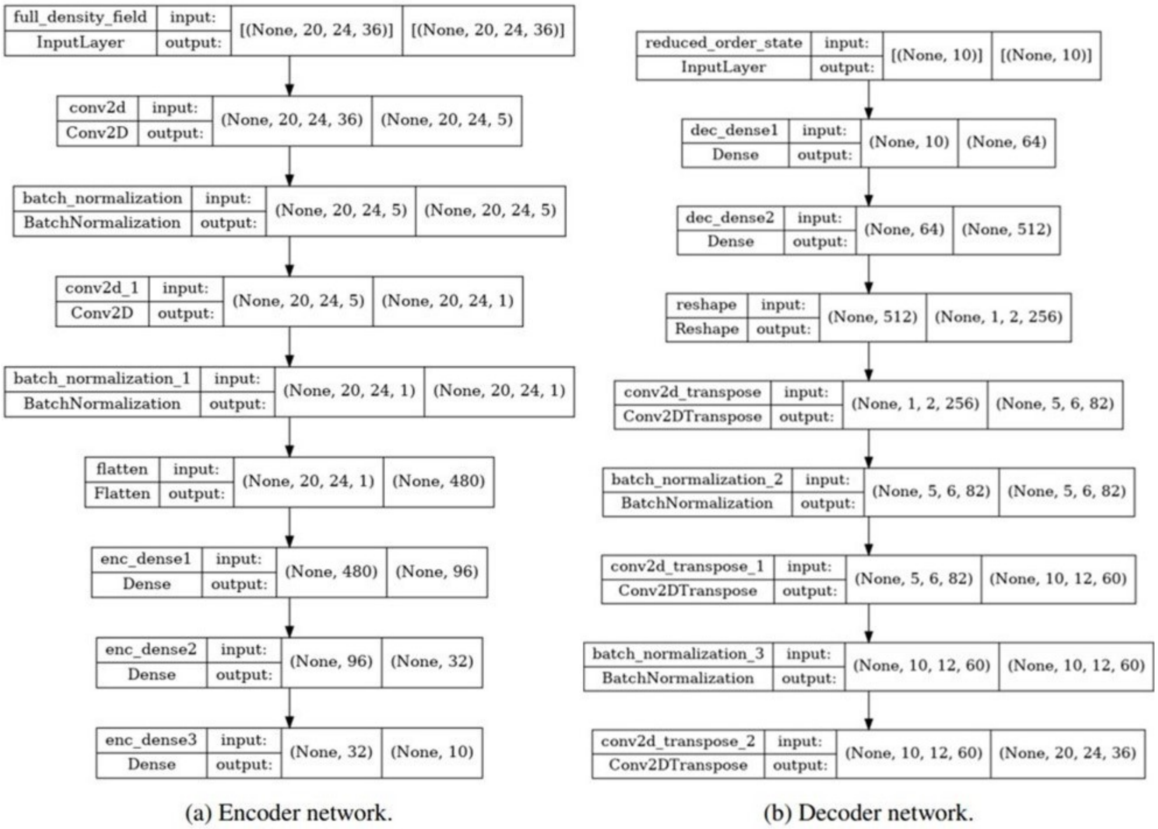
Like for the POD encoding, the densities are standardized prior to encoding. To reduce the range of data magnitudes, density snapshots are standardized by taking the base-10 logarithm and calculating the statistical z-score of the snapshot elements:

$$\mathcal{Z}(\mathbf{s},t) = \frac{\log_{10}(\mathbf{x}(\mathbf{s},t)) - \log_{10}(\bar{\mathbf{x}}(\mathbf{s}))}{\log_{10}(\sigma(\mathbf{s}))} \tag{3.14}$$

Snapshots of $\mathcal{Z}(\mathbf{s},t)$ are passed into the input layer of the autoencoder. The autoencoder stages include two 2-dimensional convolution layers using nonlinear Rectified Linear Unit (ReLU) activation functions and separated by batch normalization layers to center and scale the data between layers, followed by three fully-connected (dense) layers [10]. The first two fully-connected layers also use ReLU activation functions. The following function defines the convolution used in the 2-D convolution layers with layer input x, output data g, and filter c [10]:

$$g_{i,j,k,c} = \sigma(w_c^T x_{i,j,k,c}) \tag{3.15}$$

Note that σ in Eqn. 3.15 is a function definition, *not* the standard deviation. The architecture of the NN autoencoder and decoder are shown in Figure 3-2.


full_density_field | InputLayer | input: | output: | [(None, 20, 24, 36)] | [(None, 20, 24, 36)]
conv2d | Conv2D | input: | output: | (None, 20, 24, 36) | (None, 20, 24, 5)
batch_normalization | BatchNormalization | input: | output: | (None, 20, 24, 5) | (None, 20, 24, 5)
conv2d_1 | Conv2D | input: | output: | (None, 20, 24, 5) | (None, 20, 24, 1)
batch_normalization_1 | BatchNormalization | input: | output: | (None, 20, 24, 1) | (None, 20, 24, 1)
flatten | Flatten | input: | output: | (None, 20, 24, 1) | (None, 480)
enc_dense1 | Dense | input: | output: | (None, 480) | (None, 96)
enc_dense2 | Dense | input: | output: | (None, 96) | (None, 32)
enc_dense3 | Dense | input: | output: | (None, 32) | (None, 10)
reduced_order_state | InputLayer | input: | output: | [(None, 10)] | [(None, 10)]
dec_dense1 | Dense | input: | output: | (None, 10) | (None, 64)
dec_dense2 | Dense | input: | output: | (None, 64) | (None, 512)
reshape | Reshape | input: | output: | (None, 512) | (None, 1, 2, 256)
conv2d_transpose | Conv2DTranspose | input: | output: | (None, 1, 2, 256) | (None, 5, 6, 82)
batch_normalization_2 | BatchNormalization | input: | output: | (None, 5, 6, 82) | (None, 5, 6, 82)
conv2d_transpose_1 | Conv2DTranspose | input: | output: | (None, 5, 6, 82) | (None, 10, 12, 60)
batch_normalization_3 | BatchNormalization | input: | output: | (None, 10, 12, 60) | (None, 10, 12, 60)
conv2d_transpose_2 | Conv2DTranspose | input: | output: | (None, 10, 12, 60) | (None, 20, 24, 36)

(a) Encoder network.

(b) Decoder network.

Figure 3-2: Architecture of the NN autoencoder and decoder. Reproduced from [10], courtesy of Dr. Peng Mun Siew.

Convolutional layers help to prevent overfitting, which occurs when a NN "memorizes" specific training data sets instead of learning underlying data patterns, resulting in a NN that can accurately recreate training data but struggles to generalize to conditions not directly encountered in the training data set [103]. The decoder reconstructs $\mathcal{Z}(t)$ density data from a length-r encoded ROM state $\mathbf{z}$, and the spatially-dependent log-10 mean $\bar{\mathbf{x}}(\mathbf{s})$ and standard deviation $\sigma(\mathbf{s})$ are used to reverse the data standardization, returning density snapshots $\mathbf{X}'$ after the inverse base-10 logarithm is taken.

Dynamic Mode Decomposition with Control

The Dynamic Mode Decomposition with Control (DMDc) is a linear technique used to propagate the reduced-order density model state in time, subject to the effects of space weather as described by the SW index inputs. Given a set of density snapshots, a pair of matrices can be estimated that describe the time evolution of the system (the dynamic matrix) and the impact of space weather (the input matrix). This estimation is conducted in the reduced-order space, making the process very efficient. The DMDc linear system is [33, 10]:

$$\mathbf{z}_{k+1} = \mathbf{A}\mathbf{z}_k + \mathbf{B}\mathbf{u}_k \tag{3.16}$$

where $\mathbf{z}_k$ and $\mathbf{u}_k$ are respectively the ROM state and vector of SW inputs at epoch k, $\mathbf{A}$ is the DMDc dynamic matrix, and $\mathbf{B}$ is the DMDc input matrix. To estimate $\mathbf{A}$ and $\mathbf{B}$, two matrices of time-shifted ROM states are constructed from the set of available density snapshots. A matrix of SW input vectors corresponding to the times of the density snapshots is also constructed. These matrices are defined as [10]:

$$\mathbf{Z}_1 = \begin{bmatrix} | & | & & | \\ \mathbf{z}_1 & \mathbf{z}_2 & \dots & \mathbf{z}_{n-1} \\ | & | & & | \end{bmatrix} \quad \mathbf{Z}_2 = \begin{bmatrix} | & | & & | \\ \mathbf{z}_2 & \mathbf{z}_3 & \dots & \mathbf{z}_n \\ | & | & & | \end{bmatrix} \quad \Upsilon = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_{n-1} \\ | & | & & | \end{bmatrix}$$

where n is, as previously, the number of density snapshots. As the definitions above clearly show, $\mathbf{Z}_2$ is the equivalent of the $\mathbf{Z}_1$ matrix shifted forward in time by one time unit. Referencing the DMDc linear formulation, $\mathbf{Z}_2$ can be written as follows:

$$\mathbf{Z}_2 = \mathbf{A}\mathbf{Z}_1 + \mathbf{B}\Upsilon \tag{3.17}$$

The matrices $\mathbf{A}$ and $\mathbf{B}$ advance $\mathbf{z}_k$ to $\mathbf{z}_{k+1}$ at specified discrete-time intervals.

However, for the application of density modeling, we seek a continuous-time model which can provide a ROM state at any arbitrary time. In other words, we require the continuous dynamical model [33, 10]:

$$\dot{\mathbf{z}} = \mathbf{A}_c\mathbf{z} + \mathbf{B}_c\mathbf{u} \tag{3.18}$$

where the matrices $\mathbf{A}_c$ and $\mathbf{B}_c$ are the continuous-time equivalent of the $\mathbf{A}$ and $\mathbf{B}$ matrices from the original discrete-time formulation [33]. DeCarlo (1989) provides the relation between the discrete-time and continuous-time matrices [17, 33]:

$$\begin{bmatrix} \mathbf{A}_c & \mathbf{B}_c \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \log\left(\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}\right)/T \tag{3.19}$$

where T is the time between density snapshots used to construct the system.

A weakness of the DMDc algorithm for the application of density modeling is that it is a linear algorithm, while the dynamics of the thermosphere are nonlinear. Turner et. al. (2020) addressed this problem in [103] by using a NN, which can capture nonlinear dynamics, to propagate the ROM state instead of DMDc. The NN improved over the DMDc model [103]. However, only the DMDc-based models are explored in this thesis. Using a small 1-hour time step between ROM states helps to keep the linear approximation error low, and allows hourly correction of ROM states via data assimilation, as in [33]. Although the ROMs used in this thesis do not include data assimilation, the one-hour time step is retained for consistency.

Following propagation in the reduced-order state, the new ROM state is converted back into the full-density space by reversing POD with Eqn. 3.12 or the decoder (depending on the encoding method used), and then unstandardizing the data. Densities can then be interpolated anywhere in the longitude, latitude, and altitude domain of the ROMs.

Summary and Advantages of the ROM process

In summary, a series of training-data snapshots are used to construct the POD matrices or autoencoder/decoder, which reduce each large density snapshot to a much smaller representation consisting of an r-vector $\mathbf{z}$. The ROM states derived from the training data snapshots are used to construct the DMDc matrices, which capture the unperturbed time evolution of the ROM state as well as the perturbing impact of space weather on the ROM state. With a sufficiently extensive training data set, the ROM formulation is valid beyond the time range of the training data.

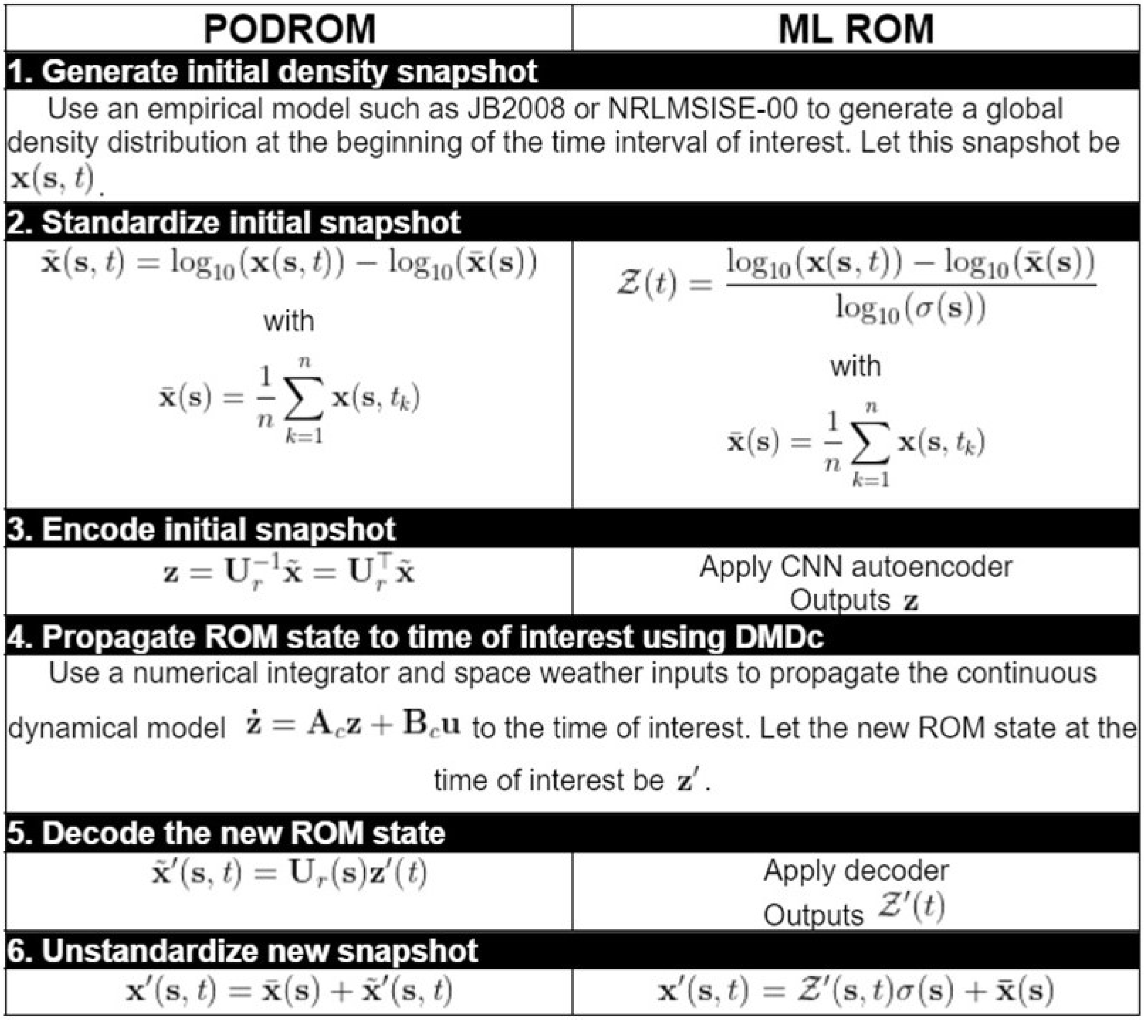
The data compression achieved by the encoding process is tremendous. Using typical values for resolution in latitude, longitude, and altitude, a single-density snapshot in the full-density space is a 3-D grid of 17,280 double-precision floating-point numbers (135 KB). Considering that the typical time resolution of density data snapshots is 1 hour, a week's worth of density snapshots is about 22.1 MB of data, and data covering a full solar cycle ($\sim$12 years) is almost 13.5 GB of data.

In contrast, when stored as a ROM state, a density snapshot can be represented as an r-vector, where r is the number of spatial modes included in the model. Typical values of r could be 10 to 30, so one density snapshot can be represented in 80-240 B, with a low single-digit truncation error in the density snapshot reconstructed from the ROM state (about 1% mean error using $r = 30$ [59]). ROMs do carry the additional overhead of storing the $\mathbf{A}_c$, $\mathbf{B}_c$, and $\mathbf{U}_r$ matrices and $\bar{\mathbf{x}}$ for POD, and the autoencoder/decoder for ML ROMs. For a POD ROM with $r = 30$, the three matrices and $\bar{\mathbf{x}}$ are about 4.09 MB; however, these need only be calculated and stored once. Including these matrices, the memory needed to store one week of ROM states with $r = 30$ is $\sim$4.14 MB. Storing 12 years of data as ROM states requires only about 28.1 MB, or about 0.2% of what is needed for storing raw density data sets.

The computational advantage of conducting the prediction step of the algorithm in the reduced-order space is also significant. DMDc could be conducted in the full density space, but this would require much larger $\mathbf{A}_c$ and $\mathbf{B}_c$ matrices. Using the same assumed density field size of 17,280 data points and 23 space weather inputs (a typical value for the JB2008-based PODROM, see Section 3.4.3), one step of DMDc in the full density space would require approximately 600 million individual operations to perform the matrix multiplication. In contrast, using $r = 30$, one DMDc step in the reduced-order space requires just over 3 thousand operations, or 0.0005% of the operations which would be needed for DMDc in the full-density space.

Once the encoding and DMDc algorithms have been trained on a set of density

Table 3.5: Modeling thermospheric density using a ROM.



PODROM	ML ROM
1. Generate initial density snapshot	
Use an empirical model such as JB2008 or NRLMSISE-00 to generate a global density distribution at the beginning of the time interval of interest. Let this snapshot be $\mathbf{x}(\mathbf{s}, t)$.	
2. Standardize initial snapshot	
$\tilde{\mathbf{x}}(\mathbf{s}, t) = \log_{10}(\mathbf{x}(\mathbf{s}, t)) - \log_{10}(\bar{\mathbf{x}}(\mathbf{s}))$ with $\bar{\mathbf{x}}(\mathbf{s}) = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}(\mathbf{s}, t_k)$	$\mathcal{Z}(t) = \frac{\log_{10}(\mathbf{x}(\mathbf{s}, t)) - \log_{10}(\bar{\mathbf{x}}(\mathbf{s}))}{\log_{10}(\sigma(\mathbf{s}))}$ with $\bar{\mathbf{x}}(\mathbf{s}) = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}(\mathbf{s}, t_k)$
3. Encode initial snapshot	
$\mathbf{z} = \mathbf{U}_r^{-1}\tilde{\mathbf{x}} = \mathbf{U}_r^{\top}\tilde{\mathbf{x}}$	Apply CNN autoencoder Outputs $\mathbf{z}$
4. Propagate ROM state to time of interest using DMDc	
Use a numerical integrator and space weather inputs to propagate the continuous dynamical model $\dot{\mathbf{z}} = \mathbf{A}_c\mathbf{z} + \mathbf{B}_c\mathbf{u}$ to the time of interest. Let the new ROM state at the time of interest be $\mathbf{z}'$.	
5. Decode the new ROM state	
$\tilde{\mathbf{x}}'(\mathbf{s}, t) = \mathbf{U}_r(\mathbf{s})\mathbf{z}'(t)$	Apply decoder Outputs $\mathcal{Z}'(t)$
6. Unstandardize new snapshot	
$\mathbf{x}'(\mathbf{s}, t) = \bar{\mathbf{x}}(\mathbf{s}) + \tilde{\mathbf{x}}'(\mathbf{s}, t)$	$\mathbf{x}'(\mathbf{s}, t) = \mathcal{Z}'(\mathbf{s}, t)\sigma(\mathbf{s}) + \bar{\mathbf{x}}(\mathbf{s})$

snapshots, the ROM can be used as an independent density model. An empirical model, such as JB2008, is used to generate an initial global density distribution, from which an initial ROM state $\mathbf{z}_0$ is calculated. Then, the ROM state is propagated from $\mathbf{z}_0$ at some time step over the time period of interest, using the relevant space weather inputs (or predicted values of those inputs). Table 3.5 summarizes the basic steps of the ROM density modeling process.

3.4.3 DESTO and DESQU Toolboxes

The MATLAB version of the PODROM code used in this thesis is publicly available in the Density Estimation Toolbox (DESTO)[10] [33, 28]. The PODROM code used in this thesis is a translation of that code into the Julia language. The Python version of the ML ROM model code is publicly available in the Dynamic Data-Driven Thermospheric Density Estimation with Quantified Uncertainties (DESQU)[11] toolbox [66], and was called into the main Julia code with the package PyCall.jl.[12]

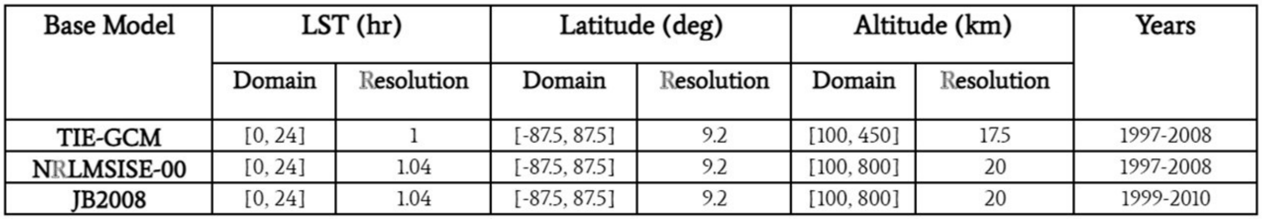
The density data used to train the PODROMs was generated using the JB2008, NRLMSISE-00, and TIE-GCM models, and the ML ROM was trained on JB2008 data. Using each of these models, training snapshots were generated at hourly intervals over a 12-year period (to cover a full solar cycle), resulting in more than 105,000 snapshots of training data [33]. In these models, local solar time (LST) was substituted for the longitude coordinate. Table 3.6 describes the features of the training data sets.

[10] https://zenodo.org/record/3634245

[11] https://github.com/SWQU/DESQU

[12] https://github.com/JuliaPy/PyCall.jl

Table 3.6: PODROM training data parameters, from [33].

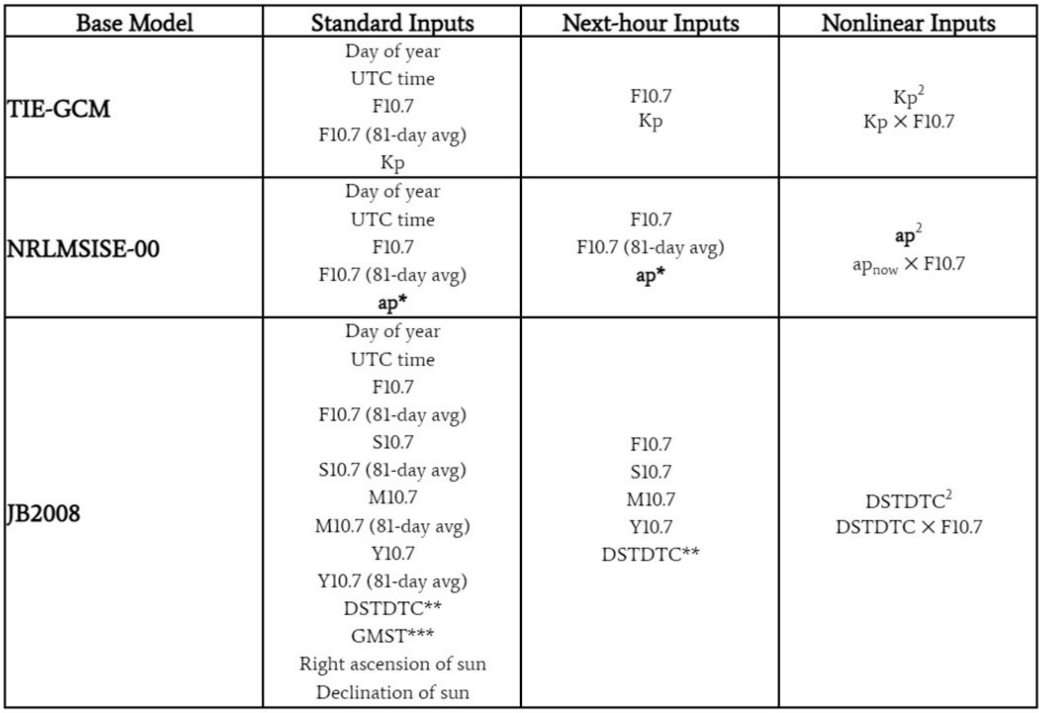


Base Model	LST (hr)		Latitude (deg)		Altitude (km)		Years
	Domain	Resolution	Domain	Resolution	Domain	Resolution	
TIE-GCM	[0, 24]	1	[-87.5, 87.5]	9.2	[100, 450]	17.5	1997-2008
NRLMSISE-00	[0, 24]	1.04	[-87.5, 87.5]	9.2	[100, 800]	20	1997-2008
JB2008	[0, 24]	1.04	[-87.5, 87.5]	9.2	[100, 800]	20	1999-2010

The different date range for JB2008 training data was due to gaps in space weather input data in the year 1998.

The ROM DMDc prediction step requires space weather inputs. In addition to the inputs used by the original model used to generate training snapshots, additional inputs were added, including nonlinear space weather inputs [33]. Table 3.7 describes the space weather inputs used.

Table 3.7: ROM space weather inputs, reconstructed from [33].



Base Model	Standard Inputs	Next-hour Inputs	Nonlinear Inputs
TIE-GCM	Day of year UTC time F10.7 F10.7 (81-day avg) Kp	F10.7 Kp	Kp^2 Kp × F10.7
NRLMSISE-00	Day of year UTC time F10.7 F10.7 (81-day avg) **ap***	F10.7 F10.7 (81-day avg) **ap***	$\textbf{ap}^2$ ap_{now} × F10.7
JB2008	Day of year UTC time F10.7 F10.7 (81-day avg) S10.7 S10.7 (81-day avg) M10.7 M10.7 (81-day avg) Y10.7 Y10.7 (81-day avg) DSTDTC** GMST*** Right ascension of sun Declination of sun	F10.7 S10.7 M10.7 Y10.7 DSTDTC**	DSTDTC^2 DSTDTC × F10.7

*Vector of 8 *ap* indices up to 57 hours before current time [33].
**Temperature change calculated from Dst index [10].
***Greenwich Mean Sidereal Time [33].

For complete details and code for the ROMs and training processes, see [33, 62, 103] and the DESTO/DESQU packages [28, 66].

3.4.4 Reduced-Order Model Implementation

The first step of ROM implementation was running the model setup to prepare the dynamic and input matrices, spatial density modes, space weather inputs, and other ROM components. An initial snapshot of global densities was encoded, using either POD or an autoencoder, to produce the initial ROM state. For the NRLMSISE-00 PODROM, this snapshot was generated using the NRLMSISE-00 model. This was a change from the original DESTO NRLMSISE-00 PODROM, which used the JB2008 model for ROM state initialization. The reason for the change was to eliminate the JB2008 influence on the NRLMSISE-based model. The JB2008 model was used for the JB2008 POD and ML ROMs, and for the TIE-GCM POD ROM, since running the TIE-GCM model to generate an initial snapshot was too computationally expensive.

Next, the space weather inputs and the dynamic/input matrices prepared in ROM setup were used to propagate the initial ROM state over the entire period of the simulation, at hourly intervals beginning from the start time of the reentry simulation. The end product of this propagation was a matrix containing ROM state vectors at one-hour time intervals. The ROM state matrix was passed into the satellite propagation routine.

At each time step in the satellite propagation process, the ROM state matrix was interpolated in time using a linear interpolation scheme, giving a ROM state at the time of interest. The resulting ROM state vector was converted back into a density field, either by summing the product of the time-dependent ROM state coefficients with the respective spatial mode values at the location of interest (for the POD ROMs), or by applying the decoder algorithm (for the ML ROM). The density field was interpolated to the satellite's position, and this density value was passed into the drag calculation subroutine.

3.5 Uncertainty Quantification and Monte Carlo

Monte Carlo (MC) simulation provides a non-intrusive method of uncertainty quantification for complex algorithms [38]. Randomly varying inputs subject to physically realistic constraints for a large number of iterations produce a distribution of outputs that can approximate the true uncertainty distribution of the algorithm results. For example, varying the initial satellite position within a range of reasonable TLE error values over repeated simulations can simulate the distribution of reentry prediction results expected in the presence of TLE uncertainty. This allows the uncertainty in the reentry predictions to be quantified to estimate margins of error. By varying the simulation inputs that are perturbed, we can also evaluate the sensitivity of the system to errors in multiple inputs. Monte Carlo simulations are very computationally expensive, since they converge only as $1/\sqrt{N}$, with N being the number of simulation runs [38].

Due to limited computational resources, MC simulation in this thesis is restricted to only the UARS case over a period of 3 days prior to reentry, and to only three density models. Uncertainty from initial-state (TLE) uncertainty and space weather index uncertainty is examined with MC simulation. Initial state uncertainty is selected for examination due to the known inaccuracies in TLEs. Space weather index error is included to evaluate the sensitivity of reentry predictions to noise in SW inputs, simulating the use of predicted SW index values. Ballistic coefficient uncertainty is not included in MC simulations, but is evaluated by varying the BC ±25%. BC uncertainty is selected for analysis due to its significant impact on reentry predictions, combined with the difficulty of determining BC because of satellite attitude and drag coefficient uncertainties.

Because bias and uncertainty distribution types are unknown for these three factors, a Gaussian (normal) error distribution centered at zero is adopted for sampling each input.

3.5.1 TLE Uncertainty

In a 2008 study of TLE uncertainty across the entire TLE catalog, Flohrer et. al. found average LEO TLE 1-sigma uncertainty values in the radial, along-track, and out-of-plane directions to be 0.102 km, 0.471 km, and 0.126 km, respectively [22]. Correlation between errors in these three dimensions is unknown, so the three directions are assumed to be independent. For each iteration of the MC simulation using TLE uncertainty, a sampled TLE error value is drawn from three independent distributions constructed for each of the three directional components. These distributions are zero-centered Gaussian distributions, with standard deviations defined by the Flohrer's findings. A sample initial-position uncertainty distribution for $N = 1000$ is plotted in 3 dimensions in Fig. 3-3. Adding uncertainty to the initial position creates a "cloud" of initial position points, centered on the unperturbed TLE location.

The vector of sampled errors in each component (which can each be positive or negative) is transformed from the local-vertical, local-horizontal (LVLH, also called RTN) coordinate frame into the ECI coordinate frame using a coordinate transformation function from the Julia SatelliteToolbox package. No information is available on the quantification of TLE-derived velocity uncertainty, so velocity uncertainty is assumed to be zero in all components. The ECI vector of TLE position uncertainty is added to the satellite's ECI initial position, as defined by the initial TLE. Satellite initial positions in the Monte Carlo simulation, therefore, vary in all three dimensions about the TLE-defined position.

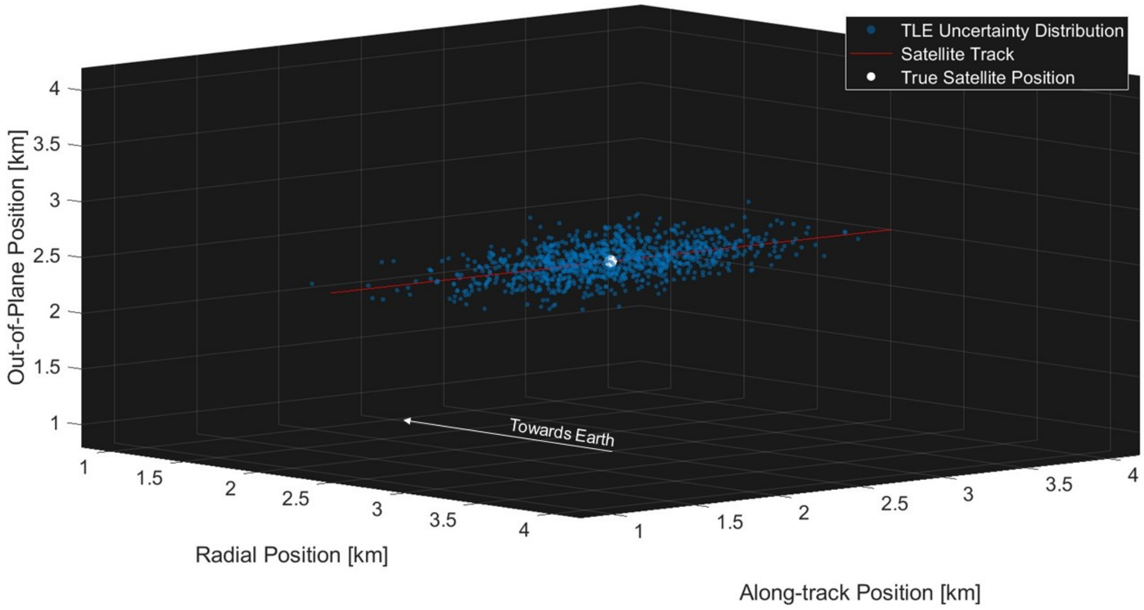


Figure 3-3: Illustration of TLE uncertainty distribution in 3 dimensions for a satellite at point [2.5, 2.5, 2.5].

3.5.2 BC Uncertainty

Because drag is linearly related to BC, the relationship of BC to reentry time is more straightforward than the relationships between reentry time and space weather or initial state uncertainty. The relationship between BC and reentry time is not generally linear; a higher BC results in quicker orbital decay, which lowers the satellite's orbit into higher-density regions of the atmosphere, compounding the impact of the higher BC non-linearly. However, the BC-reentry time relationship is straightforward in the sense that, all else being equal, a higher BC will always result in an earlier reentry time and vice versa. Therefore, reentry prediction uncertainty stemming from BC uncertainty can be modeled by varying BC up and down by some uncertainty value. This captures a window of possible reentry times, and is far more efficient than MC simulation. Pardini and Anselmo used this technique for estimating reentry time uncertainty in [75, 77].

For the UARS case, Pardini and Anselmo varied the BC by ±20% to obtain a reentry uncertainty window. The 1-sigma uncertainty in UARS SG BCs over the week prior to reentry was ±24% of the mean value, which accords relatively well with the 20% BC uncertainty estimate used by Pardini and Anselmo. In this thesis, the UARS BC was varied by ±25% as a conservative estimate on 1-sigma BC uncertainty. UARS simulations were run for all nominal durations with the varied BCs, and using the JB2008, JB8ROM, and ML ROM density models.

3.5.3 Space Weather Uncertainty

The three density models used for the Monte Carlo simulation were the JB2008, JB2008-based POD ROM, and JB2008-based ML ROM. These models take the F10.7, Y10.7, S10.7, and M10.7 indices, 81-day averaged values of these indices, and DST-DTC. The indices perturbed were the F10.7, Y10.7, S10.7, and M10.7 indices. Because the uncertainty distribution values were normal distributions centered at zero, no perturbation was added to the 81-day average values. Quantifying correlations between these indices is out of the scope of this paper, so uncertainty in each index value is treated independently.

Standard deviations are estimated from the CIRA-08 reference space weather data tables, which cover 143 months, just over a full 11-year solar cycle [100]. These tables provide minimum, mean, and maximum values of F10.7, Y10.7, S10.7, and M10.7 indices in each month of the solar cycle. To determine standard deviation, the spread between minimum and maximum index values for each month was assumed to represent a ±3-sigma spread. The one-sigma value for each index was estimated to be $\frac{1}{6}$(max-min) for the respective month of the solar cycle. For each simulation, the month in the solar cycle was determined to find the relevant minimum and maximum index values, using December 2008 and December 2019 as the respective start dates of Solar Cycles 24 and 25 [81, 96].

Noise values were generated for one-hour windows throughout the simulation duration and added directly to the SW indices. Because the JB2008 model uses some time-lagging SW inputs while the ROMs use real-time data, the magnitude of the perturbations are not equal for each step of the simulations. Some of the time lags in the inputs taken by JB2008 are greater than the MC simulation duration of 3 days, so the noise values in some of the indices do not overlap at all between the JB2008 and ROM models. Since the results of all the runs are examined as a whole rather than individually, and all noise values are randomly drawn from the same distribution, the difference in SW noise values ingested by the models is not a concern. The SW perturbation process is also described in [10]. Nonlinear ROM SW inputs were calculated from the perturbed values of the relevant indices.

Another difference between SW noise ingestion process for the ROMs versus JB2008 occurs because the ROMs require initialization using a density model snapshot from JB2008 or NRLMSISE-00. This initial snapshot is encoded and used as the starting ROM state, which is then updated at hourly intervals as it is propagated across the simulation duration. Since the ROM state is changed by applying updates rather than discarding the previous ROM state, each state vector preserves historical data from previous ROM states. The initial snapshot used to generate the first ROM state, therefore, has a persistent impact on the ROM results throughout the reentry simulations. The JB2008 model has no equivalent initialization process.

This creates two different ways that space weather noise can be ingested into the ROM: either it can be ingested only into ROM state updates, or it can be ingested into the ROM state initialization as well, by feeding noisy space weather data into the JB2008 model when generating the initial snapshot. Including space weather noise in the ROM state initialization may magnify the impact of space weather noise. The initial-state noise impacts the entire "history" of the ROM states propagated from that initial ROM state. This impact adds to the impact of space weather noise in the ROM state updates. Simulations were run using both methods of SW noise ingestion.

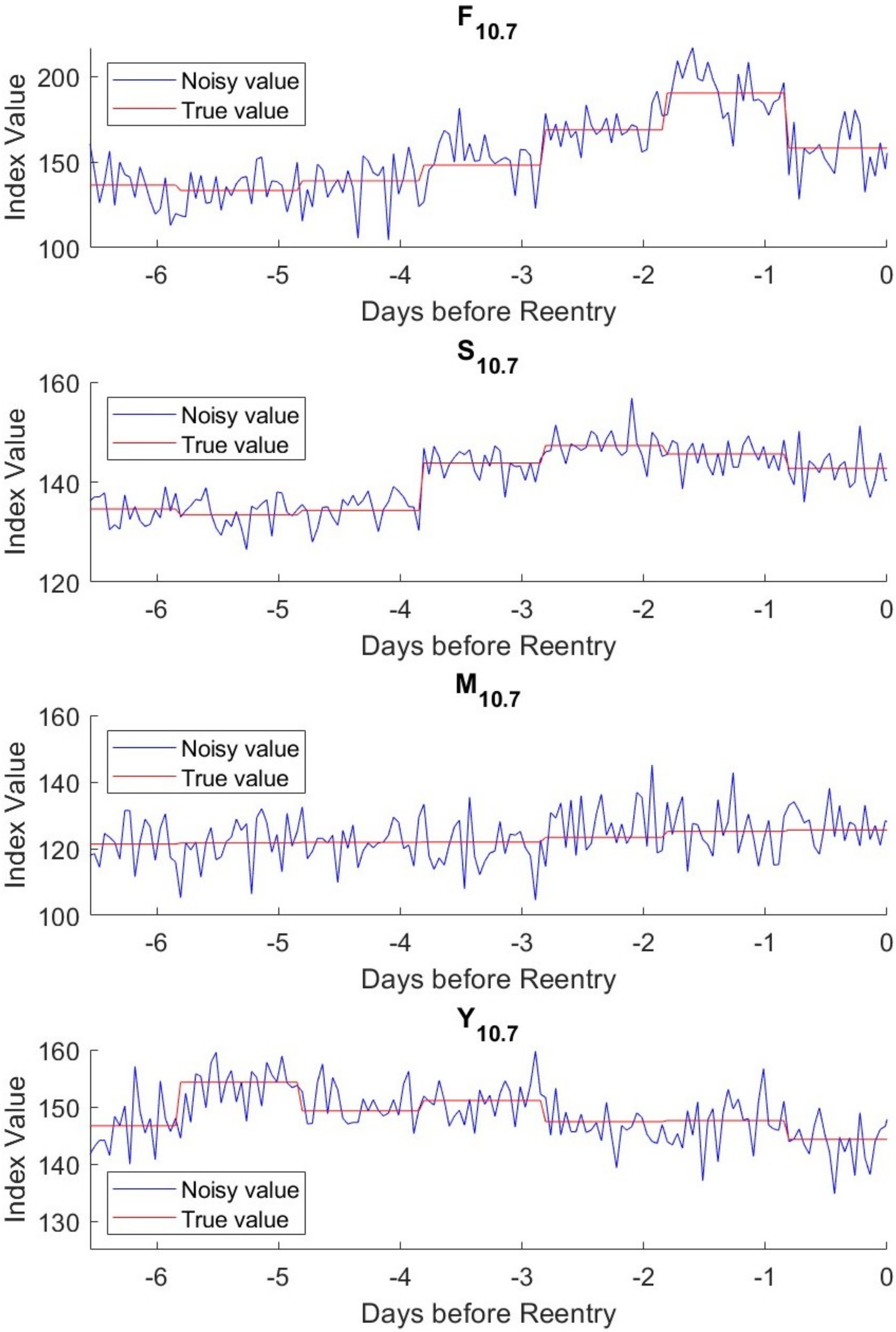


Figure 3-4: Example of a set of noisy SW values for the UARS test case. Unique sets of noisy SW values were used for each MC iteration. True values are plotted for reference.

3.6 Simulations and Expected Results

Simulation setups were chosen according to the following points of interest: 1) uncertainty quantification/sensitivity analysis to examine features of the reentry prediction process and characteristics of ROM behavior; 2) impact of different atmospheric density models on reentry simulation accuracy; 3) impact of different ballistic coefficient models on reentry simulation accuracy; and 4) evolution of reentry prediction accuracy as the simulation start time approaches the reentry time. Both MC simulations and "standard" simulations (single propagator runs, as opposed to Monte Carlo simulations) were used for point 1, while only standard simulations were used for points 2 through 4.

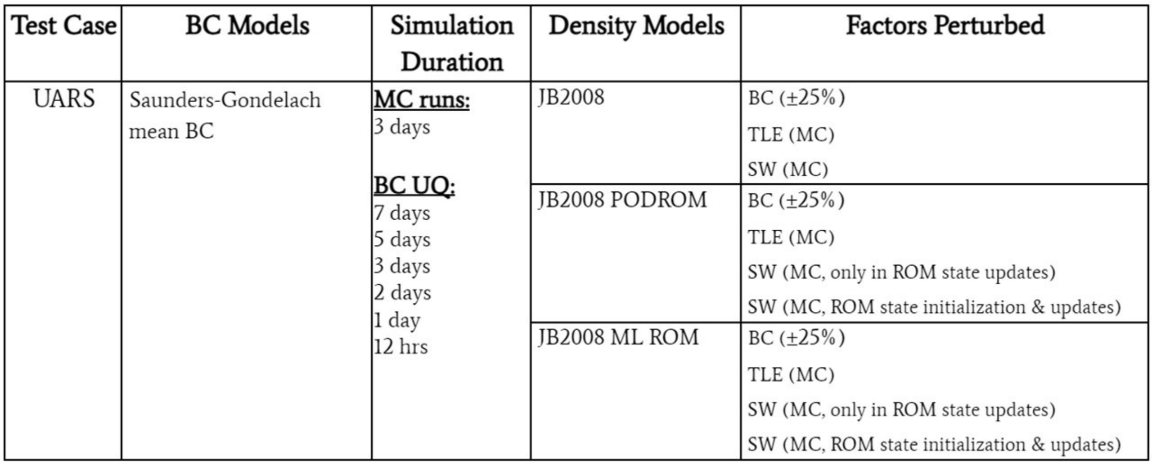
Simulation Selections

Due to the high computational cost of MC simulation (each of which involves running 150 standard simulations), only one test case (UARS), one BC model (the SG mean BC), and one simulation duration (3 days) was used for the TLE-based and SW-bases MC simulations. Three density models were selected for UQ: the JB2008, JB8ROM, and ML ROM. Because all three of these are based on JB2008, they accept the same base set of space weather inputs (although the ROMs include additional nonlinear inputs). This allows uncertainty in space weather to be introduced into the same set of inputs for consistency across the density models.

TLE uncertainty and space weather uncertainty are investigated with MC simulations, due to the more complex relationships between these factors and the drag force. BC uncertainty was explored by varying the BC by ±25% instead of using MC simulation. This reduced the run time needed and allowed the impact of BC uncertainty to be evaluated for all 6 prediction windows, instead of only for the 3-day prediction window.

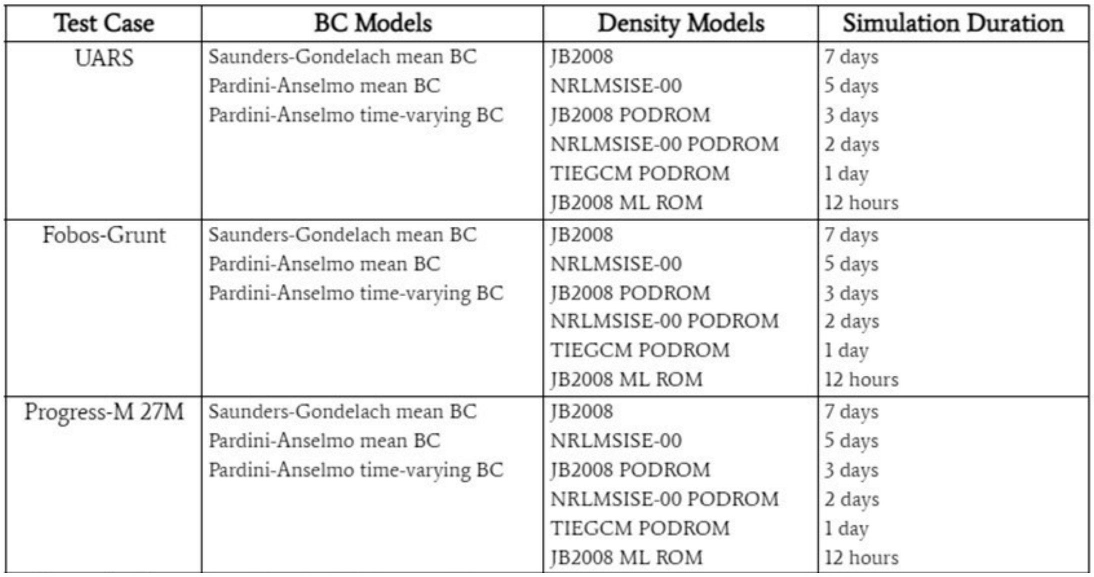
Further MC simulation would be a good direction for future work, but insufficient computational resources were available during the completion of this thesis. Table 3.8 shows the UQ simulations run. All combinations of the factors listed in Table 3.8 were run with $N = 150$.

Table 3.8: Uncertainty quantification simulations.



Test Case	BC Models	Simulation Duration	Density Models	Factors Perturbed
UARS	Saunders-Gondelach mean BC	**MC runs:** 3 days **BC UQ:** 7 days 5 days 3 days 2 days 1 day 12 hrs	JB2008	BC (±25%) TLE (MC) SW (MC)
			JB2008 PODROM	BC (±25%) TLE (MC) SW (MC, only in ROM state updates) SW (MC, ROM state initialization & updates)
			JB2008 ML ROM	BC (±25%) TLE (MC) SW (MC, only in ROM state updates) SW (MC, ROM state initialization & updates)

Table 3.9: Standard simulations.



Test Case	BC Models	Density Models	Simulation Duration
UARS	Saunders-Gondelach mean BC Pardini-Anselmo mean BC Pardini-Anselmo time-varying BC	JB2008 NRLMSISE-00 JB2008 PODROM NRLMSISE-00 PODROM TIEGCM PODROM JB2008 ML ROM	7 days 5 days 3 days 2 days 1 day 12 hours
Fobos-Grunt	Saunders-Gondelach mean BC Pardini-Anselmo mean BC Pardini-Anselmo time-varying BC	JB2008 NRLMSISE-00 JB2008 PODROM NRLMSISE-00 PODROM TIEGCM PODROM JB2008 ML ROM	7 days 5 days 3 days 2 days 1 day 12 hours
Progress-M 27M	Saunders-Gondelach mean BC Pardini-Anselmo mean BC Pardini-Anselmo time-varying BC	JB2008 NRLMSISE-00 JB2008 PODROM NRLMSISE-00 PODROM TIEGCM PODROM JB2008 ML ROM	7 days 5 days 3 days 2 days 1 day 12 hours

The final three points of interest were explored with “standard” simulations. Six different density models were used: JB2008, NRLMSISE-00 (abbreviated NRLM00), JB2008 PODROM (JB8ROM), NRLMSISE-00 PODROM (NRLROM), TIE-GCM

PODROM (TIEROM), and JB2008 ML ROM (ML ROM). Three different BCs were used for standard simulations: Saunders-Gondelach mean BC, Pardini-Anselmo mean BC, and Pardini-Anselmo time-varying BC. In addition, six different nominal prediction windows were used, ranging from 7 days to 12 hours prior to reentry, to mimic the recurring predictions conducted during IADC reentry prediction campaigns. Table 3.9 summarizes the standard simulations performed; all combinations of all factors described in Table 3.9 were run, for a total of 324 standard simulations.

Expected Results

Based on the preliminary results in [10], ROMs are expected to show performance relatively comparable to empirical models. Because the ROMs presented in this thesis do not use data assimilation, they are not necessarily expected to outperform the density models on which they were trained. In fact, the ROMs may show slightly worse results overall due to truncation error from discarding higher-order spatial density modes present in the empirical model data used for ROM training, since only 10 modes ($r = 10$) are used for the ROMs in this thesis. However, ROMs should be relatively competitive with empirical models in terms of reentry prediction accuracy, and should have an advantage in run time. Only the TIEROM is expected to possibly outperform empirical models, as it was trained on a physics-based model.

Ballistic coefficient models are expected to have a major impact on reentry predictions. The NRLM00 is expected to have an advantage in prediction accuracy on BC models that absorb significant NRLM00 model error during the estimation process. This advantage is expected to manifest as the NRLM00 showing better performance on some BC models, while other density models perform worse with the same BC. Similarly, the JB2008 is expected to have an advantage over other density models when JB2008-based BC estimates are used in the Fobos-Grunt test case.

The JB2008, JB8ROM, and ML ROM reentry predictions are expected to show relatively similar sensitivity to uncertainty in BC, and to and uncertainty in initial state. The ROMs are expected to show more sensitivity to SW than JB2008 when SW noise is included in the ROM initialization process, since initial ROM state "history" persists across later ROM states. For the same reason, ROMs are expected to show less sensitivity to SW noise than the JB2008 when SW noise is used only in ROM updates. BC uncertainty is expected to yield the greatest reentry time uncertainty of the three factors tested.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Uncertainty Quantification

4.1 Initial-State Uncertainty

Because of the low quality of TLE data, propagating initial states derived from TLEs can result in significant errors in the predicted satellite state after hours or days of propagation. To investigate the effect of TLE error on reentry time predictions, Monte Carlo simulations were performed with error added to the TLE initial position in three dimensions, as described in Section 3.5.1. The UARS test case with the SG mean BC and a nominal 3-day prediction window were chosen for the MC simulations. 150 iterations were run using the JB2008, JB2008 PODROM (JB8ROM), and the JB2008 ML ROM. Figure 4-1 shows the range of reentry trajectories predicted by each density model. The region between the earliest and the latest reentry prediction is shaded. The unperturbed results (with no error added to the initial state) are indicated with a solid line.

The significance of the range between earliest and latest reentry times is that this spread represents the amount of reentry time variation that could be attributed to the initial-state error. If the earliest and latest reentry times enclose the true reentry epoch (shown by a vertical dotted line), then TLE initial-state uncertainty could account for the entirety of the difference between the given density model's prediction and the true reentry time. This is the case for the JB8ROM and the JB2008. For both of these density models, not only do the earliest/latest predictions enclose the true reentry epoch, but the reentry trajectory ranges are nearly centered on the true reentry epoch. This means the difference between the JB2008 and JB8ROM predictions and the true reentry time is significantly less than the amount that could be attributed to error in the initial position.

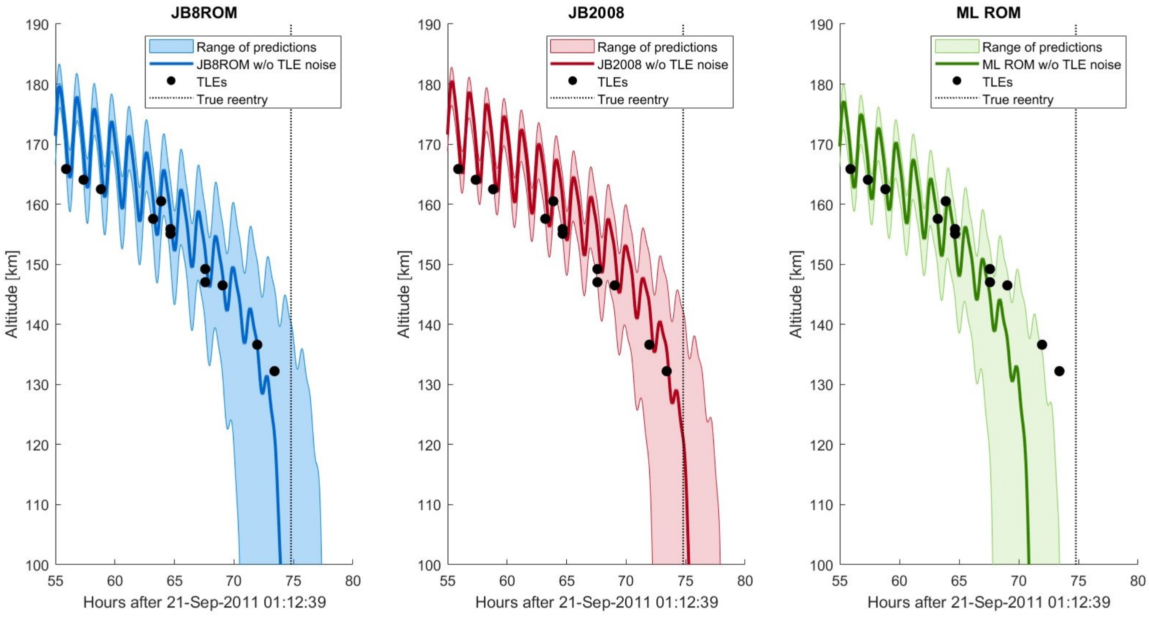


Figure 4-1: UARS reentry trajectory ranges predicted by three density models during $N = 150$ Monte Carlo simulation with radial, along-track, and out-of-plane error added to initial position.

This is not the case for the ML ROM, which underestimates the UARS residual lifetime over a three-day window. The range of reentry times attributable to TLE error does not include the true reentry time. This can be interpreted as follows: assuming the zero-centered Gaussian TLE error model is a reasonable characterization of TLE uncertainty, at least some of the error in the ML ROM's reentry prediction must come from some source *other* than TLE error. It could come from BC error, or from inherent density model error. Note that none of the error can be attributed to space weather input prediction error in this case, because true space weather inputs (a posteriori) were used for the simulations in Figure 4-1.

Reentry time distributions are plotted in Figure 4-2.

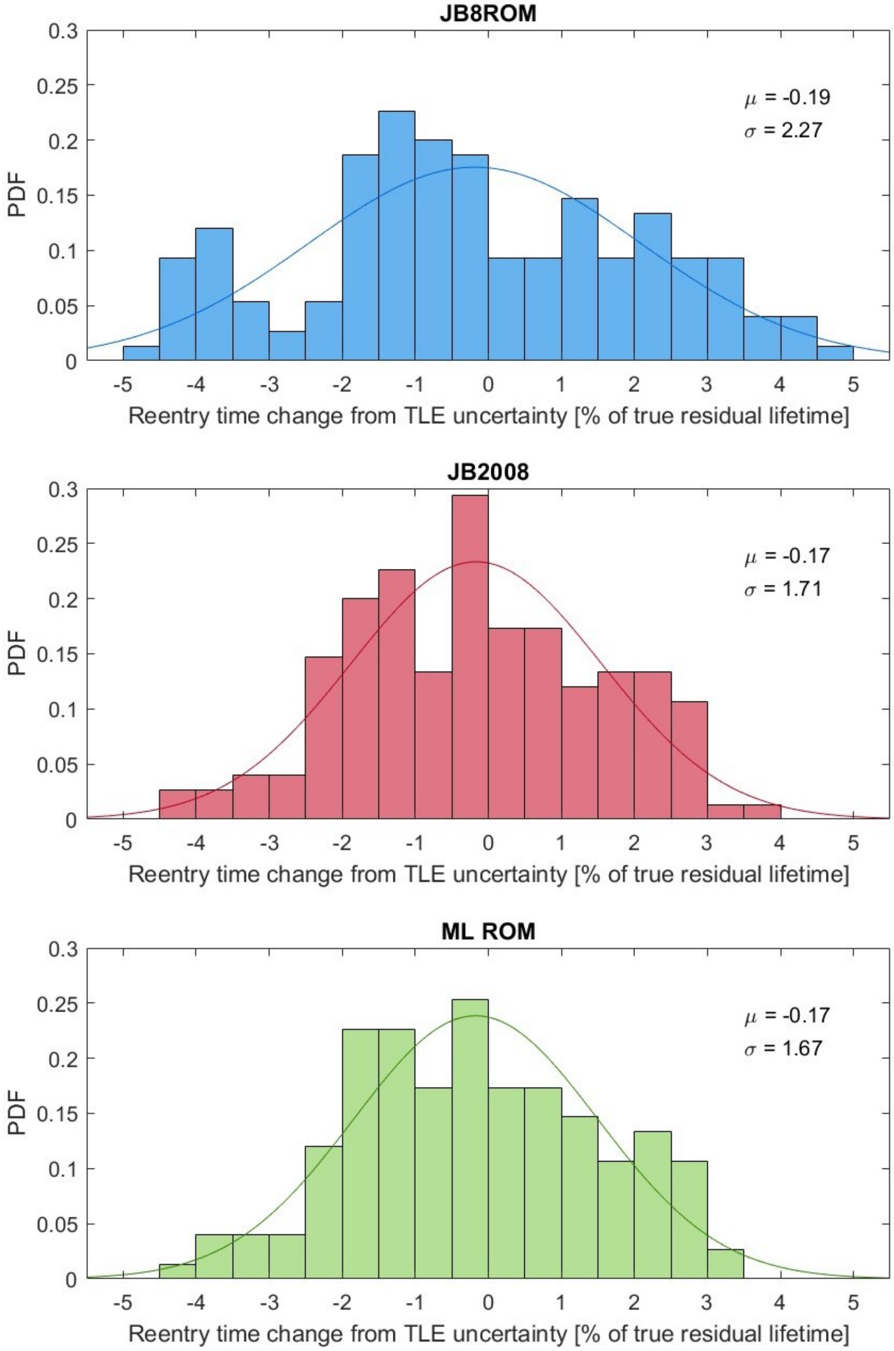


Figure 4-2: Distribution of change in reentry time predicted by three density models during $N = 150$ Monte Carlo simulation with radial, along-track, and out-of-plane error added to TLE. Change in reentry time given relative to unperturbed reentry time predicted by each respective model, as a percent of true residual lifetime.

Reentry time distributions are plotted in a relative sense. The quantity plotted is difference from unperturbed (no TLE error added) reentry time predicted by each respective model, given as a percent of the true residual lifetime. This quantity is defined by:

$$\frac{t_{r,i} - t_r}{t_{r,true} - t_0} \times 100\% \tag{4.1}$$

where $t_{r,i}$ is the reentry time predicted by the ith MC iteration, t_r is the unperturbed reentry time prediction of the respective density model, t_0 is the simulation start time, and $t_{r,true}$ is the true reentry time. Figure 4-2 therefore represents the distributions of percent error in residual lifetime resulting from TLE error, for each density model. The mean and standard deviation of each distribution is marked on its plot, and a Gaussian curve with the same mean and standard deviation is plotted on the same axes for reference.

None of the distributions well match a Gaussian (normal) distribution. In each distribution, three distinct "peaks" appear: one below zero, one close to zero, and one above zero. One possible explanation for this pattern is that it may stem from the simultaneous perturbation of the initial states in three dimensions, with the three-peak pattern representing the summation of three underlying reentry time distributions resulting from TLE perturbation in the radial, along-track, and out-of-plane directions. The second possibility is that the peaks may represent perigees in successive orbits of the UARS. Since perigee is the lowest point of the orbit, where densities are highest, the probability of reentry may be higher during "bands" of time centered on successive perigee passes. The distribution peaks in most cases are separated by (very approximately) 2% of residual lifetime, which for a 3-day prediction window corresponds to about 1.5 hours. This is also the approximate orbital period of an object in LEO, so the spacing of the peaks is about equal to the expected spacing between successive perigees of the reentering UARS.

All of the distributions are centered close to but slightly less than zero. According to Figure 4-2, TLE error tends to create a minor underestimation bias of about 0.2%. However, with 150 samples, this finding is not statistically significant at the 10% significance level. Ideally, MC simulations should be performed with a higher number of iterations to confirm this finding.

The standard deviation in reentry times is around 1.7% for the JB2008 and the ML ROM, which produce very similar distributions. The JB8ROM reentry time distribution has a greater spread, with about 2.3% standard deviation. The JB8ROM also has a small but notable peak at the -4% mark, indicating that a cluster of individual perturbed JB8ROM simulations underestimated residual lifetime compared to the unperturbed JB8ROM, by about 4%. Neither of the other density models display such a peak as far from zero. The greater spread of the JB8ROM means that it is slightly more sensitive to initial-state error than the JB2008 and the ML ROM. This could indicate greater variations in density on small time and space scales in the JB8ROM than in its counterparts. Again, increasing the number of MC runs would be useful to confirm the greater sensitivity of the JB8ROM to TLE initial state error.

As a final note on TLE uncertainty, it's important to consider the physical implications of the reentry time error magnitudes discussed above. A 1-sigma uncertainty value of 2% at three days prior to reentry (equivalent to ∼90 minutes) due to TLE error may sound relatively small, but could create a significant error in terms of ground risk assessment. 90 minutes is the approximate period of LEO RSOs, meaning that a 2% error corresponds to position uncertainty equivalent to about one entire orbit of the reentering spacecraft after only three days of propagation. Such a large position uncertainty can make risk assessment very difficult.

4.2 Ballistic Coefficient Uncertainty

As discussed in Section 3.5.2, the BC was varied by ±25% for UARS reentry simulations. These simulations were performed with the JB2008, JB8ROM, and ML ROM density models, over the same prediction windows as the standard simulations. Reentry trajectories are shown in Figure 4-3. Altitudes shown are geocentric.

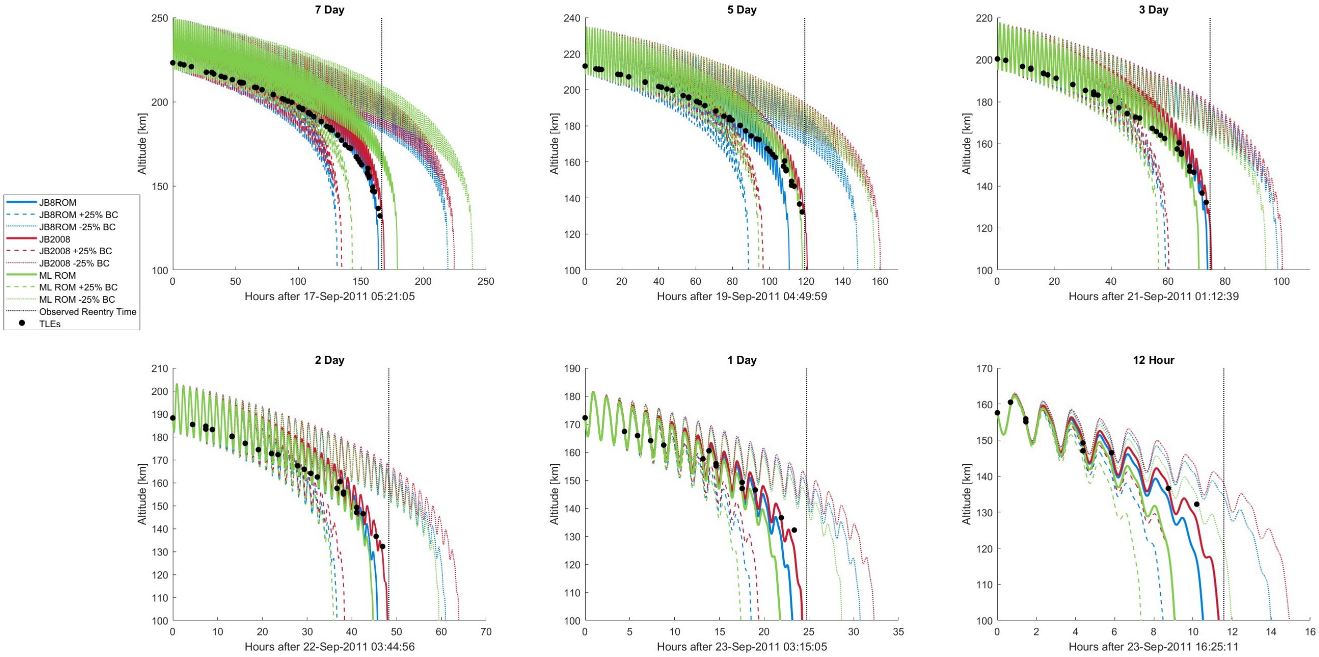


Figure 4-3: Reentry trajectories predicted by three density models with Saunders-Gondelach mean BC and ±25% BCs.

As expected, there is an inverse relationship between BC and residual lifetime. A higher BC results in greater drag force, inducing an earlier reentry, while a lower BC decreases the drag force and leads to a longer reentry time. SpaceX utilized this relationship during the February 2022 geomagnetic storm, when Starlink satellites were commanded into an "edge-on" attitude which lowered their BCs by reducing their effective frontal area [54]. This was intended to compensate for the increase in drag from higher densities during the storm, and prevent premature reentry. For 11 of the satellites, the maneuver was successful in delaying reentry long enough for the satellites to outlast the storm and recover [21]. Similarly, the predicted UARS reentry is delayed by a lower BC and expedited by a higher BC. However, the change in residual lifetime is not linear with respect to the change in BC.

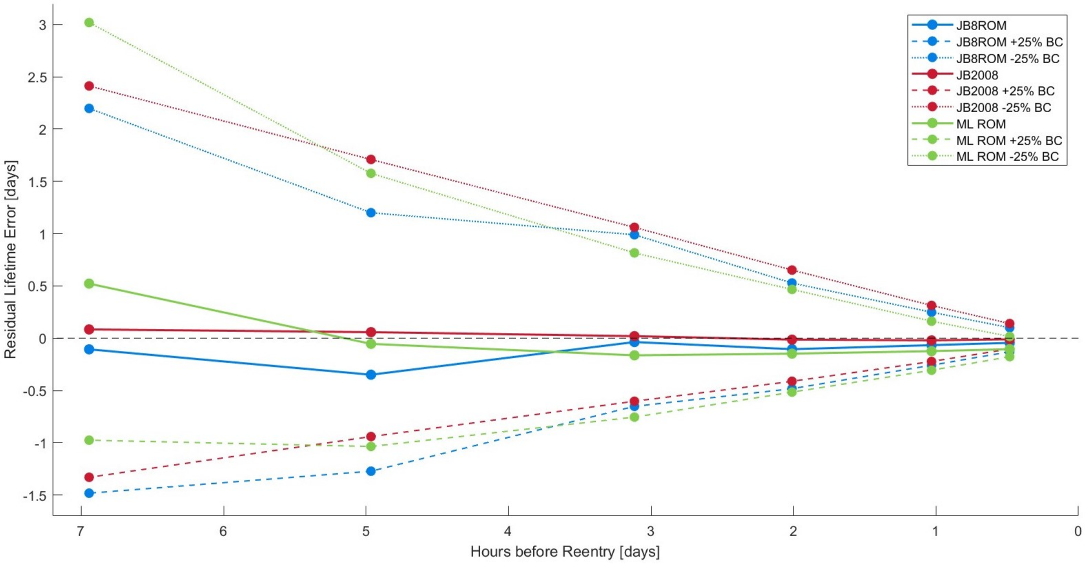


Figure 4-4: Residual lifetime predicted by three density models with Saunders-Gondelach mean BC and ±25% BCs.

Fig. 4-4 demonstrates that, assuming a 25% uncertainty in BC, all of the residual lifetime prediction error present in all three models at every prediction window can be accounted for purely by BC uncertainty. It is possible (though unlikely) that any one density model is perfectly accurate, and its residual lifetime error is entirely due to BC error. (It is not, however, possible that *all* density models are exactly correct; obviously, they would agree identically if this were the case.)

For reference, prediction results obtained by Pardini and Anselmo (2013) on the UARS reentry are presented in Figure 4-5 [5]. Uncertainty windows in this figure were produced by varying the BC by ±20% [5]. The results obtained by Pardini and Anselmo over the final 7 days of the UARS lifetime are very similar to the JB8ROM results shown in Fig. 4-4, except that the JB8ROM shows overall lower error and higher uncertainty. The lower error is likely due at least in part to the use of a posteriori SW data and TLE data for BC fitting, while the higher uncertainty is due to the larger BC uncertainty of ±25% used in Fig. 4-4.

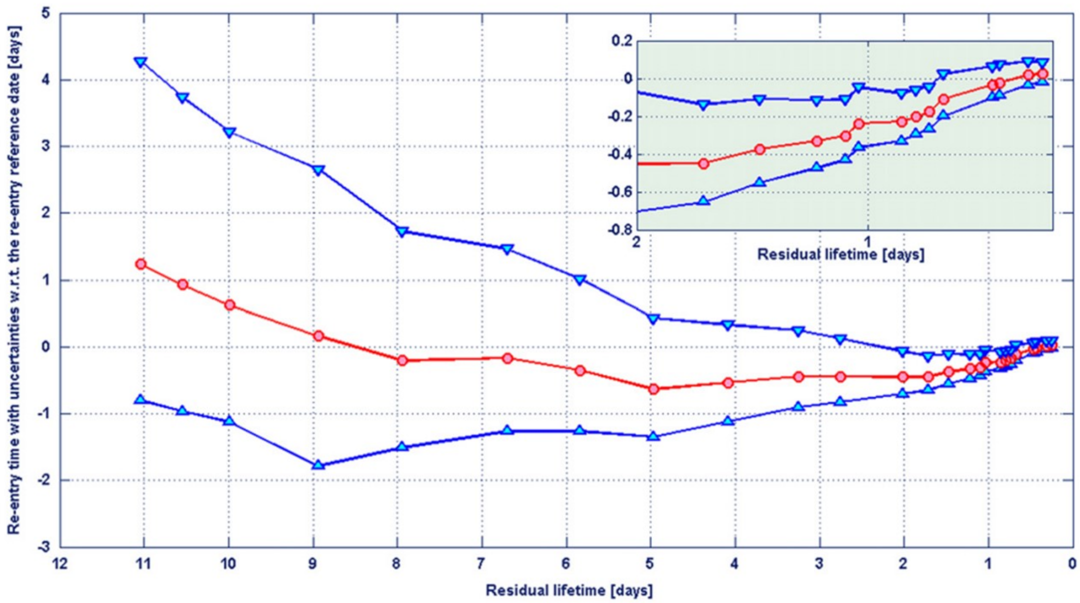


Figure 4-5: Reentry time prediction study by Pardini and Anselmo (2013) for the UARS reentry.[1] Red indicates prediction, and blue indicates uncertainty window. Figure is from [5].

The assumption of a 1-sigma estimate of 25% for BC uncertainty may be overly conservative, in which case the results in Fig 4-4 would overestimate the reentry time uncertainty attributable to BC. Although Pardini and Anselmo did use a similarly conservative variation of ±20% of the BC value, the actual standard deviation they found in the UARS' BC was 13% over the reentry campaign [75]. However, the evolution of the UARS BC was not random around a steady mean value; instead, it decreased steadily towards the end of the campaign [75]. This kind of behavior could not be well captured by varying the BC by ±13%. Pardini and Anselmo found that ±20% uncertainty yielded a "satisfactory" residual lifetime uncertainty window, but a larger uncertainty of ±30% would have given "more regular and conservative evolution of the upper bound" [75].

[1] Figure reproduced in accordance with agreement between MIT and publisher Elsevier. For more information, see https://libraries.mit.edu/scholarly/publishing/using-published-figures/.

The first conclusion regarding BC uncertainty, therefore, is contingent upon the accuracy of the BC uncertainty estimate. *If* ±25% is an accurate characterization of the uncertainty in the UARS BC, then the error in estimated residual lifetime due to atmospheric density model error for the JB2008, JB8ROM, and ML ROM is smaller than the BC-induced uncertainty in residual lifetime, for all reentry prediction windows. The practical implication of this finding is that for the UARS test case, improving the BC estimate would be more effective than improving the density model, if the goal is to reduce residual lifetime uncertainty.

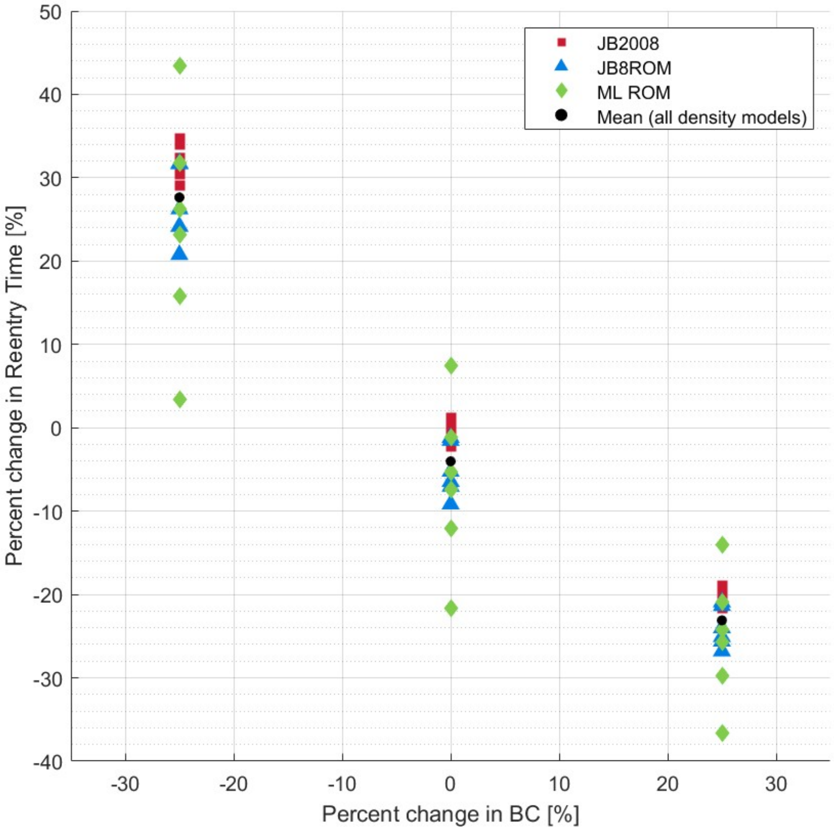


Figure 4-6: Residual lifetime percent error predicted by three density models with Saunders-Gondelach mean BC and ±25% BCs.

Figure 4-6 shows a scatter plot of percent change in predicted residual lifetime as a function of percent change in BC, for all three models and all six prediction windows.

Fig. 4-6 illustrates the second notable conclusion, which is the nonlinear relationship between BC and reentry time. Underestimation of the BC relative to its true value tends to create larger errors in predicted residual lifetime than overestimation by an equal amount. On average, a 25% decrease in the BC leads to a 32% increase in the predicted residual lifetime, while a 25% increase in BC leads to only a 19% average decrease in predicted residual lifetime.

This is due to the approximately exponential decrease of atmospheric density with altitude. An object with a lower BC spends a longer time at higher altitudes, because the lower BC causes its orbit to decay more slowly. But the amount of additional time spent at higher altitudes is not linear in BC, because density is (approximately) exponentially lower at higher altitudes. An incremental increase in altitude results in a greater percent change in density than does an incremental decrease in altitude (assuming the same starting altitude). Therefore, an incremental increase in the amount of time spent at higher altitudes causes a larger relative change in average drag force than does an incremental decrease in the amount of time spent at higher altitudes. The rate of change in average drag force, like the atmospheric density to which it is directly proportional, decreases approximately exponentially with higher altitude (and thus with time spent at higher altitude). A BC slightly higher than the true BC will produce lower residual lifetime prediction error, on average, than a BC which is lower than the true BC by the same amount.

This BC study highlights the significant impact that BC estimates have on the success of reentry prediction. In practice, real-time BC estimation is more challenging than the methods presented in this thesis, which use TLEs that are not available ahead of time. Even given TLEs for the time range of interest, physically-accurate BC estimation is difficult due to the tendency of the BC to absorb atmospheric density model error during the estimation process. Choosing an effective BC model therefore depends on the application. Pairing a BC model with the optimally compatible density model can artificially conceal density model error, which is a problem when validating density models. However, using a BC that is optimally compatible with the density model of choice is important for accuracy in practice, when the priority is compensating for density model error to achieve accurate reentry prediction.

4.3 Space Weather Uncertainty

The final uncertainty source examined is space weather inputs to density models. To understand the purpose of the following analysis, it is necessary to distinguish between two separate error types associated with using SW indices as density model inputs. The first problem is that that the true index values are proxies which may not well represent the physical solar/geomagnetic processes they are meant to describe [20]. Because of this, and also based on the density model's internal handling of the index values, the density model's response to a particular proxy value may not accurately characterize the true density response [20], *even when* the true (observed) value of the SW index is used. The key point here is that even true SW inputs may lead to inaccurate density predictions. For the purposes of this thesis, this first type of error associated with SW is considered inherent density model error (although it could, at least in part, be considered "inherent SW proxy error").

The second problem with SW inputs is prediction error. This factor comes into play when a density forecast is required, necessitating predictions of future SW index values as density model inputs. On top of the SW-associated inherent error in density modeling, predicting index values can introduce error into the actual SW index values. The difference between these error types can be summarized briefly: "type 1" error is when you have the correct SW inputs, but still predict the wrong density; "type 2" error is when you incorrectly predict the SW inputs, causing inaccurate density predictions. Type 1 (inherent density model) uncertainty is always present, while type

2 is only present when using predicted SW proxies instead of true observed values.

The following MC analysis investigates the impact of type 2 SW error on reentry predictions using the JB2008, JB8ROM, and ML ROM. SW proxy prediction error is represented by random noise drawn from a zero-centered Gaussian model described in Section 3.5.3. The perturbed SW values are centered on the true index values, with prediction bias assumed to be zero. The sensitivity of reentry time predictions of the three models to random noise in SW values, representing prediction error, is presented in the following sections. As explained in Section 3.5.3, adding SW error to ROMs can be accomplished in two ways: first, by adding noise in both the ROM state initialization and the propagation, and second, by adding SW noise only during ROM state propagation (initializing on true SW values). The first of these methods is presented in Section 4.3.1, and the second is presented in Section 4.3.2.

4.3.1 Noise in ROM state Initialization

First, noisy SW data was used as density model inputs for the reentry simulation, including in the ROM state initialization process for the ML ROM and JB8ROM. JB2008 has no equivalent initialization process, so at each point the JB2008 was run directly on noisy SW values. Figure 4-7 shows the spread between earliest and latest reentry trajectories predicted by each model in an MC simulation of 150 iterations.

SW noise clearly has a much larger impact on the two ROMs than on the JB2008 model, which shows only small deviations from its unperturbed prediction. On the other hand, both ROMs show deviations of up to a few hours in their reentry predictions as a result of the added SW index noise. Note that this is not a reflection of inherent density model error, but rather of the models' sensitivity to random zero-centered error in SW inputs. Distributions of space-weather-noise-induced reentry time change are shown in Figure 4-8.

As in Section 4.1, reentry time change is shown as defined in Eqn. 4.1. For each

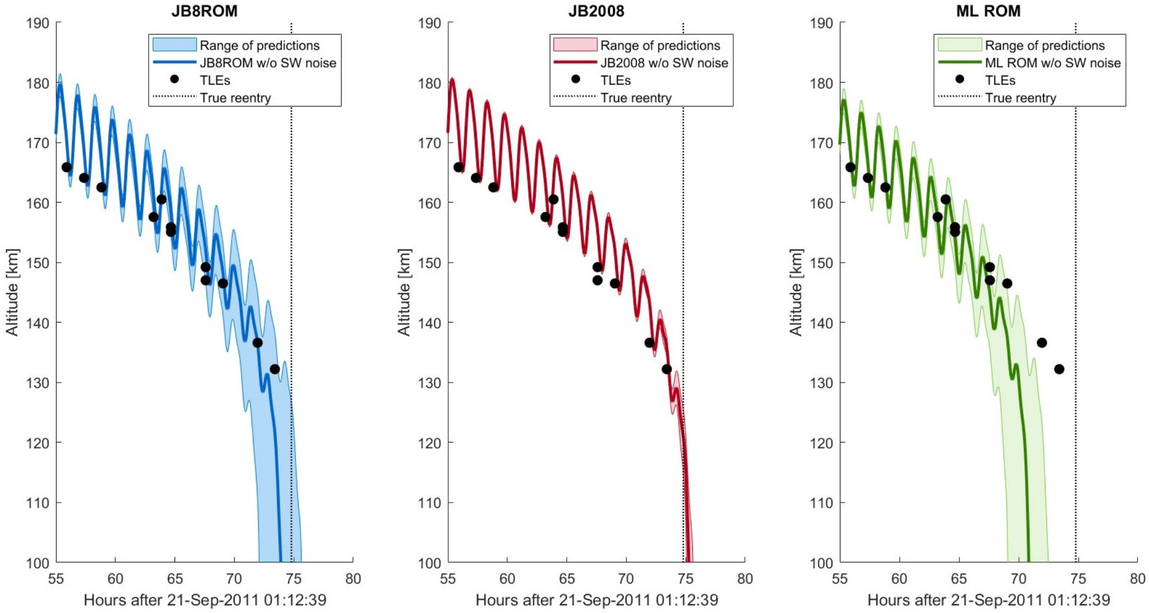


Figure 4-7: UARS reentry trajectory ranges predicted by three density models (with noise in ROM initialization) during Monte Carlo simulation with Gaussian zero-centered noise added to F10.7, S10.7, M10.7, and Y10.7.

MC iteration, this quantity answers the question: By what percent of the residual lifetime does each model's reentry time prediction change, due to SW index noise? As observed in Figure 4-7, the ROM predictions are significantly more sensitive to SW noise. Although the ROM reentry time changes center very close to zero, like the JB2008 model, the ROMs have a 1-sigma uncertainty of nearly 1% due to SW noise. The 1-sigma uncertainty of the JB2008 is almost an order of magnitude lower than the ROMs, at close to 0.1%.

However, this is not the end of the story. As noted previously, these results were obtained by adding SW noise both to ROM state initialization and to ROM state propagation. As described in Section 3.4.2 and Table 3.5, ROMs are initialized by creating a global density snapshot with an empirical density model (JB2008, in the case of the JB2008-based ROMs), and encoding that snapshot with POD or an autoencoder. The encoded snapshot becomes the initial ROM state, and it is propagated across the time window of interest to generate the full set of ROM states.

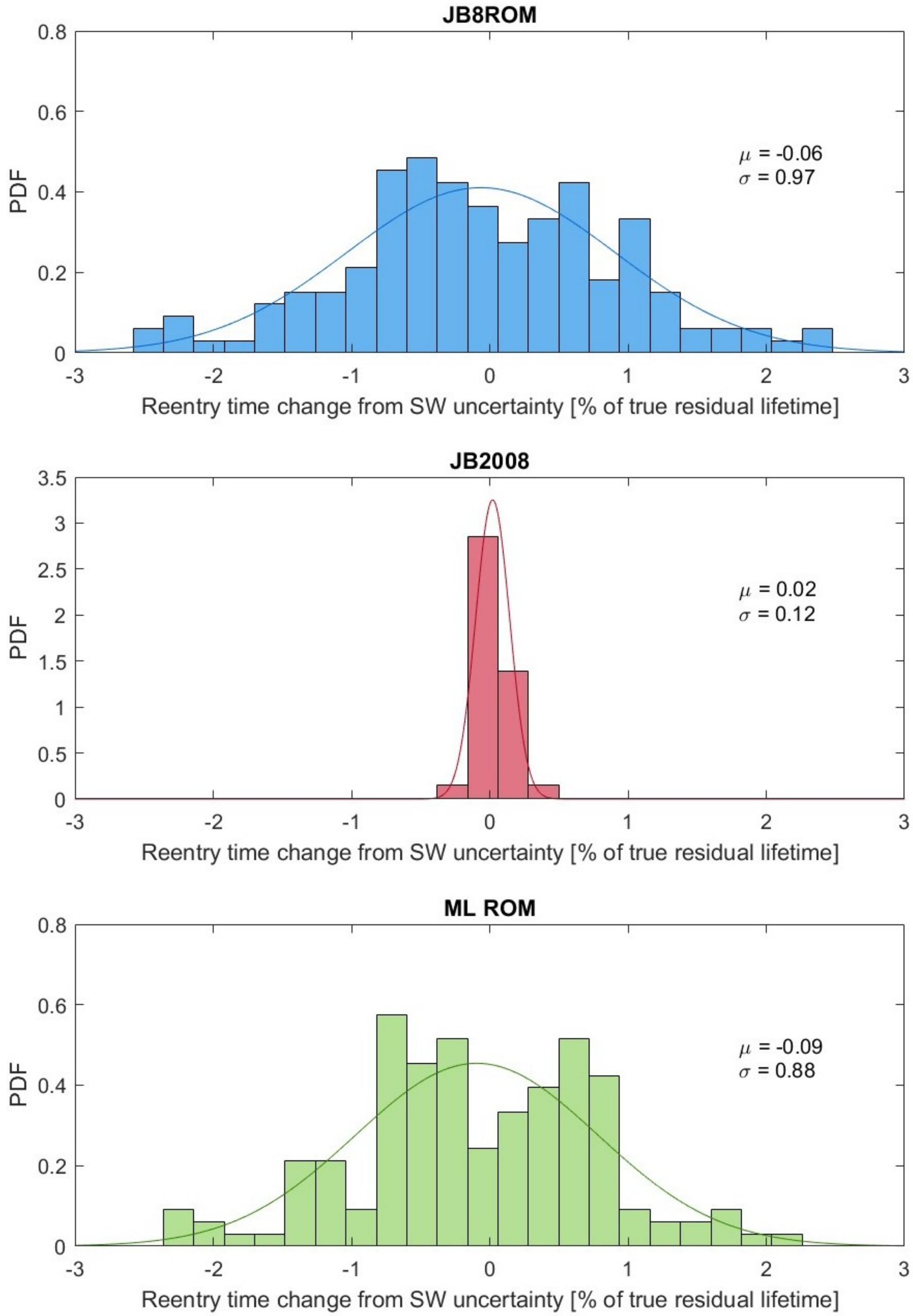


Figure 4-8: Distribution of change in reentry time predicted by three density models (with noise in ROM initialization) during $N = 150$ Monte Carlo simulation with Gaussian zero-centered noise added to F10.7, S10.7, M10.7, and Y10.7. Change in reentry time given relative to unperturbed reentry time predicted by each respective model, as a percent of true residual lifetime.

In Figs. 4-7 and 4-8, SW noise is included in this process by generating a JB2008 snapshot using perturbed SW inputs, and encoding that snapshot as the initial ROM state.

As the ROM state is propagated across the time period of interest using DMDc, the initial ROM state is *not* discarded and replaced, but rather is updated at each 1-hour time step according to 1) the time evolution described in the DMDc dynamic matrix and 2) the SW inputs throughout the time window of propagation (which are incorporated using the DMDc input matrix). The ROM state at a given point in time is therefore influenced by the history of previous ROM states. When the ROM is run from an erroneous initial ROM state (for example, one generated with SW noise), the initial error persists in later ROM states generated with DMDc propagation. This persistent error is in addition to the ROM state update errors induced by noisy SW inputs during DMDc propagation. The JB2008 has no equivalent to the initial ROM state error. Therefore, using noise in the ROM state initialization places the ROMs at an automatic disadvantage in terms of sensitivity to SW noise, as compared to the JB2008.

The other method of ingesting SW noise into ROMs is to initialize the ROMs on true SW inputs, and then propagate the initial (unperturbed) ROM state using noisy SW inputs. The results of this method are presented next.

4.3.2 No Noise in ROM state Initialization

ROMs were run from initial states generated using JB2008 with true SW inputs. Zero-centered Gaussian noise was added to all SW inputs, as previously, except that true SW inputs were used to generate the initial ROM state. This method of SW noise ingestion produced very different results. Density model reentry trajectory spreads due to SW noise incorporated in this manner are shown in Figure 4-9. Note that the JB2008 results included for reference are the same results shown in Section 4.3.1.

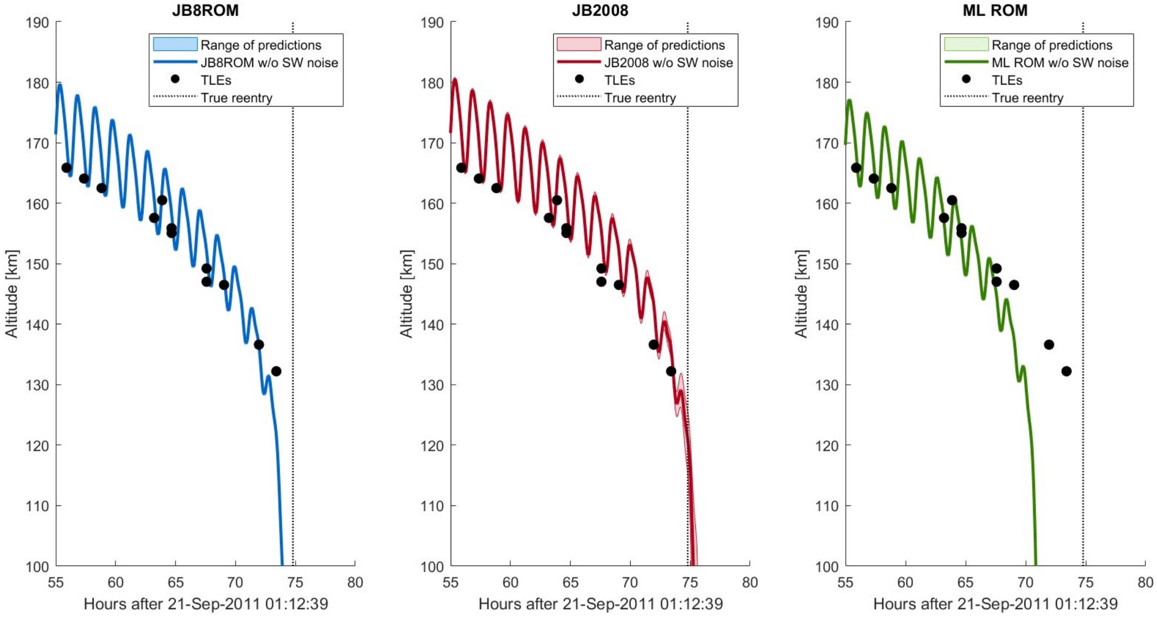


Figure 4-9: UARS reentry trajectory ranges predicted by three density models (no noise in ROM initialization) during Monte Carlo simulation with Gaussian zero-centered noise added to F10.7, S10.7, M10.7, and Y10.7.

As Fig. 4-9 shows, ROM trajectory spread due to space weather noise is indistinguishably small when no noise is included in the ROM state initialization process. Reentry time change distributions are shown in Fig. 4-10. As previously, reentry time change is defined by Eqn. 4.1. Mean and standard deviations of each distribution are shown on the plots as μ and σ, respectively. Gaussian (normal) distributions with the same μ and σ are overlaid for reference. The overlaid distribution on the JB8ROM is scaled down by a factor of ⅗ for plot clarity.

Reentry time changes produced by SW noise only in ROM state updates is vanishingly small, with a standard deviation on the order of ~30-60 seconds for 3-day predictions. For all density models, reentry time changes are centered close to zero.

The interpretation of these results is that, if initialized on true SW inputs, ROMs are virtually unaffected by random noise in SW inputs. Note that this does not indicate insensitivity to SW inputs generally (a topic which is not addressed by this UQ technique), but rather insensitivity to zero-centered (unbiased) SW noise. This

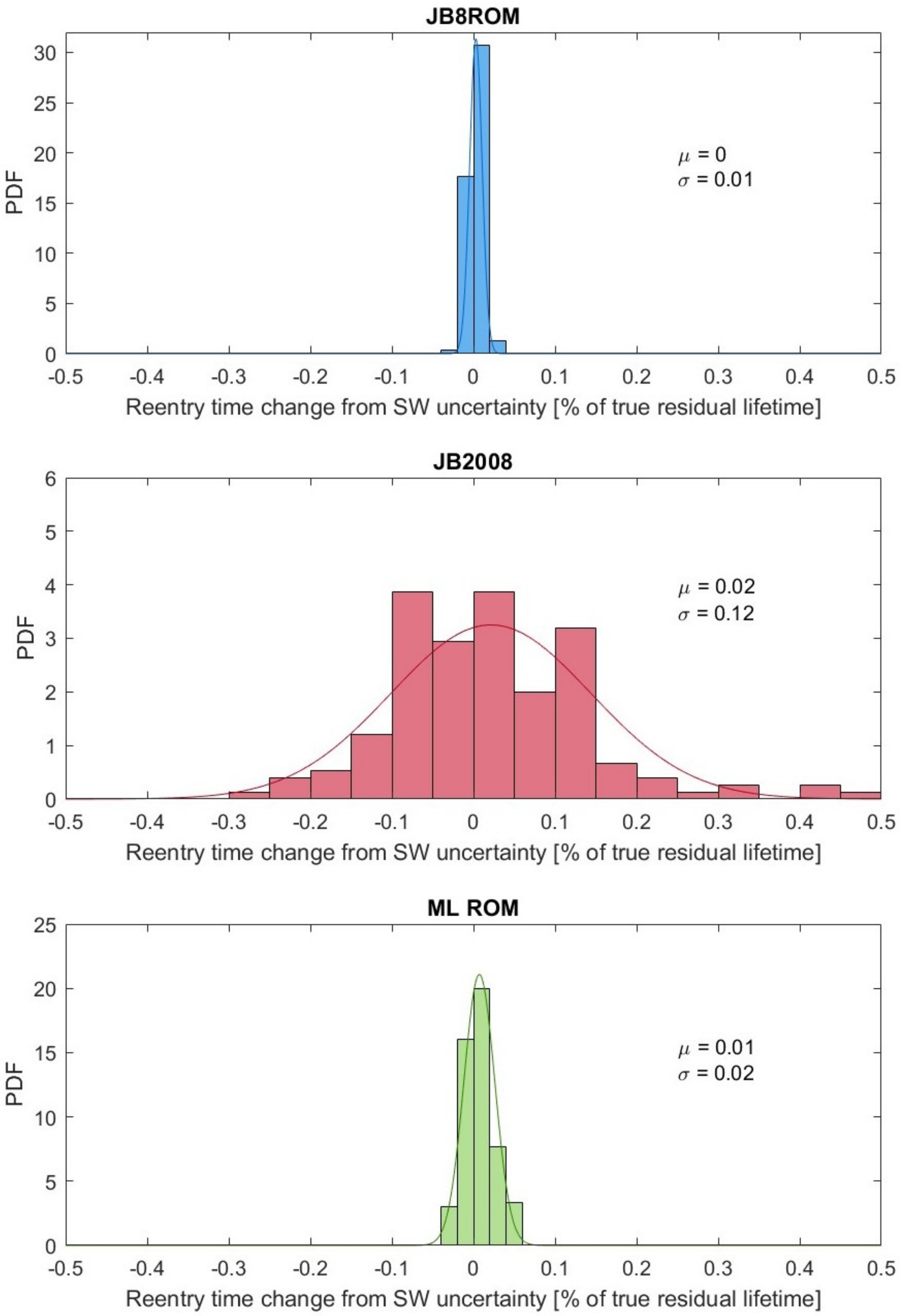


Figure 4-10: Distribution of change in reentry time predicted by three density models (no noise in ROM initialization) during $N = 150$ Monte Carlo simulation with Gaussian zero-centered noise added to F10.7, S10.7, M10.7, and Y10.7. Change in reentry time given relative to unperturbed reentry time predicted by each respective model, as a percent of true residual lifetime.

characteristic could prove very useful in density forecasting, when only predicted SW inputs, which contain error, are available as density model inputs.

There is a caveat: this analysis assumes 1) zero-centered SW noise (equivalent to no average bias in SW predictions) and 2) no correlation in errors among the perturbed SW inputs. Uncertainty quantification of SW index predictions is out of the scope of this thesis. However, error bias and correlation among predicted SW inputs are possible, in which case the density model SW UQ simulations presented here should be rerun with an updated SW index uncertainty model.

4.4 Summary of Uncertainty Quantification

In this chapter, UQ analysis was performed for three significant sources of uncertainty in reentry prediction: satellite initial positions (derived from TLEs), BCs, and SW inputs. By far the most important uncertainty source of the three was the ballistic coefficient. Uncertainties of 25% in the BC led to residual lifetime uncertainty window bounds of -19% and +32%, on average. Although 25% may be too high an estimate of BC uncertainty [75], the strong influence of BC on reentry time predictions is well documented [94, 26]. Ballistic coefficients which were too low were found to yield greater reentry time errors than BCs which were too high.

The second most influential uncertainty source on reentry time was the initial position of the satellite, which was examined with MC analysis using physically reasonable values of LEO TLE position uncertainty from [22]. Over 3-day prediction windows, satellite initial-state uncertainty led to 1-sigma residual lifetime uncertainties of around 2%. No significant reentry time bias was produced by TLE position uncertainty. The JB8ROM was slightly more sensitive to satellite initial-state uncertainty than the JB2008 and ML ROM.

Finally, SW uncertainty was the smallest uncertainty source, producing no more than 1% 1-sigma uncertainties over three days even when SW noise was included in ROM state initialization. SW-induced uncertainty had less impact on the JB2008 than on ROMs with noisy initial states. When SW noise was not included in ROM state initialization, ROMs were virtually unaffected by SW noise, showing 1-sigma uncertainties of $\sim$30-60 seconds (about 1/10 the magnitude of JB2008 1-sigma uncertainties). The SW analysis had the additional effect of demonstrating the strong, lasting impact of the initial ROM states on ROM density estimations.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Reentry Simulations

5.1 Upper Atmosphere Research Satellite (UARS)

The distinguishing feature of the UARS reentry was the relative stability of its ballistic coefficient throughout most of the reentry campaign, followed by a sudden BC decrease of almost 40% in the last two days of its lifetime [75]. Changes in BC during the reentry process are common during uncontrolled reentries [77] like that of UARS. The drag coefficient c_d may change as the satellite's orbital altitude decreases due to greater density and differences in the chemical composition of the lower atmosphere. Additionally, the satellite may begin to tumble, or components such as solar panels may break off, changing its effective frontal area and/or mass. The UARS' terminal drop in BC is captured in both the Pardini-Anselmo (PA) BC estimates and the Saunders-Gondelach (SG) BC estimates. Ballistic coefficient models used for UARS reentry simulations are plotted in Figure 5-1. SG BC estimates are also included for reference.

The interpretation of the plotted mean BC values is that the BC for a simulation starting in the x (time) range of each horizontal line is given by that line's vertical coordinate. For example, as SG mean BC simulation starting less than 7 and more

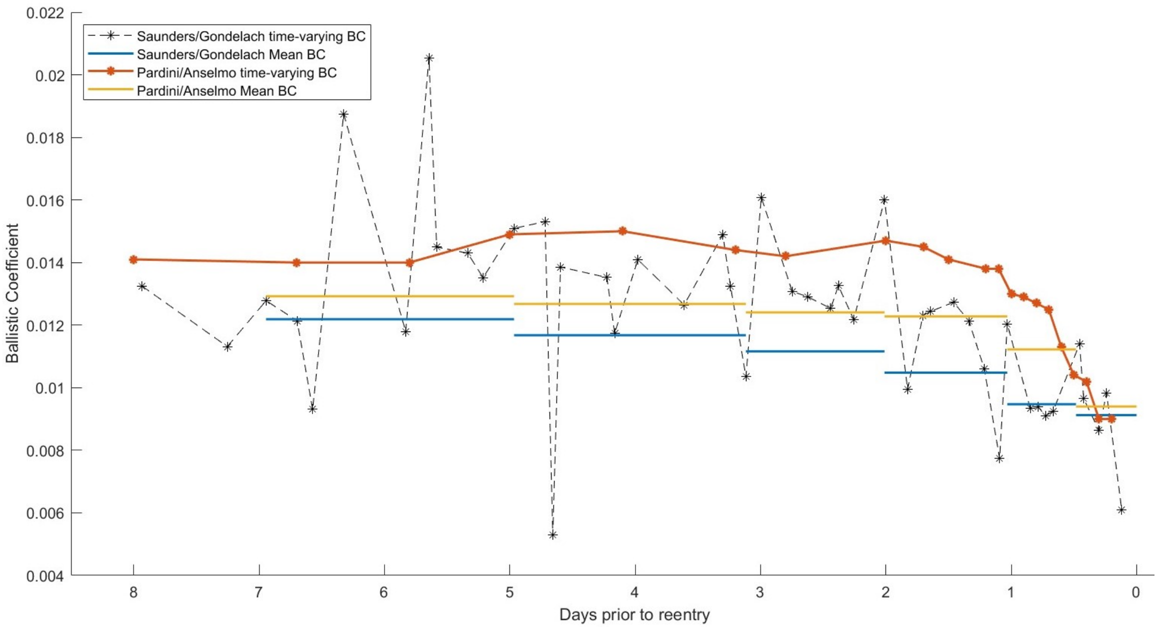


Figure 5-1: UARS ballistic coefficient models.

than 5 days prior to reentry would use the BC value of 0.012 indicated by the y-coordinate of the leftmost blue line. The mean BC value is invariant throughout the simulation, but its constant value depends on the start time of the simulation. The plotted lines indicate the constant BC value for simulations beginning in that time range. Note that each mean BC value is computed by taking the mean of all BC estimates between the simulation start time and the true reentry epoch; they are not rolling mean values.

Although the SG BC estimates (black) show greater variation than the PA estimates (orange), both sets of BC values capture the same general trend of relative stability followed by a terminal drop in BC. This trend is captured in the mean BC models, with the mean BC over the remaining simulation lifetime dropping as the remaining lifetime decreases. The SG technique estimates lower BCs overall than the PA technique, and also indicates a more gradual terminal drop in BC. The PA time-varying BC model (orange) remains higher than either of the mean BC models until within the final 24 hours of the UARS lifetime, at which point it drops steeply.

Since ballistic coefficient is directly proportional to drag force, the SG mean BC can be expected to produce the lowest drag force estimates and thus the latest reentry times, while the PA time-varying BC will yield the earliest reentry times, until within the final 24 hours.

5.1.1 Saunders-Gondelach Mean BC

Figure 5-2 shows a snapshot of UARS reentry trajectories computed with each of the 6 density models using the SG mean BC, for 6 different prediction window durations ranging from nominal 7 days to 12 hours prior to the true reentry epoch. Altitudes shown are geocentric. Note that these durations are nominal. Simulations can only be started at times when a TLE initial state is available, and TLE availability is non-uniform in time. Simulations are run from the nearest available TLE to the nominal start time of the simulation, as explained in Section 3.3.2. True reentry epoch is indicated on each plot with a vertical dotted line.

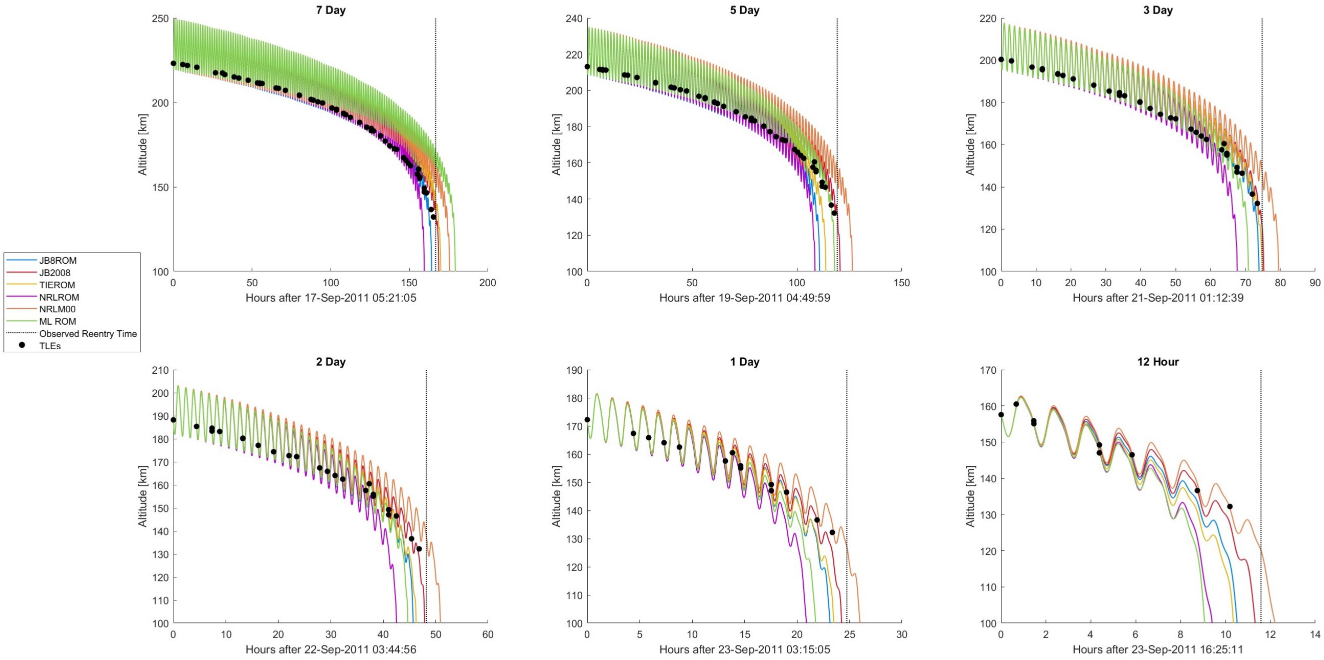


Figure 5-2: UARS reentry trajectories predicted with the Saunders-Gondelach mean BC using 6 different atmospheric density models and 6 different simulation durations.

Using the SG mean BC, reentry predictions of the 6 models center close to the true reentry epoch for all simulation durations. No significant uniform trend towards overestimation or underestimation emerges from this first-glance look at UARS results, which may indicate that the BC does not contain a significant bias. Figures 5-3 and 5-4 shows percent and absolute error of the density models over time. Note that negative percentages in Figure 5-3 indicate a predicted entry time prior to the true reentry, while positive values indicate a predicted reentry time later than the true reentry. For the UARS, Pardini and Anselmo found mostly negative prediction error (predicted reentry time was earlier than true reentry epoch) [75]. the predictions shown in Figs. 5-3 and 5-4 validate Pardini and Anselmo's finding.

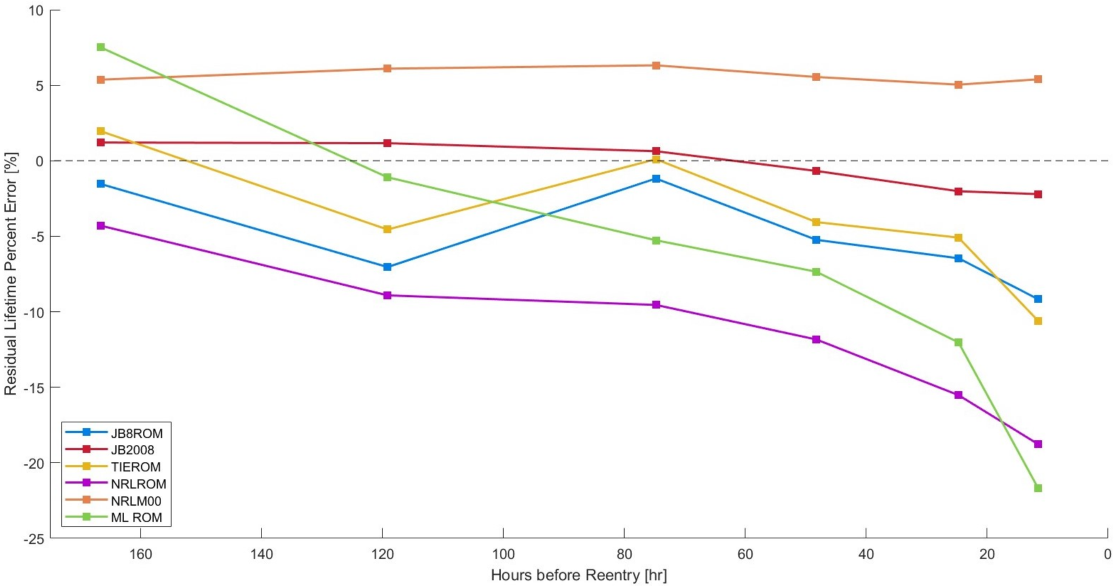


Figure 5-3: Percent error (%) of each model over time using the Saunders-Gondelach mean BC.

Using the SG mean BC, the best-performing model overall is the JB2008, which demonstrates a maximum absolute percent error of 2.2% at 12 hours prior to reentry. Up until the final prediction at 12 hours prior to reentry, the overall runner-up is the TIEROM, with a maximum error of 5.1% and mean absolute percent error (MAPE) of 3.2% as compared to the third-place NRLM00 maximum error of 6.3% and MAPE

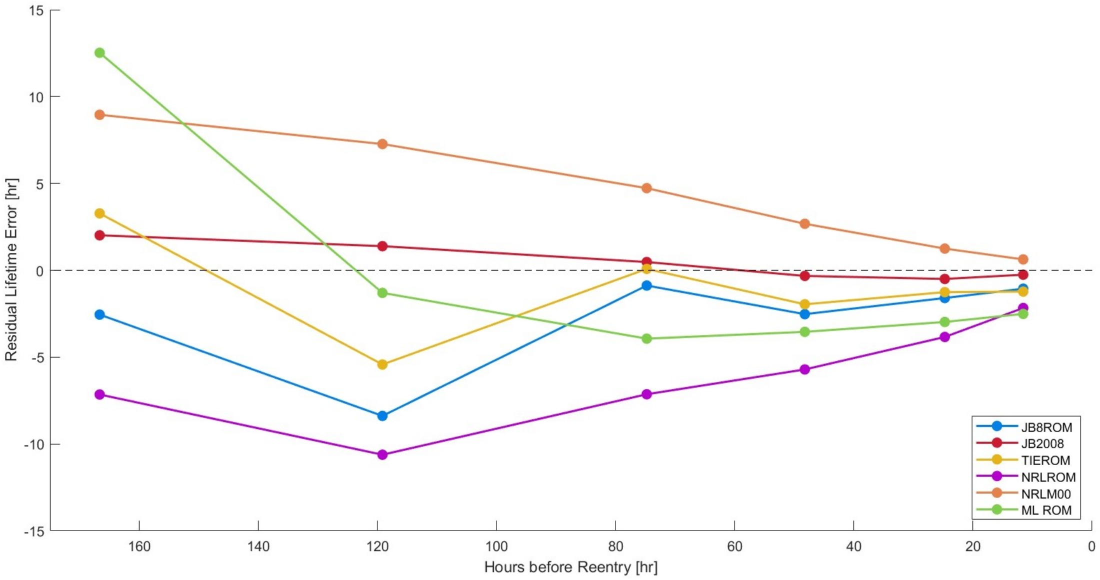


Figure 5-4: Absolute error (hrs) of each model over time using the Saunders-Gondelach mean BC.

of 5.7% over the same time frame. However, the performance of the TIEROM suffers significantly at 12 hours prior to reentry, with its error more than doubling to 10.6%, while the NRLM00 error remains low at 5.4%.

The JB8ROM follows the same performance trend as the TIEROM, at a consistent negative bias of equal to 1-3% up to 12 hours prior to reentry. The physical interpretation of this is that the JB8ROM consistently predicts reentries earlier than the TIEROM by an amount 1-3% of the true residual lifetime. At 12 hours prior to reentry, the JB8ROM is slightly better than the TIEROM, with an error of 9.2%.

The ML ROM and the NRLROM perform worst overall, but still provide reasonably low percent error (under 10%) for longer simulation time frames of 3+ days (for the NRLROM) and 2+ days (for the ML ROM). In the final 2 days of the UARS lifetime, the error of both models increases to around 20%. Both the NRLROM and ML ROM also experience an increasing trend towards underestimation of the true reentry epoch, in terms of percent error, as the prediction window decreases.

In terms of absolute error, all of the models do show a general convergence towards the true reentry epoch as prediction window decreases, as shown in Fig. 5-4. In addition, all of the density models meet the ±20% error benchmark for "competitive performance," as established in Section 2.1, across all prediction windows. The sole exception is the ML ROM at 12 hours prior to reentry (21.7% error). In addition, three of the models, the JB2008, NRLM00, and JB8ROM, meet the "good performance" benchmark of ±10% error across all prediction windows, and the TIEROM only fractionally misses out on this benchmark with 10.6% error at 12 hours before reentry. Finally, only the JB2008 model meets the ±5% metric for "excellent performance" for all simulation durations. The TIEROM meets this standard for simulation durations of 2 days or longer. The NRLM00 and JB8ROM are close contenders, with errors of 5-6.3% and 1.2-9.2%, respectively.

5.1.2 Pardini-Anselmo Mean BC

Figure 5-5 shows the UARS trajectories predicted by the six density models using the PA mean BC.

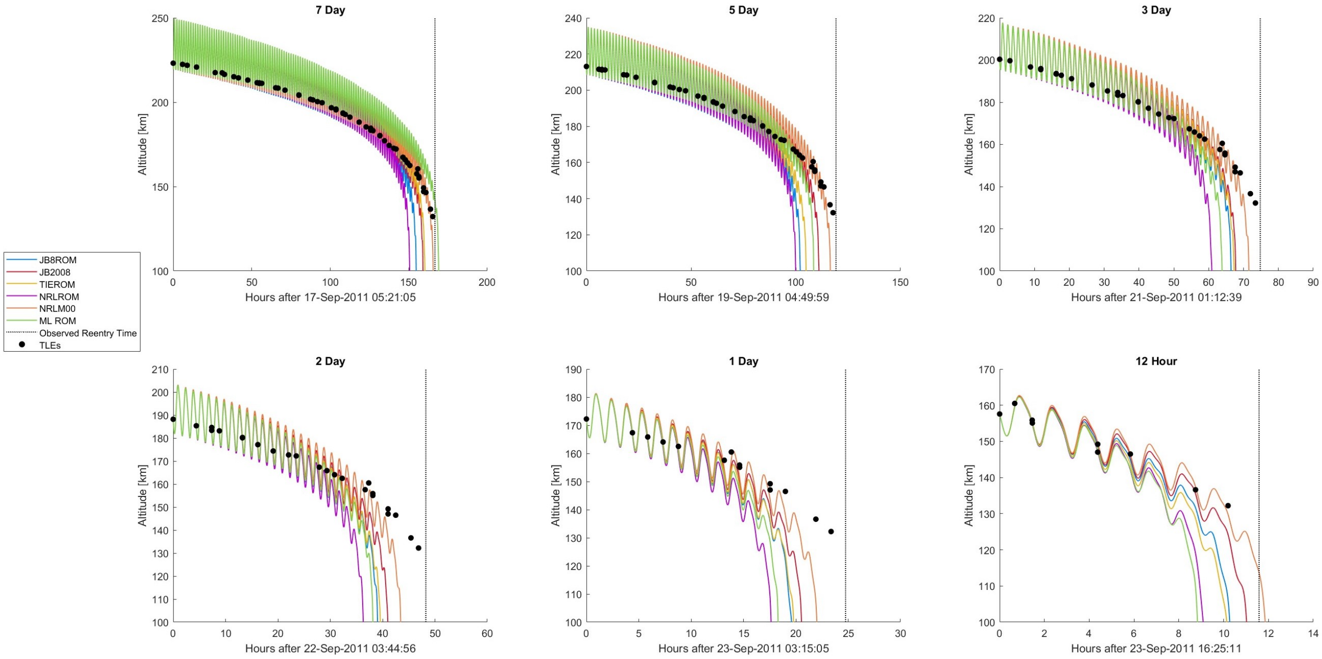


Figure 5-5: UARS reentry trajectories predicted with the Pardini-Anselmo mean BC using 6 different atmospheric density models and 6 different simulation durations.

Because the PA mean BC is higher than the SG mean BC, it is no surprise that the density models predict earlier reentry times with this BC.

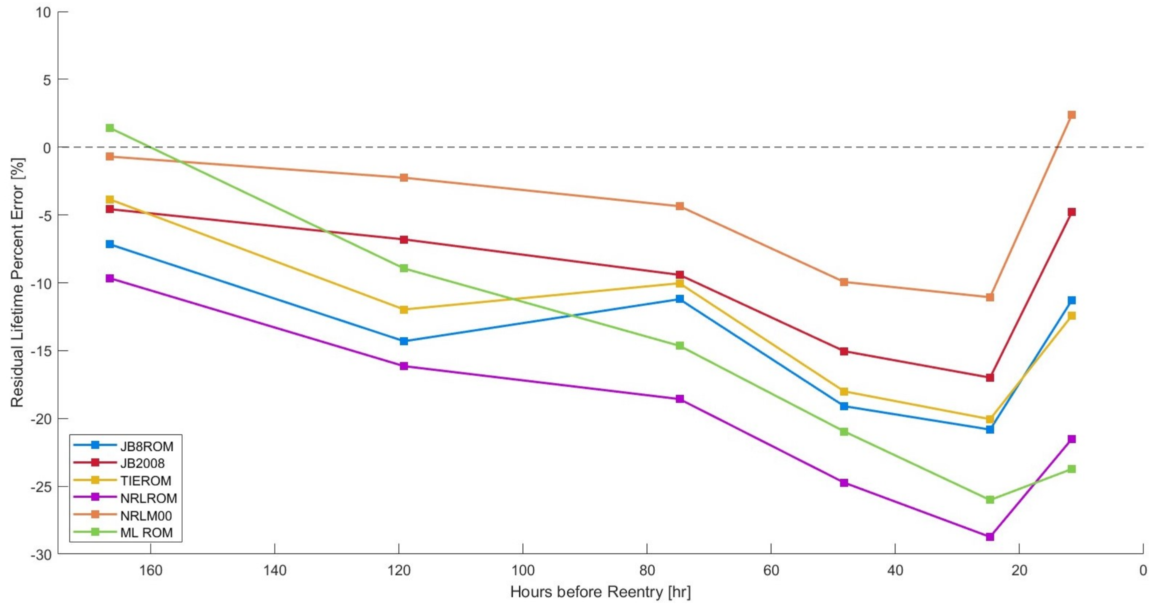


Figure 5-6: Percent error (%) of each model over time using the Pardini-Anselmo mean BC.

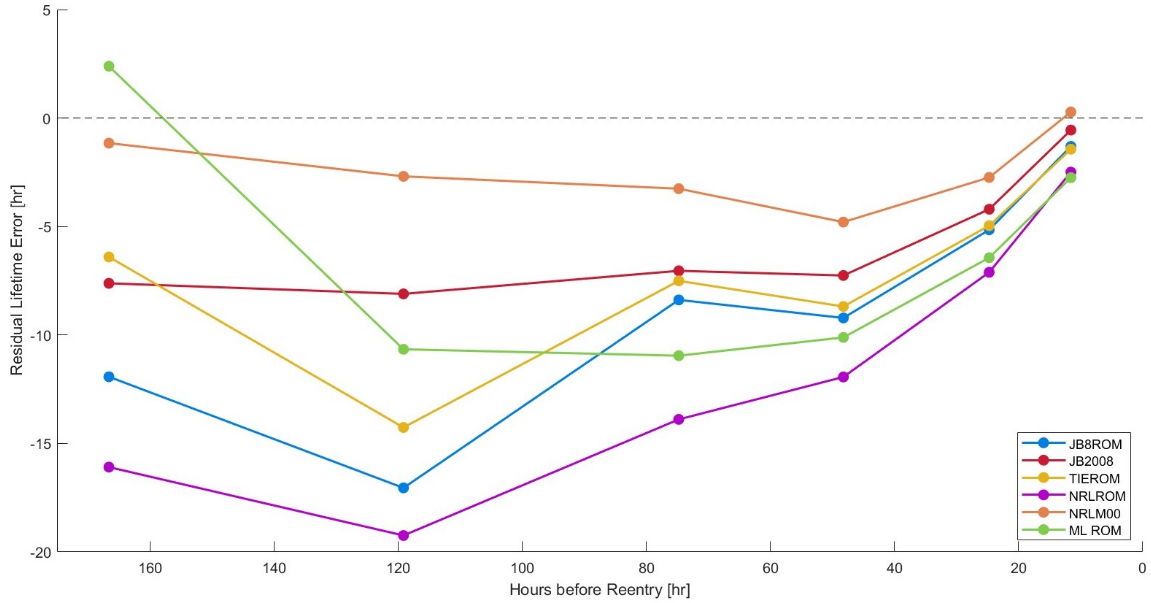


Figure 5-7: Absolute error (hrs) of each model over time using the Pardini-Anselmo mean BC.

Overall percent error for all of the density models is generally higher for this BC model. Even the best-performing model, the NRLM00, exhibits errors of 10% or more for some simulation durations. All of the models show greatest relative error at 1 day prior to reentry, which is also the time at which the PA mean BC has the greatest disagreement with the SG mean BC. All models experience an improvement in performance at 12 hours before reentry, at which time the PA mean BC comes into closer agreement with the SG mean BC. The steeper and later terminal decrease in the PA BCs means that the average BC for the simulation duration does not fully reflect the drop in BC until very shortly before reentry.

For all prediction windows, the JB2008 follows the same performance trend as the NRLM00, with a 4-7% negative (underestimation) bias. Similarly, the JB8ROM follows the TIEROM at a slight negative bias until 12 hours before reentry, as in the SG mean BC results. Once again, the ROMs demonstrate their best performance over longer prediction windows. At 7 days prior to reentry, both the TIEROM (3.8%) and the ML ROM (1.4%) give lower error than the JB2008 (4.6%). However, ROM error is higher for shorter prediction windows, especially for the ML ROM and NRLROM, which experience maximum errors of 26% and 28.7%, respectively.

Almost all of the models underestimate the reentry at almost all times. In fact, only two individual predictions (ML ROM at 7 days and NRLM00 at 12 hours) overshoot the true reentry epoch. Again, this is consistent with Pardini and Anselmo's results [75]. This could indicate either 1) a consistent tendency of all the density models to overestimate densities, or 2) a tendency of the PA BC to overestimate the true BC. Further testing is needed to distinguish which of these is the most probable cause.

Although Fig. 5-7 shows that absolute error does converge towards zero for all of the models as the prediction window decreases, this convergence is not consistent over time for any of the models. The PODROMs all experience their highest absolute error at 5 days prior to reentry, and the absolute error of JB8ROM and TIEROM improves briefly at 3 days before deteriorating again at 2 days. Even the NRLM00 converges inconsistently, with its worst absolute error at 2 days prior to reentry.

The JB2008, NRLM00, and TIEROM all exhibit sub-20% error over all prediction windows, with the JB8ROM just missing this standard with 20.1% error at 1 day prior to reentry. However, none of the models have errors of less than 10% at all times.

5.1.3 Pardini-Anselmo time-varying BC

Figure 5-8 shows the predicted reentry trajectories for UARS using the PA time-varying BC model.

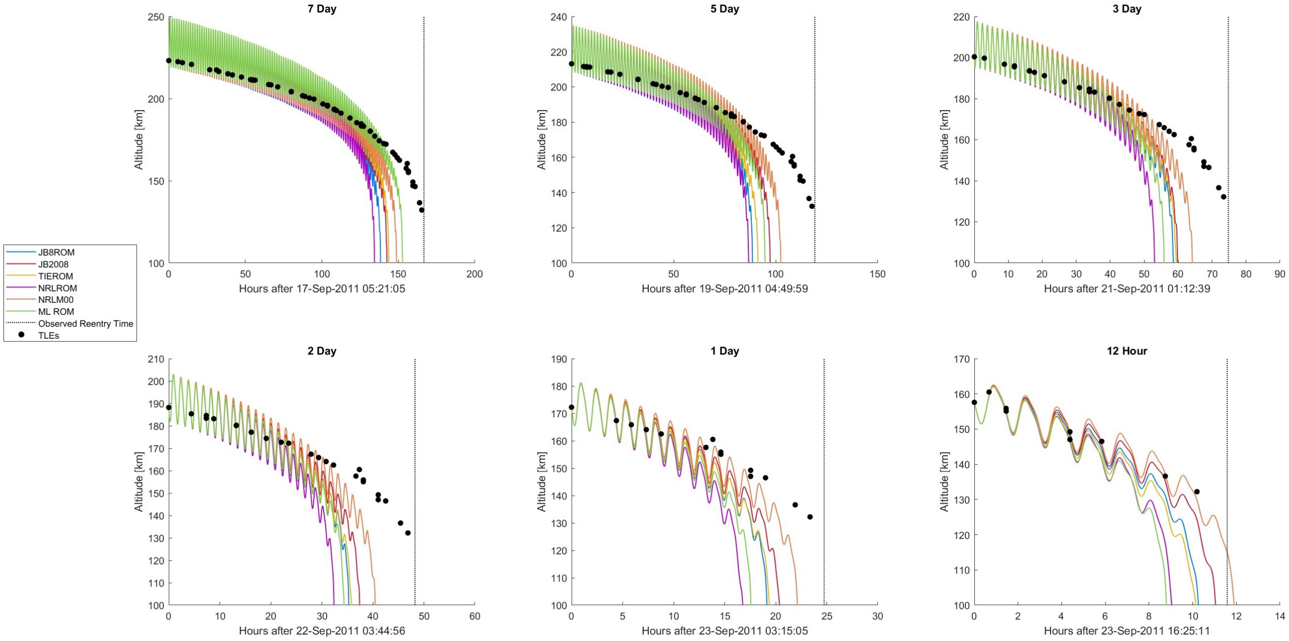


Figure 5-8: UARS reentry trajectories predicted with the Pardini-Anselmo time-varying BC using 6 different atmospheric density models and 6 different simulation durations.

Reentries predicted with this model are even more premature than those predicted with the PA mean BC, up until 24 hours prior to reentry. This reflects the fact that the PA time-varying BC remains higher than the PA mean BC until shortly before the reentry, because the terminal BC drop influences the mean BC for all simulation durations. Simulations using the time-varying BC do not experience this drop until the last few days prior to reentry.

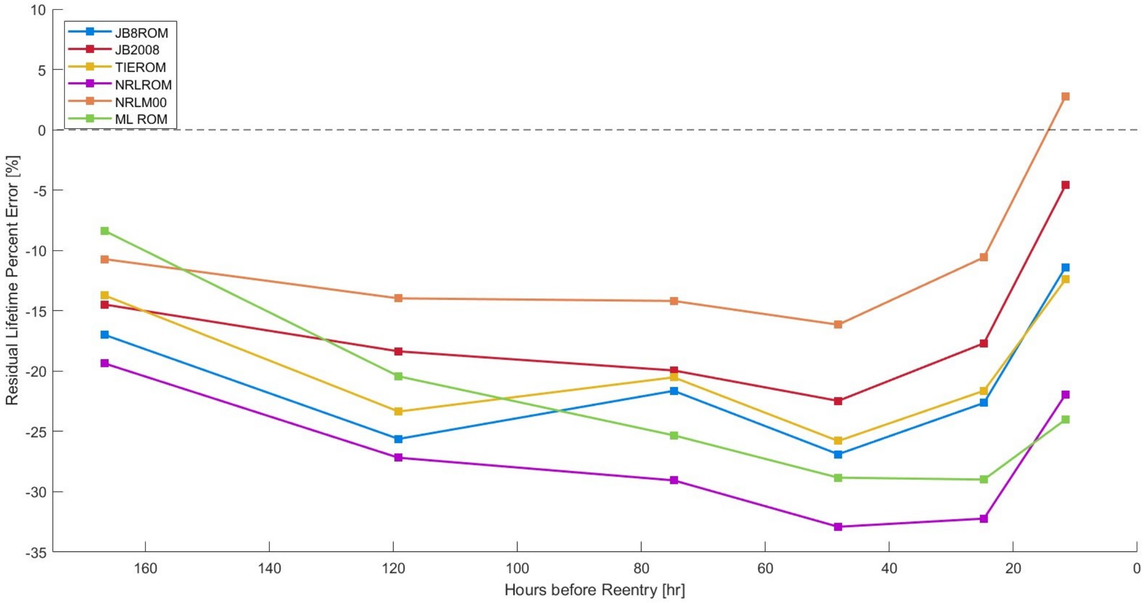


Figure 5-9: Percent error (%) of each model over time using the Pardini-Anselmo time-varying BC.

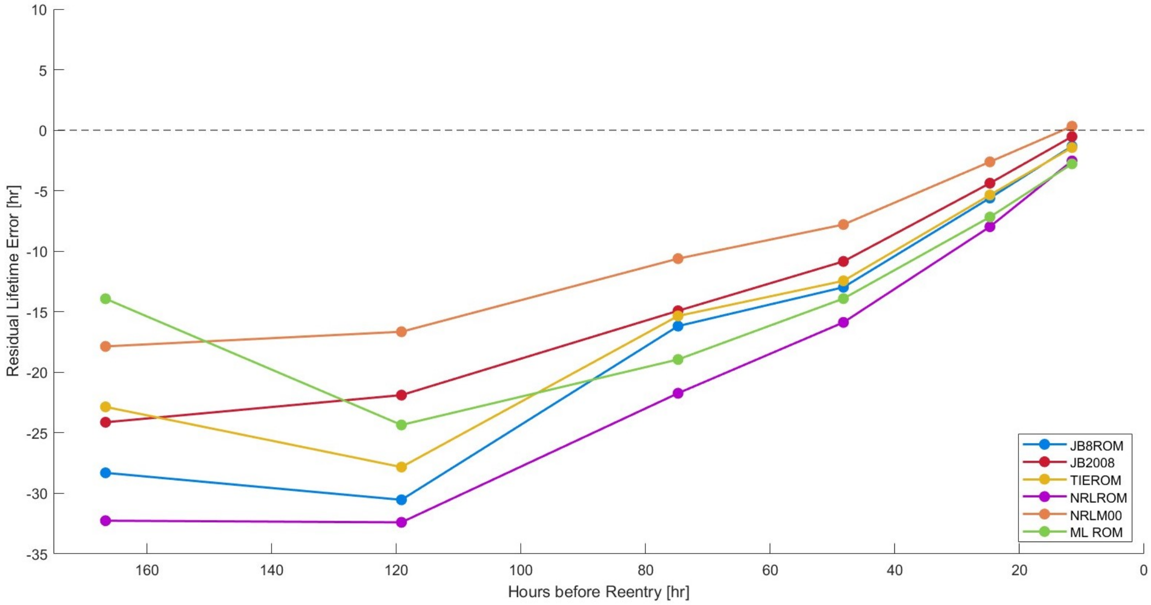


Figure 5-10: Absolute error (hrs) of each model over time using the Pardini-Anselmo time-varying BC.

The primary difference between the mean and time-varying PA BC results is the poorer performance of all models over longer prediction windows using the time-varying BC. The time-varying BC is higher for longer simulation durations, resulting in underestimation of the true residual lifetime by up to 32 hours (NRLROM) at 7 days prior to reentry. For longer simulation durations, the generally higher time-varying BC causes many of the models to predict reentry before the terminal BC drop even begins, so this BC feature is entirely missed in some simulations. All of the ROMs experience their highest absolute error at 5 days prior to reentry, after which all the models converge steadily towards the true reentry epoch in terms of absolute error, as the terminal drop in BC begins to affect the simulation results.

The highest relative error for all density models except the ML ROM occurs at 2 days prior to reentry, while the ML ROM experiences slightly greater percent error at 1 day prior to reentry. Starting at 1 day prior to reentry, the time-varying BC results converge closely to those of the PA mean BC, since at this point the two BC models are very close (see Fig. 5-1).

Apart from the higher error for longer simulation durations, there are no major differences between the results obtained with the PA mean and time-varying BCs. All models follow the same general performance trends, and the rank of model performance for each prediction window is the same as in the mean BC case. Results obtained with the PA time-varying BC were relatively poor for all models. Only the NRLM00 (which was used in the CDFIT BC estimation routine) meets the ±20% error benchmark for all simulation durations, and only 3 individual predictions feature error below 10% (ML ROM at 7 days, and NRLM00/JB2008 at 12 hours).

5.1.4 UARS Overall Results

Figure 5-11 shows the mean absolute percent error across all simulation durations, for each density model and each BC model.

For the UARS test case, the SG mean BC model was the most successful for

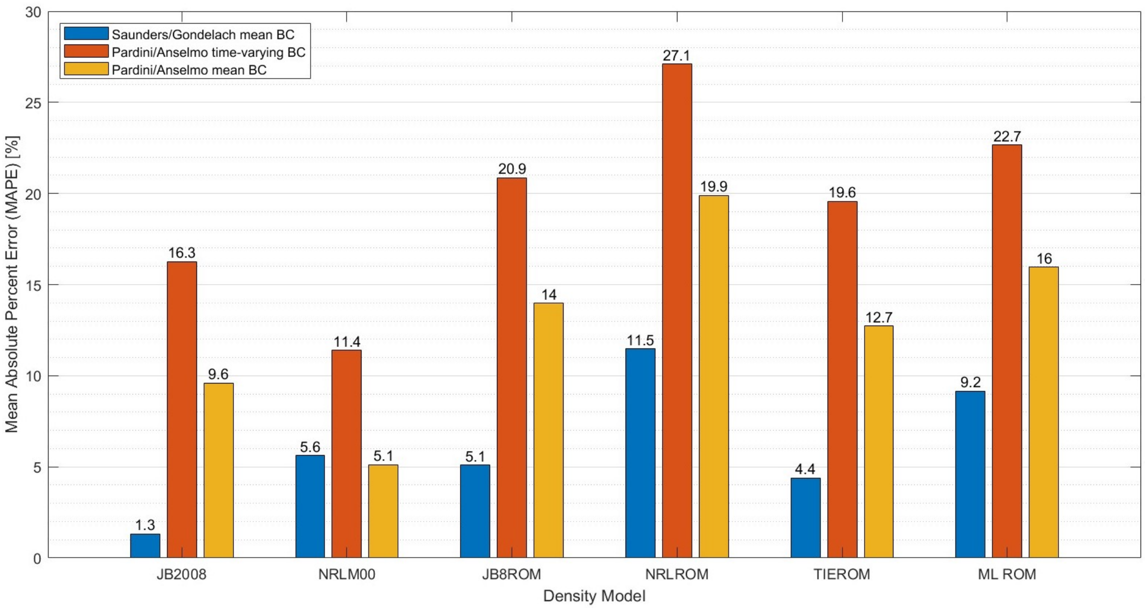


Figure 5-11: Average residual lifetime prediction MAPE across all simulation durations, for all density models and all BC models, for the UARS test case.

most density models by a wide margin. The only exception is the NRLM00, which performed marginally better with the PA mean BC. All models demonstrated their worst performance with the PA time-varying BC. In most cases, all density models reacted similarly to all BC models, meaning that a BC model change either improved performance for most density models, or worsened performance across most models. The NRLM00 was the exception, as mentioned earlier. This likely stems from the use of NRLM00 in BC estimation for all of the BC models. Depending on the NRLM00 error bias during the BC fitting window, and the particulars of the BC estimation algorithm, it is likely that some BC models absorbed more NRLM00 error than others. The NRLM00 would tend to perform better using BCs which compensated for NRLM00 inaccuracies due to density model error absorbed during fitting. On the other hand, non-NRLM00 density models are expected to suffer from the presence of NRLM00 error in the BC.

The best performance observed across all density-BC model combinations for the UARS test case was the JB2008 with the SG mean BC model, which achieved an impressively low 1.3% MAPE across all prediction windows. The second-best density model with this BC was the TIEROM, with a MAPE of 4.4%, and the JB8ROM at 5.1%. The NRLM00 achieved good performance with both mean BC models (5.6% and 5.1%), but it is uncertain to what extent these results are due to the NRLM00's advantage resulting from its use in BC estimation. All the models except the NRLROM achieved the ±10% benchmark, in terms of MAPE, using the SG BC. The NRLROM missed this standard only slightly, with a MAPE of 11.5%. This means that most of the density models are showing good performance relative to typical reentry prediction accuracy, as described in Section 2.1. All models met the ±20% standard in terms of MAPE using both mean BC models; however, only half the models (JB2008, NRLM00, and TIEROM) achieved this standard with the time-varying BC. The TIEROM had very similar MAPE performance to the JB8ROM. These two density models agreed in MAPE to within 1.5% for all BC models.

Results are further split into short prediction windows (1 nominal day or less) and long prediction windows (nominal 2+ days).

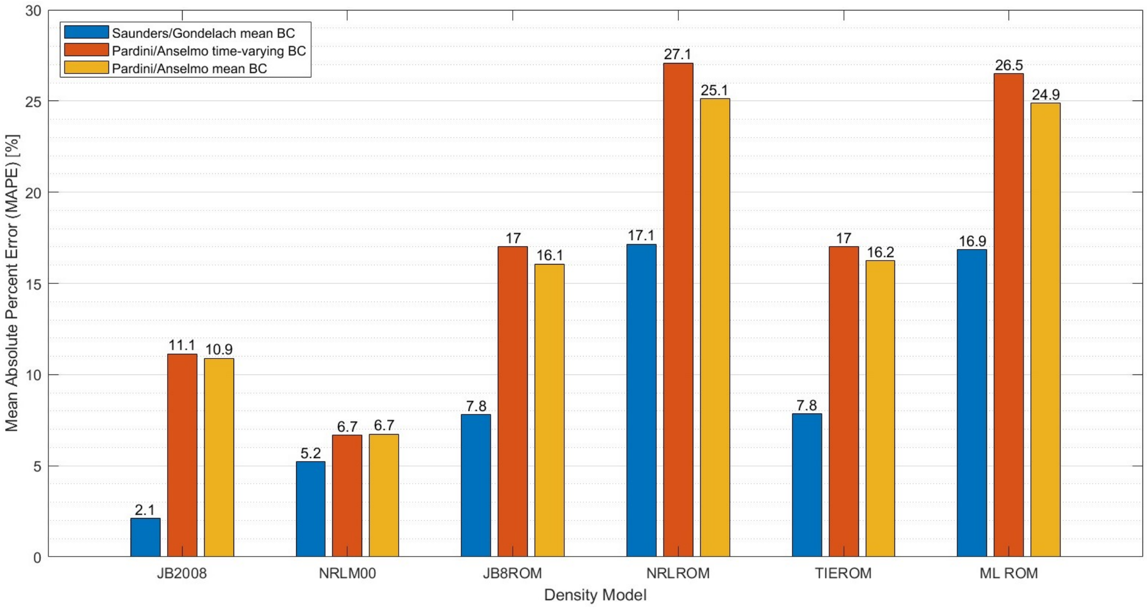


Figure 5-12: UARS residual lifetime prediction MAPE for all density and BC models for short prediction windows of one nominal day or less.

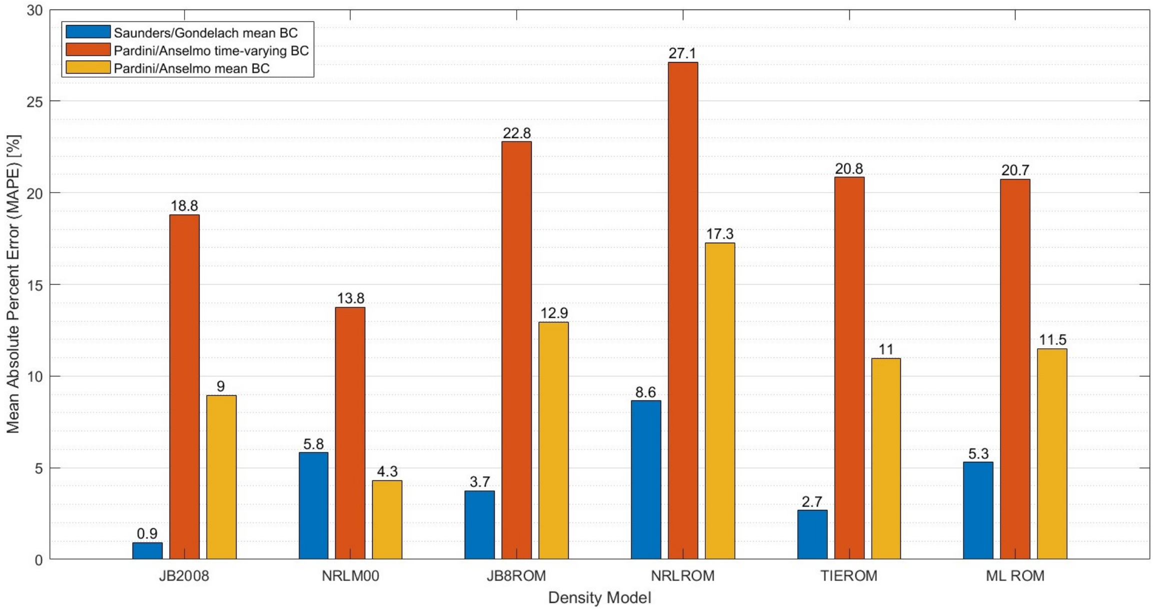
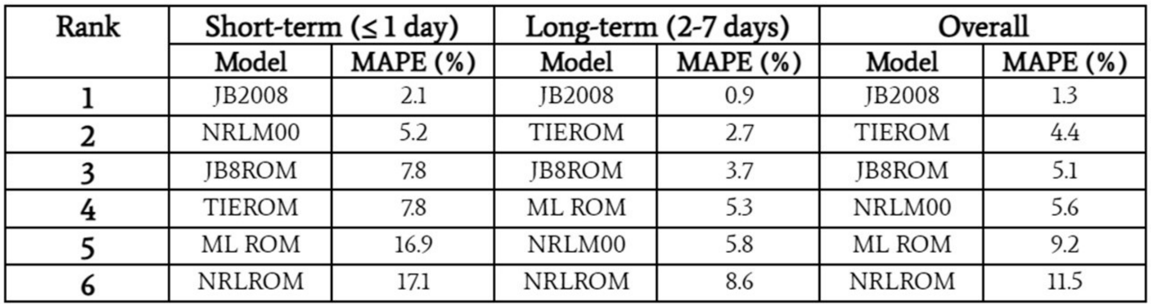


Figure 5-13: UARS residual lifetime prediction MAPE for all density and BC models for longer prediction windows of 2-7 nominal days.

For both mean BC models, all of the ROMs have lower MAPE for longer prediction windows. The improvement in ROM MAPE for long prediction windows over short windows with the mean BC models ranged from 3.2% (JB8ROM with PA mean BC) to 13.4% (ML ROM with PA mean BC). The same trend was not observed for the time-varying BC, where most ROMs did as well or better on shorter simulations. The difference across BC models is likely because longer simulations using the higher time-varying BC often yielded reentries prior to the terminal BC drop, causing larger prediction errors.

Like the ROMs, the JB2008 model also did better over longer prediction windows for the mean BC models. The performance difference was smaller (1-2%) than for most of the ROMs; however, the JB2008 MAPE was the lowest of all the models for both short and long prediction windows. Both the JB2008 and the NRLM00 did relatively worse on longer simulations with the time-varying BC. The NRLM00 had low MAPE for longer simulations with the PA mean BC, but experienced greater MAPE for long simulations with the SG mean BC.

Table 5.1: Density models ranked by MAPE performance on UARS test case with Saunders-Gondelach mean BC.



Rank	Short-term (≤ 1 day)		Long-term (2-7 days)		Overall	
	Model	MAPE (%)	Model	MAPE (%)	Model	MAPE (%)
1	JB2008	2.1	JB2008	0.9	JB2008	1.3
2	NRLM00	5.2	TIEROM	2.7	TIEROM	4.4
3	JB8ROM	7.8	JB8ROM	3.7	JB8ROM	5.1
4	TIEROM	7.8	ML ROM	5.3	NRLM00	5.6
5	ML ROM	16.9	NRLM00	5.8	ML ROM	9.2
6	NRLROM	17.1	NRLROM	8.6	NRLROM	11.5

Table 5.1 ranks models by MAPE for short-duration, long-duration, and overall performance for the UARS simulations, using the overall best-performing BC model, the SG mean BC.

Density model performance by run time is shown in Figure 5-14, averaged across all simulation durations and all BC models, in units of seconds of run time per hour of simulation.

Note that the run times shown are for the entire propagation process with each model, and are *not* run times for the density models alone. On average, orbit propagation with the PODROMs is an impressive 70.4% faster than the empirical JB2008 and NRLM00 models, while the ML ROM is an average of 28.4% faster. The ML ROM used in this thesis was not optimized for run time, which could make the model faster. The run time advantage of the ROMs is most pronounced over long time frames, when the overhead cost of model setup is negligible. For very short simulations (nominal 12 hours), the ML ROM run time is approximately on par with the empirical model run times due to the overhead of ML ROM setup. For more details and complete run time plots, see Appendix A.

In light of the fact that the ROMs performed better on longer simulations than shorter simulations, and that two of the PODROMs met the ±5% error in terms of MAPE for long simulations with the SG mean BC, the fast run time makes ROMs a strong prospect for longer reentry simulations. Although the JB2008 retains an advantage of 1.8-2.8% MAPE over the top-performing ROMs for long reentry simulations with the SG BC, it also takes an average of 9.7 hours to run a 1-week simulation with JB2008 using HPOP. The JB8ROM or TIEROM can run the same simulation in about 2.5 hours.

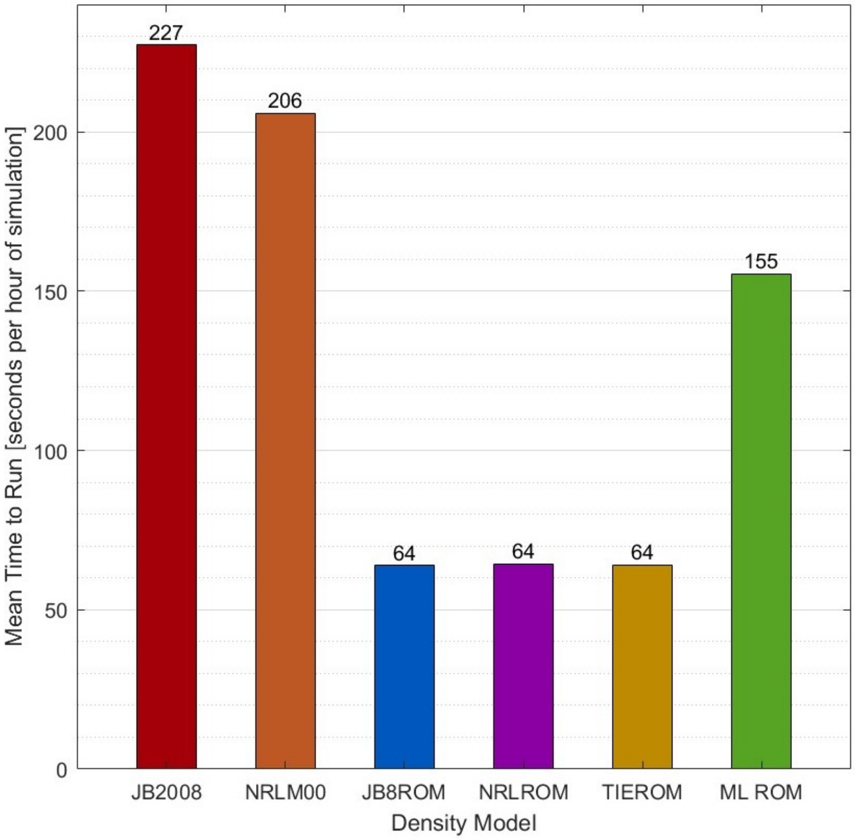


Figure 5-14: Average HPOP run time by density model for UARS test case.

From an operational perspective, this represents the ability to run 3-4 orbit propagations with a PODROM in the time it would take to run 1 propagation with JB2008 or NRLMSISE-00, with only a small tradeoff in accuracy (especially over long time frames). The increasing congestion of LEO requires the ability to perform large amounts of orbit propagations on a continual basis to deconflict the trajectories of active satellites with other orbiting objects, such as debris and other dead or active satellites. The fast performance of the ROMs, at comparable accuracy to prevailing empirical models, provides greater propagation ability per unit of available run time. This gives an opportunity to perform orbit propagation for reentry prediction or conjunction analysis a) over longer time windows, b) more frequently, c) for a greater number of objects, or d) some combination of the above, while still requiring less total run time than needed to use JB2008/NRLM00 for shorter, less frequent, and fewer analyses. Achieving this performance does require a suitable BC model, as evidenced in the varying efficacy of ROMs depending on BC. However, the necessity for good BC estimation is not a problem unique to ROMs. Saunders et. al. (2009) found in a study of 4 reentries that an accurate BC was "the prime factor in accurately predicting re-entry dates" [94].

5.1.5 A Note on Reentry Locations

An example of the sensitivity of reentry location to reentry time may be useful to illustrate the challenge of ground impact risk assessment. Figure 5-15 shows plots of predicted ground tracks and 100-km reentry points for the UARS test case, with the SG mean BC. Only three density models (JB8ROM, JB2008, and ML ROM) are shown in order to maintain plot clarity. The final few orbit ground tracks and the 100-km reentry point for each model are shown for 7-day and 12-hour predictions. As Figure 5-15 shows, major differences in reentry location can result from relatively minor differences in reentry time, such as the 48-minute difference between the 12-hour predictions of JB8ROM and JB2008.

Note that the plotted reentry locations should not be taken as ground impact locations. Surviving pieces of the UARS would travel significant distances over the final 100 km of altitude before ground impact. Figure 5-15 is presented only to provide context on the extreme sensitivity of reentry location to reentry time, due to the high orbital velocities in LEO.

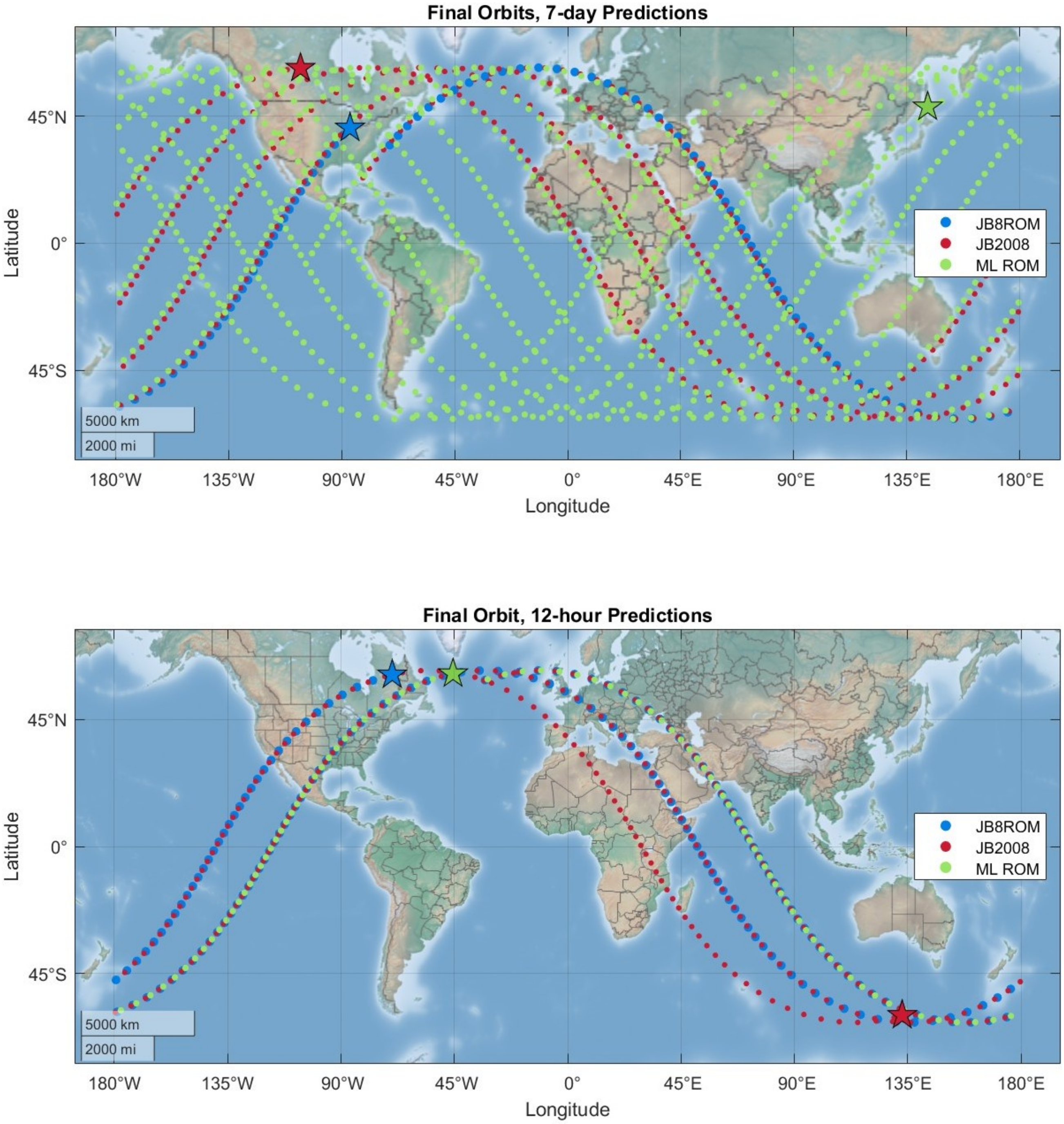


Figure 5-15: Ground tracks and 100-km reentry locations (starred) predicted for UARS.

5.2 Fobos-Grunt

The Fobos-Grunt ballistic coefficient proved challenging to model with the SG technique, which produced a non-physical (negative) BC value at approximately 1 day prior to reentry, followed immediately by a sharp spike in BC. This behavior may be due to TLE error, despite the use of TLE filtering. The nonphysical BC value was not included in the mean SG BC model. This exclusion was partially responsible for the different trends in BC between the SG BC model and the PA BC models shortly prior to reentry, when the SG method shows elevated BCs while the PA method predicted a slight drop in BC, as seen in Figure 5-16. This led to disagreement between the BC models, especially between 3 days and 12 hours prior to reentry. A plot of the BC models is shown in Figure 5-16, along with the SG BC estimates for reference.

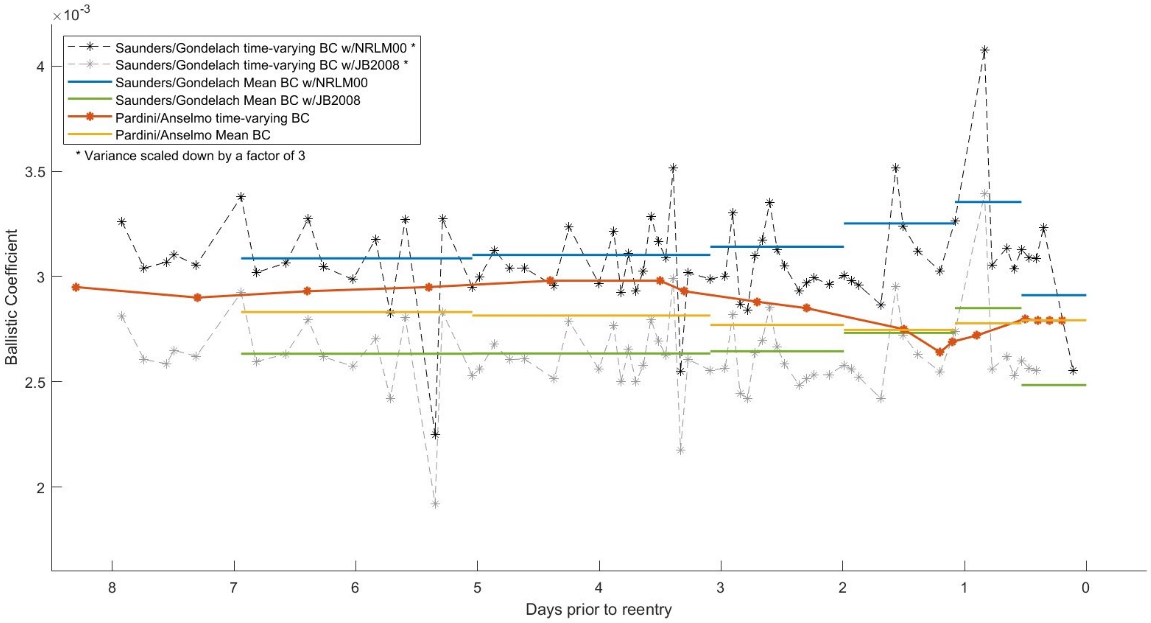


Figure 5-16: Fobos-Grunt ballistic coefficient models.

Due to the difficulty in estimating a BC for this test case, one additional BC model was included for the Fobos-Grunt. BC values obtained using the SG BC estimates generated with NRLM00 showed an especially heavy bias towards the NRLM00 for the Fobos-Grunt case, so SG BC estimates were also generated using JB2008 as the density model in the AIDA propagator for BC estimation. The additional BC permits evaluation of how much of the non-NRLM00 density models' residual lifetime prediction error was due to absorbed NRLM00 bias in the BC. A mean BC model was produced with both the NRLM00-based and the JB2008-based estimates, and reentry simulation was performed for Fobos-Grunt using both BCs.

In Fig. 5-16, the variance of the SG BC estimates, using both NRLM00 and JB2008, was scaled down around their mean value by a factor of 3 to allow all BCs to be plotted on the same axes. (This scaling is why the raw SG estimates with JB2008 appear to be higher than their mean value, just prior to reentry.) Note that the reason for the JB2008-based estimates ending slightly before the NRLM00-based estimates with the SG technique is due to altitude restrictions on the MATLAB version of the JB2008 model used. Propagation from the final, low TLEs resulted in decay of the orbital altitude below the allowed altitude range of the model, and BC estimation could not continue. The interpretation of the mean BCs is the same as for UARS: the BC for a simulation *starting* in the x (time) range of each horizontal line is given by that line's y-coordinate.

The SG BC model with NRLM00 yields the highest BCs across all simulation durations, so this BC model is expected to give the earliest reentry times. Note that Pardini and Anselmo observed mostly negative reentry prediction errors for Fobos-Grunt in [75], meaning that reentry predictions using their BC estimates typically showed premature reentries, as in the UARS case. Therefore, the SG BC with NRLM00 can also be expected to yield premature reentry predictions, because it is even higher than the PA BC estimates.

Since Pardini and Anselmo also used NRLM00 in their BC estimation process, the trend they observed towards underestimation of residual lifetime could indicate that NRLM00 chronically underestimated density during the time period of the Fobos-

Grunt reentry, resulting in artificially elevated BCs from estimation techniques using NRLM00. To evaluate the extent to which the NRLM00 effects BCs estimated using that density model, an SG mean BC model using JB2008 instead of NRLM00 for the estimation process was included for the Fobos-Grunt.

As Fig. 5-16 shows, the BC estimates generated with JB2008 are significantly lower than any of the NRLM00-generated BCs for most of the prediction windows. Only at 1-2 days prior to reentry is the JB2008-generated BC model close to or slightly above the PA BCs, and it remains consistently well below the SG NRLM00 mean BC. In fact, the SG BCs estimated with JB2008 are, on average, 14% lower than those estimated with NRLM00! Clearly, the density model used in the BC estimation process does significantly impact the BC, as Saunders et. al. (2012) observed in [95]. These differences in BC have a significant impact on reentry predictions.

5.2.1 Saunders-Gondelach Mean BC with NRLM00

Figure 5-17 shows Fobos-Grunt reentry trajectories produced by each density model for each simulation duration using the SG mean BC. Altitudes shown are geocentric.

The SG mean BC, which is the highest BC overall, does indeed result in premature reentry predictions for most of the atmospheric density models. The one exception is NRLM00, which is the density model used for BC estimation. For every prediction window, the NRLM00 predicts reentry closest to the true reentry epoch by a large margin. As mentioned previously, these results may indicate that the higher BC is due to NRLM00 underestimating densities during this time frame, requiring a higher BC to fit the TLE data. Although NRLM00 is known to underestimate densities under some conditions, such as during geomagnetic storms [21], an examination of archived space weather data[1] revealed no storms or major geomagnetic disturbances during the week of 8-15 January, 2012. It is nonetheless probable, however, that the

[1] https://www.spaceweatherlive.com/en/archive.html

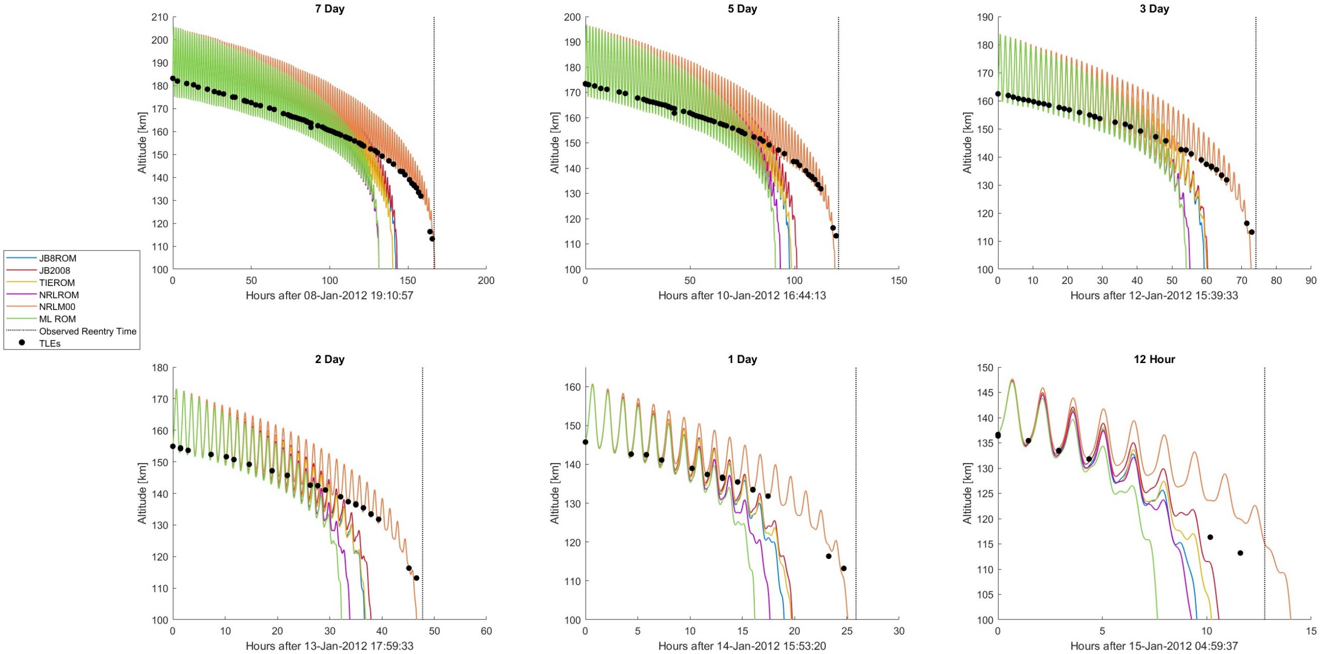


Figure 5-17: Fobos-Grunt reentry trajectories predicted with the Saunders-Gondelach mean BC using 6 different atmospheric density models and 6 different simulation durations.

excellent performance of the NRLM00 with the SG BC was due to absorbed density model error. This is a known feature of the SG method, and is the reason why Saunders et. al. recommend using the same density model for propagation as for BCs estimation when using this method [95].

Percent error and absolute error for each density model are shown in Figure 5-18 and 5-19.

As observed in Figure 5-17, the NRLM00 density model is the best-performing model for all prediction windows, with 3% error or less up until 1 day prior to reentry and a maximum error of 9.8% at 12 hours prior to reentry. In comparison, the next-best individual predictions of any other models using this BC were the 7-day predictions for JB2008, JB8ROM, and TIEROM, which were in the range of 14.2-15.8% error. All of the PODROMs and the JB2008 experienced their worst predictions in terms of percent error at 1 day prior to reentry, ranging from 23-24% error for the JB2008 and TIEROM up to almost 32% for the NRLROM. The ML ROM, which

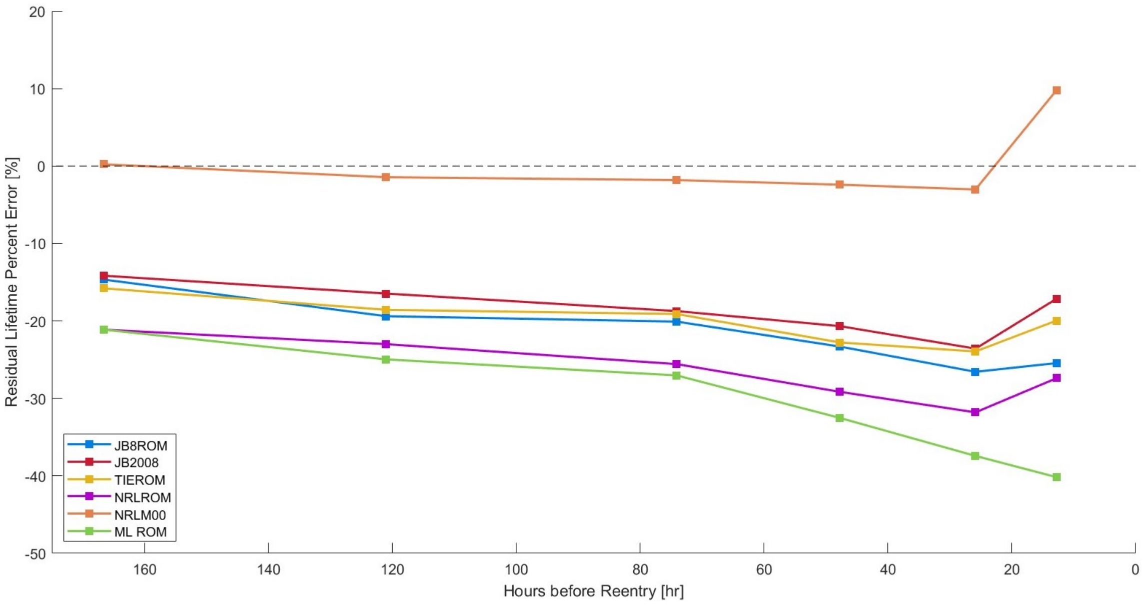


Figure 5-18: Percent error (%) of each model over time using the Saunders-Gondelach mean BC.

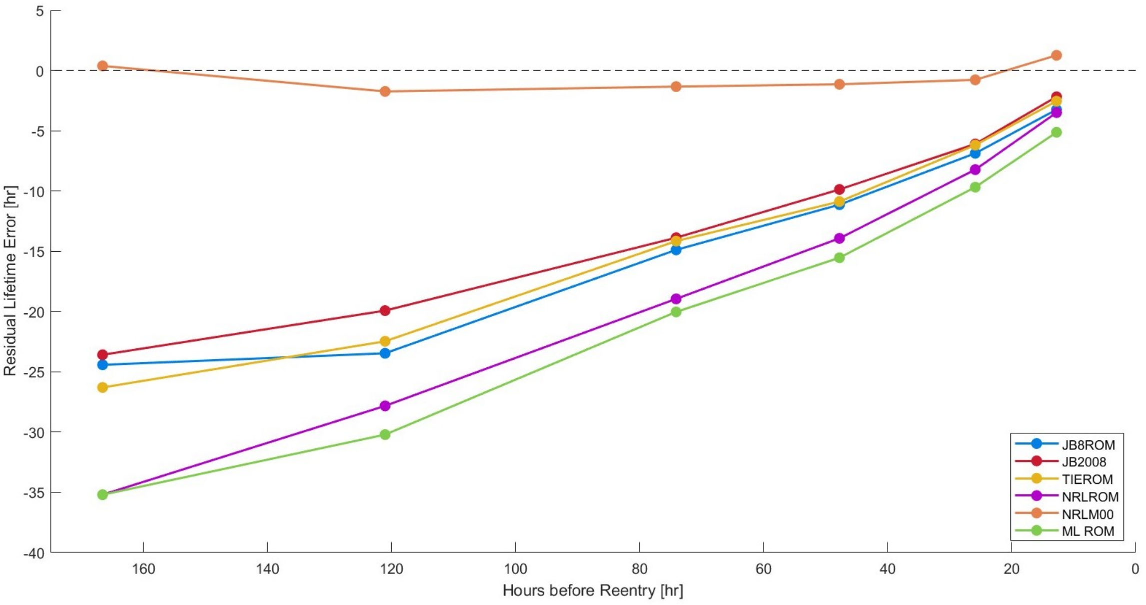


Figure 5-19: Absolute error (hrs) of each model over time using the Saunders-Gondelach mean BC.

was overall the worst-performing model with this BC, deteriorated to a maximum of just over 40% absolute percent error at 12 hours prior to reentry. Both the JB8ROM and the TIEROM tracked JB2008 behavior at a slight negative bias, with the TIEROM overall closer than JB8ROM to JB2008.

Although the relative performance of most of the density models in terms of percent error is underwhelming with the SG BC, the absolute error does steadily decrease with proximity to the reentry for all models except the NRLM00. Nonetheless, the overall performance of most of the density models on Fobos-Grunt with the SG mean BC was poor. None of the density models except NRLM00 remained within ±20% error for all predictions, and the NRLROM and ML ROM had no predictions within ±20% error with this BC model. The NRLM00 had errors within ±10% for all predictions, and within ±5% for most predictions.

5.2.2 Pardini-Anselmo Mean BC

Figure 5-20 shows Fobos-Grunt trajectories predicted by all models with the Pardini-Anselmo mean BC.

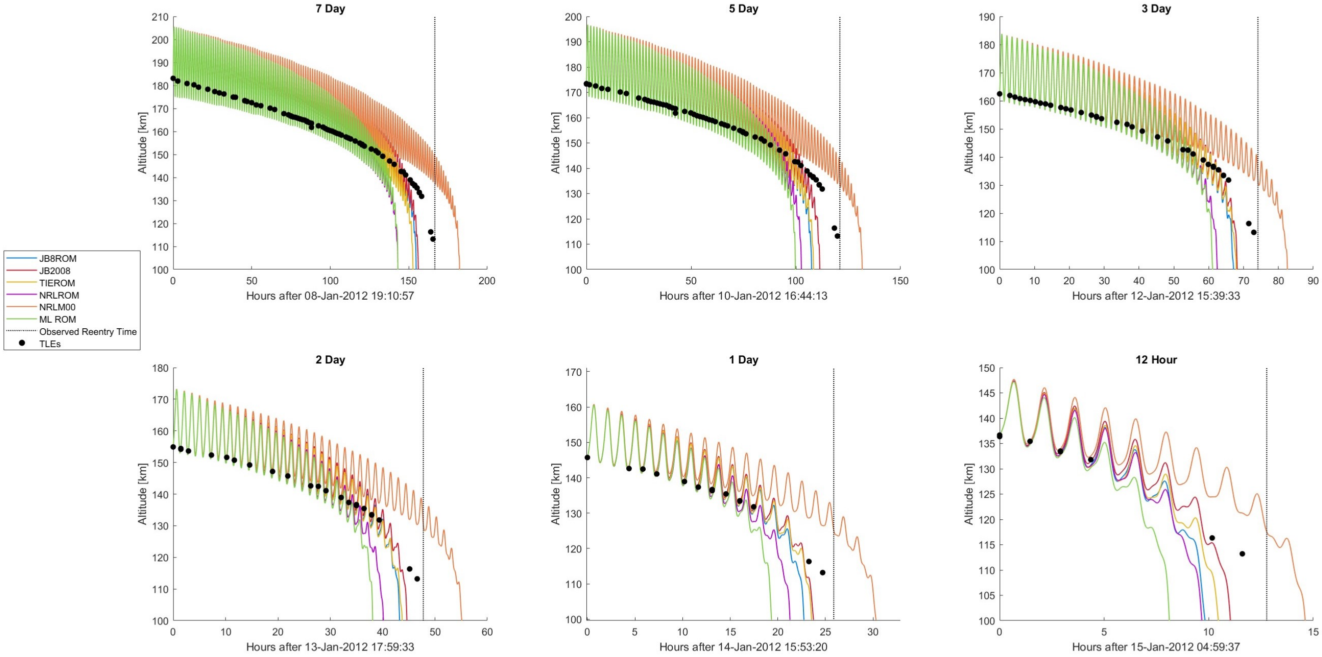


Figure 5-20: Fobos-Grunt reentry trajectories predicted with the Pardini-Anselmo mean BC using 6 different atmospheric density models and 6 different simulation durations.

With the overall lower BCs of the PA mean BC model, most of the density models' predictions are closer to the true reentry epoch. However, as Pardini and Anselmo found in their own Fobos-Grunt reentry predictions [75], most of the predictions still result in underestimation of the residual lifetime. The NRLM00 is the exception; in every case, it overestimates the residual lifetime. These results show that NRLM00 underestimates atmospheric densities relative to the other density models.

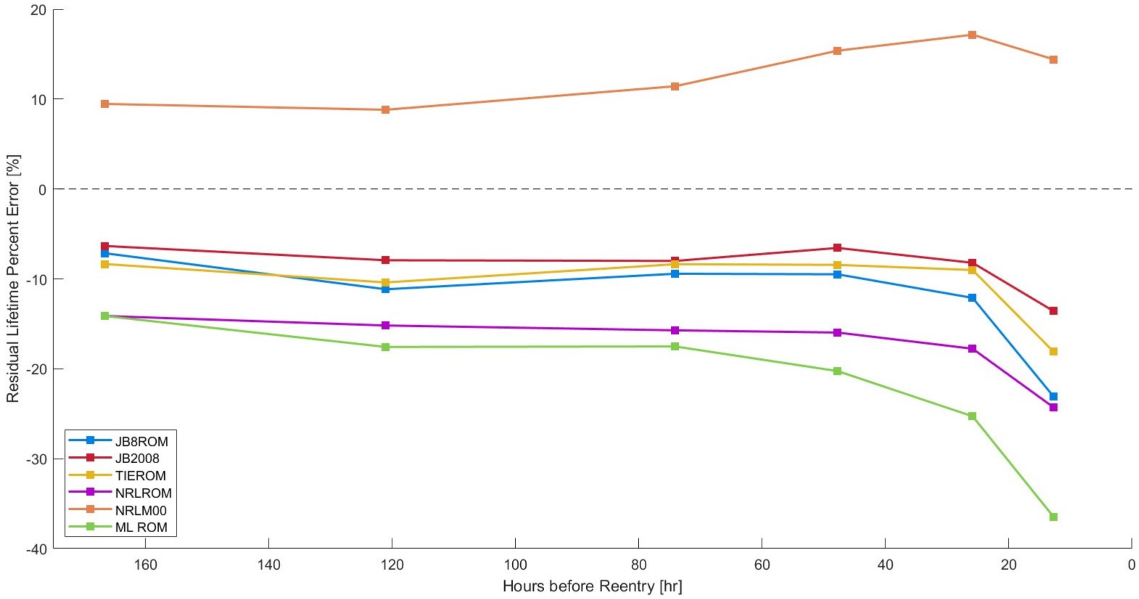


Figure 5-21: Percent error (%) of each model over time using the Pardini-Anselmo mean BC.

The JB2008 model has the lowest overall percent error, with a maximum relative error of 13.6% at 12 hours prior to reentry and most errors at or below about 8%. Most of the density models experience their maximum relative error at 12 hours prior to reentry. The NRLM00 is the exception, and has its greatest error of 17.2% at 1 day before reentry. The JB8ROM and TIEROM track the performance of the JB2008 at a small bias, as in the Fobos-Grunt simulations with the SG BC. The three density models agree to within 4% error up until 12 hours prior to reentry, at which time the spread of the three models increases to 10% error.

The similarity between JB8ROM and TIEROM was also observed in the UARS

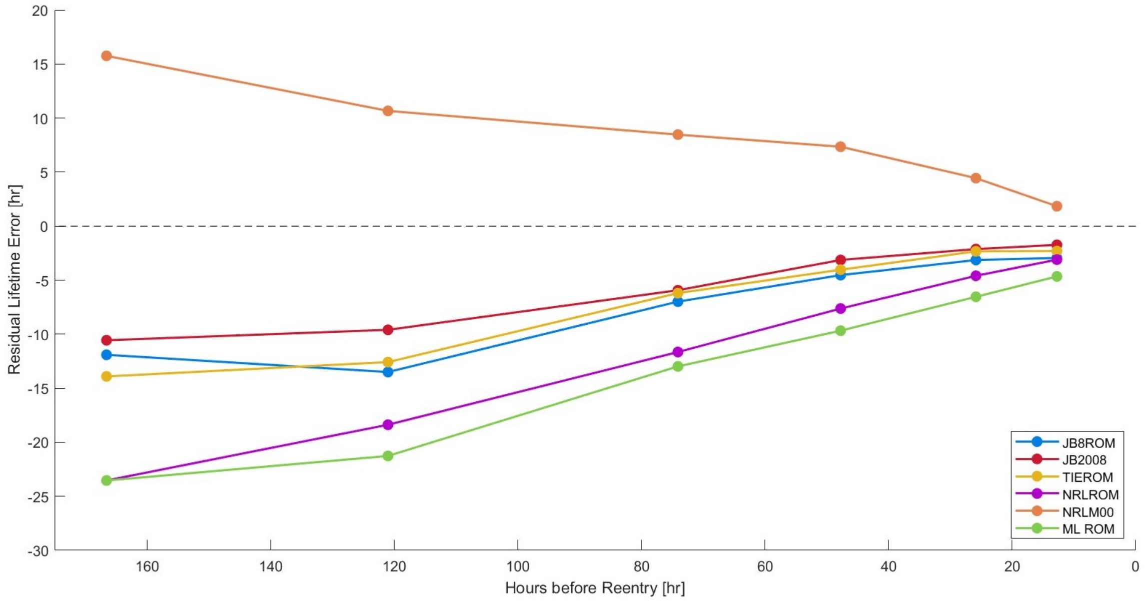


Figure 5-22: Absolute error (hrs) of each model over time using the Pardini-Anselmo mean BC.

test case. Because it was trained and initialized on JB2008, the JB8ROM is expected to agree well with the JB2008 model. The TIEROM, although it was trained on TIE-GCM data, likely yields similar results because it, like the JB8ROM, was initialized with POD on a JB2008 snapshot. As UQ results showed in Section 4.3, initial ROM states have a persistent impact on ROM results. The ML ROM has the highest error overall, with a maximum error of 36.5%. However, the ML ROM has lower error for longer prediction windows.

Absolute error for all models converges towards zero as the prediction window shrinks. The NRLM00, JB2008, and TIEROM model relative error remains under ±20% for all reentry prediction windows, while the JB8ROM and NRLROM fail this standard only in their 12-hour predictions, which have 23.1% and 24.3% absolute percent error, respectively. None of the density models demonstrate error under ±10% for all predictions, although the JB2008 is close, with >10% error only for one prediction window. None of the density models has less than 5% residual lifetime error for any of the simulation durations using the PA mean BC.

5.2.3 Pardini-Anselmo Time-Varying BC

Figure 5-23 shows the Fobos-Grunt reentry predictions for all density models using the PA time-varying BC model.

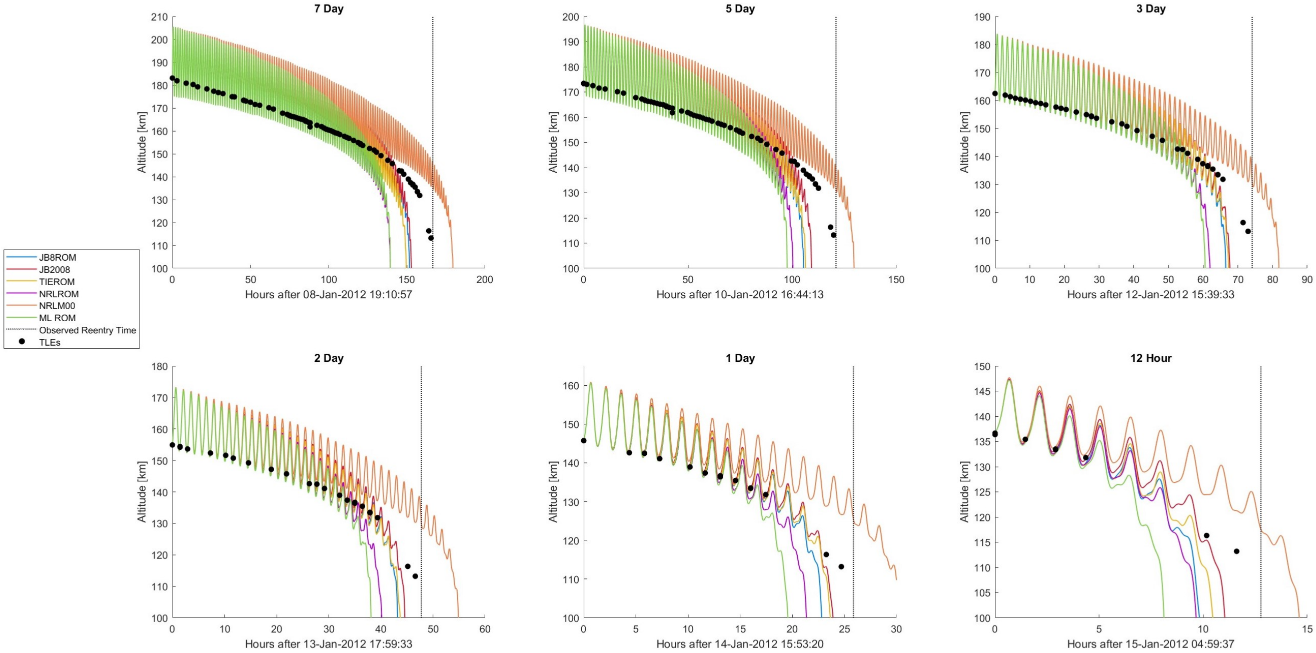


Figure 5-23: Fobos-Grunt reentry trajectories predicted with the Pardini-Anselmo time-varying BC using 6 different atmospheric density models and 6 different simulation durations.

An initial inspection of Figure 5-23 shows that the results produced with the PA time-varying BC are similar to those with the PA mean BC. All density models except for the NRLM00 predict premature reentries, while the NRLM00 overestimates residual lifetime of the Fobos-Grunt.

As in the PA mean BC case, the JB2008 has the least residual lifetime relative error overall. For longer prediction windows ≥3 days, the JB2008 has slightly higher error (by about 1%) than it does with the mean BC, while for shorter prediction windows the JB2008 error is very close or slightly lower than with the mean BC. This result closely corresponds to the relative differences in the PA mean and time-varying BCs. Where the time-varying BC is higher than the mean, at ≥3 days prior to reentry, the

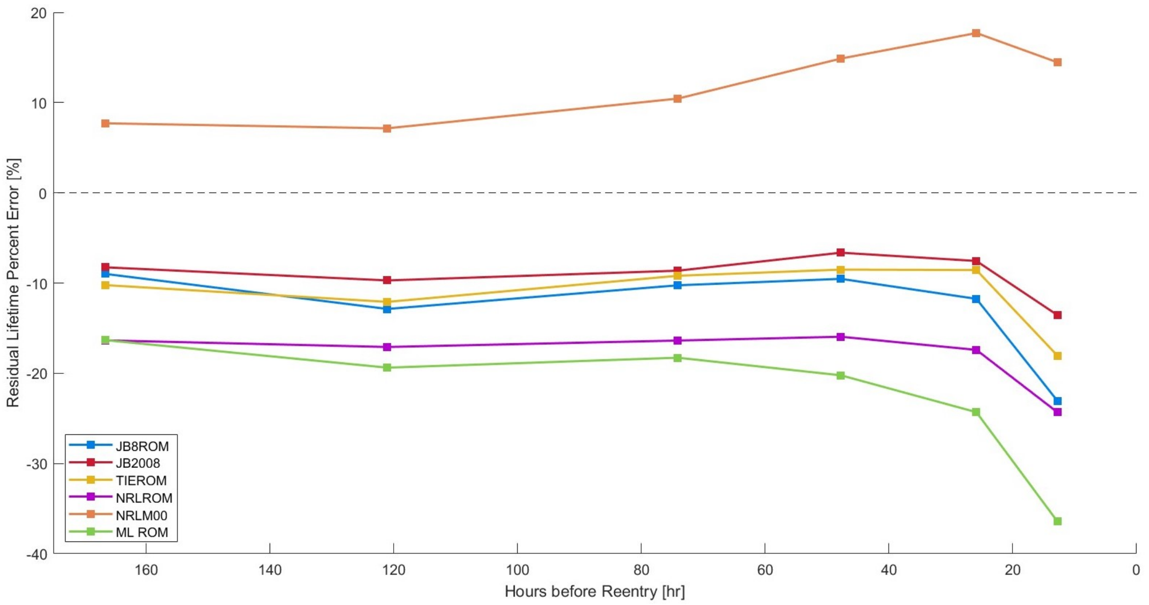


Figure 5-24: Percent error (%) of each model over time using the Pardini-Anselmo time-varying BC.

JB2008 performs worse with the time-varying BC than with the mean BC. Where the two BCs are very close or the time-varying BC is lower, at <3 days prior to reentry, the JB2008's performance is equivalent or slightly better with the time-varying BC. Given that the mean BC results in underestimation of residual lifetime, this is the expected result; a lower BC results in a later predicted reentry time. The same is true for all of the ROMs, and again, this is the expected result. When the density models are underestimating residual lifetime, a higher BC will worsen performance by resulting in earlier reentry predictions, while a lower BC will improve performance by causing later reentries. For the NRLM00, which overestimates residual lifetime, the opposite is true: higher BCs slightly improve performance, while lower BC worsens performance.

Despite these slight differences, the performance of all density models is relatively similar for the two PA BC models. None of the density models' residual lifetime prediction error changes by more than 2% for any prediction window for the PA time-varying BC as compared to the PA mean BC. This is the reflection of the relative

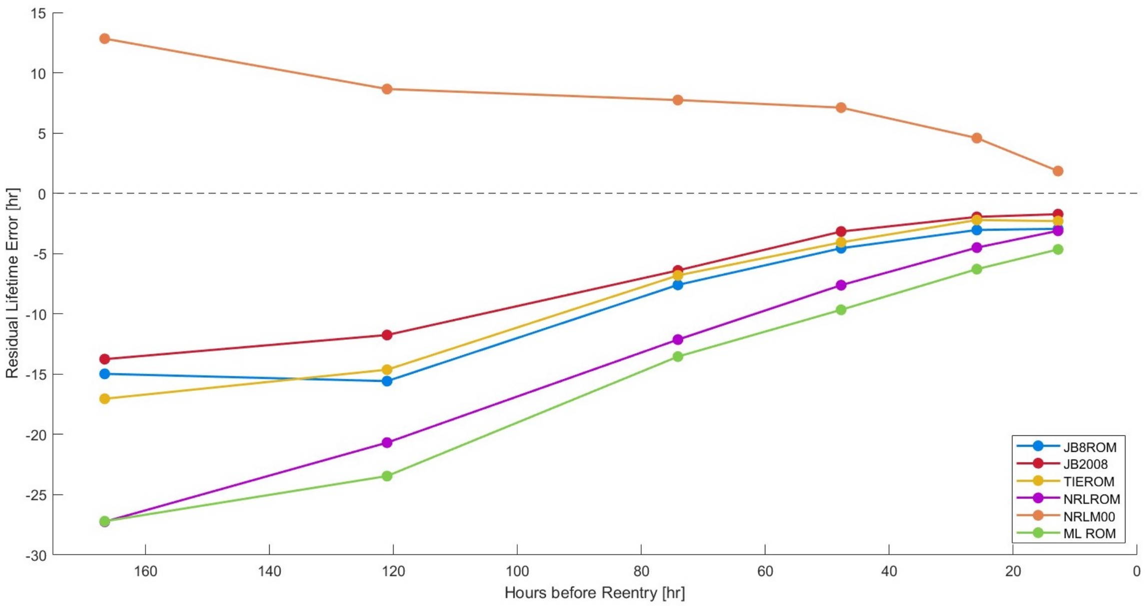


Figure 5-25: Absolute error (hrs) of each model over time using the Pardini-Anselmo time-varying BC.

agreement between the two PA BC models, which differ by no more than ∼6% across all prediction windows.

5.2.4 Saunders-Gondelach Mean BC with JB2008

Because of the poor performance of most density models using the SG mean BC with NRLM00, results with the JB2008-derived SG mean BC are presented here for comparison. Reentry trajectories using the JB2008-based BC model are shown in Figure 5-26.

The effect of the much lower BCs estimated using the JB2008 density model is to "shift" all reentry predictions forward in time, so that the JB2008 model is generally in best alignment with the TLE data. This result is expected, since the SG BC estimation process fits BCs to TLE data. The NRLM00, which predicts much lower densities than the JB2008 over the Fobos-Grunt reentry time frame, considerably overestimates the residual lifetime of the probe using this BC model.

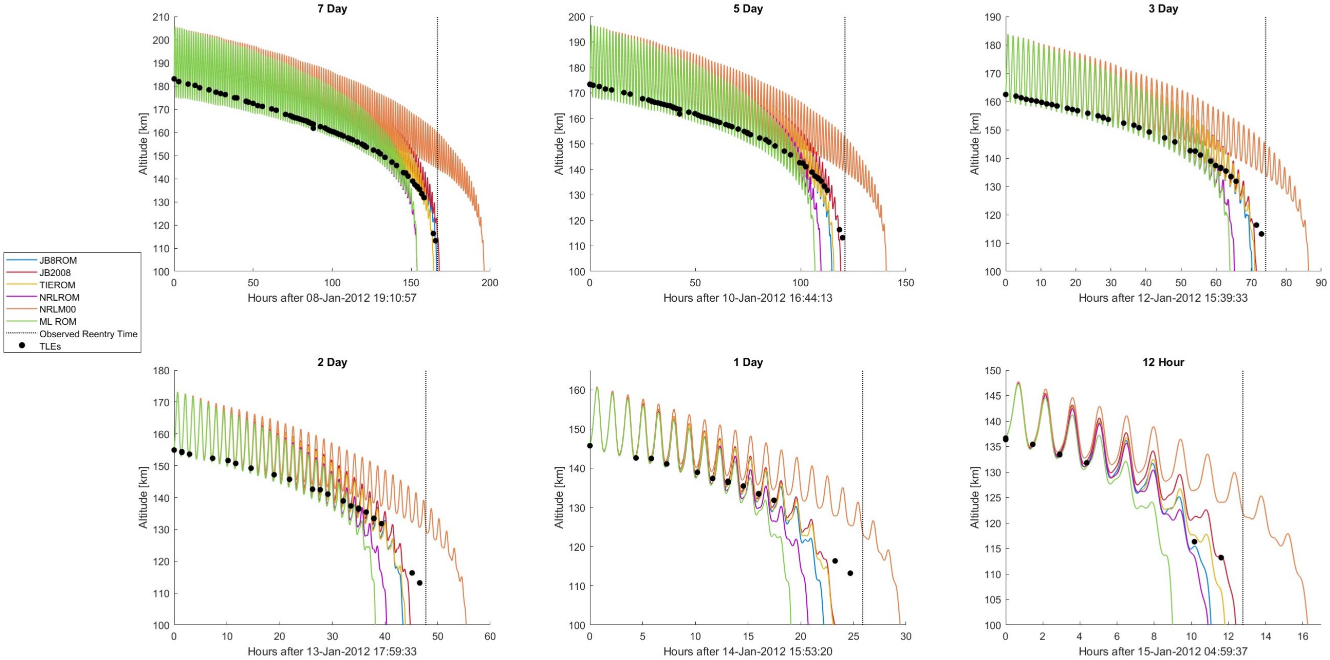


Figure 5-26: Fobos-Grunt reentry trajectories predicted with the Saunders-Gondelach mean BC fitted with the JB2008 density model, using 6 different atmospheric density models and 6 different simulation durations.

The residual lifetime percent error curves for all density models are similarly shifted from those generated with the NRLM00-based SG mean BC. All of the density models except for NRLM00 improve by an average of about 12-14% in terms of residual lifetime relative error using this BC model, as compared to the equivalent NRLM00-based model. As expected, the performance of the NRLM00 worsens by a similar amount (14.8%, on average). Apart from this shift, the general performance trends in all models in terms of percent error are the same as with the NRLM00-based SG BC model. For example, most of the models still exhibit a tendency to underestimate residual lifetime, and the JB2008, JB8ROM, and TIEROM remain in very good agreement until 12 hours prior to reentry.

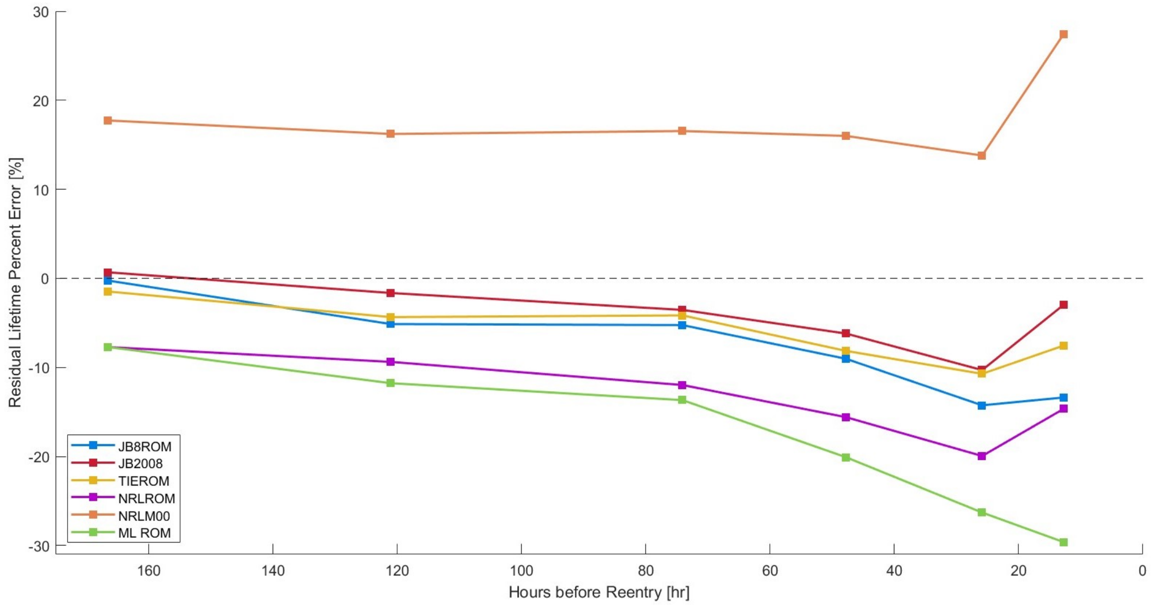


Figure 5-27: Percent error (%) of each model over time using the Saunders-Gondelach mean BC derived with the JB2008 density model.

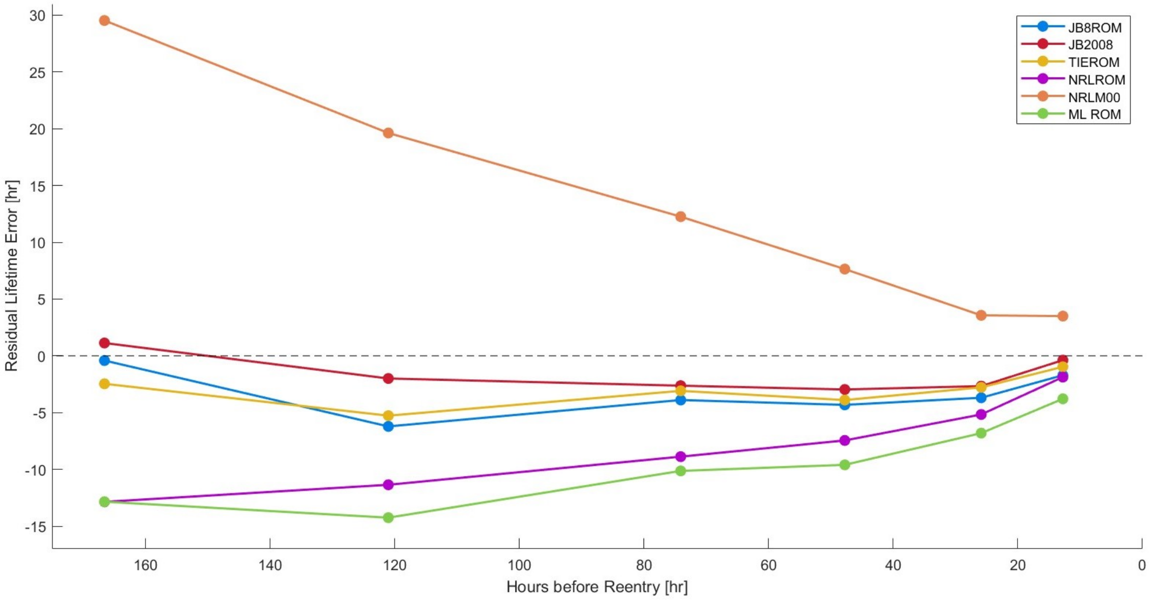


Figure 5-28: Absolute error (hrs) of each model over time using the Saunders-Gondelach mean BC derived with the JB2008 density model.

5.2.5 Overall Results

Figure 5-29 shows Fobos-Grunt mean absolute percent error across all simulation durations, with each density model-BC combination.

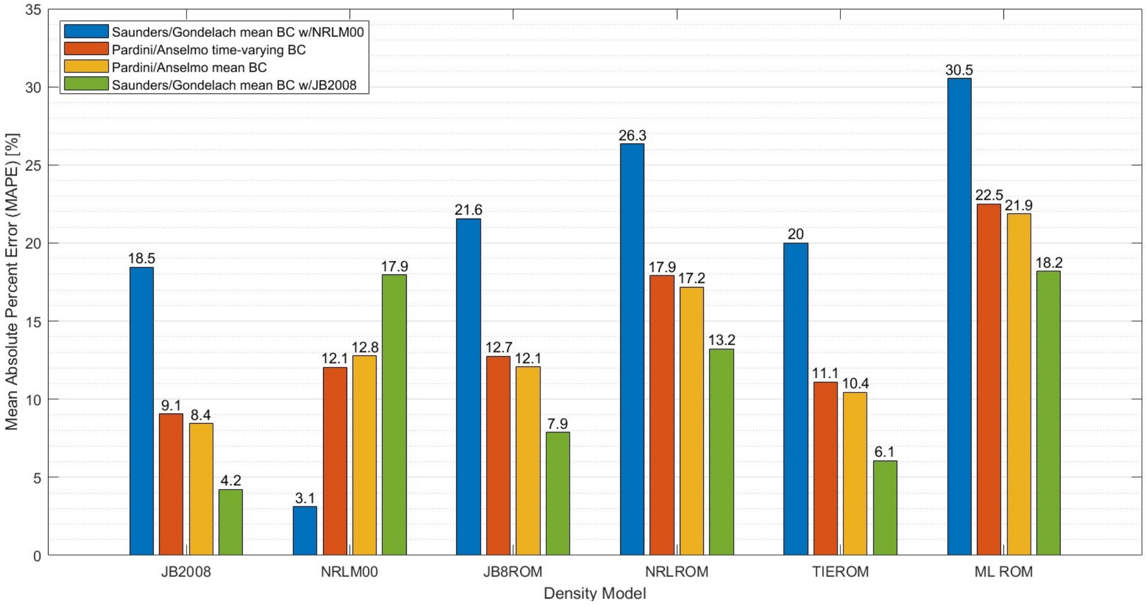


Figure 5-29: Average residual lifetime prediction MAPE across all simulation durations, for all density models and all BC models, for the Fobos-Grunt test case.

For all of the models except NRLM00, the PA BC models are more successful than the NRLM00 SG BC by about 8-10% in terms of MAPE. The NRLM00 reverses this trend, performing 9-9.7% better in terms of MAPE with the NRLM00 SG BC than with the PA BCs. The ROMs and JB2008 perform slightly (0.6-0.7%) better with the mean PA BC than with the time-varying BC, while the NRLM00 performs worse by the same margin. On the other hand, all models except the NRLM00 perform about 4% better with the SG JB2008 BC than with either PA BC. The NRLM00 exactly reverses the performance trend of all other models used across all four BC models. This likely indicates of NRLM00 error absorption into the NRLM00-based BC models, with the NRLM00 performing better and other density models worse on BCs which have absorbed more NRLM00 error.

The best performance of any density-BC model combination was 3.1% MAPE with the NRLM00 and the NRLM00 SG BC. However, using this BC, none of the other models achieved 10% MAPE or lower, and only two other models (JB2008 and TIEROM) achieved within 20% MAPE. The next-best performer is, unsurprisingly, the JB2008 paired with the JB2008-based mean BC at 4.2% MAPE. With this BC, the JB2008 is followed by the TIEROM with 6.1% MAPE and the JB8ROM at 7.9%. With the PA BCs, the best performer is JB2008, at 9.1% and 8.4% MAPE with the time-varying and mean BCs respectively. The next-best performer was the TIEROM, with 11.1% and 10.4% MAPE, respectively. As in the UARS test case, the JB8ROM gave very similar results to the TIEROM, with MAPE agreeing to within 1.8% between the two models for all BCs. The JB8ROM performance was also similar to the NRLM00 for both of the PA BCs, on which the two density models agreed to within 0.7% MAPE. For the Fobos-Grunt test case, the NRLROM and ML ROM lagged in performance with all BCs.

For this test case, only the NRLM00 with the NRLM00 SG BC and the JB2008 with the JB2008 SG BC reached the ±5% standard in terms of MAPE. These two density-BC model combinations are expected to yield good results. An inevitable consequence of estimating BC estimation by using BC as a fit parameter is that the BC will absorb density model error during the estimation process, and will therefore be optimally compatible with that density model when used in propagation [95]. However, the JB8ROM and TIEROM both met the ±10% standard by a significant margin when using the JB2008-based BC, and all models achieved ±20% MAPE with this BC. For the Pardini-Anselmo BCs, only the JB2008 reached the ±10% performance standard, although the TIEROM was close at 11.1% and 10.4% on the two PA BCs. All of the density models except the ML ROM performed to within ±20% MAPE with the PA BCs, though the ML ROM was close to this standard as well.

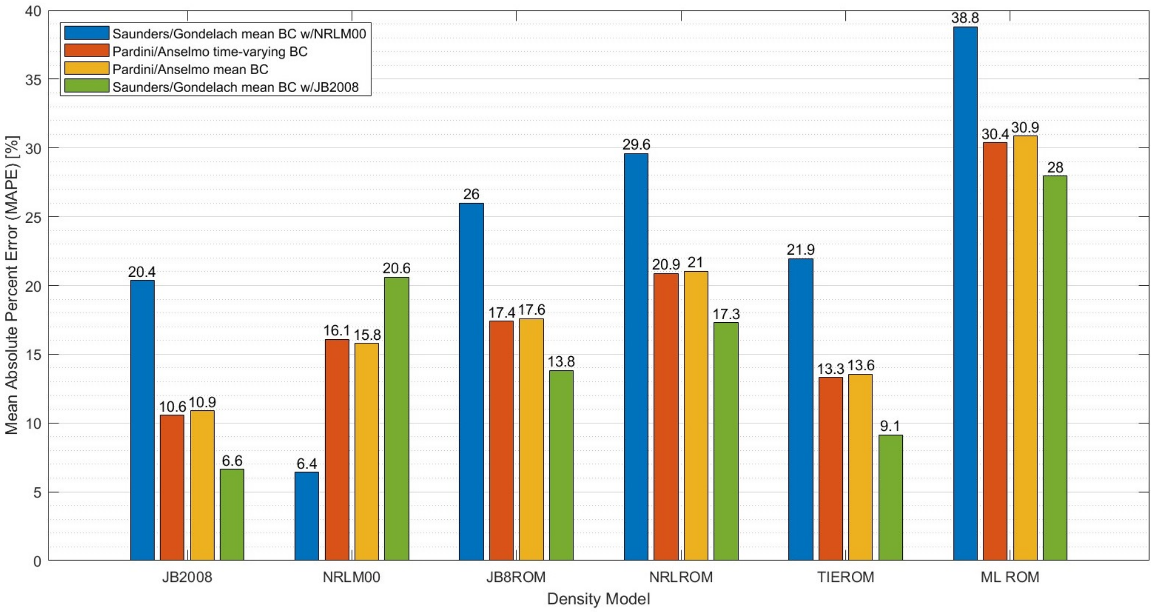


Figure 5-30: Fobos-Grunt residual lifetime prediction MAPE for all density and BC models for short prediction windows of one nominal day or less.

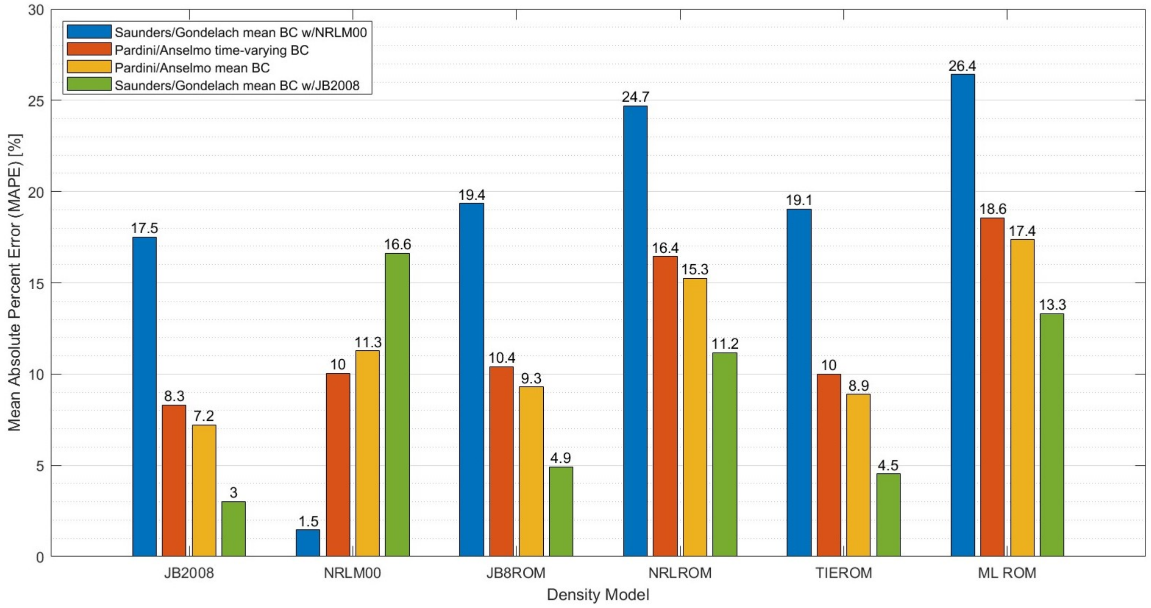


Figure 5-31: Fobos-Grunt residual lifetime prediction MAPE for all density and BC models for longer prediction windows of 2-7 nominal days.

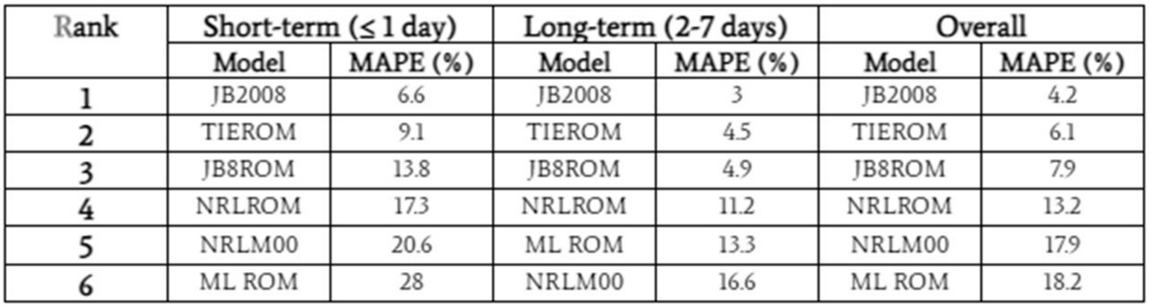
All density-BC model combinations performed better over long prediction windows than over short prediction windows. This could indicate either larger errors in density models, or larger BC errors, during the last day of the Fobos-Grunt's lifetime. The

ML ROM saw the greatest performance difference, with 11-15% lower MAPE over long prediction windows than over short prediction windows, for all BCs. The JB2008 saw the least performance difference, with an average of 3.1% lower MAPE for long prediction windows than for short prediction windows.

Over long prediction windows, the JB2008 and TIEROM met the ±10% MAPE standard for both PA BCs. The JB8ROM meets this standard for the PA mean BC and barely misses for the time-varying BC, while the NRLM00 is within ±10% for the time-varying BC. Using the JB2008-based SG mean BC, the JB2008, JB8ROM, and TIEROM all met the ±5% mark in terms of MAPE. This is considered very low error for residual lifetime estimation. The NRLM00 with the SG NRLM00 BC met this standard as well. All density-BC model combinations are within ±20% MAPE over long time windows, except the ML ROM and NRLROM with the SG NRLM00 BC. Over short prediction windows, the JB2008, NRLM00, JB8ROM, and TIEROM met the ±20% standard for both PA BCs. The JB2008 and all PODROMs achieve ±20% MAPE over short prediction windows with the JB2008-based BC, and the JB2008 and TIEROM also meet the ±10% mark. None of the density-BC model combinations achieved ±5% MAPE over short prediction windows.

Table 5.2 ranks MAPE performance for the Fobos-Grunt simulations using the overall best-performing BC model, the SG JB2008 mean BC.

Table 5.2: Density models ranked by MAPE performance on Fobos-Grunt test case with Pardini-Anselmo mean BC.



Rank	Short-term (≤ 1 day)		Long-term (2-7 days)		Overall	
	Model	MAPE (%)	Model	MAPE (%)	Model	MAPE (%)
1	JB2008	6.6	JB2008	3	JB2008	4.2
2	TIEROM	9.1	TIEROM	4.5	TIEROM	6.1
3	JB8ROM	13.8	JB8ROM	4.9	JB8ROM	7.9
4	NRLROM	17.3	NRLROM	11.2	NRLROM	13.2
5	NRLM00	20.6	ML ROM	13.3	NRLM00	17.9
6	ML ROM	28	NRLM00	16.6	ML ROM	18.2

Orbit propagation run time by density model for the Fobos-Grunt simulations is shown in Figure 5-32, averaged across BC models and prediction windows.

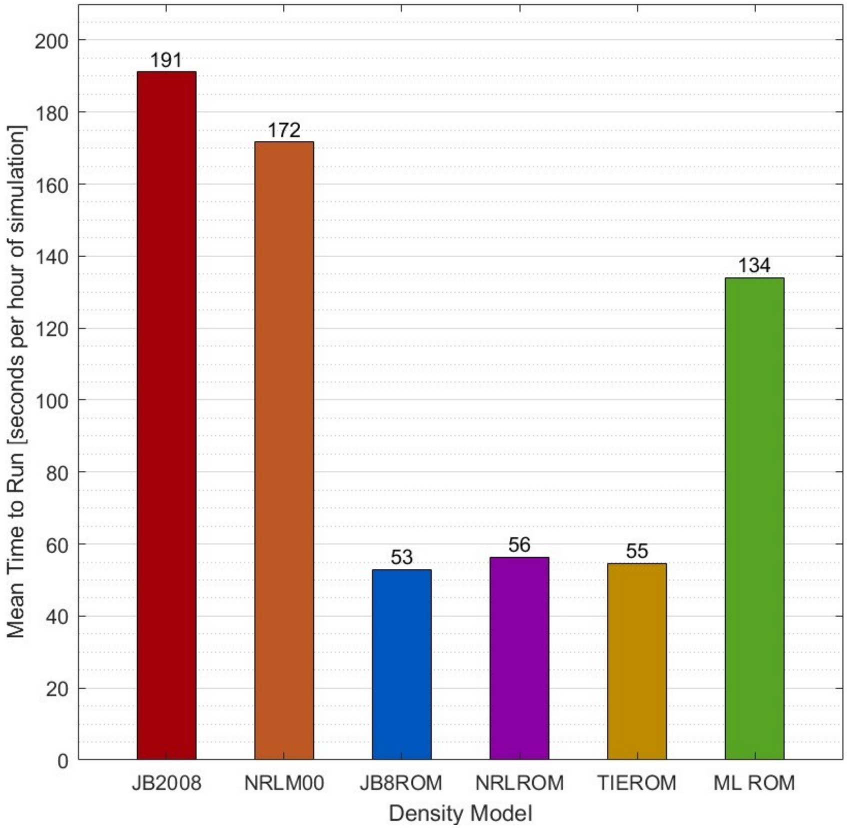


Figure 5-32: Average HPOP run time by density model for Fobos-Grunt test case.

HPOP performance with all density models is slightly faster but comparable to run time performance observed in the UARS test case. PODROM propagation averages 69.8% faster and the ML ROM 26.0% faster than propagation with empirical models.

None of the density-BC model combinations perform remarkably well with NRLM00-based BCs, except for the NRLM00 with the SG BC. However, all other density models perform relatively poorly with this BC model and much better with the JB2008-based BC model, likely indicating a high level of NRLM00 error absorption during the BC estimation process for NRLM00-based BCs. In addition, most of the density models chronically underestimated the Fobos-Grunt's residual lifetime for all BC models and prediction windows. This accords with Fobos-Grunt reentry predictions made by Pardini and Anselmo, where 15 out of 20 reentry predictions underestimated the residual lifetime of Fobos-Grunt [75].

Once again, with a suitable BC model, at least two ROMs achieve comparable performance to empirical models at a drastically reduced run time, offering the potential for greater orbit propagation capacity over the same time frame. Additionally, the TIEROM and JB8ROM show excellent performance (±5% MAPE) over long prediction windows.

5.3 Progress-M 27M

The Progress-M 27M (PM27M) test case was distinguished in this thesis by its higher initial orbital eccentricity of 0.175, making it the most eccentric of the five test cases. This may make estimation of its BC more challenging than for other test cases, since TLEs tend to experience a bias in error between perigee and apogee [95]. However, this effect is most pronounced in Highly Elliptical Orbits (HEO) [22, 95], which PM27M is not. By many standards, such as those in [109] and [22], the orbit of PM27M is still considered low-eccentricity or near-circular, so the impact of perigee-apogee TLE bias is unlikely to be large. A plot of BC estimates used for the PM27M case is shown in Figure 5-33.

Once again, note that for each mean BC simulation, the BC value is held constant throughout the entire simulation. Mean BC values in Fig. 5-33 are shown as a function of start time. Recall that the interpretation of the mean BCs is that the BC for a simulation starting in the x (time) range of each horizontal line is given by that line's value. The mean BC value is invariant throughout the simulation, but its constant value depends on the start time of the simulation.

In this figure, the variance of the SG time-varying BC around its mean value has again been scaled down by a factor of 3 to allow all BCs to be shown on the same plot.

The most notable feature of this plot is the spike in BC predicted by both methods of BC estimation between 17 and 20 hours prior to reentry. Because the mean BC is sensitive to outliers, this spike has a notable impact on calculations of mean BC. Using the SG BC estimation method, this spike is especially dramatic – the outlying value jumps around 200% above the previous and next estimated BCs.

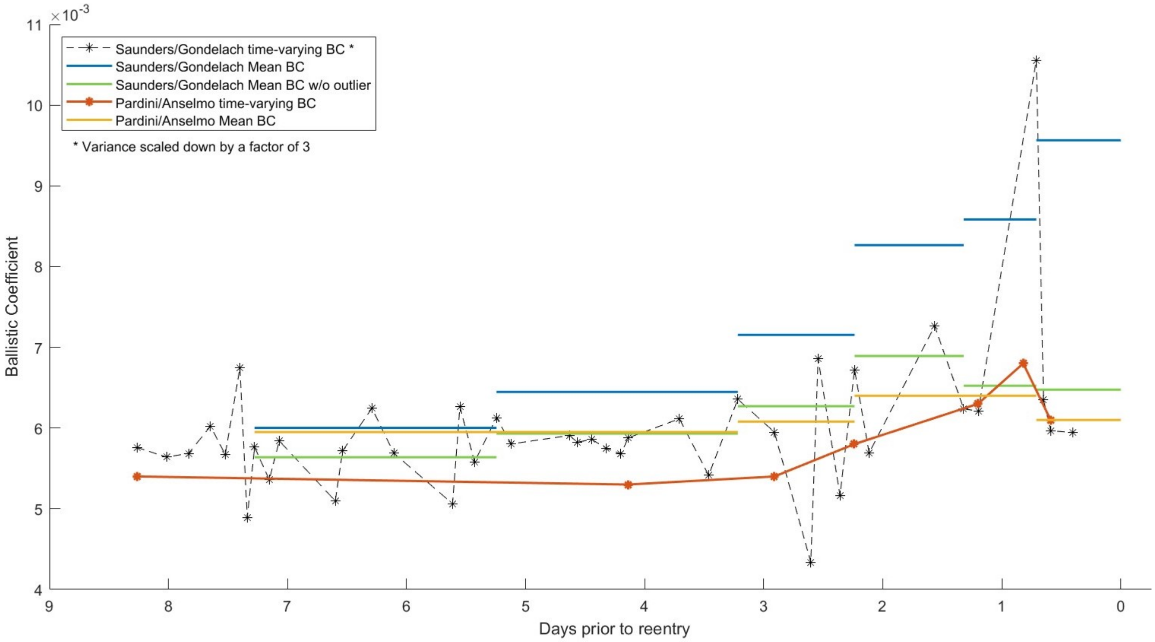


Figure 5-33: PM27M Ballistic coefficient estimates.

For reference, the SG mean BC is calculated with the outlying value replaced by the previous highest BC value, and the result is plotted in green. Note that the BC plotted in green was not used in any simulations, and is shown here only for comparison. Exclusion of the outlying BC estimate results in a 6% decrease in the SG mean BC value for a 7-day simulation, and up to a 32% decrease for shorter simulations. Interestingly, the SG mean BC with the outlier excluded remains relatively close to the PA mean BC, with a maximum difference of 8% and a mean difference of 2%. Additionally, both show a terminal decrease in BC, unlike the SG mean BC with the outlier included (in blue).

The spike in BC is not necessarily erroneous, and it is also present in the PA BC estimates (orange). Variations in BC during reentry are common for uncontrolled objects [77] as the object traverses different atmospheric regimes at a changing velocity, and potentially undergoes changes in mass/area due to breakup. However, it is possible that the SG method overestimates the BC increase in the case of PM27M. In any event, since the mean BC method is not time-weighted, the impact of this short-lived BC increase on PM27M reentry predictions using the SG mean BC may be disproportionate. Time-weighting of the SG BC estimates might be a more effective technique for this test case. Based on Fig. 5-33, we can expect that reentry simulations using the SG mean BC will predict earlier reentries than simulations run with the PA mean BC. Furthermore, the SG mean BC with the outlying BC value replaced, also plotted in Fig. 5-33, shows that the earlier reentry predictions are almost entirely due to the impact of the BC spike on the mean BC value. Without the outlier, the SG mean BC would be similar to the PA mean BC, yielding later reentry times.

5.3.1 Saunders-Gondelach Mean BC Results

Figure 5-34 shows the reentry trajectories predicted by each model over nominal simulation durations ranging from 7 days to 12 hours, using the SG mean BC model.

The SG mean BC yields significant underestimations of the reentry time, with all models uniformly forecasting premature reentries over all simulation durations. Given the disproportionate impact of the BC spike on the SG mean BC model due to the lack of BC estimate weighting, this result is not surprising. Percent error and absolute error of the density models over time are shown in Figures 5-35 and 5-36. Again, negative percent error indicates residual lifetime underestimation, where predicted reentry time is premature with respect to the true reentry time. Altitudes shown are geocentric.

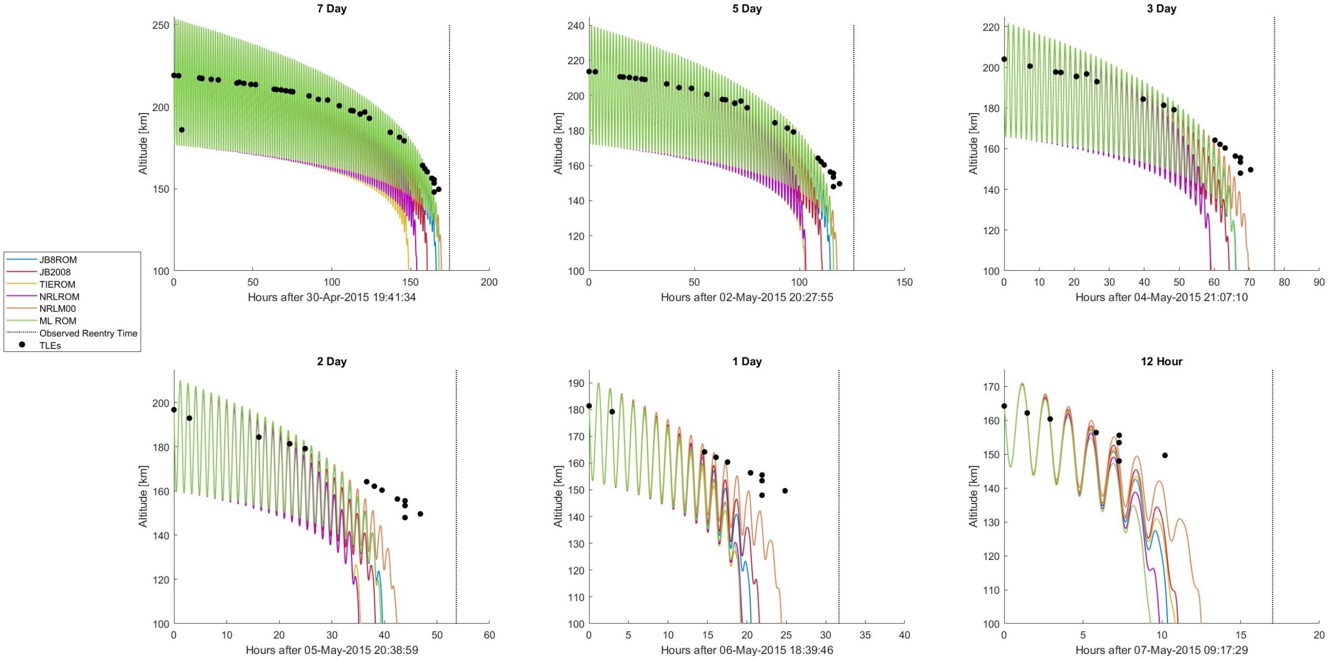


Figure 5-34: PM27M reentry trajectories predicted with the Saunders-Gondelach mean BC using 6 different atmospheric density models and 6 different simulation durations.

Two major trends visible in Fig. 5-35 are that the NRLM00 density model has consistently the lowest error over time, and that the percent error of all models increases with time. The first observation is an expected result for the NRLM00 SG BC, as also observed in Section 5.2.1. The mean of BC values estimated using the NRLM00 retain absorbed error caused by any bias in the density model.

Over longer prediction windows (nominal 7 days to 5 days prior to reentry), the ML ROM and JB8ROM are close runners-up to the NRLMSISE-00 model, with absolute percent errors of 4-5% at 7 days and 8-9% at 5 days, as compared to 3% and 6% error in NRLM00 for the same prediction windows. The JB2008-based ROMs begin to diverge from NRLM00 predictions at 3 days prior to reentry, but remain runners-up until 1 day prior to reentry, after which time the JB2008 model takes over the second-best position after NRLM00.

The JB2008 model follows the same trend in percent error over time as NRLM00, but with a consistent underestimation bias as compared to NRLM00. For prediction windows between 7 and 2 days, the JB2008-based PODROM and ML ROM predictions are bounded above by NRLM00 and below by JB2008.

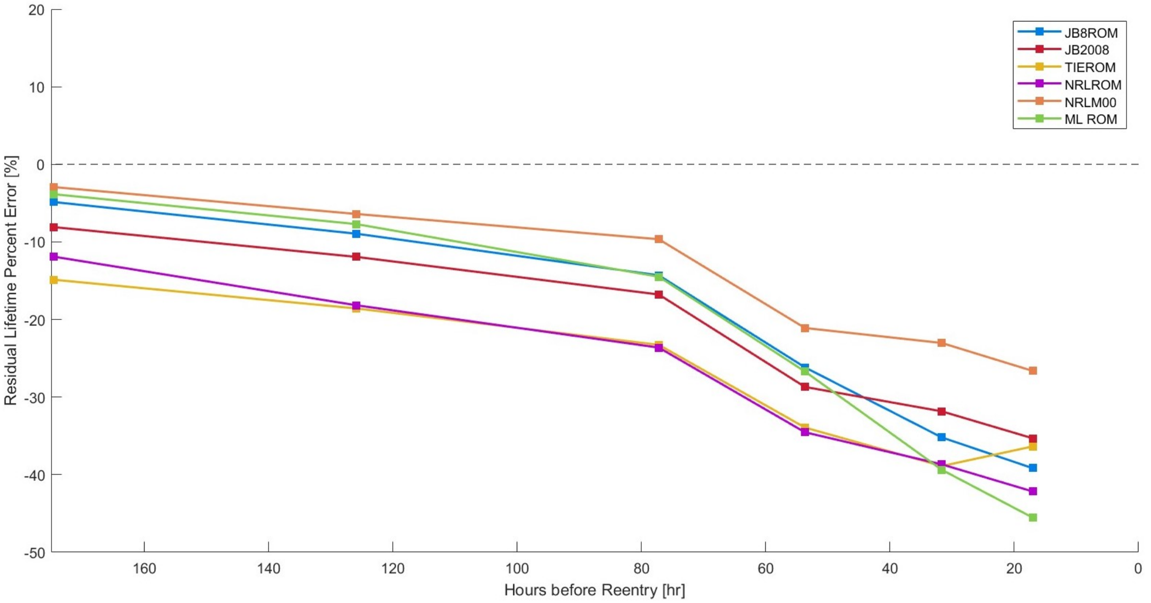


Figure 5-35: Percent error (%) of each model over time using the Saunders-Gondelach mean BC.

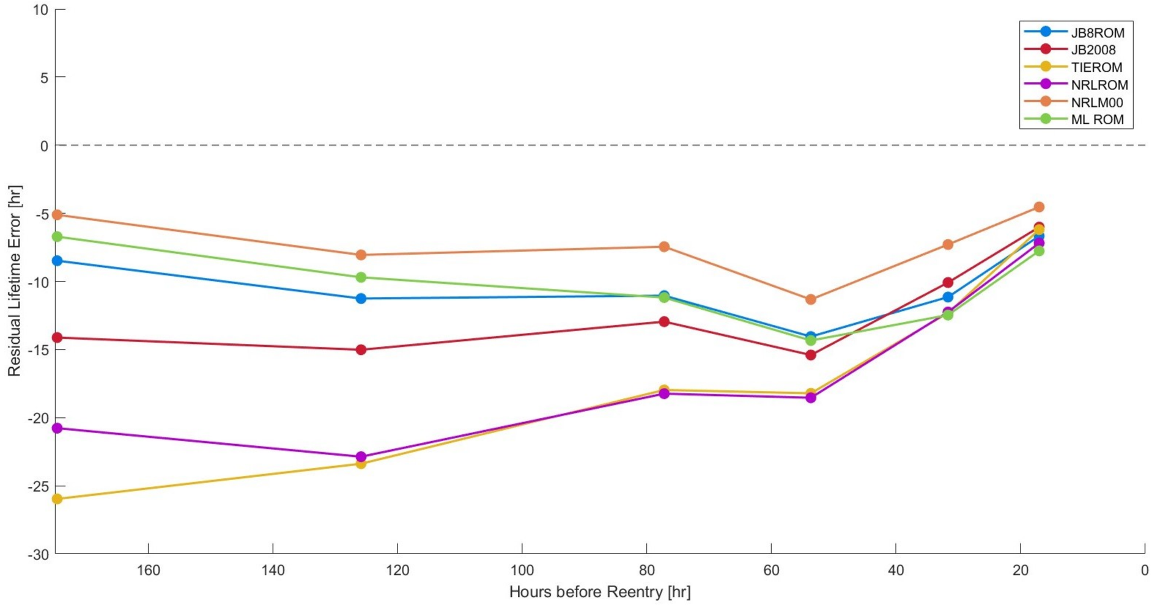


Figure 5-36: Absolute error (hrs) of each model over time using the Saunders-Gondelach mean BC.

Fig. 5-36 shows that over the final 2 days of the PM27M's lifetime, absolute residual lifetime error of all models shrinks, but this convergence is slow. Therefore, the relative error of the reentry predictions for most of the models continues to grow during this time, reaching around 35-45% (underestimation) for most of the models 12 hours prior to reentry. The NRLM00 retains an almost 9% advantage over the second-best model at 12 hours prior to reentry. The one exception to the overall growth in absolute percent is the TIEROM, which experiences a 2.5% reduction in relative error between its final two reentry predictions. The final performance of the TIEROM at a nominal 12 hours prior to reentry is within 1% of the JB2008, which is the second-best performer at 12 hours, after the NRLM00. Although the ML ROM performed well over longer prediction windows, its performance quickly drops over short windows to a percent error of almost 46% at 12 hours prior to reentry.

The performance of most of the models on the PM27M test case using the SG mean BC technique is disappointing. Maximum residual lifetime percent errors at 12 hours prior to reentry reach nearly 50%, much worse than the target performance of 20% or better for all models and simulation durations. However, the poor performance of all models, including the NRLM00 (with which the BCs were estimated) is very possibly a reflection of the disproportionate impact of the 200% increase in estimated BC occurring during the previously-discussed BC spike. Taking the mean BC over the prediction window with this value included yields a 7% BC increase over a 7-day prediction window, and the impact becomes increasingly pronounced as the prediction window narrows.

5.3.2 Pardini-Anselmo Mean BC

Figure 5-37 shows reentry trajectories predicted by all density models for PM27M using the PA mean BC model.

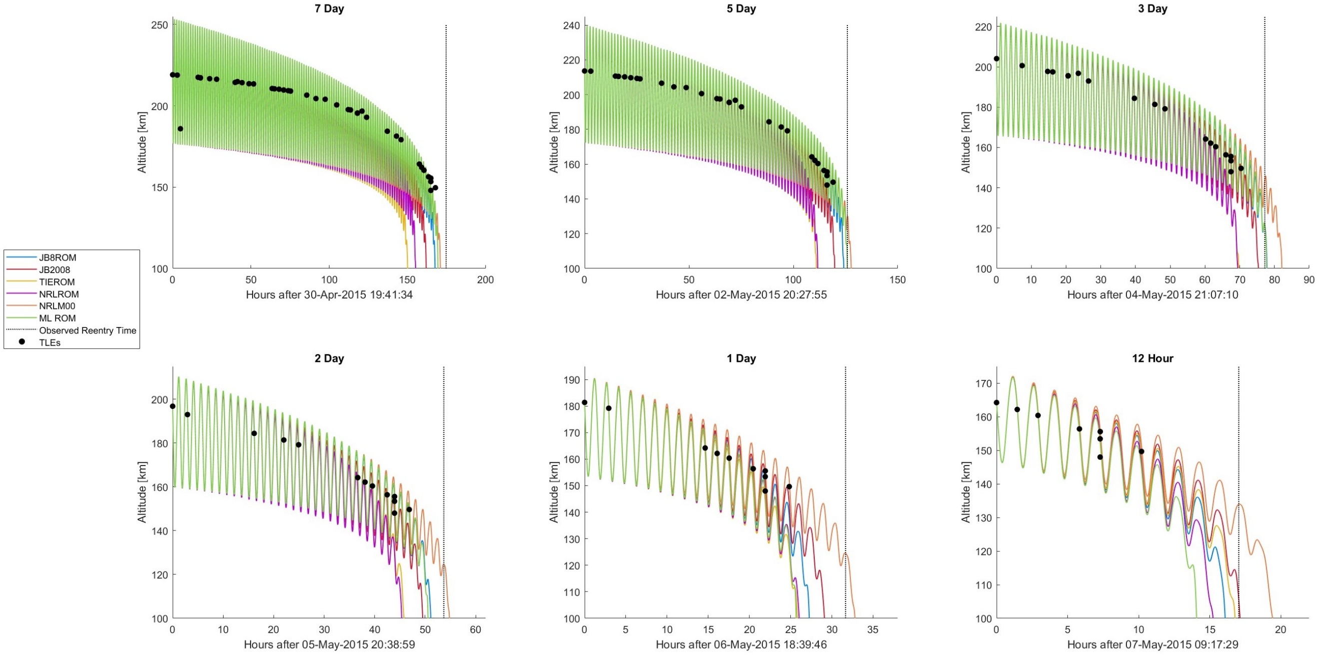


Figure 5-37: PM27M reentry trajectories predicted with the Pardini-Anselmo mean BC using 6 different atmospheric density models and 6 different simulation durations.

It is immediately apparent that the PA mean BC model is more effective at modeling the BC of PM27M than the SG BC model. The lower mean BC is due both to the smaller spike in BC predicted by Pardini and Anselmo's CDFIT algorithm as compared to the BC spike predicted by the SG mean BC method, and also to the overall lower BCs estimated with the PA method.

The two empirical models, JB2008 and NRLM00, show a very similar trend in percent error over time, with the NRLM00 biased 5-13% higher than JB2008. In fact, this trend is quite strong – the Pearson correlation coefficient (a statistical measure of correlation between two variables, with 1 being perfect correlation) for the relative error of these two models is 0.85. From 7 to 2 days prior to reentry, the JB2008-based ROMs are bounded between the NRLM00 and JB2008 models, resulting in a lower mean percent error than either empirical model over this time frame. Between 7 and 2 days prior to reentry, the JB2008 and NRLM00 models have a MAPE of 5.6% and 3%, respectively, while the JB8ROM and ML ROM have lower MAPE of 2.8% and

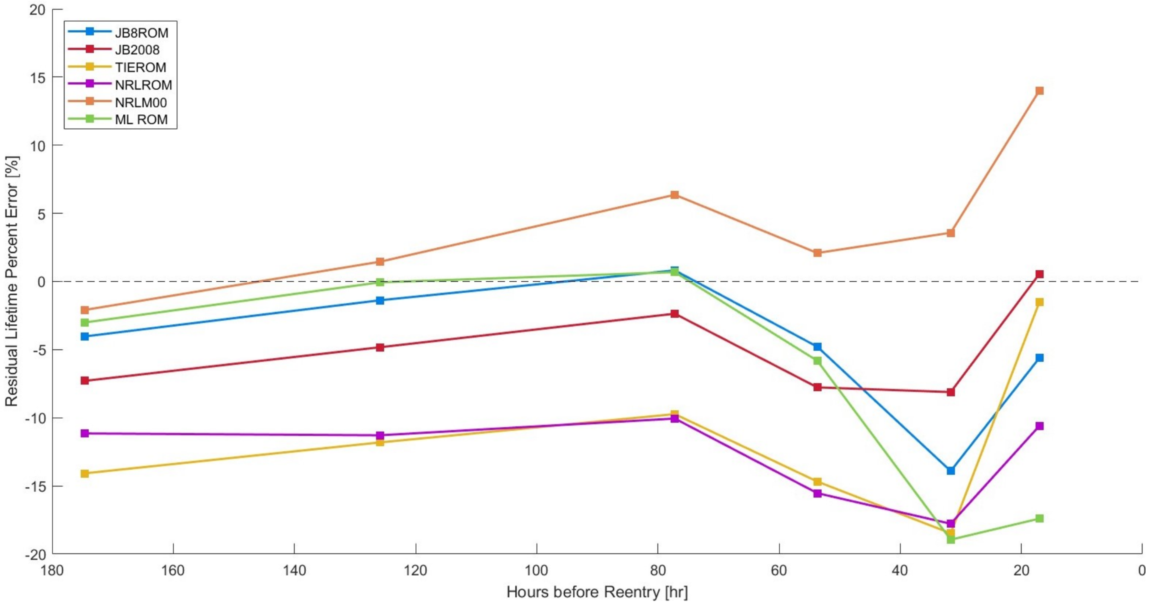


Figure 5-38: Percent error (%) of each model over time using the Pardini-Anselmo mean BC.

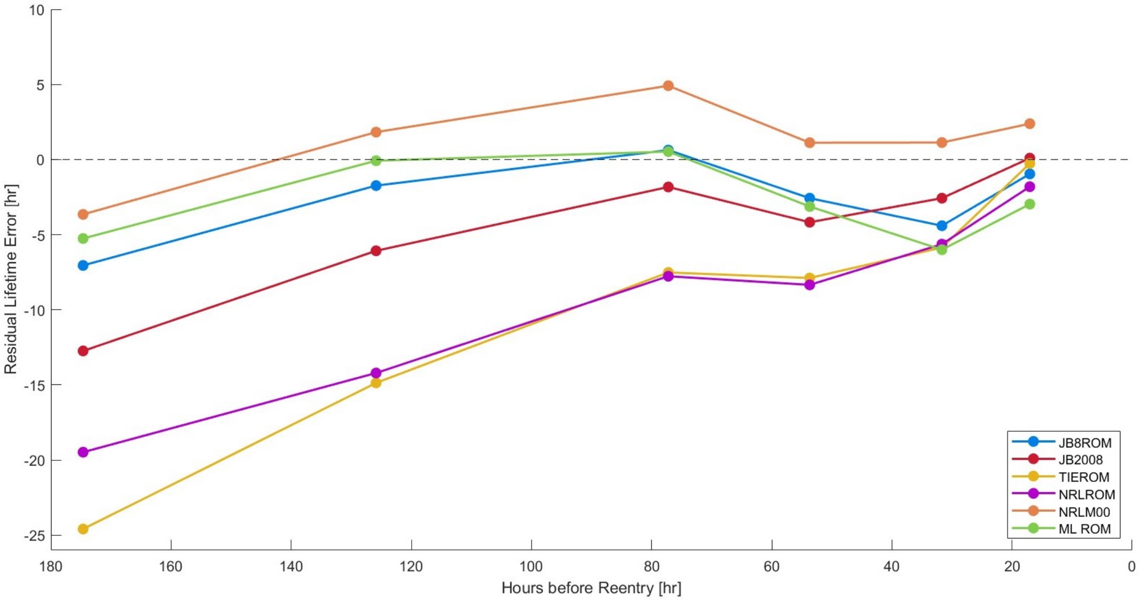


Figure 5-39: Absolute error (hrs) of each model over time using the Pardini-Anselmo mean BC.

2.4% over the same period. The tendency of the JB2008-based ROMs to be bounded by JB2008 above and NRLM00 below over the 2-7 day prediction windows was also observed when using the SG mean BC model.

All of the models predict a later reentry time 12 hours before reentry than they did at 1 day prior to reentry, reflecting the terminal drop in BC for this model (see Fig. 5-33). This feature is especially pronounced in the case of the TIEROM, which at 12 hours predicted reentry 5.8 hours later than it did at 24 hours prior to reentry. This change brought the TIEROM's final prediction into close agreement (∼22 minutes / 2% residual lifetime error) with the final prediction of the JB2008 model. With the exception of the NRLMSISE-00 model, all models made their worst predictions in terms of relative error at 1 day prior to reentry. Absolute error (in hours) did not smoothly decrease for shorter prediction windows; only half of the models (JB2008, TIEROM, and NRLROM) had their lowest absolute error at 12 hours prior to reentry.

Using the PA mean BC, all density models' reentry predictions stay within ±20% error for all prediction windows. Only the JB2008 model meets the "good performance" standard of ±10% across all prediction windows. However, for longer prediction windows of 2-7 days, the JB8ROM is the only model to consistently achieve excellent performance of ±5% error for all predictions (the ML ROM is also very close, but misses out with an error of 5.8% at 2 nominal days prior to reentry). For the same period, the JB2008 and NRLM00 models have maximum absolute percent errors of 7.8% and 6.4%, respectively, and also have higher MAPE, as previously discussed.

5.3.3 Pardini-Anselmo Time-Varying BC

Figure 5-40 shows the reentry trajectories predicted by each model over nominal simulation durations ranging from 7 days to 12 hours, using the PA time-varying BC model.

Reentry predictions using the PA time-varying BC model appear reasonably close to those made with the PA mean BC. The main difference is the generally later reentry times for longer prediction windows, reflecting the diminished influence of the BC

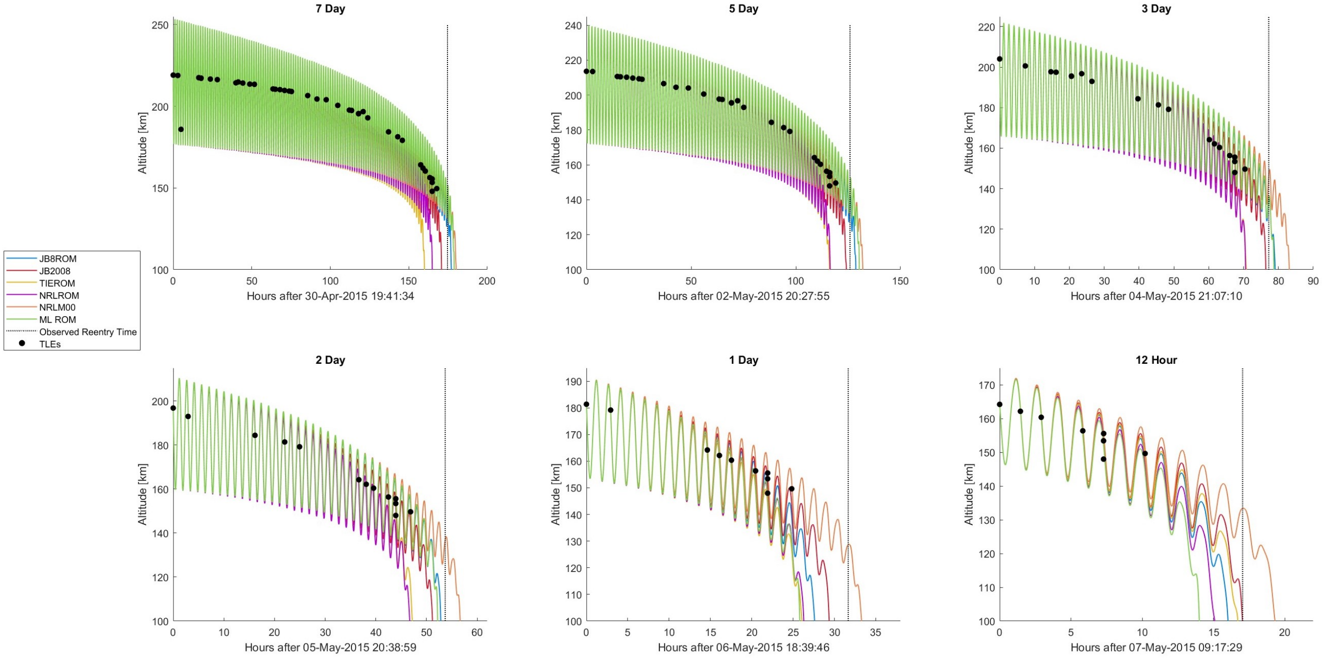


Figure 5-40: PM27M reentry trajectories predicted with the Pardini-Anselmo time-varying BC using 6 different atmospheric density models and 6 different simulation durations.

spike in the PA time-varying versus the PA mean BC.

Error trends with the PA time-varying BC are also very similar to those observed with the PA mean BC.

The JB2008 model has the lowest relative error overall, although the JB8ROM and ML ROM rival the performance of the JB2008 to within about ±2% until the final 24 hours prior to reentry. The relative performance of all models except the NRLM00 is worst at 24 hours prior to reentry, with all models excluding the NRLM00 recovering slightly at 12 hours before reentry.

The TIEROM shows an especially pronounced performance improvement from 24 to 12 hours prior to reentry, with a 15% decrease in its relative error over this time window. The improvement is so significant that at the nominal 12-hour mark, the TIEROM is the second-best model, after the JB2008. This is consistent with the

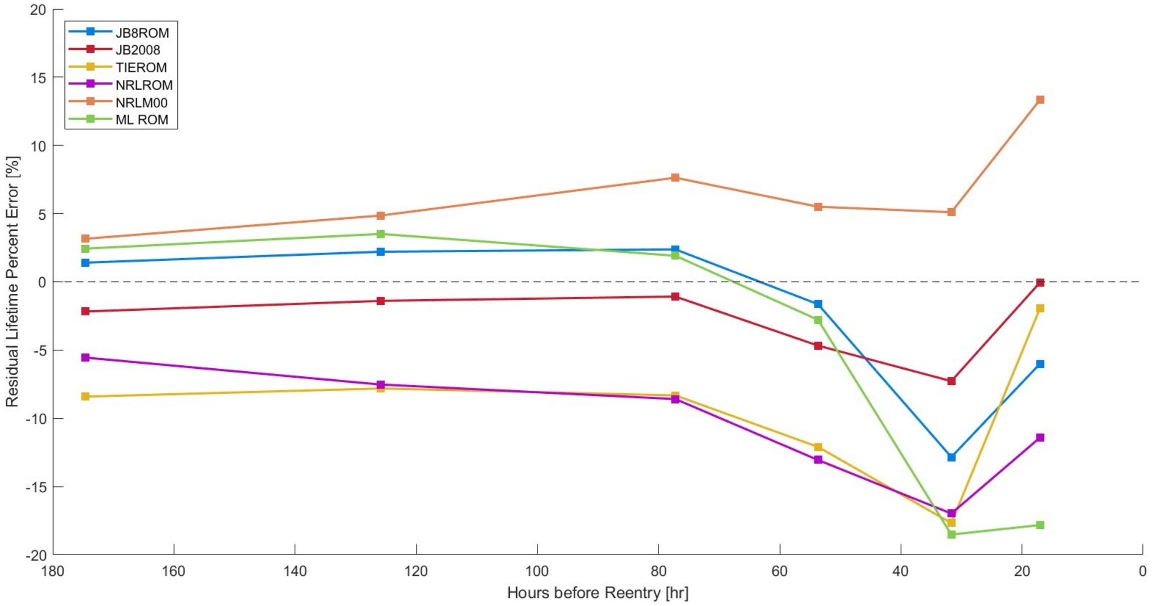


Figure 5-41: Percent error (%) of each model over time using the Pardini-Anselmo time-varying BC.

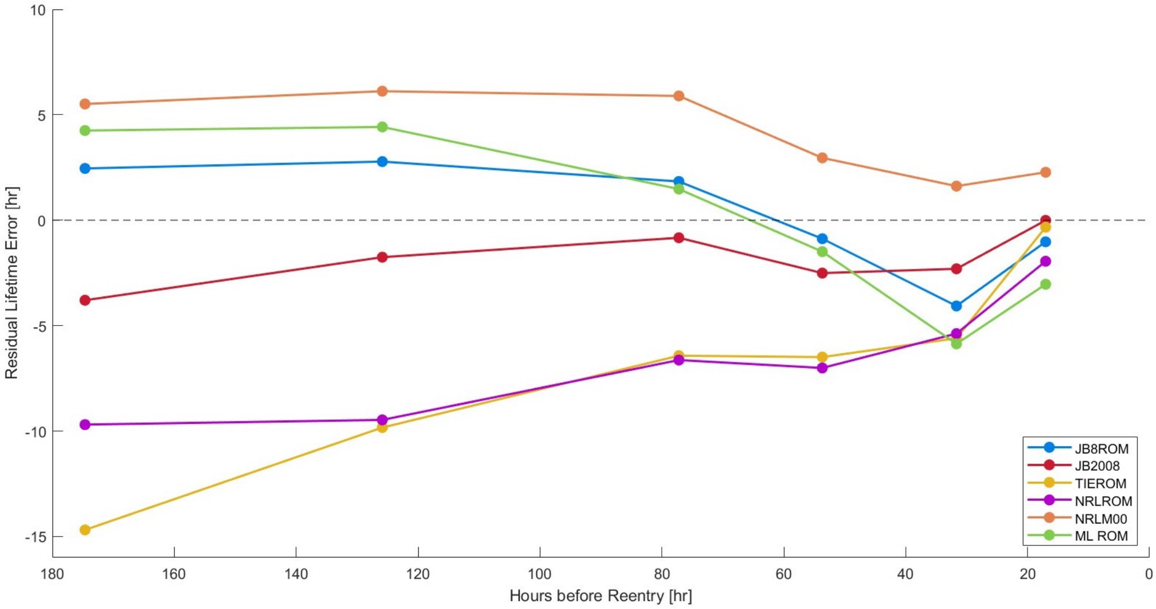


Figure 5-42: Absolute error (hrs) of each model over time using the Pardini-Anselmo time-varying BC.

pattern of its behavior in both of the other BC cases, where the TIEROM demonstrated worse performance until shortly before reentry, at which point its relative error dropped suddenly and the TIEROM yielded results very similar to those of the JB2008 model. One possible explanation for this behavior is the influence of the JB2008-based initial ROM state, which may be stronger over short prediction windows than long windows; however, this is not confirmed.

A comparison with the JB8ROM, which is also initialized with JB2008, shows that it, like the TIEROM, echoes JB2008 performance for shorter simulations, albeit at a diminished intensity. Both TIEROM and JB8ROM show their highest error at the 24-hour mark but improve at the 12-hour mark, like the JB2008 model. The JB8ROM is expected to show similar behavior to the JB20088, since it is trained and initialized on that model. However, it is uncertain why the impact of JB2008 initialization on the TIEROM appears so pronounced over short simulation durations for the Progress-M 27M particularly.

In terms of absolute error, most of the models converge towards the true reentry epoch as the simulation start time approaches the reentry time. Exceptions include the JB8ROM and ML ROM, which experience their highest absolute error of about -4 and -6 hours (underestimating the true residual lifetime), respectively, at 24 hours prior to reentry. Additionally, at 12 hours prior to reentry, the absolute error of the NRLM00 increases slightly over its 24-hour absolute error.

The overall accuracy of the six models is quite strong over long prediction windows, with all models maintaining ±9% or lower residual lifetime prediction error for nominal 7-day, 5-day, and 3-day prediction windows. Performance begins to diverge at 2 nominal days prior to reentry, when the absolute percent error of the TIEROM and NRLROM increases to 12-13%. Over the last 24 hours, model performance is increasingly inconsistent. However, all models stay within the benchmark of ±20% error for all prediction windows using the PA time-varying BC model. Only the JB2008 meets the ±10% benchmark for all prediction windows, though both JB2008-based ROMs maintain sub-5% error between 7 and 2 days prior to reentry, as with the PA mean BC.

5.3.4 Overall Results

The mean absolute percent error for all density and ballistic coefficient models for the Progress-M 27M case are shown in Figure 5-43.

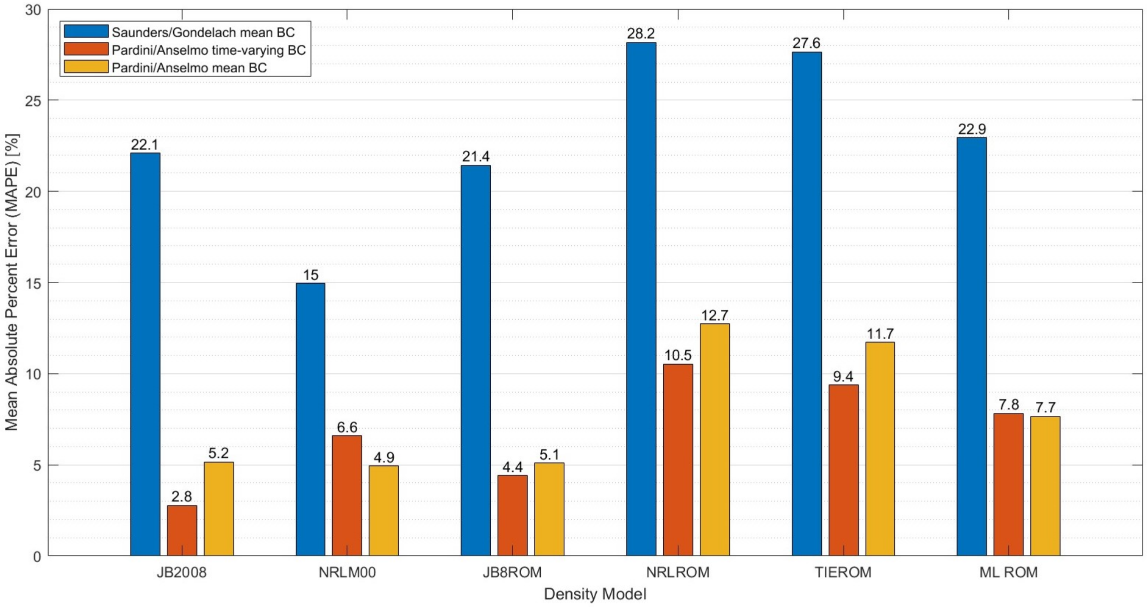


Figure 5-43: Mean absolute percent error (MAPE) across all prediction window lengths, for all models and all BCs.

As in previous test cases, a single BC model emerges as most successful for the majority of the density models, with NRLM00 as an exception. In this case, the most successful BC model is the PA time-varying BC, while the SG BC results in the poorest performance for all density models. Another difference in this test case is that the ML ROM, like the NRLM00, has lower MAPE with the PA mean BC, though only by a tiny fraction of a percent. As with previous cases, NRLM00 exhibits an opposite performance trend to other density models across the different BCs, tending to do worse on BC models which improve results for other density models.

The best performance across all density-BC model combinations came from the JB2008, with 2.8% residual lifetime MAPE across all prediction windows. This performance was achieved with the time-varying BC. For the same BC, the next-best model was JB8ROM with 4.4% MAPE, followed by NRLM00 at 6.6% and ML ROM at 7.8%. Once again, the NRLROM was the only model which did not achieve the ±10% "good performance" benchmark in terms of MAPE averaged across all simulation durations, though for the time-varying BC it was very close at 10.5% MAPE. All models met the ±20% standard by a wide margin for both of the PA BCs, but only the NRLM00 had <20% MAPE with the SG BC. The JB8ROM and JB2008 met the <5% "excellent performance" standard with the time-varying BC in terms of MAPE, and missed out by only 0.1-0.2% for the PA mean BC. The NRLM00 also had <5% MAPE for the PA mean BC.

Unlike in the UARS case, the TIEROM and JB8ROM were not in close agreement in terms of MAPE across all prediction windows. Instead, the TIEROM behaved very similarly to the NRLROM, with MAPE agreement to within 1.1% for all BC models. The JB8ROM, on the other hand, showed very similar MAPE performance to the JB2008 across all BC models, agreeing to within 1.6% for the time-varying BC and within 0.7% on the mean BC models.

Next, results were split into MAPE over long and short prediction windows.

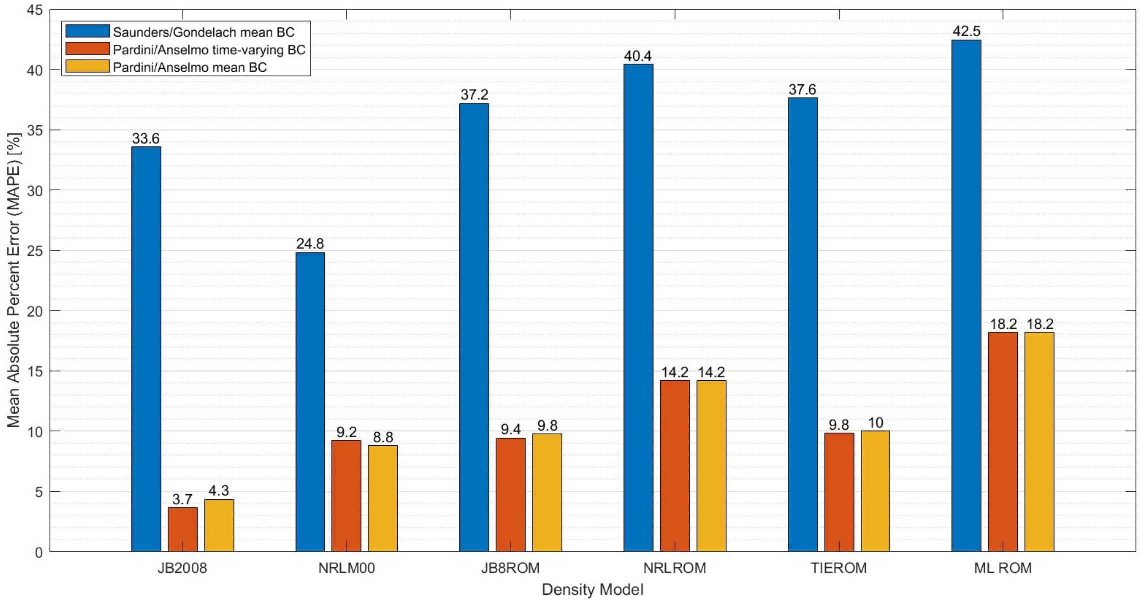


Figure 5-44: PM27M residual lifetime prediction MAPE for all density and BC models for short prediction windows of one nominal day or less.

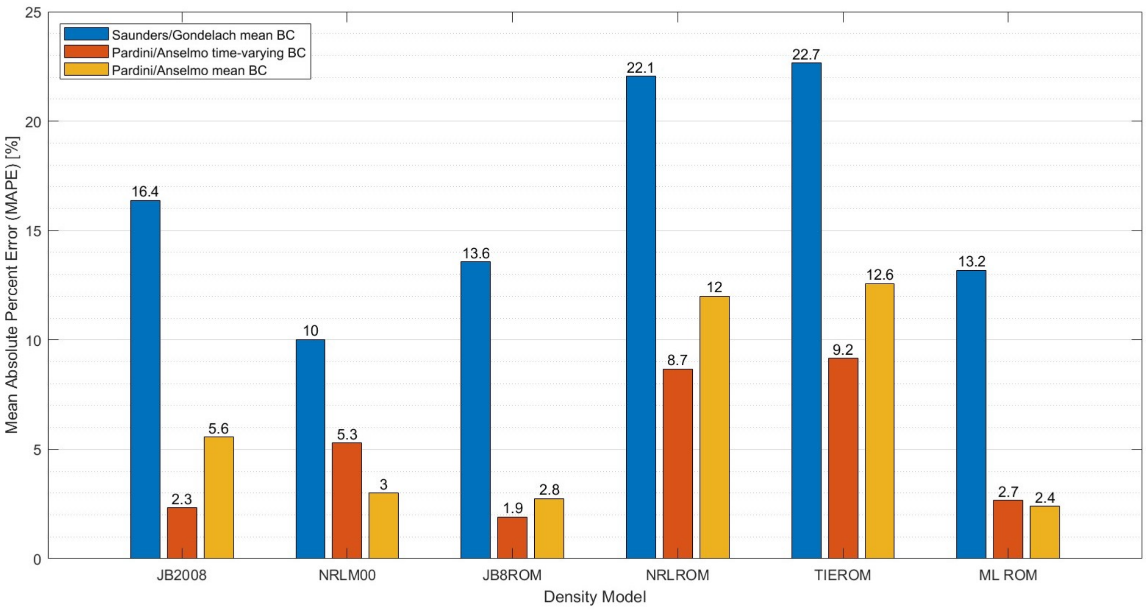


Figure 5-45: PM27M residual lifetime prediction MAPE for all density and BC models for longer prediction windows of 2-7 nominal days.

Although the two empirical models were the top performers in terms of short-term reentry prediction MAPE for all BCs, density model performance was very different over long prediction windows. With the exception of the TIEROM paired with the PA mean BC, every single ROM-BC combination performed better in terms of MAPE over long prediction windows than over short prediction windows. The reduction in MAPE ranged from 0.6% (TIEROM with time-varying BC) to 29.3% (ML ROM with Saunders-Gondelach BC). Most ROM-BC combinations had lower MAPE by >5% over long prediction windows. The ML ROM had an astonishing 20% average reduction in MAPE for long predictions as compared to short prediction windows, across all BCs. In fact, the ROM advantage for long prediction windows brought the JB8ROM into the lead for the time-varying BC, and the ML ROM into the lead for the PA mean BC for longer prediction windows.

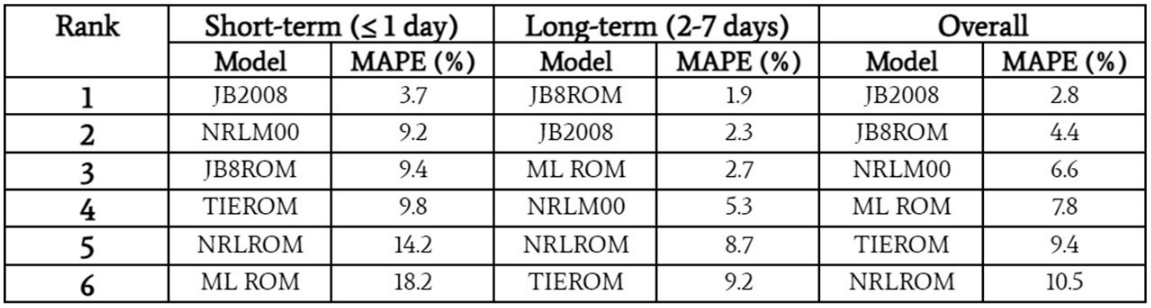
Furthermore, over long prediction windows, the JB8ROM outperformed the JB2008 in terms of MAPE for all BCs and the NRLM00 for two out of three BCs, while the ML ROM outperformed both the JB2008 and NRLM00 for two out of three BCs.

Even for the time-varying BC, where JB2008 out-performed the ML ROM, the difference between the two density models' MAPE was only 0.4%. Over long prediction windows, the JB8ROM and ML ROM were the only two density models to meet the ±5% "excellent performance" standard in terms of MAPE for multiple BC models; the JB2008 and NRLM00 each achieved this standard for only one BC model.

Interestingly, the TIEROM-JB8ROM similarity observed in the UARS test case re-emerges in short-term Progress-M 27M reentry predictions, with these two models agreeing in MAPE to within 0.4% for all BCs. For long-term predictions, however, the TIEROM behaved like the NRLROM, with MAPE agreement to within 0.6% across all BCs.

Table 5.3 ranks models by MAPE for the overall best BC model, the PA time-varying BC.

Table 5.3: Density models ranked by MAPE performance on PM27M test case with Pardini-Anselmo time-varying BC.



Rank	Short-term (≤ 1 day)		Long-term (2-7 days)		Overall	
	Model	MAPE (%)	Model	MAPE (%)	Model	MAPE (%)
1	JB2008	3.7	JB8ROM	1.9	JB2008	2.8
2	NRLM00	9.2	JB2008	2.3	JB8ROM	4.4
3	JB8ROM	9.4	ML ROM	2.7	NRLM00	6.6
4	TIEROM	9.8	NRLM00	5.3	ML ROM	7.8
5	NRLROM	14.2	NRLROM	8.7	TIEROM	9.4
6	ML ROM	18.2	TIEROM	9.2	NRLROM	10.5

HPOP run time per hour of simulation time, averaged across all BCs and all prediction windows, is shown in Figure 5-46.

Once again, the PODROMs dominate in terms of computational speed, averaging 70.1% faster than the empirical models, while the ML ROM averages 28.6% faster. The advantage of ROMs for longer simulations is even more striking in this test case, where the JB8ROM achieves both the lowest MAPE on the leading BC model, and the fastest run time.

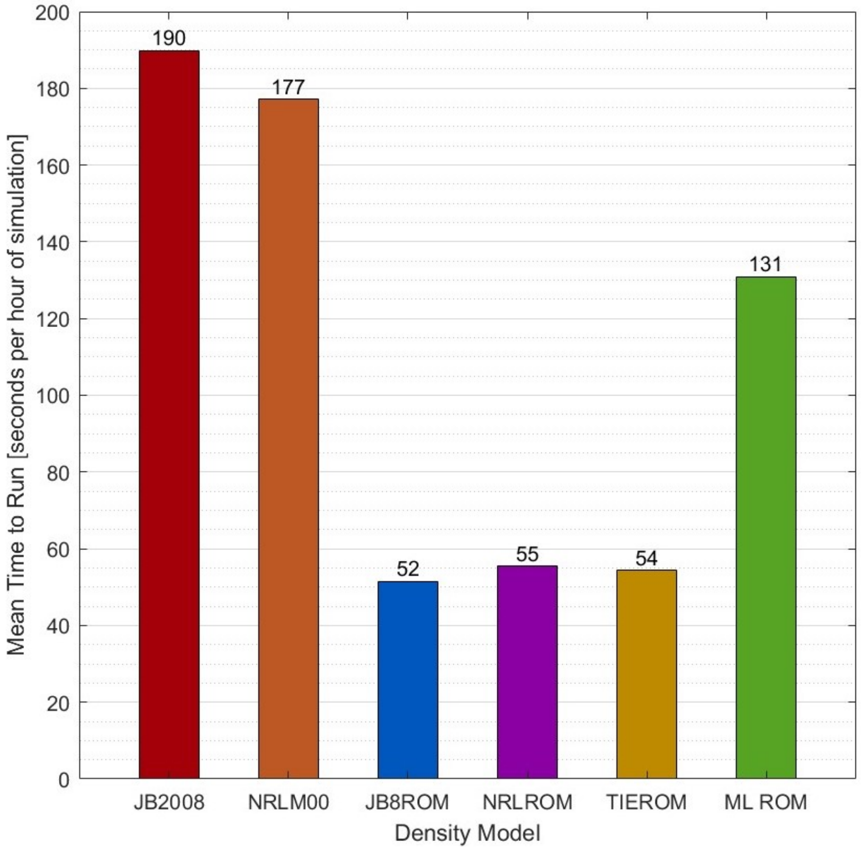


Figure 5-46: Average HPOP run time by density model for PM27M test case.

5.3.5 A Final Note on Reentry Locations

Figure 5-47 gives a final example of the sensitivity of reentry location to reentry time. Ground tracks and 100-km locations for the final few orbits of Progress-M 27M are shown for nominal 7-day and 12-hour simulations with JB8ROM, JB2008, and ML ROM. As Figure 5-47 demonstrates, reentry location is both highly sensitive to, and nonlinearly related to, reentry time. A reentry time which is incorrect by an integer multiple of the orbital period will provide a better reentry location estimate than a reentry time estimate which is off by only a fraction of an orbital period. This is clearly demonstrated in the bottom plot of Fig. 5-47. The JB8ROM reentry time is only about 60 minutes before the JB2008 prediction, but the two models' 100-km-altitude location predictions are over 10,000 km apart. On the other hand, the

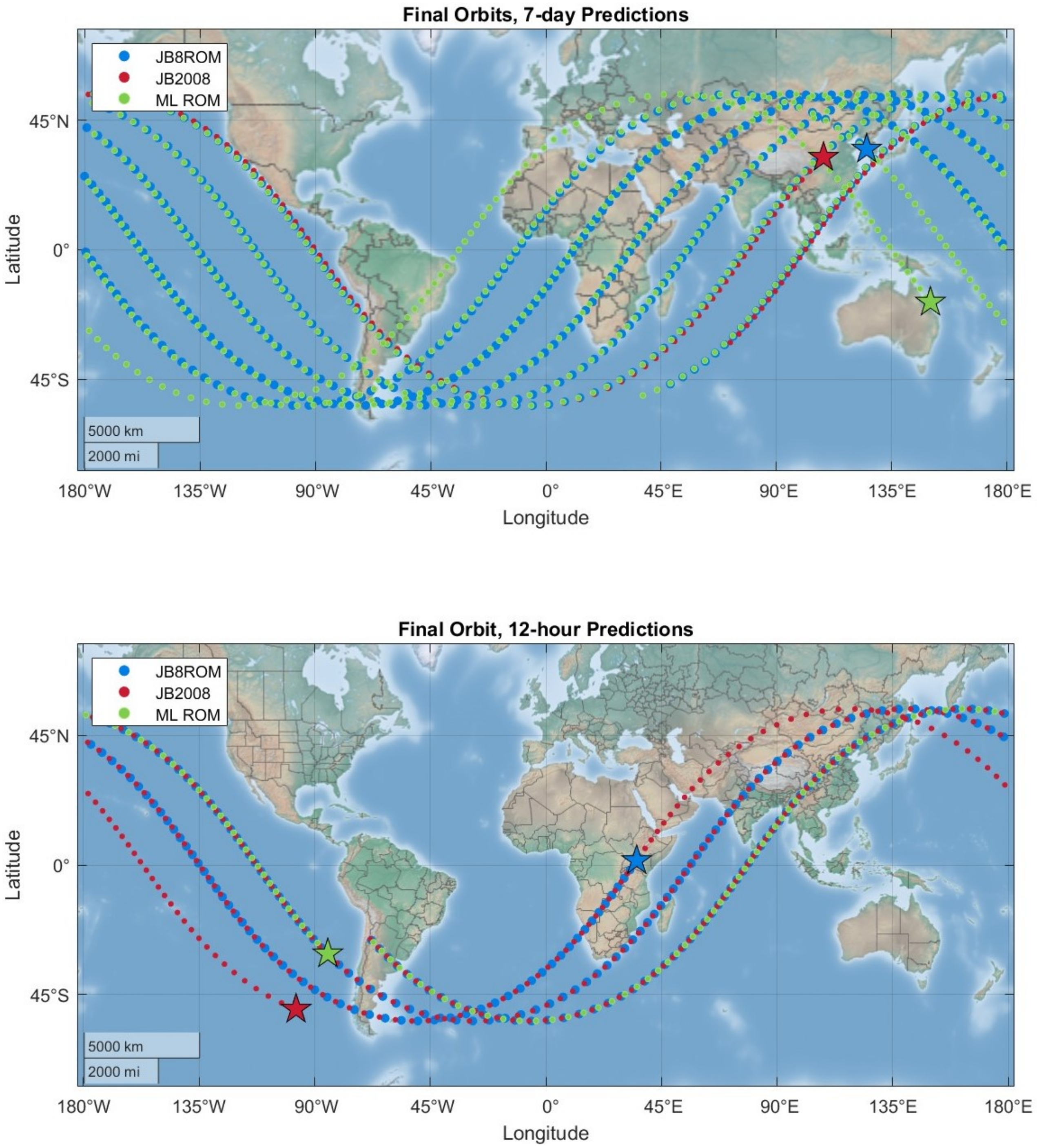


Figure 5-47: Ground tracks and 100-km reentry locations (starred) predicted for PM27M with Pardini-Anselmo time-varying BC.

ML ROM predicts reentry 3 hours (about 2 orbital periods) before the JB2008, and its 100-km location prediction is much closer to JB2008 due to the periodic nature of orbital motion.

For reference, the JSpOC reentry assessment of the Progress-M 27M found that its 80-km reentry point was 350-1300 km off the coast of south Chile [77], which is in good agreement with the 100-km reentry location estimated by the JB2008 in Fig. 5-47.

5.4 Summary

This chapter presented residual lifetime prediction estimates with four ROMs and two empirical density models for the UARS, Fobos-Grunt, and Progress-M 27M reentries. Reentry simulations were run for all density models and 3-4 BC models, over prediction windows ranging from a nominal week to 12 hours prior to reentry.

One noteworthy observation from these results is the importance of an effective BC model. This was also studied in Chapter 4, where BC uncertainty was found to be the primary contributor to reentry time uncertainty among the three uncertainty sources examined. In the current chapter, no single BC model was found to be the most effective for all test cases. Further study of BC modeling is out of the scope of this paper, but the need for improved BC estimates is a clear conclusion of the preceding results.

In most cases, JB2008 provided the most accurate reentry time predictions. ROMs were not found to generally outperform both of the empirical models, but the JB8ROM and TIEROM in particular showed comparable performance to the empirical models (within ~1-4% overall MAPE in most cases) when paired with the best-performing BC models. This accords with expectations for ROMs trained on empirical models. Without data assimilation, each ROM aims to replicate the density model it was trained on, at a small relative error due to truncation of spatial modes.

The only model which could reasonably be expected to perform better than the empirical models was the TIEROM, which was trained on a physics-based model. In two out of the three test cases, TIEROM did show excellent performance. In terms of MAPE for these two cases (UARS and Fobos-Grunt), TIEROM beat out all models except for the JB2008 in simulations using the overall best BC model. However, the TIEROM never outperformed the JB2008 in terms of MAPE. Chapter 4 provides an explanation for this result. The UQ analysis in Section 4.3 amply demonstrates the strong persistent impact of the initial ROM state on ROM performance. Due to the prohibitive computational expense of running TIE-GCM to generate snapshots for ROM state initialization, the TIEROM was initialized on JB2008. The lasting impact of the JB2008-generated initial ROM state may explain why TIEROM did not outperform JB2008, and also helps explain the similarities observed between the JB2008, JB8ROM, and TIEROM throughout the rest of this chapter.

Like the TIEROM, the ML ROM showed good performance in two of three test cases. In the PM27M test case, the ML ROM even outperformed all other models over long prediction windows for the PA mean BC, with a very low 2.4% MAPE. However, its performance was quite inconsistent. The ML ROM typically had >10% higher MAPE over short prediction windows as compared to long windows, and on the Fobos-Grunt test case, the ML ROM achieved <10% error on only one individual prediction on one BC. In fact, the ML ROM produced both some of the lowest and the highest individual errors observed among all density models throughout this thesis. Further work is needed to determine why the ML ROM's behavior was so inconsistent.

The JB8ROM was a consistently strong performer in all three test cases. Although the TIEROM slightly outperformed the JB8ROM on 2 out of 3 test cases, its performance was worse for the Progress-M 27M. The JB8ROM also showed the best individual performance of any ROM, when it surpassed all density other models with an impressive 1.9% MAPE over long reentry prediction windows on the Progress-M 27M test case, using the PA time-varying BC.

The NRLROM was the only ROM to show relatively poor performance throughout, though it was still able to achieve <10% MAPE for at least one BC on long prediction windows for two of three test cases. An unexpected finding for the NRLROM is that it did not tend to agree well with the NRLM00 model. Further research is needed to determine the cause of this unexpected behavior, but it should be noted that the NRLROM presented here deviated from the original DESTO formulation by using a NRLM00 snapshot for ROM state initialization, as opposed to the JB2008 snapshot used in the original version. This change may have worsened the performance of the NRLROM.

Figure 5-48 displays the percent error distributions for each density model across all BCs, prediction windows, and test cases. Mean percent error μ, standard deviation of percent error σ, and mean absolute percent error (MAPE) are displayed on each plot. For reference, the desired qualities for a density model percent error distribution are 1) low μ, indicating no significant bias in reentry predictions (accuracy); 2) low

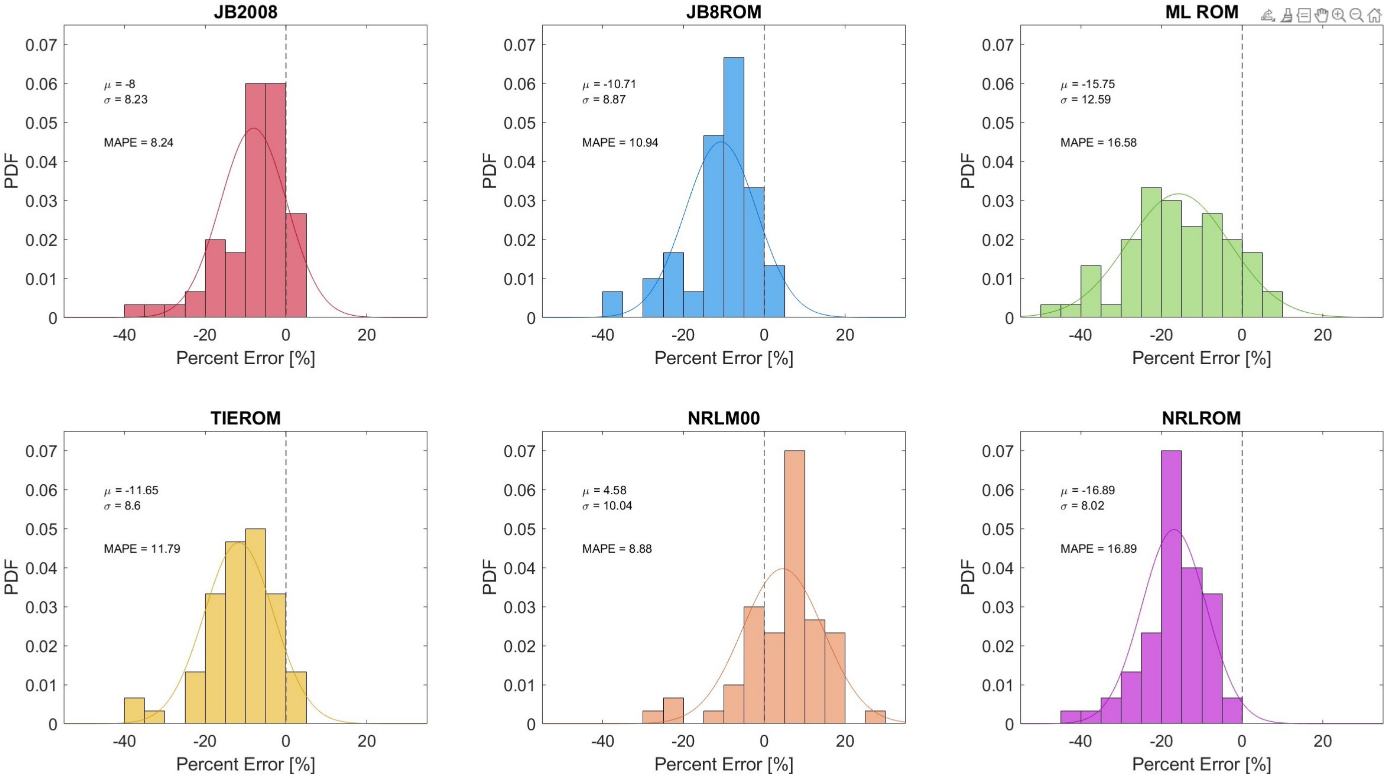


Figure 5-48: Percent error distributions by density model, for all BCs, all prediction windows, and all test cases.

σ, indicating that most predictions are consistent with each other (precision); and 3) low MAPE, indicating generally low error. Normal distributions with the same μ and σ are overlaid for reference. Note that because all BC models are included in this plot, even those which were very ineffective on some test cases, errors will be appear generally high in Fig. 5-48.

NRLM00 displayed the least overall bias. However, this is an expected result, given that NRLM00 was the density model used in BC estimation for all but one BC model on one test case (Fobos-Grunt with the JB2008-based SG BC). The JB8ROM and TIEROM had somewhat more bias (2.7-3.7%) than the JB2008, while the NRLROM and ML ROM had significant bias. All the PODROMs and JB2008 had similar standard deviations of 8-9%, while the NRLM00 and ML ROM had higher variance. The finding of higher standard deviation of error for the NRLM00 agrees with with the results of Marcos et. al. (2006), who found a relatively high data-to-model error standard deviation for the NRLM00 [55]. The ML ROM has the greatest variance, resulting from its inconsistent predictions.

With the exception of the NRLM00, all of the density model error distributions are centered below zero, showing a general tendency to underestimate residual lifetime across all test cases, BCs, and prediction windows. The explanation for this may be, again, the use of NRLM00 in BC estimation. The NRLM00 shows a tendency to *overestimate* residual lifetime, meaning that this model predicts the lowest densities. When NRLM00 is used for BC estimation, the lower densities of the NRLM00 result in higher BCs to compensate for the low densities when fitting BCs to observed data. The higher BCs then induce earlier reentries when paired with other density models, causing the underestimation bias. Except for the ML ROM, all of the density models produced left-tailed distributions. Large errors tended to be underestimations rather than overestimations, relative to each model's mean percent error.

Although the ROMs did not generally outperform both empirical models in terms of reentry time prediction accuracy, they dominated in terms of run time performance. HPOP orbit propagation with PODROMs averaged more than 70% faster than empirical models. The ML ROM was slower than PODROMs, but was still at least 26-28% faster than empirical models when used in HPOP. Additionally, optimizing the ML ROM for run time could improve its speed. Consistent with the results of Arenillas et. al. (2020), NRLM00 was found to be faster on average than JB2008 [6], but still much slower than the ROMs.

An interesting and potentially useful feature of ROM performance was lower relative error over prediction windows >2 days in length. In almost every single case, ROMs showed notably lower MAPE for long than for short prediction windows. Over long prediction windows, the JB8ROM never had more than 2% higher MAPE than JB2008 for the best BC models, and never more than 4% higher MAPE than JB2008 for any BC model. In fact, for one test case (the PM27M), the JB8ROM outperformed the JB2008 in terms of MAPE for all BC models over longer prediction windows. As previously noted, the ML ROM tended to do much better for long than for short prediction windows. The TIEROM also achieved its best performance of 2.7% MAPE on long prediction windows for the UARS case with the SG BC.

Significant work remains in the area of orbit propagation for reentry prediction with ROMs. However, the results presented in this chapter validate the ability of ROMs, even without data assimilation, to provide reentry prediction performance relatively comparable to that of empirical models, at a drastically reduced run time.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

Conclusions

6.1 Key Findings and Importance

Reentry prediction is an increasingly important area for the application of atmospheric density models, as the rising population of objects in LEO leads to a greater risk of casualties and/or damage from debris surviving for RSO reentries. However, atmospheric density models contain large uncertainties, making accurate reentry prediction difficult, and can moreover be computationally expensive to run. This thesis has presented an extensive application of reduced-order atmospheric density modeling to the problem of reentry prediction for three uncontrolled satellite test cases, along with UQ for three key sources of reentry prediction uncertainty. The primary conclusion of the results presented herein is that, even without data assimilation, ROMs are able to provide reentry prediction accuracy within a few percent MAPE of JB2008 and NRLMSISE-00, with run times averaging up to 70% less than the empirical density models. The dramatically lower computational/memory cost of ROMs is one of their key advantages. The relative orbit propagation run time advantage of the ROMs is shown in Figure 6-1.

Three out of the four ROMs tested (JB8ROM, TIEROM, and ML ROM) provided residual lifetime MAPE of <10% on at least two out of three test cases, using the leading BC models for each case. According to the standards established in Section 2.1 based on a review of reentry prediction accuracy, this is considered good performance. The JB8ROM and TIEROM showed good performance on all three using at least one BC model, and the JB8ROM further achieved <8% MAPE with at least one BC on all test cases, and excellent performance of ∼5% MAPE on two of the three test cases. ML ROM achieved excellent performance at times, but was unexpectedly inconsistent across prediction windows and test cases. Also unexpectedly, the NRLROM did not agree well with the NRLM00. These unexpected ML ROM and NRLROM results may warrant further inquiry in future work. Overall, the NRLROM was the least accurate ROM.

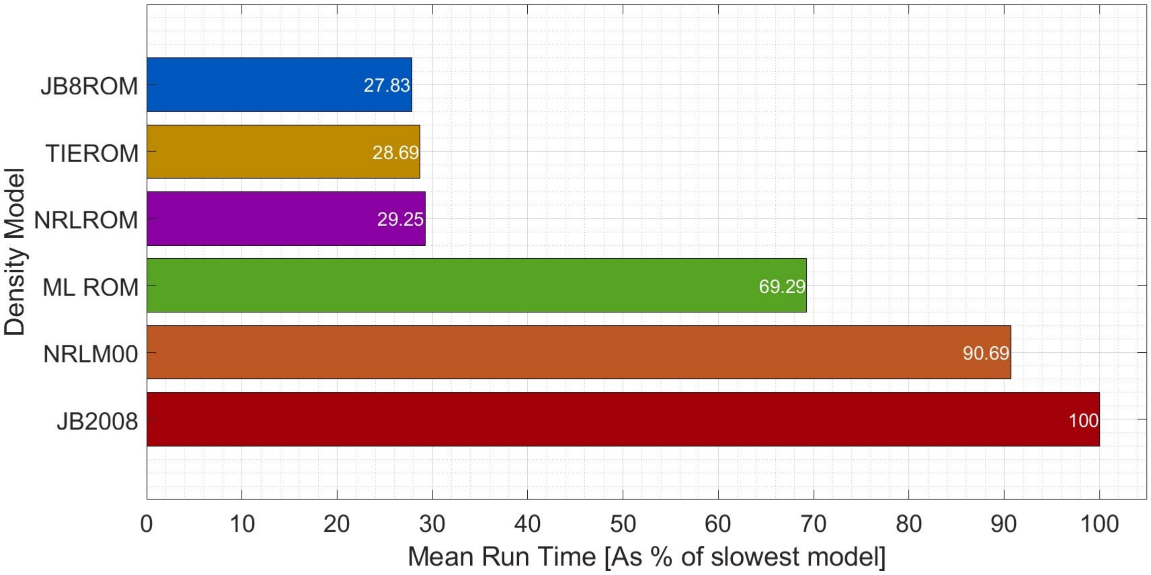


Figure 6-1: Run time per hour of simulation for all density models, as a percent of slowest model's run time per simulation hour. Averaged across all BCs, prediction windows, and test cases.

The ballistic coefficient was found in both Chapter 4 and Chapter 5 to be a key factor in the success of reentry predictions, but none of the BC models used in Chapter 5 was consistently effective across all test cases. In addition, the density model absorbed into the BC during estimation was found to significantly influence the reentry prediction results. Reentry prediction performance was generally best when orbit propagation was conducted with the same density model used for BC estimation, consistent with recommendations by Saunders et. al. (2012) [95]. In the UQ analysis of Chapter 4, underestimation of the BC resulted in greater reentry time errors than overestimation. Also in Chapter 4, initial position error was found to influence reentry predictions much less than BC.

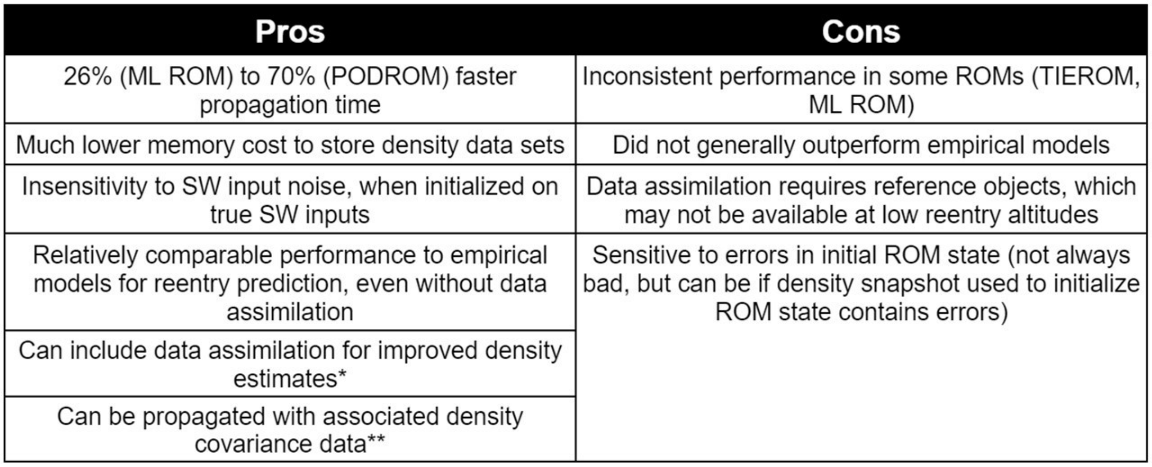
Across all test cases and all BCs, all density models except the NRLM00 showed a tendency to underestimate the residual lifetime. The underestimation bias was likely a result of using NRLM00 as the density model in BC estimation for most of the BC models, which lead to higher BCs to compensate for the lower densities estimated by NRLM00. Consistent with previous findings [55], NRLM00 was found to have a relatively high standard deviation of error. JB2008 and PODROMs had lower standard deviations. The inconsistent predictions of the ML ROM caused it to have the largest percent-error standard deviation of all the density models used.

When initialized on true SW inputs, ROMs were found to be effectively insensitive to noise in the SW inputs used for updating ROM states with DMDc. This quality could be very useful for density forecasting, as it shows that variance in SW prediction error would have little impact on ROM orbit propagation results, as long as the ROM was initialized on true SW inputs. In this regard, the ROMs were more successful than the JB2008 model, which was about 10 times as sensitive as the ROMs to SW input noise. SW UQ analysis also revealed the heavy impact of the initial ROM state on overall ROM performance. This finding was further illustrated by the relatively similar performance of the PODROMs which were initialized on JB2008 data (TIEROM and JB8ROM).

ROMs were found to have lower relative error for prediction windows of $\geq$2 days than on shorter prediction windows. This finding is very complementary to 1) the extremely fast run time of ROMs and 2) their insensitivity to SW noise when initialized on true SW data. In combination, these results make ROMs a strong prospect for longer-duration reentry predictions. Over long prediction windows using predicted SW inputs, ROMs may tend to show good results and less SW noise sensitivity relative to empirical models, and will also run much faster.

Table 6.1 summarizes the advantages and drawbacks of ROMs for orbit propagation and reentry prediction.

Table 6.1: Pros and cons of ROM density models for orbit propagation.



Pros	Cons
26% (ML ROM) to 70% (PODROM) faster propagation time	Inconsistent performance in some ROMs (TIEROM, ML ROM)
Much lower memory cost to store density data sets	Did not generally outperform empirical models
Insensitivity to SW input noise, when initialized on true SW inputs	Data assimilation requires reference objects, which may not be available at low reentry altitudes
Relatively comparable performance to empirical models for reentry prediction, even without data assimilation	Sensitive to errors in initial ROM state (not always bad, but can be if density snapshot used to initialize ROM state contains errors)
Can include data assimilation for improved density estimates*	
Can be propagated with associated density covariance data**	

* See [33, 61, 34]. ** See [32, 30].

The greatest advantage of the ROM density models presented in this thesis is their low computational demand, so this work will likely be most important for applications with very limited memory and/or computational resources. ROMs are much faster to run than empirical models, and ROM order reduction using POD stores density data in a very compressed form, even when the overhead of storing $\mathbf{U}_r$ and $\bar{\mathbf{x}}$ for POD decoding is considered. Although the order reduction (POD or autoencoder) and propagation (DMDc or NN) steps of ROMs have been used together throughout this thesis, it is important to note that these processes can be separated. POD could be performed after the fact to store density forecasts, without significant truncation error for a reasonable r-value. For example, 48-hour WAM-IPE density forecasts[1]

[1] https://www.swpc.noaa.gov/products/wam-ipe

could be encoded into a very low-memory format with POD. This could be useful for low-memory/low-resource applications, such as satellite onboard operations.

Based on the findings in this thesis, the author recommends consideration of ROM density models, especially the JB8ROM and TIEROM, for orbit propagation and reentry predictions in the following scenarios:

1. Longer-term predictions of 2-7 days, especially when SW index values may contain noise.

2. Applications with limited memory/computational resources. Users may consider applying only the POD encoding technique to their own density predictions for low-memory storage.

3. Applications where frequent, quick-to-run density estimates are preferred over infrequent but slightly more accurate estimates.

6.2 Strengths and Weaknesses of the Research

The work in this thesis improved upon the previous work of [10] in several key ways. First, test cases were selected for data availability. All three test cases used had associated IADC reentry campaigns and published reentry studies [75, 77]. For each test case, JSpOC published true reentry epochs with ±1 minute uncertainty [75, 77]. Dimensions and masses of the test cases were also well-known.

Secondly, this work conducted reentry predictions over a series of 6 prediction windows, beginning at 7 days prior to reentry and moving progressively closer to reentry, up to 12 hours prior. In contrast, previous work included only nominal 5-day predictions [10]. The inclusion of multiple prediction windows is much more representative of operational reentry campaigns, during which frequent predictions are made based on updated initial states and ballistic coefficients, as seen in [75, 77].

The work in this thesis also added several BC models for reentry prediction. Although no single BC model was found to be consistently most effective across all test cases, this measure did serve to highlight 1) the importance of the BC model in reentry prediction and 2) the significance of the density model used in BC estimation.

Another major advancement of this thesis was the inclusion of run time assessments for orbit propagation with ROMs as compared to empirical density models. The quantified run time advantage of ROMs over empirical models in orbit propagation is a new finding.

Finally, this thesis added UQ for reentry prediction uncertainty sources. This addition allowed for a comparison of the effects of BC, initial position, and SW uncertainties on reentry time predictions for two ROMs and JB2008. SW MC simulations revealed the relative insensitivity of ROM orbit propagation to SW noise, which has not been studied in previous work. SW UQ also revealed the persistent impact of the initial ROM state on ROM orbit propagation results.

Despite these major advancements, the author recognizes several areas for improvement in the research process. First, a sample set of only three test cases (chosen due to time and computational constraints) is quite low. The test cases used did allow for good data availability, but did not reflect 1) a significant range of varying solar/geomagnetic conditions; 2) variation in orbit type (such as the inclusion of highly-elliptical orbits); or 3) significant variation of individual object parameters (all test cases were large, uncontrolled objects, with BCs $\sim 10^{-2}$ or 10^{-3}). The exclusion of prediction windows $>$7 days represents another restriction on the scope of this work. Reentry prediction campaigns often start more than 7 days prior to reentry, so the longer prediction windows excluded from this study are certainly relevant for judging reentry prediction performance. Again, this limitation was due to constraints on time and computational resources.

Another limitation of the work in this thesis was the lack of information on uncertainty in predicted SW indices. The assumption of a zero-centered Gaussian noise model, with no correlation between error in the various SW indices, may not well characterize the uncertainty distribution actually present in SW index predictions. Therefore, the finding of ROM insensitivity to SW noise is restricted in scope by the simplistic uncertainty model used. The areas here identified as drawbacks of the research methodology in this thesis represent areas of improvement for future work.

6.3 Directions for Future Research

Besides the areas for improvement discussed above, a number of directions are available for future research on ROM application in orbital propagation for reentry prediction and other applications. Figure 6-2 shows recommended areas for future work, organized by an intended goal.

First, given the significant impact of the ballistic coefficient on reentry prediction performance, another avenue for further research is to generate Saunders-Gondelach BCs using each density model, and to pair each BC with its respective density model for reentry predictions, in accordance with the recommendation of Saunders et. al. (2012) [95]. This would circumvent the problem of absorbed NRLM00 density error in the BC causing increased error for non-NRLM00 density models. The expected outcome of pairing each density model with its respective BC for reentry prediction is improved overall performance for most or all of the density models at all prediction windows.

Another opportunity for future work, which would advance ROMs further towards operational application, is to remove the a posteriori availability of data assumed in this thesis. TLEs and space weather data are not available ahead of time for BC estimation and density model input, respectively. To create a realistic predictive scenario, BCs would be estimated using only TLEs prior to the start time of the simulation. Ideally, this experimentation would be performed using the aforementioned SG BC estimates paired with their respective density models. Additionally, true space weather inputs would be replaced with predicted index values available prior to the reentry simulation. These replacements would introduce additional uncertainty, in line with the uncertainty realistically expected in an operational environment. Using ROMs in a truly predictive sense, without a posteriori data availability, would better exemplify ROM performance in practice.

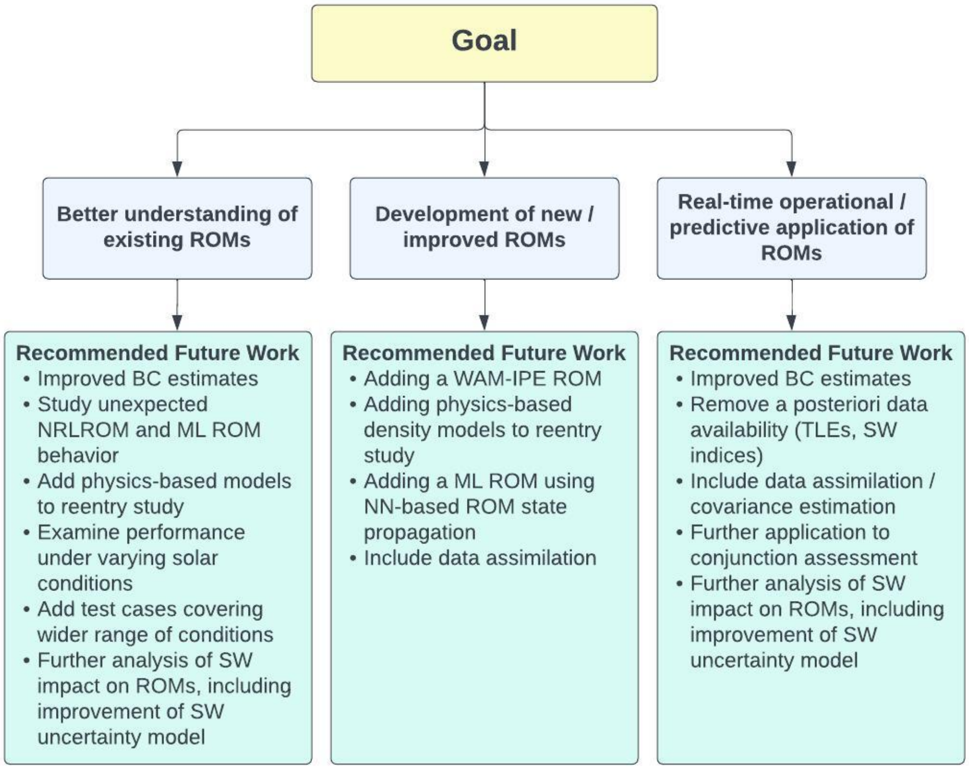

Goal
Better understanding of existing ROMs
Development of new / improved ROMs
Real-time operational / predictive application of ROMs
Recommended Future Work
• Improved BC estimates
• Study unexpected NRLROM and ML ROM behavior
• Add physics-based models to reentry study
• Examine performance under varying solar conditions
• Add test cases covering wider range of conditions
• Further analysis of SW impact on ROMs, including improvement of SW uncertainty model
Recommended Future Work
• Adding a WAM-IPE ROM
• Adding physics-based density models to reentry study
• Adding a ML ROM using NN-based ROM state propagation
• Include data assimilation
Recommended Future Work
• Improved BC estimates
• Remove a posteriori data availability (TLEs, SW indices)
• Include data assimilation / covariance estimation
• Further application to conjunction assessment
• Further analysis of SW impact on ROMs, including improvement of SW uncertainty model

Figure 6-2: Recommended directions for future work, based on the goal of interest.

To use ROMs to greater advantage, further work should be conducted on data-assimilating ROMs for use in orbit propagation. Assimilation of data using a Kalman filter allows ROMs to make real-time density corrections based on observations from reference objects, which reveal current drag conditions. Data assimilation was found to make ROMs more accurate than empirical models [34]. Additionally, ROM density uncertainty can be propagated with the density estimate using this technique, as described in [32], which can help define accurate uncertainty windows for propagated RSOs. In practice, applying this technique to reentry prediction might be challenging due to lack of reference objects at comparable altitudes. Additionally, the effect of data assimilation on run time (a key advantage of ROMs) is unknown. However, both conjunction assessment and reentry prediction using data-assimilating ROMs are worthwhile areas of investigation.

Other areas of future interest include training a POD and/or ML ROM on the NOAA WAM-IPE density model for reentry prediction; further study of the inconsistency of ML ROM performance; adding physics-based density models, such as TIE-GCM or WAM-IPE to this reentry prediction study; including a ML ROM with neural network ROM state propagation instead of DMDc, to fully realize the benefits of density field non-linearity captured by the ML autoencoder; studying ROM performance on reentries during both high and low solar conditions and other investigations of SW impact on ROMs; and continuation of the work in [30, 32] by further application of ROMs to conjunction analysis.

Code Availability

Code and results data for this thesis are available on Github. Please contact Nicolette Clark at nlclark2021@gmail.com for access.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Appendix A

A.1 UARS Run Time Plots

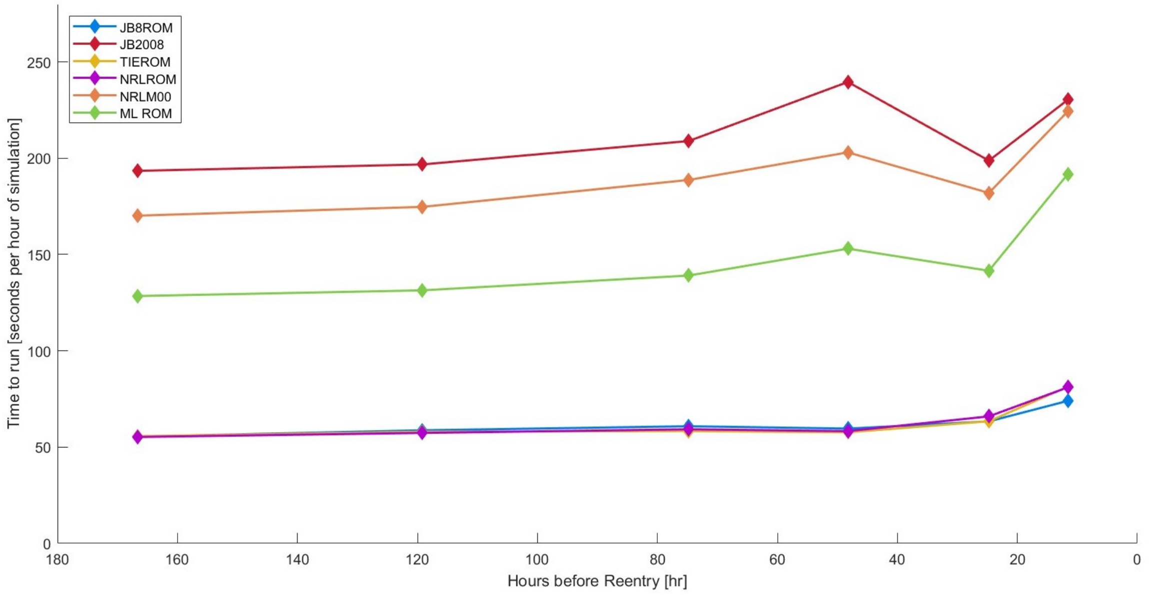


Figure A-1: Run time performance of all density models on UARS test case for all simulation durations for the Saunders-Gondelach mean BC simulation set.

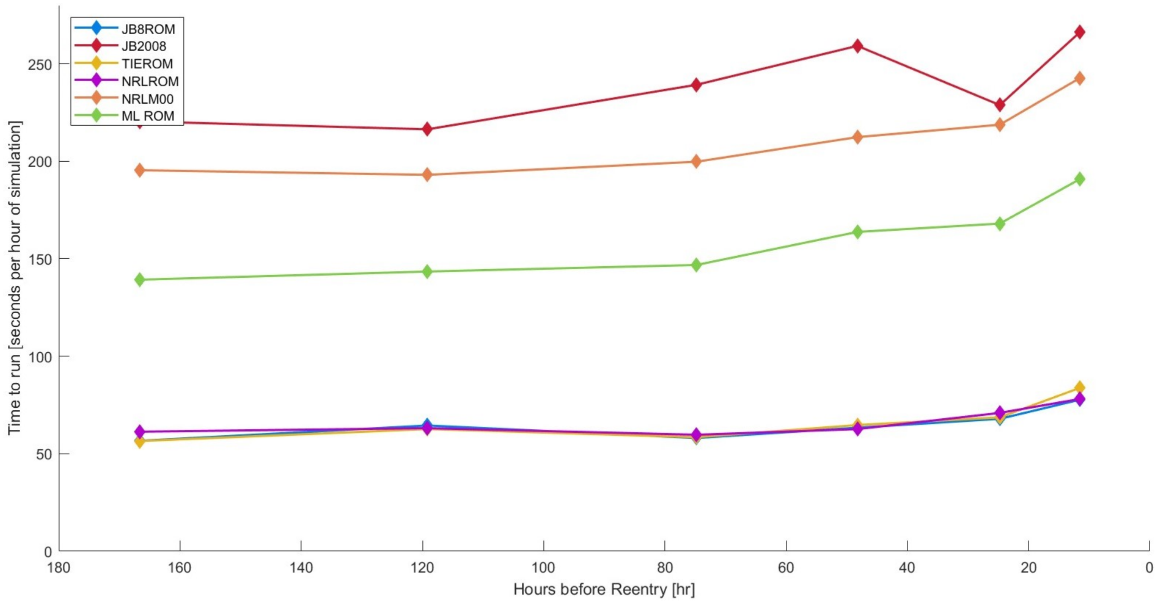


Figure A-2: Run time performance of all density models on UARS test case for all simulation durations for the Pardini-Anselmo mean BC simulation set.

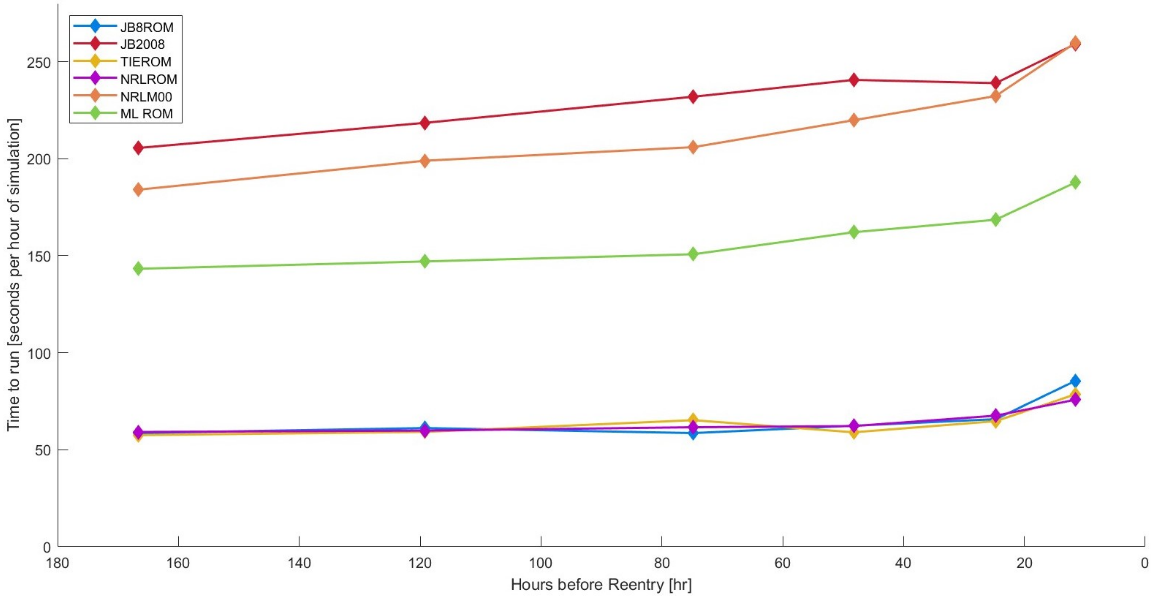


Figure A-3: Run time performance of all density models on UARS test case for all simulation durations for the Pardini-Anselmo time-varying BC simulation set.

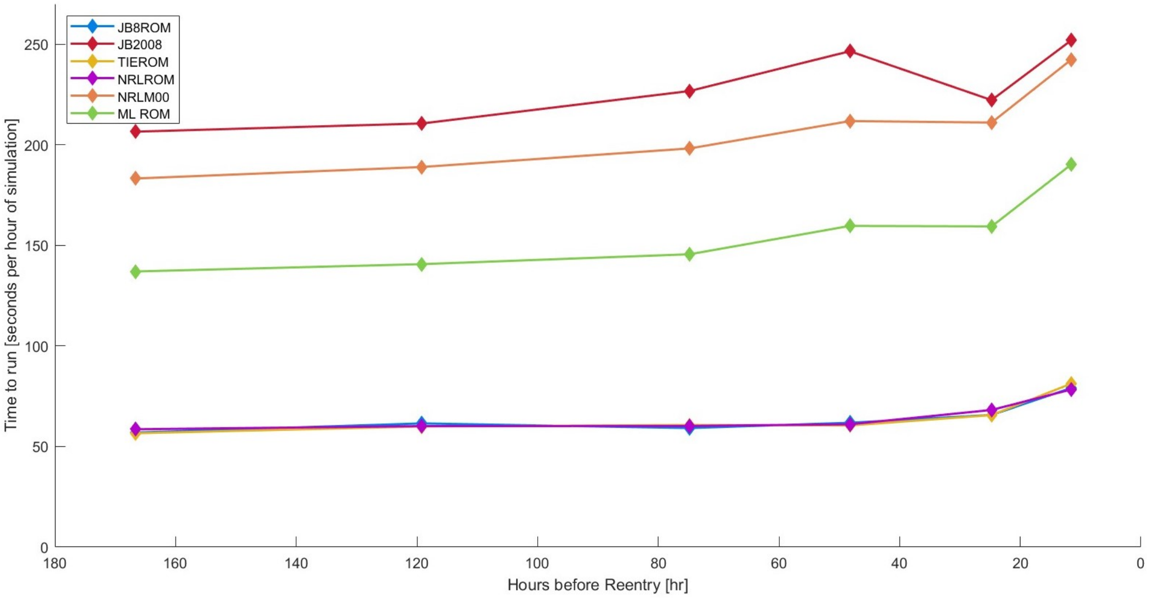


Figure A-4: Run time performance of all density models on UARS test case for all simulation durations, averaged across all BC models.

A.2 Fobos-Grunt Run Time Plots

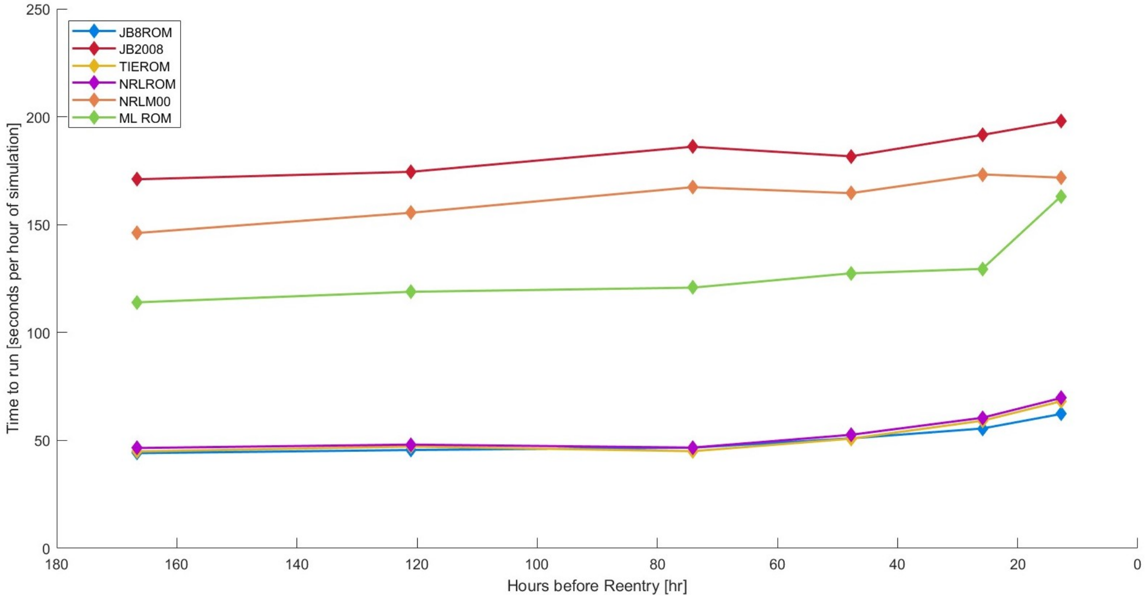


Figure A-5: Run time performance of all density models on Fobos-Grunt test case for all simulation durations for the Saunders-Gondelach mean BC simulation set.

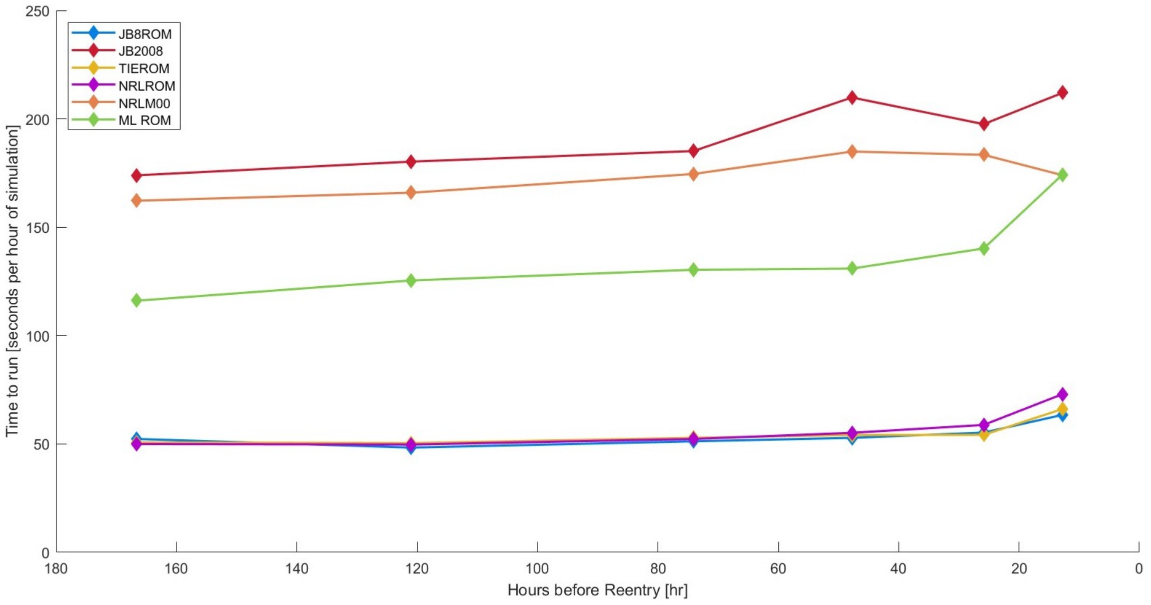


Figure A-6: Run time performance of all density models on Fobos-Grunt test case for all simulation durations for the Pardini-Anselmo mean BC simulation set.

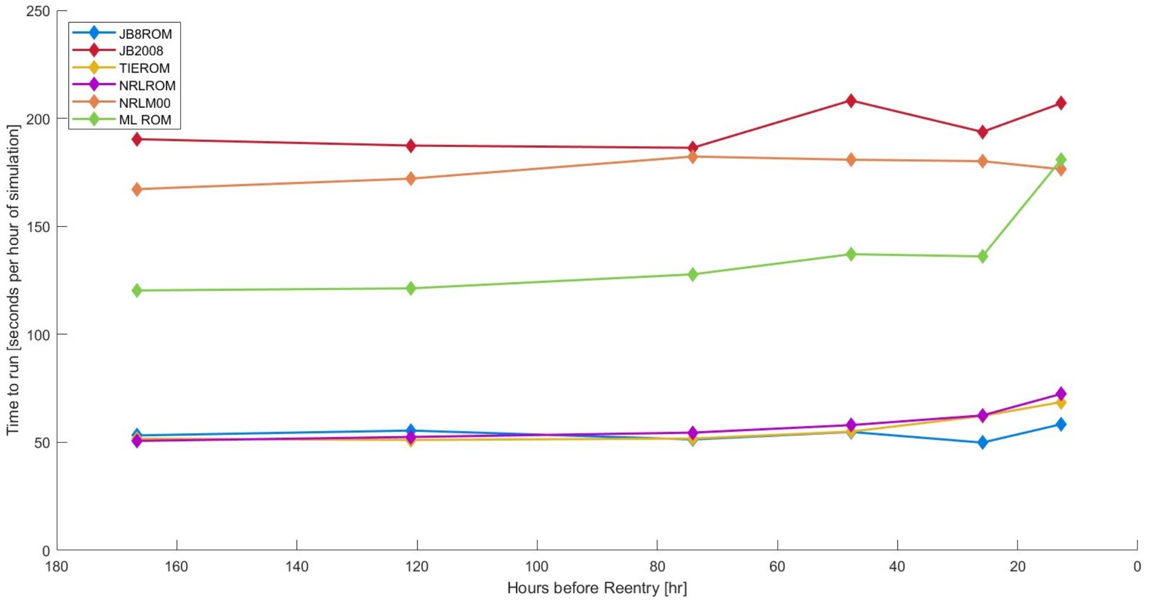


Figure A-7: Run time performance of all density models on Fobos-Grunt test case for all simulation durations for the Pardini-Anselmo time-varying BC simulation set.

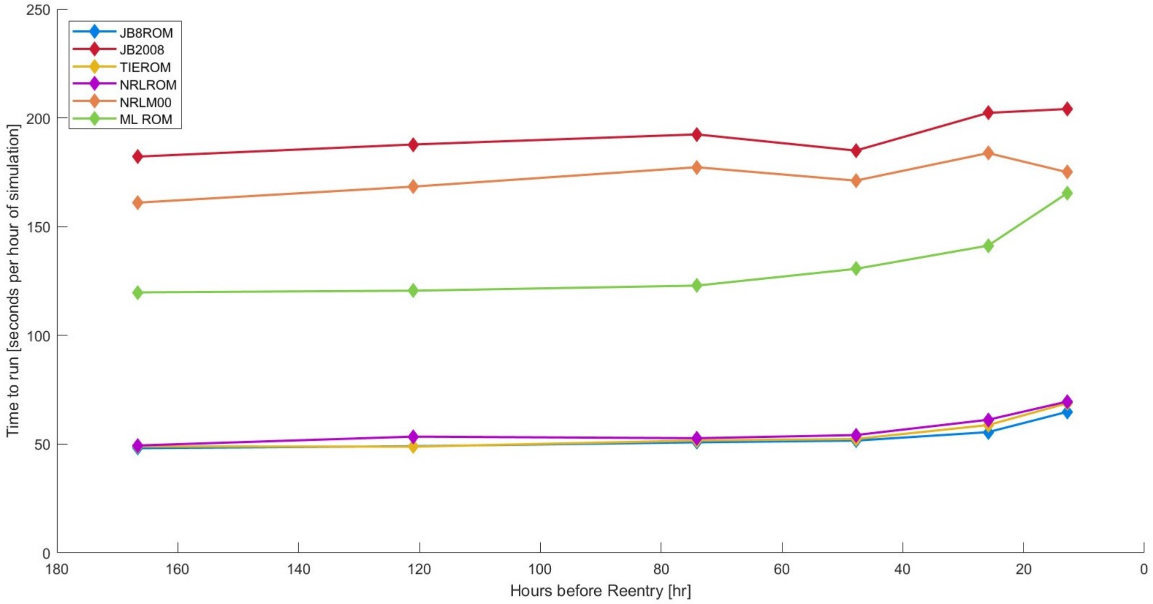


Figure A-8: Run time performance of all density models on Fobos-Grunt test case for all simulation durations for the simulation set using Saunders-Gondelach mean BC derived with the JB2008 density model.

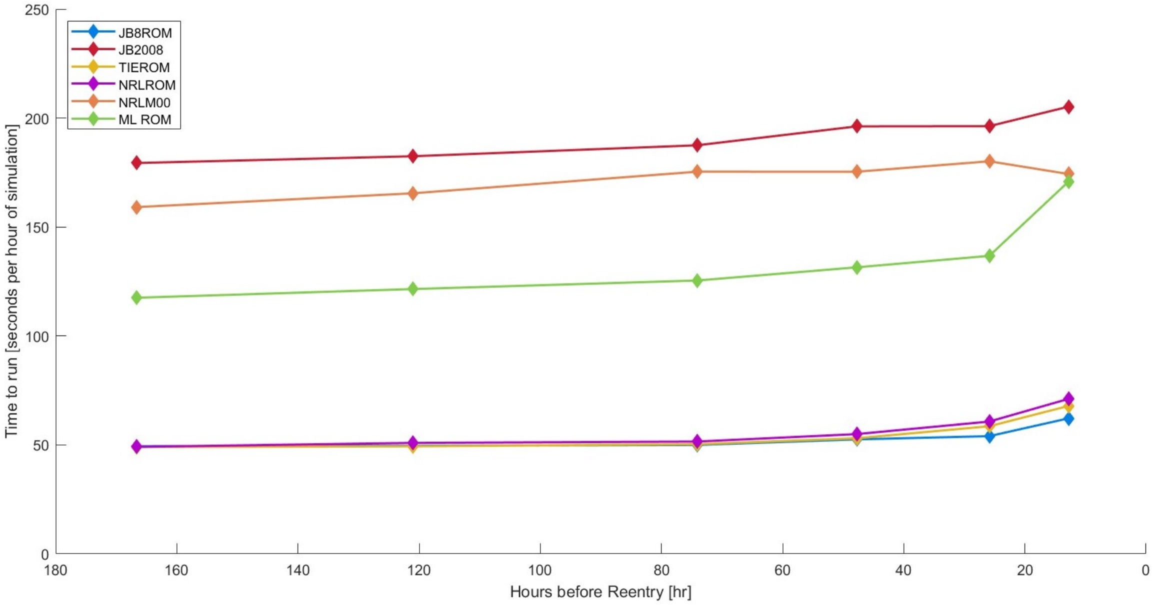


Figure A-9: Run time performance of all density models on Fobos-Grunt test case for all simulation durations, averaged across all BC models.

A.3 Progress-M 27M Run Time Plots

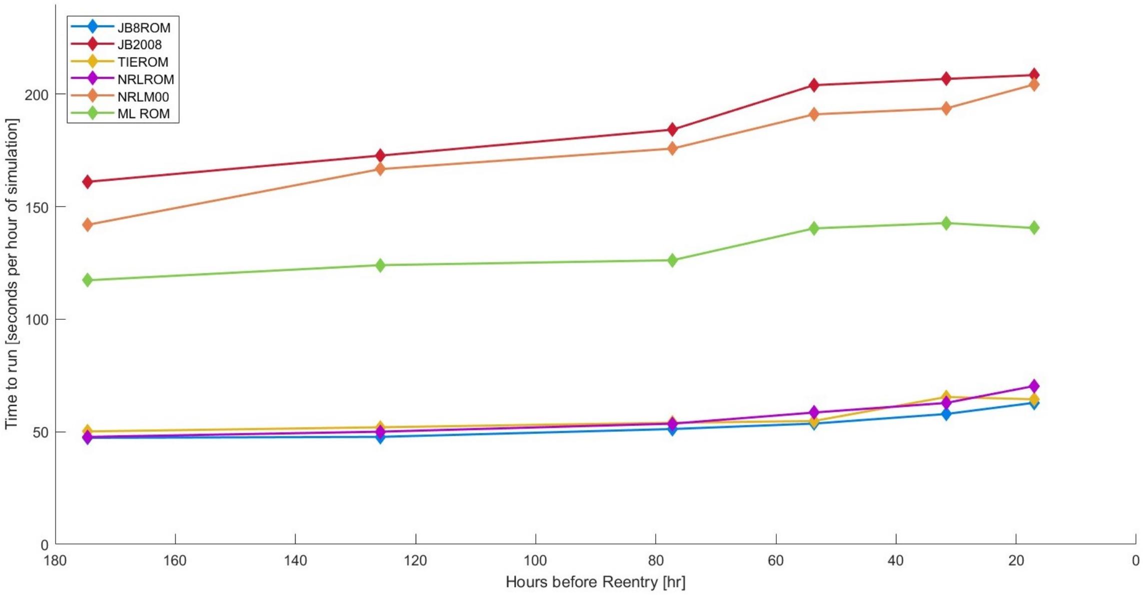


Figure A-10: Run time performance of all density models on Progress-M 27M test case for all simulation durations for the Saunders-Gondelach mean BC simulation set.

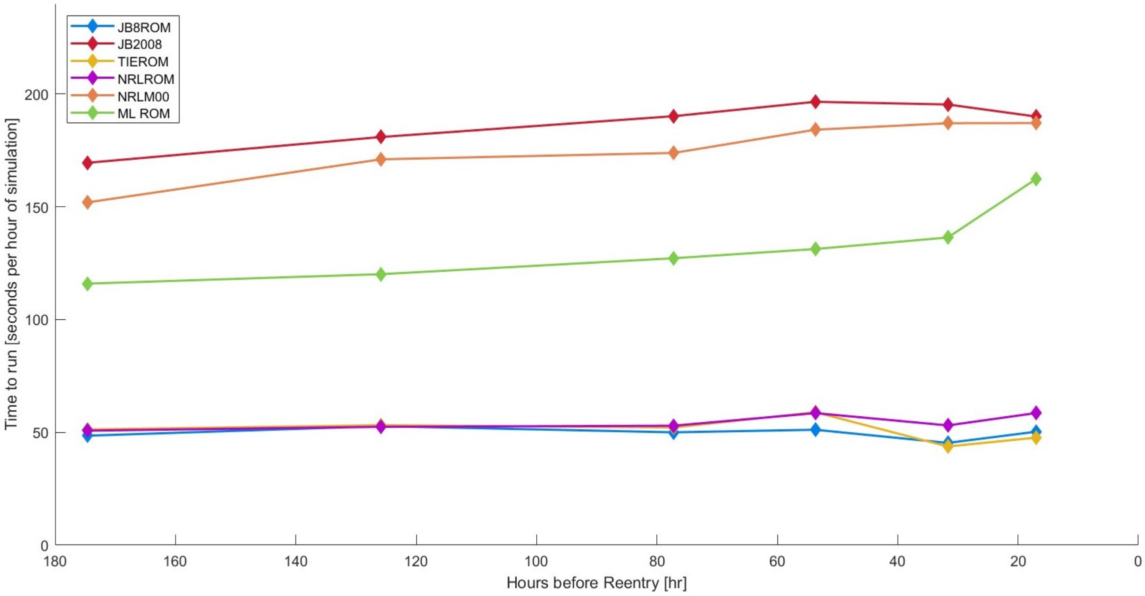


Figure A-11: Run time performance of all density models on Progress-M 27M test case for all simulation durations for the Pardini-Anselmo mean BC simulation set.

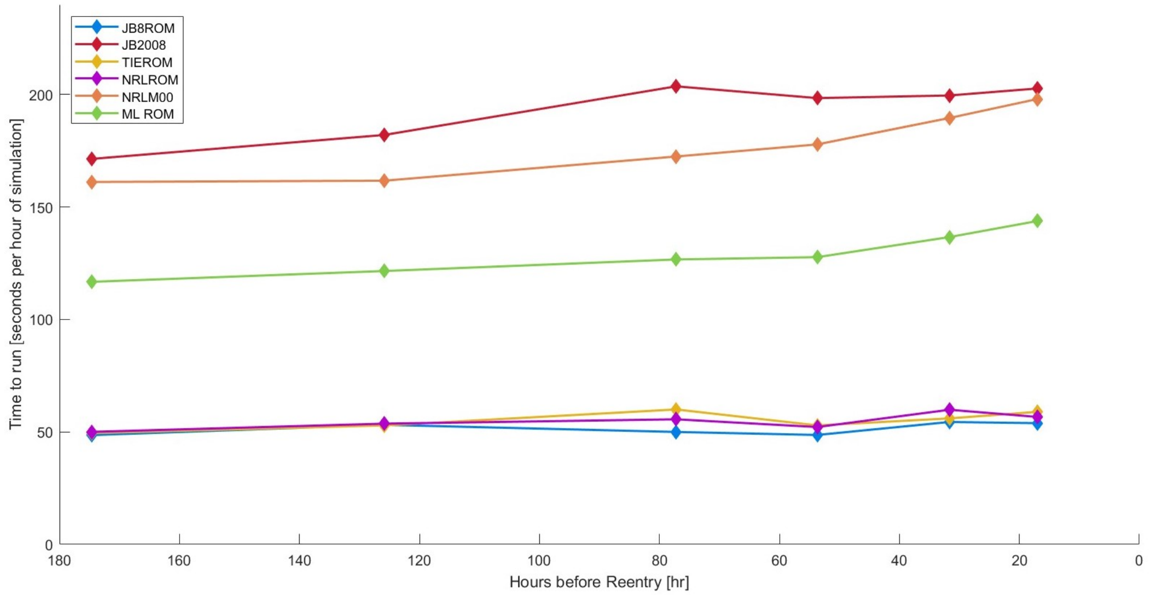


Figure A-12: Run time performance of all density models on Progress-M 27M test case for all simulation durations for the Pardini-Anselmo time-varying BC simulation set.

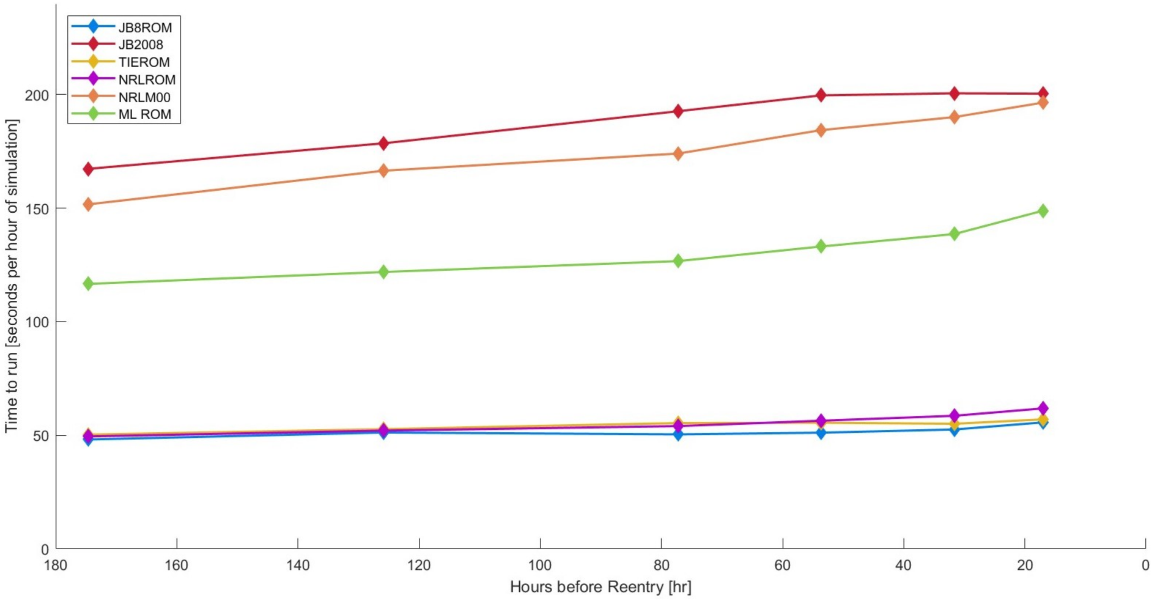


Figure A-13: Run time performance of all density models on Progress-M 27M test case for all simulation durations, averaged across all BC models.

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] National Aeronautics and Space Administration. Frequently Asked Questions: Orbital Debris. https://www.nasa.gov/news/debris_faq.html.

[2] National Aeronautics and Space Administration. Sputnik 1. https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1957-001B.

[3] European Space Agency. About Space Debris. https://www.esa.int/Space_Safety/Space_Debris/About_space_debris.

[4] European Space Agency. How many space debris objects are currently in orbit? https://www.esa.int/Space_Safety/Clean_Space/How_many_space_debris_objects_are_currently_in_orbit.

[5] Luciano Anselmo and Carmen Pardini. Satellite reentry predictions for the Italian civil protection authorities. *Acta Astronautica*, 87:163–181, 2013. https://www.academia.edu/18028749/Satellite_reentry_predictions_for_the_Italian_civil_protection_authorities.

[6] Daniel Arenillas, Raúl González, Noelia Ortiz, Sandra Negrín, Sean Bruinsma, David Jackson, and Claudia Stolle. Implementation and assessment of a new blended whole atmosphere model in reentry services for space surveillance tracking operations as part of SWAMI H2020 project. In *5th International Space Debris Re-entry Workshop*, volume 1. ESA, 2020. https://conference.sdo.esoc.esa.int/proceedings/isdrw05/paper/11.

[7] Jason H.M. Beedle, Christopher E. Rura, David G. Simpson, Hale I. Cohen, Valmir P. Moraes Filho, and Vadim M. Uritsky. A User's Guide to the Magnetically Connected Space Weather System: A Brief Review. *Frontiers in Astronomy and Space Sciences*, 8, 2022. https://doi.org/10.3389/fspas.2021.786308.

[8] Christophe Bonnal, Jean-Marc Ruault, and Marie-Christine Desjean. Active debris removal: Recent progress and current trends. *Acta Astronautica*, 85:51–60, 2013. https://doi.org/10.1016/j.actaastro.2012.11.009.

[9] Bruce Bowman, W. Kent Tobiska, Frank Marcos, Cheryl Huang, Chin Lin, and William Burke. A New Empirical Thermospheric Density Model JB2008 Using New Solar and Geomagnetic Indices. In *AIAA/AAS Astrodynamics Specialist*

Conference and Exhibit, 2008. https://arc.aiaa.org/doi/abs/10.2514/6.2008-6438.

[10] Julia Briden, Nicolette Clark, Peng Mun Siew, Richard Linares, and Tzu-Wei Fang. Impact of Space Weather on Space Assets and Satellite Launches. In *Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference*, 2022. https://amostech.com/TechnicalPapers/2022/Atmospherics_Space-Weather/Briden.pdf.

[11] Sean Bruinsma, Daniel Arnold, Adrian Jäggi, and Noelia Sánchez-Ortiz. Semi-empirical thermosphere model evaluation at low altitude with GOCE densities. *Journal of Space Weather and Space Climate*, 7, 2017. https://doi.org/10.1051/swsc/2017003.

[12] Michael Byers, Ewan Wright, Aaron Boley, and Cameron Byers. Unnecessary risks created by uncontrolled rocket reentries. *Nature Astronomy*, 6:1093–1097, 2022. https://doi.org/10.1038/s41550-022-01718-8.

[13] Eun-Jung Choi, Sungki Cho, Deok-Jin Lee, Siwoo Kim, and Jung Hyun Jo. A Study on Re-entry Predictions of Uncontrolled Space Objects for Space Situational Awareness. *Journal of Astronomy and Space Sciences*, 34:289–302, 2017. https://doi.org/10.5140/JASS.2017.34.4.289.

[14] Stefano Cicaló, James Beck, Edmondo Minisci, Francesca Guerra, Ian Holbrough, Stijn Lemmens, Annalisa Riccardi, and Massimiliano Vasile. GOCE radar-based Orbit Determination for Re-Entry Predictions and comparison with GPS-based POD. In *7th European Conference on Space Debris*, volume 7. ESA Space Debris Office, 2017. https://conference.sdo.esoc.esa.int/proceedings/sdc7/paper/315.

[15] Stefano Cicaló and Stijn Lemmens. Radar-based Re-Entry Predictions with very limited tracking capabilities: the GOCE case study. In *4th International Space Debris Re-entry Workshop*, volume 1. ESA, 2018. https://conference.sdo.esoc.esa.int/proceedings/isdrw04/paper/19.

[16] Christopher P. Clark. A Feasibility Study of CubeSat Architectures for Space Debris Removal from Low Earth Orbit. Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2021. https://hdl.handle.net/1721.1/138733.

[17] Raymond A. DeCarlo. *Linear systems: A state variable approach with numerical implementation*. Prentice Hall, Inc, Upper Saddle River, NJ, USA, January 1989.

[18] P. Dodin, P. Minvielle, and J.P. Le Cadre. Estimating the ballistic coefficient of a re-entry vehicle. *IET Radar Sonar Navig.*, 1:173–183, 2007. https://doi.org/10.1049/iet-rsn:20060064.

[19] Matteo Emanuelli. Space Debris and Meteorite Forecast for Safer Aviation, July 2014. http://www.spacesafetymagazine.com/space-on-earth/malaysia-flight-370/space-debris-meteorite-forecast-safer-aviation/.

[20] John T. Emmert. Thermospheric mass density: A review. *Advances in Space Research*, 56:773–824, September 2015. https://doi.org/10.1016/j.asr.2015.05.038.

[21] Tzu-Wei Fang, Adam Kubaryk, David Goldstein, Zhuxiao Li, Tim Fuller-Rowell, George Millward, Howard Singer, Robert Steenburgh, Solomon Westerman, and Erik Babcock. Space Weather Environment During the SpaceX Starlink Satellite Loss in February 2022. *Space Weather*, 20, November 2022. https://doi.org/10.1029/2022SW003193.

[22] Tim Flohrer, Holger Krag, and Heiner Klinkrad. Assessment and Categorization of TLE Orbit Errors for the US SSN Catalogue. In *Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference*, 2008. https://amostech.com/TechnicalPapers/2008/Orbital_Debris/Flohrer.pdf.

[23] University Corporation for Atmospheric Research Center for Science Education. The Thermosphere. https://scied.ucar.edu/learning-zone/atmosphere/thermosphere.

[24] Jason L. Forshaw, Guglielmo S. Aglietti, Thierry Salmon, Ingo Retat, Mark Roe, Christopher Burgess, Thomas Chabot, Aurélien Pisseloup, Andy Phipps, Cesar Bernal, François Chaumette, Alexandre Pollini, and Willem H. Steyn. Final payload test results for the RemoveDebris active debris removal mission. *Acta Astronautica*, 138:326–342, 2017. https://doi.org/10.1016/j.actaastro.2017.06.003.

[25] Jeff Foust. NASA planning to spend up to $1 billion on space station deorbit module, March 2023. https://spacenews.com/nasa-planning-to-spend-up-to-1-billion-on-space-station-deorbit-module/.

[26] J. Geul, E. Mooji, and R. Noomen. Analysis of Uncertainties and Modeling in Short-Term Reentry Predictions. *Journal of Guidance, Control, and Dynamics*, 41:1276–1289, 2018. https://doi.org/10.2514/1.G003258.

[27] Jacco Geul, Erwin Mooij, and Ron Noomen. TLE uncertainty estimation using robust weighted differencing. *Advances in Space Research*, 59:2522–2535, 2017. https://doi.org/10.1016/j.asr.2017.02.038.

[28] David Gondelach, 2020. https://github.com/davidgondelach/DensityEstimation/tree/v1.0.

[29] David Gondelach, Aleksander Lidtke, Roberto Armellin, Camilla Colombo, Quirin Funke, and Tim Flohrer. Re-entry prediction of spent rocket bodies in GTO. In *26th AAS/AIAA Space Flight Mechanics Meeting*, February 2016.

https://www.researchgate.net/publication/299347585_Re-entry_prediction_of_spent_rocket_bodies_in_GTO.

[30] David Gondelach, Srinivas Setty, Christoph Bamann, and Paul Cefola. Atmospheric density estimation for improved orbit determination and conjunction assessment. In *8th European Conference on Space Debris*, volume 8. ESA Space Debris Office, 2021. https://conference.sdo.esoc.esa.int/proceedings/sdc8/paper/25.

[31] David J. Gondelach, Roberto Armellin, and Aleksander A. Lidtke. Ballistic Coefficient Estimation for Reentry Prediction of Rocket Bodies in Eccentric Orbits Based on TLE Data. *Mathematical Problems in Engineering*, 2017, December 2017. https://doi.org/10.1155/2017/7309637.

[32] David J. Gondelach and Richard Linares. *Atmospheric Density Uncertainty Quantification for Satellite Conjunction Assessment*. 2020. https://doi.org/10.2514/6.2020-0232.

[33] David J. Gondelach and Richard Linares. Real-Time Thermospheric Density Estimation via Two-Line Element Data Assimilation. *Space Weather*, 18(2):e2019SW002356, 2020. https://doi.org/10.1029/2019SW002356.

[34] David J. Gondelach and Richard Linares. Real-Time Thermospheric Density Estimation via Radar and GPS Tracking Data Assimilation. *Space Weather*, 19(4):e2020SW002620, 2021. https://doi.org/10.1029/2020SW002620.

[35] Francisco J. Gonzalez and Maciej Balajewicz. Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems, 2018. https://arxiv.org/abs/1808.01346.

[36] David Guglielmo, Sanny Omar, Riccardo Bevilacqua, Laurence Fineberg, Justin Treptow, Bradley Poffenberger, and Yusef Johnson. Drag Deorbit Device: A New Standard Reentry Actuator for CubeSats. *Journal of Spacecraft and Rockets*, 56(1):129–145, 2019. https://doi.org/10.2514/1.A34218.

[37] Felix R. Hoots and Ronald L. Roehrich. Spacetrack Report No. 3: Models for Propagation of NORAD Element Sets, December 1980. https://celestrak.org/NORAD/documentation/spacetrk.pdf.

[38] Matthew Horsley. Satellite Re-entry Modeling and Uncertainty Quantification. In *Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference*, 2012. https://www.osti.gov/servlets/purl/1073798.

[39] Okchul Jung, Jaedong Seong, Youyeun Jung, and Hyochoong Bang. Recurrent neural network model to predict re-entry trajectories of uncontrolled space objects. *Advances in Space Research*, 68:2515–2529, 2021. https://doi.org/10.1016/j.asr.2021.04.041.

[40] Erin Kahr, Oliver Montenbruck, and Kyle P. G. O'Keefe. Estimation and Analysis of Two-Line Elements for Small Satellites. *Journal of Spacecraft and Rockets*, 50(2):433–439, 2013. https://doi.org/10.2514/1.A32352.

[41] T.S. Kelso. Validation of SQP4 and IS-GPS-200D against GPS Precision Ephemerides. In *17th AAS/AIAA Space Flight Mechanics Conference*, 2007. https://celestrak.org/publications/AAS/07-127/AAS-07-127.pdf.

[42] Donald J. Kessler and Burton G. Cour-Palais. Collision Frequency of Artificial Satellites: The Creation of a Debris Belt. *Journal of Geophysical Research*, 83:2637–2646, 1978. https://agupubs-onlinelibrary-wiley-com.libproxy.mit.edu/doi/pdf/10.1029/JA083iA06p02637.

[43] Siwoo Kim, Byeong-Un Jo, Eun-Jung Choi, Sungki Cho, and Jaemyung Ahn. Two-phase framework for footprint prediction of space object reentry. *Advances in Space Research*, 64:824–835, 2019. https://doi.org/10.1016/j.asr.2019.05.023.

[44] Alexei Klimenko, Sean Brennan, Humberto Godinez, David Higdon, Josef Koller, Earl Lawrence, Richard Linares, David M. Palmer, Michael Shoemaker, David Thompson, Andrew Walker, Brendt Wohlberg, Moriba Jah, Eric Sutton, Thomas Kelecy, Aaron Ridley, and Craig McLaughlin. *IMPACT: Integrated Modeling of Perturbations in Atmospheres for Conjunction Tracking*, pages 139–162. https://arc.aiaa.org/doi/abs/10.2514/5.9781624101991.0139.0162.

[45] Heiner Klinkrad. On the Use of Atmosphere Models in Re-Entry Predictions. In *ESA Symposium Proceedings on 'Environment Modelling for Space-based Applications'*, September 1996.

[46] Heiner Klinkrad. Methods and Procedures for Re-Entry Predictions at ESA. In *6th European Conference on Space Debris*, volume 6. European Space Agency, 2013. https://conference.sdo.esoc.esa.int/proceedings/sdc6/paper/148.

[47] John Koetsier. Starlink Hits 250,000 Customers, Elon Musk Hints: SpaceX Booking Over $300 Million/Year, February 2022. https://www.forbes.com/sites/johnkoetsier/2022/02/14/starlink-hits-250000-customers-elon-musk-hints-spacex-booking-over-300-millionyear.

[48] Aleksander A. Lidtke, David J. Gondelach, and Roberto Armellin. Optimising filtering of two-line element sets to increase re-entry prediction accuracy for GTO objects. *Advances in Space Research*, 63:1289–1317, 2019. https://doi.org/10.1016/j.asr.2018.10.018.

[49] Aleksander A. Lidtke, David J. Gondelach, Roberto Armellin, Camilla Colombo, Hugh G. Lewis, Quirin Funke, and Tim Flohrer. Processing two line element sets to facilitate re-entry prediction of spent rocket bodies from

the geostationary transfer orbit. In *The 6th International Conference on Astrodynamics Tools and Techniques*, 2016. https://eprints.soton.ac.uk/390088/1/Lidtke.pdf.

[50] Tobias Lips and Bent Fritsche. A comparison of commonly used re-entry analysis tools. *Acta Astronautica*, 57:312–323, 2005. https://doi.org/10.1016/j.actaastro.2005.03.010.

[51] Huixin Liu, Jeff Thayer, Yongliang Zhang, and Woo Kyoung Lee. The non–storm time corrugated upper thermosphere: What is beyond MSIS? *Space Weather*, 15(6):746–760. https://doi.org/10.1002/2017SW001618.

[52] Meysam Mahooti. High Precision Orbit Propagator v2.2.1, 2022. https://www.mathworks.com/matlabcentral/fileexchange/55167-high-precision-orbit-propagator.

[53] Meysam Mahooti. Jacchia-Bowman Atmospheric Density Model, 2022. https://www.mathworks.com/matlabcentral/fileexchange/56163-jacchia-bowman-atmospheric-density-model.

[54] Tariq Malik. SpaceX says a geomagnetic storm just doomed 40 Starlink internet satellites, February 2022. https://www.space.com/spacex-starlink-satellites-lost-geomagnetic-storm.

[55] Frank Marcos, Bruce Bowman, and Robert Sheehan. *Accuracy of Earth's Thermospheric Neutral Density Models*. 2006. https://sol.spacenvironment.net/jb2008/pubs/JB2006_AIAA_2006-6167.pdf.

[56] Frank Marcos, William Burke, and Shu Lai. *Thermospheric Space Weather Modeling*. 2007. https://arc.aiaa.org/doi/abs/10.2514/6.2007-4527.

[57] Frank Marcos, Shu Lai, Cheryl Huang, Chin Lin, John Retterer, Susan Delay, and Eric Sutton. *Towards Next Level Satellite Drag Modeling*. 2010. https://arc.aiaa.org/doi/abs/10.2514/6.2010-7840.

[58] F. Landis Markley and John L. Crassidis, editors. *Fundamentals of Spacecraft Attitude Determination and Control*. Number 33 in Space Technology Library. Springer Science + Business Media, New York, 2014.

[59] Piyush M. Mehta and Richard Linares. A methodology for reduced order modeling and calibration of the upper atmosphere. *Space Weather*, 15(10):1270–1287. https://doi.org/10.1002/2017SW001642.

[60] Piyush M. Mehta and Richard Linares. A New Transformative Framework for Data Assimilation and Calibration of Physical Ionosphere-Thermosphere Models. *Space Weather*, 16(8):1086–1100, 2018. https://doi.org/10.1029/2018SW001875.

[61] Piyush M. Mehta and Richard Linares. Real-Time Thermospheric Density Estimation from Satellite Position Measurements, 2020. https://dspace.mit.edu/handle/1721.1/135426.2.

[62] Piyush M. Mehta, Richard Linares, and Eric K. Sutton. A Quasi-Physical Dynamic Reduced Order Model for Thermospheric Mass Density via Hermitian Space-Dynamic Mode Decomposition. *Space Weather*, 16(5):569–588, 2018. https://doi.org/10.1029/2018SW001840.

[63] Piyush M. Mehta, Richard Linares, and Eric K. Sutton. Data-Driven Inference of Thermosphere Composition During Solar Minimum Conditions. *Space Weather*, 17(9):1364–1379, 2019. https://doi.org/10.1029/2019SW002264.

[64] Piyush M. Mehta, Richard Linares, and Andrew C. Walker. Photometric Data from Nonresolved Objects for Improved Drag and Reentry Prediction. *Journal of Spacecraft and Rockets*, 55(4):959–970, 2018. https://doi.org/10.2514/1.A33825.

[65] Alessandro Morselli, Roberto Armellin, Pierluigi Di Lizia, and Franco Bernelli Zazzera. A high order method for orbital conjunctions analysis: Sensitivity to initial uncertainties. *Advances in Space Research*, 53:490–508, 2014. https://doi.org/10.1016/j.asr.2013.11.038.

[66] Peng Mun Siew. DESQU - Dynamic Data-Driven Thermospheric Density Estimation with Quantified Uncertainties, 2021. https://github.com/SWQU/DESQU.

[67] Andrey I. Nazarenko. How can we increase the accuracy of determination of spacecrafts lifetime? *Acta Astronautica*, 116:229–236, 2015. https://doi.org/10.1016/j.actaastro.2015.07.004.

[68] Andrey I. Nazarenko and I.V. Usovik. Instability of the solution of the problem on determining the reentry time of satellites on elliptic orbits. *Acta Astronautica*, 163:142–146, 2019. https://doi.org/10.1016/j.actaastro.2019.01.026.

[69] National Oceanic and Atmospheric Administration Space Weather Prediction Center. WAM-IPE. https://www.swpc.noaa.gov/products/wam-ipe.

[70] Victor P. Osweiler. Covariance Estimation and Autocorrelation of NORAD Two-Line Element Sets. Master's thesis, Air Force Institute of Technology, Department of Aeronautics and Astronautics, March 2006. https://scholar.afit.edu/etd/3531/.

[71] Carmen Pardini and Luciano Anselmo. Comparison and Accuracy Assessment of Semi-Empirical Atmosphere Models through the Orbital Decay of Spherical Satellites. *The Journal of the Astronautical Sciences*, 49:255–268, 2001. https://www.academia.edu/18028749/Satellite_reentry_predictions_for_the_Italian_civil_protection_authorities.

[72] Carmen Pardini and Luciano Anselmo. Performance Evaluation of Atmospheric Density Models for Satellite Reentry Predictions with High Solar Activity Levels. *Transactions of the Japan Society for Aeronautical and Space Sciences*, 46:42–46, 2003. https://doi.org/10.2322/tjsass.46.42.

[73] Carmen Pardini and Luciano Anselmo. On the accuracy of satellite reentry predictions. *Advances in Space Research*, 34:1038–1043, 2004. https://doi.org/10.1016/j.asr.2003.01.010.

[74] Carmen Pardini and Luciano Anselmo. Performances of atmospheric density models during satellite reentry prediction campaigns at sunspot minimum. 2008. https://www.researchgate.net/publication/255635678.

[75] Carmen Pardini and Luciano Anselmo. Reentry predictions of three massive uncontrolled spacecraft. 2012. https://issfd.org/ISSFD_2012/ISSFD23_CRSD1_4.pdf.

[76] Carmen Pardini and Luciano Anselmo. Re-entry predictions for uncontrolled satellites: Results and challenges. In *6th IAASS Conference – Safety is Not an Option*, May 2013. https://www.researchgate.net/publication/309202373_RE-ENTRY_PREDICTIONS_FOR_UNCONTROLLED_SATELLITES_RESULTS_AND_CHALLENGES.

[77] Carmen Pardini and Luciano Anselmo. The uncontrolled reentry of Progress-M 27M. In *Proceedings of the 8th IAASS Conference*, May 2016. https://www.researchgate.net/publication/309202499_THE_UNCONTROLLED_REENTRY_OF_PROGRESS-M_27M.

[78] Carmen Pardini, K. Moe, and Luciano Anselmo. Thermospheric density model biases at the 23rd sunspot maximum. *Planetary and Space Science*, 67:130–146, 2012. https://doi.org/10.1016/j.pss.2012.03.004.

[79] Carmen Pardini, W. Kent Tobiska, and Luciano Anselmo. Analysis of the orbital decay of spherical satellites using different solar flux proxies and atmospheric density models. *Advances in Space Research*, 37:392–400, 2006. https://doi.org/10.1016/j.asr.2004.10.009.

[80] Nikolaos K. Pavlis, Simon A. Holmes, Steve C. Kenyon, and John K. Factor. The development and evaluation of the Earth Gravitational Model 2008 (EGM2008). *Journal of Geophysical Research: Solid Earth*, 117(B4), 2012. https://doi.org/10.1029/2011JB008916.

[81] William Dean Pesnell. Predictions of Solar Cycle 24: How are we doing? *Space Weather*, 14(1):10–21, 2016. https://doi.org/10.1002/2015SW001304.

[82] J. M. Picone, A. E. Hedin, D. P. Drob, and A. C. Aikin. NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues. *Journal of Geophysical Research: Space Physics*, 107(A12):SIA 15–1–SIA 15–16, 2002. https://doi.org/10.1029/2002JA009430.

[83] Marcin D. Pilinski, Geoff Crowley, Eric Sutton, and Mihail Codrescu. Improved Orbit Determination and Forecasts with an Assimilative Tool for Satellite Drag Specification. In *Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference*, 2016. https://amostech.com/TechnicalPapers/2016/Poster/Pilinski.pdf.

[84] Tereza Pultarova and Elizabeth Howell. The number of satellites orbiting Earth could quintuple in the next decade, June 2019. https://www.technologyreview.com/2019/06/26/755/satellite-constellations-orbiting-earth-quintuple/.

[85] Liying Qian, Alan G. Burns, Barbara A. Emery, Benjamin Foster, Gang Lu, Astrid Maute, Arthur D. Richmond, Raymond G. Roble, Stanley C. Solomon, and Wenbin Wang. *The NCAR TIE-GCM*, chapter 7, pages 73–83. American Geophysical Union (AGU), 2014. https://doi.org/10.1002/9781118704417.ch7.

[86] Liying Qian and Stanley C. Solomon. Thermospheric Density: An Overview of Temporal and Spatial Variations. *Space Science Reviews*, 168:147–173, 2012. https://doi-org.libproxy.mit.edu/10.1007/s11214-011-9810-z.

[87] S.F. Rafano Carná and R. Bevilacqua. High fidelity model for the atmospheric re-entry of CubeSats equipped with the Drag De-Orbit Device. *Acta Astronautica*, 156:134–156, 2019. https://doi.org/10.1016/j.actaastro.2018.05.049.

[88] Vishal Ray, Tom E. Berger, Zach C. Waldron, Eric K. Sutton, Greg Lucas, Delores J. Knipp, Jeffrey P. Thayer, Siamak G. Hesar, and Daniel J. Scheeres. The impact of space weather on very low Earth orbit (VLEO) satellites. In *Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference*, 2022. https://amostech.com/TechnicalPapers/2022/Atmospherics_Space-Weather/Ray.pdf.

[89] Albert Reuther, Jeremy Kepner, Chansup Byun, Siddharth Samsi, William Arcand, David Bestor, Bill Bergeron, Vijay Gadepally, Michael Houle, Matthew Hubbell, Michael Jones, Anna Klein, Lauren Milechin, Julia Mullen, Andrew Prout, Antonio Rosa, Charles Yee, and Peter Michaleas. Interactive supercomputing on 40,000 cores for machine learning and data analysis. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pages 1–6. IEEE, 2018. https://ieeexplore.ieee.org/document/8547629.

[90] A. D. Richmond, E. C. Ridley, and R. G. Roble. A thermosphere/ionosphere general circulation model with coupled electrodynamics. *Geophysical Research Letters*, 19(6):601–604, 1992. https://doi.org/10.1029/92GL00401.

[91] A.J. Ridley and G. Tóth Y. Deng. The global ionosphere–thermosphere model. *Journal of Atmospheric and Solar-Terrestrial Physics*, 68(8):839–864, 2006. https://doi.org/10.1016/j.jastp.2006.01.008.

[92] Kathleen Riesling. Orbit Determination from Two Line Element Sets of ISS-Deployed CubeSats. In *29th Annual AIAA/USU Conference on Small Satellites*, 2015. https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=3222&context=smallsat&httpsredir=1&referer=.

[93] Tate Ryan-Mosley, Erin Winick, and Konstantin Kakaes. Starlink satellites: Everything you need to know about the controversial internet megaconstellation, November 2022. https://www.space.com/spacex-starlink-satellites.html.

[94] Arrun Saunders, Hugh Lewis, and Graham Swinerd. A new tool for satellite re-entry predictions. *European Space Agency, (Special Publication) ESA SP*, 672, January 2009. https://www.researchgate.net/publication/228975096_A_new_tool_for_satellite_re-entry_predictions.

[95] Arrun Saunders, Graham G. Swinerd, and Hugh G. Lewis. Deriving Accurate Satellite Ballistic Coefficients from Two-Line Element Data. *Journal of Spacecraft and Rockets*, 49(1):175–184, 2012. https://doi.org/10.2514/1.A32023.

[96] National Weather Service. Hello Solar Cycle 25, September 2020. https://www.weather.gov/news/201509-solar-cycle.

[97] Chuang Shi, Wenwen Li, Min Li, Qile Zhao, and Jizhang Sang. Calibrating the scale of the NRLMSISE00 model during solar maximum using the two line elements dataset. *Advances in Space Research*, 56(1):1–9, 2015. https://doi.org/10.1016/j.asr.2015.03.024.

[98] Mark F. Storz, Bruce R. Bowman, Major James I. Branson, Stephen J. Casali, and W. Kent Tobiska. High accuracy satellite drag model (HASDM). *Advances in Space Research*, 36(12):2497–2505, 2005. https://doi.org/10.1016/j.asr.2004.02.020.

[99] Sarah Thiele and Aaron C. Boley. Investigating the Risks of Debris-Generating ASAT Tests in the Presence of Megaconstellations. *The Journal of the Astronautical Sciences*, 69:1797–1820, 2022. https://doi.org/10.1007/s40295-022-00356-6.

[100] W Kent Tobiska, Bruce Bowman, and Dave Bouwer. *Solar and Geomagnetic Indices for Thermospheric Density Models*, chapter 4, page 27. January 2012. https://www.researchgate.net/publication/295103804_SOLAR_AND_GEOMAGNETIC_INDICES_FOR_THERMOSPHERIC_DENSITY_MODELS.

[101] Mirko Trisolini and Camilla Colombo. Integrating density-based uncertainty propagation with object-oriented re-entry models. In *8th European Conference on Space Debris*, 2021. https://www.researchgate.net/publication/351108965_Integrating_density-based_uncertainty_propagation_with_object-oriented_re-entry_models.

[102] Mirko Trisolini and Camilla Colombo. Propagation and Reconstruction of Reentry Uncertainties Using Continuity Equation and Simplicial Interpolation. *Journal of Guidance, Control, and Dynamics*, 44(4):793–811, 2021. https://doi.org/10.2514/1.G005228.

[103] Herbert Turner, Maggie Zhang, David Gondelach, and Richard Linares. Machine Learning Algorithms for Improved Thermospheric Density Modeling. In Frederica Darema, Erik Blasch, Sai Ravela, and Alex Aved, editors, *Dynamic Data Driven Applications Systems*, volume 12312, pages 143–151, Cham, 2020. Springer International Publishing. https://doi.org/10.1007/978-3-030-61725-7_18.

[104] David Vallado, editor. *Fundamentals of Astrodynamics and Applications*. Space Technology Library. Microcosm Press, Hawthorne, CA, fourth edition, 2013.

[105] David Vallado, Paul Crawford, Ricahrd Hujsak, and T.S. Kelso. *Revisiting Spacetrack Report #3*. 2006. https://arc.aiaa.org/doi/abs/10.2514/6.2006-6753.

[106] David A. Vallado and David Finkleman. A critical assessment of satellite drag and atmospheric density modeling. *Acta Astronautica*, 95:141–165, 2014. https://doi.org/10.1016/j.actaastro.2013.10.005.

[107] Benjamin B. Virgili and Stijn Lemmens. The impact of the new NRLMSIS 2.0 on re-entry predictions. In *8th European Conference on Space Debris*, volume 8. ESA Space Debris Office, 2021. https://conference.sdo.esoc.esa.int/proceedings/sdc8/paper/44.

[108] Ting Wang. Analysis of Debris from the Collision of the Cosmos 2251 and the Iridium 33 Satellites. *Science Global Security*, 18(2):87–118, 2010. https://doi.org/10.1080/08929882.2010.493078.

[109] Zhang Wei, Cui Wen, Wang Xiuhong, Wei Dong, and Liu Xing. Re-entry prediction of objects with low-eccentricity orbits based on mean ballistic coefficients. *Open Astronomy*, 29(1):210–219, 2020. https://doi.org/10.1515/astro-2020-0006.

[110] Vasiliy S. Yurasov, Andrey I. Nazarenko, Kyle T. Alfriend, and Paul J. Cefola. Reentry Time Prediction Using Atmospheric Density Corrections. *Journal of Guidance, Control, and Dynamics*, 31(2):282–289, 2008. https://doi.org/10.2514/1.26593.