SET TR 2022-001

Global Thermospheric Density Estimation using CubeSat GPS Data and a Physics-based Space Environment Model

Shaylah Mutschler

June 2022

Global Thermospheric Density Estimation using CubeSat GPS Data and a Physics-based Space Environment Model

by

Shaylah Maria Mutschler

B.S., Wright State University, 2012

M.S., University of Colorado at Boulder, 2017

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Aerospace Engineering Sciences

2022

Committee Members:

Penina Axelrad, Chair

Marcus Holzinger

Brandon Jones

Tomoko Matsuo

Eric Sutton

Mutschler, Shaylah Maria (Ph.D., Aerospace Engineering Sciences)

Global Thermospheric Density Estimation using CubeSat GPS Data and a Physics-based Space Environment Model

Thesis directed by Prof. Penina Axelrad

This dissertation introduces a novel method for using CubeSats as indirect sensors of the space environment, with a focus on thermospheric density for more accurate prediction of Low Earth Orbit (LEO) object motion. In LEO, atmospheric drag is the largest contributor to trajectory prediction error. The current thermospheric density model used in operations applies corrections to an empirical density model using observations of 75+ calibration satellites.

Recently, there has been a rise in popularity of LEO CubeSats, with several companies establishing commercial CubeSat constellations for remote sensing and communication applications. My research takes advantage of the abundance and quality of CubeSat GPS position information to infer the space environment affecting their motion.

The tool presented here is named SoleiTool to emphasize its focus on the sun-driven nature of the near-Earth space environment. SoleiTool estimates forcing parameters of the physics-based space environment model, Thermosphere-Ionosphere-Electrodynamics General Circulation Model (TIE-GCM), using CubeSat GPS Precise Orbit Determination (POD) data. Global estimated density is obtained by using the forcing parameters estimated in SoleiTool as input to TIE-GCM.

This work examines the application of SoleiTool to both a simulation scenario modeled after the full Spire CubeSat constellation (119 satellites) and a 10-satellite Spire CubeSat flight data case study, in which SoleiTool estimates the dynamic calm and storm-time global density conditions. SoleiTool performance is evaluated via comparison to the true density in the simulation scenario; and via comparison to the HASDM operational density model and Swarm mission derived density in the Spire CubeSat case study. Simulation scenario results show 5% to 25% error in the global estimated density field, with the largest error occurring during storm-time conditions. The Spire

CubeSat flight data case study has a larger density error when compared to HASDM and Swarm mission derived density along a satellite trajectory. The larger error in the flight data case study is likely due to the smaller number of satellites and occasional anomalous tool performance for high-altitude polar orbiting satellites. Suggestions for future work are provided, including modifications for potential tool performance improvement. Ultimately, the operational vision for SoleiTool is to produce global density now-casts and forecasts using the most recent POD data from tens to hundreds of LEO CubeSats to support SDA and provide real-time satellite mission support.

Dedication

To my favorite planet, Earth. May I contribute in some small way to preserving our ability to gaze at your beauty from space.

“It’s never too late to be what you might have been.” - George Eliot

“Everything that you need is already there. The idea you seek’s seed has already been planted; re-discover it with new eyes.”

Acknowledgements

I would like to thank the committee members, Marcus Holzinger, Tomoko Matsuo, Brandon Jones, and especially Eric Sutton for his guidance on Spire data and TIE-GCM.

A special thank you to my advisor, Penny Axelrad. I feel so much gratitude in my heart for the consistent support you have offered, especially during some difficult times. Thank you for making my graduate school experience such a positive one. The words I write here are no where close to fully expressing how much your mentorship has meant to me.

I appreciate my mentors, Moriba Jah, Tom Kelecy, and Ryan Weisman. Thanks to my interns, Diana Mata, Connor Ott, Stephen Albert, and especially my current intern, Brandon DiLorenzo.

Thanks, George Born, for your mentorship and also for talking me into changing my CU Boulder application to PhD instead of just Masters. I am extremely glad that I did.

Partial funding for this research was provided by the National Defense Science and Engineering Graduate (NDSEG) Fellowship.

Contents

Chapter

Appendix

Tables

Table

Figures

Figure

Chapter 1

Introduction

1.1 Motivation

In the past several decades there has been an exponential rise in the number of human-made objects in Earth orbit. These objects include active satellites, rocket bodies, and pieces of satellites from in-orbit collisions. Uncontrolled orbital debris objects make up 95% of the regularly tracked objects for space object catalog maintenance (objects $\geq$ 10cm in size) [30]. Since active debris removal is not yet feasible, it is imperative that we prevent avoidable debris-producing events such as collisions involving active satellites.

The US spends millions of dollars annually on Space Domain Awareness (SDA) in order to prevent loss, disruption, or degradation of space services and capabilities [51]. The Combined Space Operations Center (CSpOC) is responsible for the detection, identification, and daily tracking of all man-made objects in space [62]. CSpOC provides a conjunction analysis (CA) service that alerts satellite operators if their satellite exceeds a specific collision probability threshold. In fact, NASA executes space shuttle and space station debris avoidance maneuvers when the probability of collision is greater than 1 in 100,000 [19].

The LEO regime is becoming more crowded as the number of satellites continues to grow with the rising popularity and establishment of SmallSat constellations. For example, SpaceX is in the process of creating a 12,000-satellite Starlink constellation in LEO, with 1,600 SmallSats in orbit currently. Similarly, Blue Origin is planning to construct and operate a 3,200 LEO SmallSat mega-constellation to provide global low-cost internet service. To add to the complexity of the situation, some countries continue to execute ill-advised anti-satellite (i.e., ASAT) tests in LEO where they destroy their own satellite, creating hundreds of new debris objects. Recently, in November 2021, Russia performed an ASAT test, blowing up one of their defunct LEO satellites and creating hundreds of new orbital debris objects. In fact, more than 700 new debris objects originating from this event have been added to the space object catalog, with CSpOC continuing to detect, track, characterize and catalog objects from this event as of December 2021.

In order to prevent future collisions in the crowded near-Earth space environment, CSpOC must perform accurate trajectory prediction, which requires knowledge of the non-conservative forces affecting space objects. Figure 1.1 provides a visual comparison between the magnitudes of leading LEO forces as a function of orbit altitude normalized to daytime drag. Solar Radiation Pressure (SRP) is the force produced by the impact of sunlight photons on the surface of a spacecraft. SRP and drag are the only non-gravitational forces in Figure 1.1. It is evident that daytime drag is the largest non-gravitational perturbing force for orbits 200 to 600 km altitude.

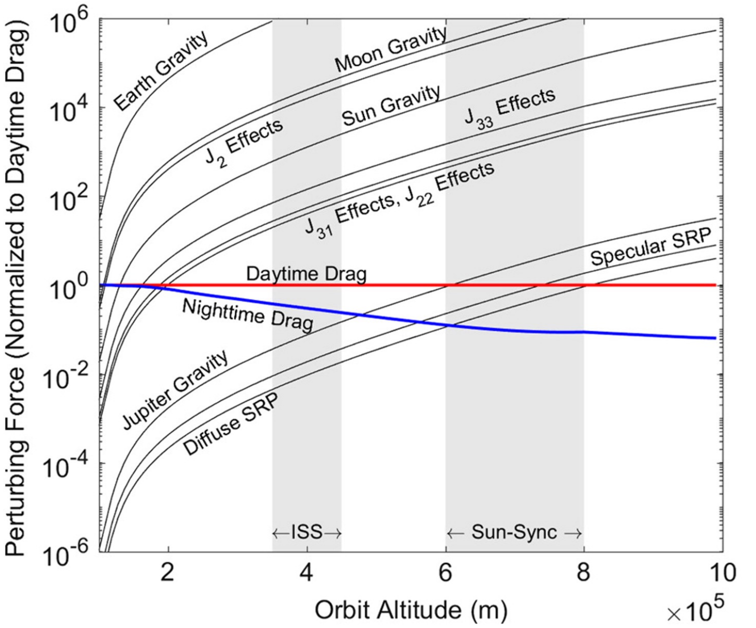

Figure 1.1: Comparison of LEO Force Magnitudes; Credit: [18]

Atmospheric drag is caused by the presence of Earth's atmosphere extending into orbital space. Atmospheric density at these altitudes is called thermospheric density, and references the thermosphere, which spans the LEO regime, ranging from about 90 to 1000 km altitude. Thermospheric density varies temporally and spatially, and is primarily driven by the dynamical behavior of space weather. Existing SDA algorithms for orbit prediction adjust their thermospheric density models based on observations of calibration satellites. Nevertheless, atmospheric drag mismodeling remains the largest contributor to LEO trajectory prediction error.

Current methods that estimate or model thermospheric density can be categorized as either empirical or physics-based approaches. Empirical models use long-term trends from measurements over an array of instruments to provide a simplified and parameterized mathematical formulation of the system. Although computationally fast, a major drawback of empirical space environment models is that they do not model the dynamics of the thermosphere and therefore are limited to historical modeling or nowcasts [21]. Physics-based models, on the other hand, model the underlying dynamics of the system, but require substantially more computing resources than empirical models. The current density model used at CSpOC is the High Accuracy Satellite Drag Model (HASDM) [6, 7, 60], and is an example of an empirical-based density model. My research aims to improve thermospheric density estimation and prediction to support SDA applications by utilizing a physics-based space environment model and leveraging observations of objects not primarily operated for the purpose of space environment monitoring.

1.2 Overview of Research

Recently, there has been a rapid rise in the number of deployed LEO CubeSats, with several commercial entities establishing CubeSat constellations for remote sensing and communication applications. For example, Spire Global operates a fleet of more than 100 LEO CubeSats, and Planet Labs currently maintains a constellation of nearly 200 CubeSats [59]. Other companies plan to establish their own CubeSat constellations; for example, Kepler Communications and Lacuna Space are planning to launch a 140 and 32-CubeSat constellation, respectively [27], [26]. This research takes advantage of the abundance and quality of GPS-based CubeSat position information to infer the space environment affecting their motion. CubeSats are particularly well suited for this application due to their simple physical geometry, which is described in terms of a 10x10x10 cm^3 volume. Typical CubeSats range from a single unit (or 1U) to upwards of 12U. For this work, Spire 3U Lemur CubeSats are utilized, which use onboard GPS receivers to provide accurate time and position estimates for operations. Spire data sets were provided by Spire Global as part of the NASA Commercial SmallSat Data Pilot Program.

I have developed a tool that estimates a global density field incorporating both CubeSat data and a physics-based space-environment model. The tool is named SoleiTool to emphasize its focus on the sun-driven nature of the near-Earth space environment. SoleiTool estimates forcing parameters of the Thermosphere-Ionosphere-Electrodynamics General Circulation Model (TIE-GCM) [57] using CubeSat GPS Precise Orbit Determination (POD) data. This is accomplished via a two-step process that includes two filters: a Batch filter and Particle Filter (PF). The Batch filter uses arcs of CubeSat POD data to sense the density affecting the CubeSat motion. Next, the PF uses the density information from the Batch filter to estimate a time-series of the TIE-GCM input parameter, Kp, while keeping a fixed F10.7. Lastly, the estimated Kp time-series and fixed F10.7 from SoleiTool are input to TIE-GCM to produce a time-history of global density. The operational vision for SoleiTool is to produce global density now-casts and forecasts using the most recent 1+ hours of POD from tens to hundreds of LEO CubeSats to support SDA and provide real-time satellite mission support.

Although the final design of SoleiTool is the primary focus of this dissertation, Appendices A and B describe earlier approaches considered. The method discussed in Appendix A is the first version of SoleiTool, in which simulated orbital debris data were used in a UKF to estimate acceleration due to drag, and then passed to a PF to estimate Kp and F10.7 TIE-GCM input parameters. We have made various modifications to SoleiTool since first considering its use with debris object tracking data. Specifically, the first filter, originally a UKF, has been replaced with a Batch filter. Appendix B describes my initial approach to thermospheric density state estimation in which a Partially Orthogonal Ensemble Kalman Filter (POEnKF) assimilates observations of 16 simulated debris objects to estimate a 3000-element spatial grid of thermospheric density in addition to the position and velocity of each object. A variation of the EnKF, the POEnKF, was applied to this research problem because the EnKF is a commonly used filter for high-dimensional nonlinear geophysical applications, such as forecasting of atmosphere and ocean systems [28].

1.3 Contributions of Research

The primary contributions of this dissertation are as follows:

[1] A novel method for utilizing CubeSat GPS information to estimate global thermospheric density is introduced.

[2] Thermospheric density information is estimated from CubeSat POD ephemeris arcs as short as 20 minutes.

[3] Method is demonstrated via a simulation scenario of the full Spire CubeSat constellation (119 satellites) and a 10-satellite case study using Spire CubeSat flight data during two historical geomagnetic storms.

Overall, the objective of SoleiTool is to improve thermospheric density modeling by assimilating CubeSat GPS information to infer drivers to a physics-based space environment model for more accurate LEO object motion prediction. In doing so, we aim to decrease the number of unnecessary satellite maneuvers executed to avoid satellites and debris objects. This not only extends satellite lifetime, but also provides for a safer space environment by preserving fuel to mitigate actual collision risks.

1.4 Thesis Overview

This dissertation is organized as follows. Chapter 2 provides a description of thermospheric density and the space weather mechanisms that drive its dynamic nature. The chapter also includes an overview of the NRLMSISE-00 (i.e., MSIS) density model, which is referenced throughout this dissertation, as well as a description of the background density model used in SoleiTool, TIE-GCM, and our unique approach to running it in such a way that reduces filter computation time. Next, atmospheric drag is described and an example of the effect of drag on a LEO satellite trajectory is presented. Chapter 2 concludes with a review of prior work, including a description of the HASDM and Swarm projects, which are used to evaluate SoleiTool performance.

Chapter 3 provides a detailed description and theoretical comparison of three nonlinear filters: the Unscented Kalman Filter (UKF), the Ensemble Kalman Filter (EnKF), and the PF. This work informed the decision of the filter types applied in the prior methods described in Appendices A and B and the final design of SoleiTool.

The design of SoleiTool is described in Chapter 4, and the remainder of this dissertation explores the application of SoleiTool to both a simulation scenario and a case study using Spire CubeSat flight data. Chapter 5 provides a description of the 119-satellite simulation scenario in which the density field during a simulated geomagnetic storm is estimated. Chapter 6 applies SoleiTool to a 10-satellite case study in which the global density field is sensed via the POD flight data from ten Spire CubeSats during two actual geomagnetic storms. The Spire CubeSat data used for assimilation are described in Section 6.1 and details about the May 2019 11-day case study are provided in Section 6.2.

The results for both the simulation scenario and case study are broken down into the following sections: Batch filter results, PF results, estimated density results, and a propagation analysis. The SoleiTool estimated density is compared to the true density for the simulation scenario results, while the SoleiTool density is compared to TIE-GCM using historically measured Kp/F10.7, HASDM, MSIS, and Swarm-derived density for the case study. Comparisons between satellite POD and the propagated trajectories of CubeSats using SoleiTool density and other density models (e.g., HASDM) are presented to demonstrate the improvement in LEO object predictions.

Chapter 7 presents a summary of findings and conclusions, as well as suggestions for future work.

Chapter 2

Thermospheric Density

Drag is the largest contributor to LEO prediction error and is driven by thermospheric density. This chapter provides the background information necessary to understand thermospheric density and what drives its dynamical nature (Sections 2.1 and 2.2). Section 2.3.2 describes the physics-based space environment model used in this work to model thermospheric density. The impact that thermospheric density has on orbits is described in Section 2.4. Lastly, Section 2.5 provides a review of prior efforts that aim to estimate and model thermospheric density using various kinds of satellite and space environment data.

2.1 Thermospheric Density

Atmospheric density describes the number of particles per unit area in the atmosphere. This work focuses on thermospheric density, which denotes the density of particles in near-Earth space. Thermospheric density is highly dynamic and depends on a number of factors, including solar cycle, solar activity, geomagnetic activity, diurnal cycle, altitude, and latitude. All of these factors are directly linked to the Sun. Solar cycle and solar activity describe the Sun's conditions, and geomagnetic activity is a measure of disturbance in the Earth's magnetic field caused by an event originating at the Sun. Solar and geomagnetic activity are discussed in detail in Section 2.2. The geometry of the Sun with respect to a location of the terrestrial density field is described by the diurnal cycle, altitude, and latitude. Thermospheric density decreases exponentially as a function of altitude, and therefore affects the trajectories of LEO objects more than higher orbital regimes.

Periods of time when there is little to no geomagnetic field disturbance and relatively low solar flux are referred to as calm conditions; whereas storm-time refers to periods of time when there is a disturbance in the near-Earth space environment caused by a geomagnetic storm or solar activity. Figure 2.1 shows an example of the density field during calm and storm-time conditions.

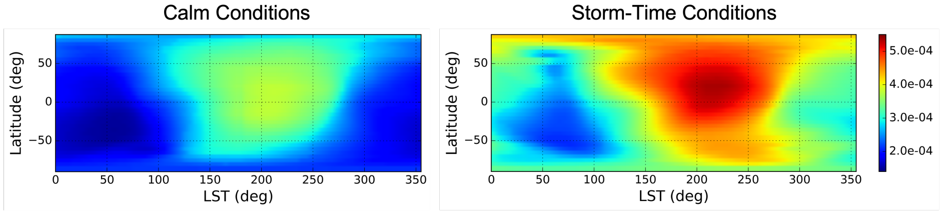

Figure 2.1: Calm and Storm-Time Density (kg/km^3)

Thermospheric density is best parameterized in terms of Local Sidereal Time (LST), latitude, and altitude because the nominal density field remains constant as a function of LST, as opposed to longitude. Density is typically highest near local noon, and lowest near local midnight and in the polar region experiencing winter. The longitude of local noon changes throughout the day, but the LST of local noon is a fixed parameter, making it favorable for density analysis.

2.2 Sun-Driven Near Earth Space Environment

Disturbances in the near-Earth space environment are driven by the Sun and its interaction with the geomagnetic field. The primary energy input in to the thermosphere is solar irradiance in the form of soft X-ray Ultra-Violet (XUV, 1-30 nm), Extreme Ultra-Violet (EUV, 30-120 nm), and Far Ultra-Violet (FUV, 120-200 nm) [36]. The frequency and magnitude of density field enhancements stemming from solar irradiance are dependent on the 11-year solar cycle and 27-day solar rotation because active regions of the sun (i.e., sunspots) produce the largest quantity of solar flux.

The second most important energy input to the thermosphere is geomagnetic activity caused by the interaction between the solar wind and the Earth's magnetic field. More specifically, Coronal

Mass Ejections (CME) and high speed solar wind cause disturbances in the Earth's magnetic field that lead to thermospheric density enhancements. Similar to solar irradiance, the frequency and magnitude of such disturbances are dependent on the 11-year solar cycle, with more disturbance-inducing events occurring during solar maximum, compared to solar minimum. CMEs are magnetized plasma bubbles that erupt from the sun and travel outward through the interplanetary medium (Figure 2.2). Depending on the origin and direction of travel, some CMEs collide with Earth's magnetic field and cause geomagnetic storms. Solar wind is a constant outflow of the Sun's coronal plasma into the interplanetary medium. Reactions between slow and fast solar wind cause High-Speed Streams (HSS) and Co-rotating Interaction Regions (CIR) (Figure 2.3). When solar wind HSS and/or CIRs interact with Earth, the field irregularities in the plasma may contribute to moderate geomagnetic storms. Both CMEs and high-speed solar wind can cause geomagnetic storms, which lead to thermospheric density enhancements. Knipp [35] provides a comprehensive text on these topics.

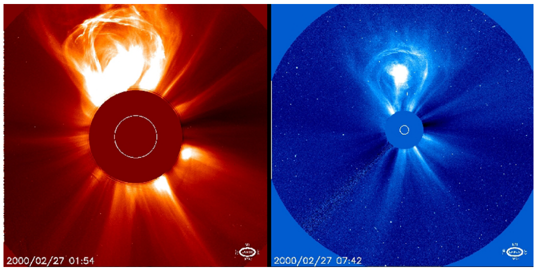

Figure 2.2: Example of a Coronal Mass Ejection captured by SOHO LASCO C2 and C3 with a 3 and 6 solar-radii field of view, respectively; Credit: SOHO ESA and NASA [73]

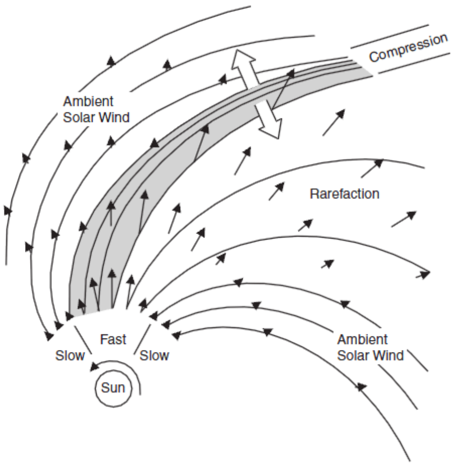

Figure 2.3: Solar wind High-Speed Streams and Co-rotating Interaction Regions: Interface between fast and slow solar wind; Credit: [72]

Geomagnetic field disturbances and solar flux are consistently measured and these measurements are typically the primary drivers of thermospheric density models. Kp describes geomagnetic activity, while F10.7 is a direct measurement of the solar radio flux [13]. Kp is a weighted mean of individual K-indices measured from several stations, scattered across mid-latitudes worldwide. The K-index is a measure of geomagnetic field disturbance via a magnetometer over a three-hour interval specific to a particular geographic location and ranges 0 to 9 [15]. More specifically, the K-index indicates the geomagnetic field fluctuations relative to a quiet day of the horizontal components. Figure 2.4 shows the measured K-indices and overall Kp-index during two geomagnetic storms in May, 2019.

F10.7 is the solar flux at a wavelength of 10.7 cm (radio spectrum), and is a useful proxy for solar EUV emissions (Figure 2.5). It ranges 50 to 300 s.f.u. and can be measured from anywhere on Earth regardless of terrestrial weather conditions because it is in the radio spectrum. EUV can only be measured directly by space-based assets such as the Solar Extreme-ultraviolet Monitor (SEM) instrument on the NASA/ESA Solar and Heliospheric Observatory (SOHO) research satellite at Earth-Sun Lagrange Point 1 (LP1) [67].

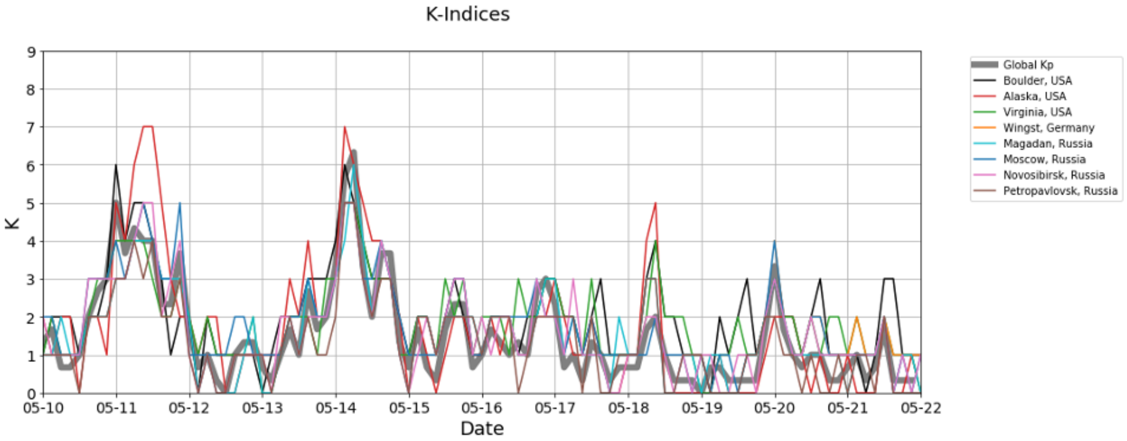

Figure 2.4: K measurements from Individual K-stations and Global Kp

Note: Showing only a selection of K measurements/stations

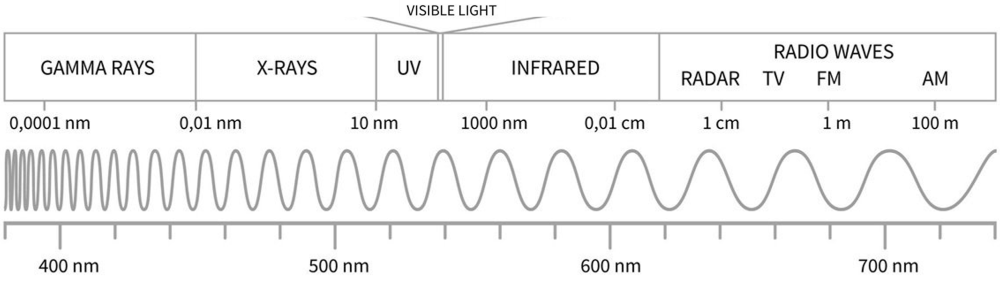

Figure 2.5: Wavelength Spectrum; F10.7 index measures 10.7 cm radio wavelength (proxy for EUV); EUV can be measured by GOES-R satellite

While Kp and F10.7 do not fully describe the near-Earth space environment, they are used as the primary inputs to most near-Earth space environment models. As currently measured, Kp and F10.7 have inherent time-resolutions of 3 hours and 1 day, respectively. Therefore, they are unable to capture some finer resolution space environment fluctuations. They also do not describe other physical drivers of thermospheric density, such as species mass density, full-spectrum emissions, temperature, etc. In the past decade, some density modeling efforts have incorporated the use of such additional drivers (e.g., HASDM); these efforts are reviewed in Section 2.4.

2.3 Thermospheric Density Models

Density models can be categorized into two model types: those that employ an empirical density model and those that use a physics-based space environment model. This section provides a description of two commonly used thermospheric density models, MSIS and TIE-GCM. MSIS is an example of an empirical model and TIE-GCM is a physics-based model. Both models are referenced throughout this dissertation and used to evaluate SoleiTool performance.

2.3.1 MSIS

The US Naval Research Laboratory Mass Spectrometer and Incoherent Scatter radar (NRLMSIS) was first introduced in 1986 and is more commonly referred to as MSIS [52]. The current version of MSIS is NRLMSISE-00, released in 2000. The MSIS model generates a density value when provided with various input parameters. These parameters include latitude, LST, altitude, day of year (DOY), F10.7 index, and the Ap index, which can be derived directly from the Kp index. MSIS is one of the models used to evaluate SoleiTool performance in Chapter 6.

2.3.2 TIE-GCM

TIE-GCM is an open source general circulation model developed by the National Center for Atmospheric Research (NCAR) [56]. It produces a three-dimensional, nonlinear representation of the coupled thermosphere and ionosphere system [63]. By adjusting its primary inputs, Kp and F10.7 indices, TIE-GCM can simulate global thermospheric mass density changes under various solar and geomagnetic activity levels. For the purpose of this work, the default model output time step and resolution is used. The model output time step is set to 30 minutes and a horizontal resolution of 5 degrees latitude and 5 degrees longitude is used, with a vertical pressure level size of a half-scale height, extending from about 100 to between approximately 500 and 700 km (depending on solar cycle/F10.7) [41].

Vertical interpolation is performed during post-processing of the model output to obtain thermospheric density at any LEO orbit altitude. During calm conditions (Kp $<$ 3 and F10.7 $<$ 80 solar flux units), the TIE-GCM density field does not extend to LEO altitudes above about 500km, so the density field is extrapolated when necessary. Figure 2.6 shows an example of interpolation and extrapolation of TIE-GCM density to altitudes not directly modeled.

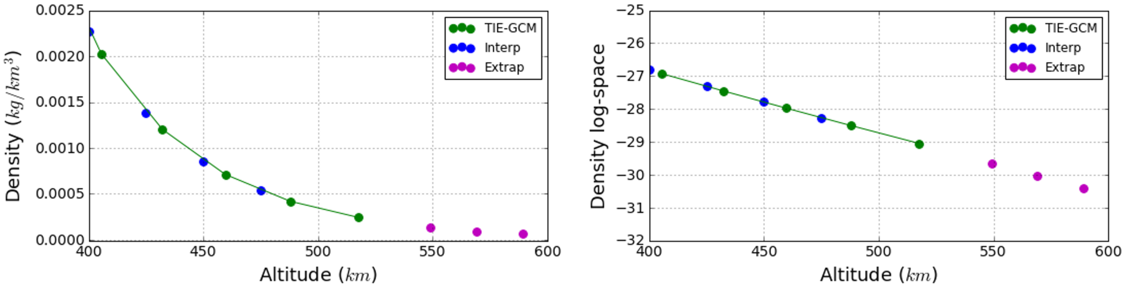

Figure 2.6: TIE-GCM Model Interpolation and Extrapolation; a) linear scale, b) logarithmic scale

For the purpose of this work, all TIE-GCM runs are completed before filter execution. TIE-GCM is run in a steady-state mode to populate a Kp-F10.7 state-space (0 to 9 Kp and 67 to 273 F10.7) with a resolution of roughly .5 Kp and 5 s.f.u for F10.7 for the span of the case study dates. This is particularly useful for the PF because a TIE-GCM density value is necessary at each state update for 50 different particle combinations of Kp and F10.7. Running TIE-GCM for all combinations at every state update time during the execution of the PF would drastically increase execution time.

2.3.2.1 TIE-GCM: Nominal vs. Steady-State Mode

TIE-GCM is typically run by inputting the time-series of historically measured Kp and F10.7 indices. When TIE-GCM is run in this nominal mode, the density field dynamic response to fluctuations in the Kp and F10.7 index can take hours. This is not the case when analyzing output from steady-state TIE-GCM runs. The steady-state density field is generated by initializing TIE-GCM with each combination of Kp and F10.7 days before the case study such that the density field reaches a diurnally reproducible state by the onset of the case study; the density field during the case study is produced by continuing to hold each Kp-F10.7 combination constant for the case study duration. The steady-state TIE-GCM density field reflects the state of the space environment after the density field has fully responded to the geomagnetic field disturbance and/or fluctuation in solar flux. Steady-state TIE-GCM runs and steady-state Kp are unique to SoleiTool and are only intended to be utilized within SoleiTool to produce a density field.

In SoleiTool, the TIE-GCM runs to populate the Kp-F10.7 state-space for the length of the case study are run with steady-state Kp and F10.7. Therefore, when a density value for a particular Kp and F10.7 (at a particular time and location) is desired, the density value can be retrieved from a density database populated via previously executed TIE-GCM runs. Due to the nature of the run, TIE-GCM steady-state mode does not necessarily capture the full range of transient thermospheric behavior. Some suggestions for implementing nominal-mode TIE-GCM in SoleiTool are discussed in future work (Section 7.2).

Figure 2.7 provides a theoretical example of the density along the trajectory of a satellite that is generated using TIE-GCM in nominal and steady-state mode. A fixed F10.7 of 75 s.f.u. and the Kp shown in the figure are used as input parameters to TIE-GCM to produce the density field response. The steady-state mode density field response to an increase in Kp is immediate, whereas the nominal model density field response is gradual.

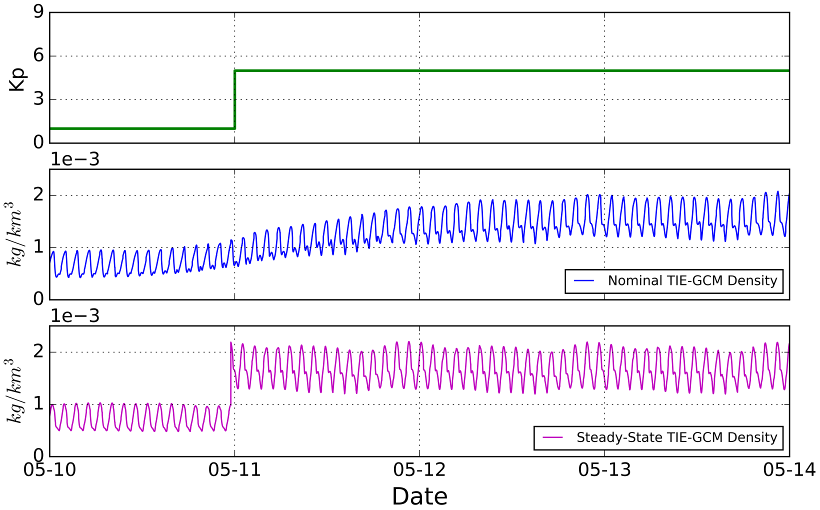

Figure 2.7: Nominal vs. Steady-State TIE-GCM density along a satellite trajectory and the Kp used as input to TIE-GCM

Figure 2.8 shows an example of the difference in the behavior of the steady-state Kp and the nominal Kp, as well as the global density field response. The magenta line shows the steady-state Kp index and the green line shows the nominal Kp index. Both the steady-state and nominal Kp indices are initially small. The nominal Kp index increases first, indicating the beginning of a geomagnetic storm. The steady-state Kp increases later as the density field responds to the geomagnetic field disturbance. There is a clear peak in the nominal Kp which represents the height of geomagnetic field disturbance. The steady-state Kp peak and mean global density peak occur later when the density field response to the geomagnetic field disturbance is at its maximum. The nominal Kp decreases first as the geomagnetic storm ends, followed by the steady-state Kp slowly tapering off as the density field enhancement subsides.

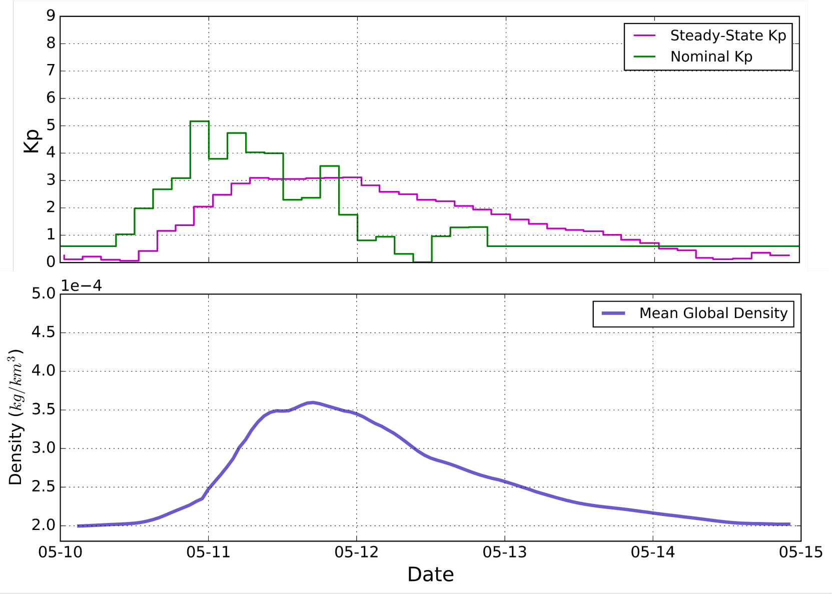

Figure 2.8: Steady-State Kp vs. Nominal Kp Index; Mean global density at 450 km altitude

2.4 Atmospheric Drag

An object in LEO experiences atmospheric drag caused by particles in the atmosphere colliding with the surface of the object. In fact, atmospheric drag is the second largest perturbation on LEO object motion; with J2 effects caused by the non-spherical Earth being the largest perturbation [18]. Drag acts primarily in the opposite direction of the velocity vector and effectively dissipates orbit energy. The magnitude of the force due to drag is directly dependent on neutral density. This is generally modeled as follows:

$$a_{drag} = -\frac{1}{2}\rho\frac{C_D A}{m}v_{rel}^2\frac{\vec{v}_{rel}}{|\vec{v}_{rel}\,|} \tag{2.1}$$

where

$$\vec{v}_{rel} = \frac{d\vec{r}}{dt} - \vec{\omega}_{\oplus} \times \vec{r} \tag{2.2}$$

is the velocity vector relative to an atmosphere co-rotating along with the Earth; C_D is the coefficient of drag; A is the exposed cross-sectional area; m is the object's mass; $\vec{\omega}_{\oplus}$ is the mean motion of the Earth's rotation; $\vec{r}$ is the object position vector in an Earth Centered Inertial reference frame;

and ρ is thermospheric density [68]. The drag coefficient, mass, and area of the object are typically combined to form the ballistic coefficient ($\beta = \frac{C_D A}{m}$). C_D and A can vary in time as a function of spacecraft attitude. For this work, C_D is computed using the force model of Sutton [61] with an accommodation coefficient of $\alpha = 0.93$, which takes spacecraft physical characteristics and attitude into account. The cross-sectional area, A, is computed using a spacecraft body model and attitude information.

To demonstrate the effect that drag has on a LEO trajectory, Figure 2.9 provides the position and velocity difference at the end of a 20-minute arc between a satellite propagated with a more enhanced density field (higher Kp) and a density field with calmer conditions (lower Kp). The satellite has a ballistic coefficient of 0.05 m^2/kg, which is similar in magnitude to the ballistic coefficients of Spire CubeSats. The density fields are produced in TIE-GCM using a constant F10.7=80 with Kp=1, 3, or 5. A larger position difference indicates a more observable drag effect. The y-axis is the satellite orbit altitude. As orbit altitude increases the difference in drag effect decreases and therefore becomes less observable. This analysis informs the minimum arc length limitation of 20 minutes applied in the Batch portion of SoleiTool, described in Chapter 4. If the drag effect difference is within the CubeSat POD measurement noise, information about the thermospheric density influencing the satellite orbit is not observable.

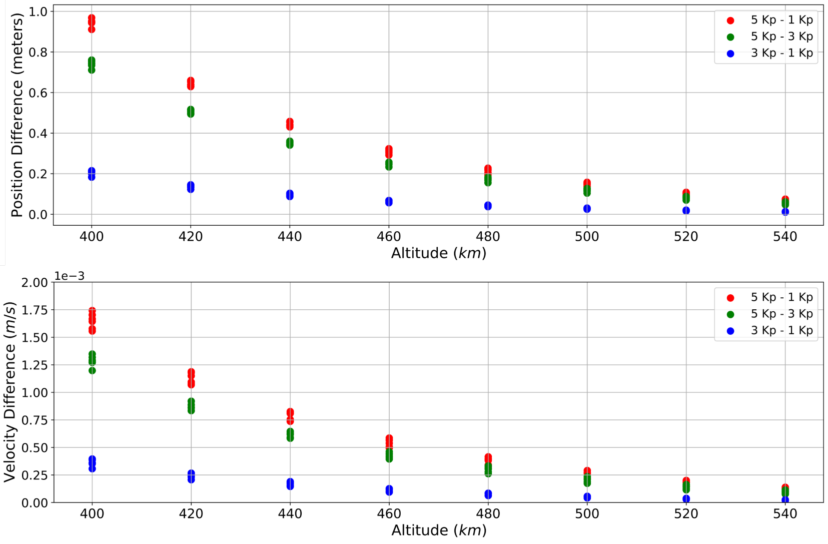

Figure 2.9: Drag Effect on LEO Satellite; Position and Velocity difference between propagation with Kp = 1, 3, and 5 density field as a function of orbit altitude

2.5 Density Retrieval and Modeling

2.5.1 Data Assimilation with Empirical Density Models

The operational High Accuracy Satellite Drag Model (HASDM) is an example of a purely empirical-based approach to thermospheric density modeling. HASDM is the operational density model used by the US Space Force (USSF) Combined Space Operations Center (CSpOC). HASDM currently estimates a time-series of ten 3-hour global temperature correction coefficients to an empirical density model using real-time ground-based radar measurements of 75-80 carefully selected calibration satellites (payloads and debris objects) in a batch fit [6, 7, 60]. Intensive sensor tasking is made available for this effort, which allows for the collection of approximately 500 observations per day per calibration satellite. HASDM decouples the ballistic coefficient and the density parameter by first solving for the long-term averaged ballistic coefficient of each satellite, averaging almost 3200 previously estimated values for each calibration satellite [7]. In addition to estimating temperature correction coefficients, HASDM also utilizes solar and geomagnetic indices as input to the model. Solar indices include the standard F10.7, as well as S10, M10, and Y10, which describe solar flux at wavelengths of 26-34 nm (EUV), 160 nm (FUV Schumann-Runge photospheric irradiance proxy), 121.6 nm (Lyman-alpha chromospheric/transition region irradiance), and 0.1-0.8 nm (X-ray coronal irradiance) [66]. Geomagnetic indices, Ap and Dst, are also used. Ap is derived directly from the Kp index, and Dst is an indication of magnetic activity originating from the Earth's ring current. HASDM temperature correction coefficients are not made available to the public, but solar indices have been publicly available by Space Environment Technologies (SET) since 2012 and the HASDM global density outputs were recently released by SET [66]. Even with operational use of HASDM, satellite drag remains the largest contributor to error in the CSpOC LEO trajectory predictions, therefore, providing motivation for continued exploration of other modes of density estimation and modeling.

Newer density modeling approaches include Mehta and Linares [43], who developed an approach that aims to combine the strength of a physics-based model in its predictive capability, with the speed of an empirical model. They introduced a reduced-order model to represent a high-dimensional thermospheric density model using a smaller number of parameters. The dimensionality reduction was achieved via Proper Orthogonal Decomposition. This approach was first demonstrated using an empirical model, MSIS, as the underlying upper atmosphere model with the intention that a physics-based model will replace MSIS in the future. More recently, Mehta et al. [45] have extended this work by using CHAMP and GRACE measurements to calibrate the MSIS model. High-dimensional state estimation is performed via a reduced order modeling approach to determine the contributions of thermosphere composition and temperature to model-data differences. Their results show that MSIS over-predicts atomic oxygen and helium.

In [21], Gondelach and Linares estimate global density using two reduced-order density models, generated with empirical models, MSIS and Jacchia-Bowman 2008 (JB2008). Measurements in the form of range and range-rate of 20 objects (active and defunct satellites for which the ballistic coefficient is known) and GPS positions of 10 satellites are assimilated to estimate a reduced state. A constant ballistic coefficient is estimated for the 10 satellites that provide the GPS measurements. The densities derived using the range-rate measurements of 20 objects is compared to the densities derived using GPS data of 10 satellites. It is found that the latter has better performance. The orbit and daily averaged density, estimated using GPS measurements, is computed along the orbits of Swarm satellites and compared to Swarm-derived density to assess the performance of their method. They report a daily 5% 1-σ error in the estimated density when the GPS data for 10 satellites are used as measurements, improved from 22%, 14%, and 10% for MSIS, JB2008, and TLE-estimated densities. They did not provide a comparison to HASDM or a physics-based density estimation approach in this effort.

2.5.2 Data Assimilation with Physics-Based Models

Incorporating a physics-based model requires sophisticated data assimilation methodologies and substantially more computing power compared to empirical models. In the last decade, advances in computing technology have made it possible to explore density modeling and estimation methods that do this. For example, Morozov et al. [46] assimilated CHAMP neutral mass density measurements to estimate a driver, F10.7, to the physics-based Global Ionosphere-Thermosphere Model (GITM) using an Ensemble Adjustment Kalman Filter (EAKF). Matsuo [40] subsequently showed that ionospheric electron density observations can be used to infer the F10.7 driver parameter of TIE-GCM, to supplement sparse thermospheric mass density observations.

Pilinski et al. [53] combine the strength of multiple physics-based models (referred to as super-ensemble members) and several empirical models to estimate driver parameters, F10.7 and Ap index, in an EAKF using Two-Line Element sets (TLEs) of 75 calibration satellites (objects for which the shape and ballistic coefficient are known).

Sutton [62] assimilated neutral density data from CHAMP to estimate atmospheric drivers as inputs to TIE-GCM. This approach provides a time-history of TIE-GCM Kp and F10.7 drivers that can be used to model global density, with the potential capability for a 3-hour forecast using the most recent estimated drivers. It was shown that this method improves model-data agreement both near and away from the orbit of the assimilated data.

2.5.3 Density Retrieval using GPS Data

An important development in the past decade is the utilization of GPS-based orbit information from LEO satellites to estimate thermospheric density. Most of these efforts estimate density only along the observed satellite trajectories, and do not assimilate the GPS data to model density globally. For example McLaughlin [42] used POD information, derived from continuous GPS data, to estimate density along the trajectories of CHAMP, GRACE, and TerraSAR-X. This was done by estimating ballistic coefficients and corrections to various baseline empirical models (i.e., MSIS), simultaneously. Density estimates were found to have better agreement with accelerometer data than empirical models. This work also revealed that there are high frequency density variations (i.e., nocturnal peaks) that are captured by the accelerometers, but are non-existent or only slight bumps in the the POD-derived and HASDM densities.

Similarly, Kuang et al. [37] use GPS-based POD data from CHAMP, GRACE, SAC-C, Jason-2, TerraSAR-X and COSMIC satellite to derive thermospheric density along a satellite trajectory. This was accomplished by estimating an averaged coefficient of drag and an atmospheric drag acceleration correction to an underlying nominal empirical thermospheric density model, the Drag Temperature Model (DTM-2000). The corrected drag acceleration and constant drag coefficient allow for the extraction of thermospheric density along each satellite trajectory. They report that the estimated density data from CHAMP, GRACE, SAC-C and TerraSAR-X have high correlations with a nominal thermospheric density model. On the other hand, the estimated density data from COSMIC and Jason-2 showed errors larger than the actual signal, making it of no value. This is due to the higher orbit altitude of COSMIC and Jason-2, 500-800 km and 1348 km altitude, respectively, compared to CHAMP, GRACE, SAC-C and TerraSAR-X with a 250-440 km, 480 km, 440 km, and 520 km altitude, respectively. Calabia and Jin [12] used reduced-dynamic POD to fill-in gaps in GRACE thermospheric mass densities. They showed that the precision of the POD-derived densities have good agreement with accelerometer-based mass densities (<10% relative error), and better performance than that of MSIS [10, 11, 12].

After finding several anomalies in the Swarm satellite mission accelerometers, members of the Swarm mission team, van den IJssel et al. [69], developed a method to determine thermospheric density along the tracks of three Swarm satellites (orbits ranging from 430 to 510 km altitude) using GPS data. This is done via a two-step process in which the non-gravitational accelerations are derived using GPS data and gravity field models in the first step, and converted into thermospheric densities in the second step. This information is used to apply corrections, with a resolution of about 20 minutes, to the anomalous accelerometer data. The thermospheric density values obtained from the Swarm GPS data are located only along the three Swarm satellite trajectories and there was no assimilation of the data in combination with a space environment model. The derived densities are available as a secondary product of the mission. Evaluation of the Swarm-derived densities was performed via comparison with MSIS, which showed good agreement. In Chapter 6, we compare SoleiTool estimated density to Swarm-derived density along the Swarm-C trajectory.

2.5.4 Data Assimilation with Physics-Based Model using GPS Data

The methods using GPS data that have been discussed thus far do not assimilate density data for the benefit of modeling density for other LEO objects; but rather, use satellite GPS data to estimate density only along the trajectories of those same satellites. The approach taken by Sutton et al. [64] combines GPS information with assimilation and a physics-based model to extend estimated density beyond that of the individual satellite trajectories. They used 78 days of POD information for three Spire CubeSats in late 2018 to retrieve a time-series of the effective energy dissipation rate (EDR) with a resolution of one orbital period. Orbital variations (EDRs) are used to obtain thermospheric density along the satellite trajectory with a cadence of about 2-2.5 hours. To obtain a global density field estimate, EDRs are assimilated to estimate driver parameters (most recent daily F10.7 and a 6-hour Kp index) to TIE-GCM. Sutton et al. evaluated the performance of their method by quantifying the agreement between the EDR-derived density and Swarm-derived density along the trajectory of Swarm-C. The EDR-derived density outperforms TIE-GCM using historically measured F10.7 and Kp, however, HASDM still has the best agreement with Swarm density.

Like [64], SoleiTool combines the strength of a physics-based method, satellite GPS information, and data assimilation to estimate global density. In this work, we aim to advance what has been accomplished in terms of the temporal resolution of an estimated global density field. We demonstrate SoleiTool's ability to utilize POD information from CubeSats to sense a global density field during calm and storm-time conditions. SoleiTool extracts density information from POD arcs as short as 20 minutes to estimate input parameters to TIE-GCM with a resolution of 1 to 1.5 hours. This finer time resolution allows SoleiTool to capture smaller density features than previous methods, which is beneficial for more accurate LEO trajectory predictions.

Chapter 3

Nonlinear Filter Comparison

3.1 Nonlinear Filters

Nonlinear systems pose challenges in estimation that require techniques and approaches to handle non-Gaussian distributions due to nonlinear dynamics and/or measurements. This chapter compares three filters used for nonlinear systems: the Unscented Kalman Filter (UKF), the Ensemble Kalman Filter (EnKF), and Particle Filter (PF).

An obvious commonality between the three filters is that they take advantage of a particle representation of the N-dimensional state PDF. The number of particles each uses is an obvious difference among the filters. The UKF uses $2N + 1$ weighted "particles," or sigma points, to represent the first and second moments of the non-Gaussian PDF. The PF employs at least 10^N weighted particles to capture the full non-Gaussian PDF, while the EnKF uses an ensemble of state realizations, or "particles," of size greater than $2N + 1$ and less than 10^N. The number of particles in each filter is a direct indication of the level of non-linearity the filter maintains. In other words, the filter with the smallest number of particles makes more linear assumptions and therefore experiences a greater loss of information in nonlinear conditions.

For ease of comparison, similar notation is used for each filter. For example, χ represents the set of sigma points in the UKF, the ensemble in the EnKF, and the particles in the PF.

3.2 Unscented Kalman Filter

The UKF was introduced in 1997 as a linear estimator that also adapts elegantly to nonlinear systems [31]. The UKF utilizes a deterministic set of particles, called sigma points, whose distribution is determined via the state covariance [9]. An unscented transform is applied in the measurement update in order to avoid truncation of the nonlinear dynamics higher order terms, as would occur in the Kalman Filter and Extended Kalman Filter (EKF). The UKF is considered a combination of a Monte Carlo (MC) and an EKF [25].

The UKF uses $2N+1$ weighted sigma points to represent the state PDF. The sigma points capture the first two moments, the mean and uncertainty, of the state PDF. Therefore, the sigma points are not a full representation of the state PDF when the state is non-Gaussian. The UKF is able to maintain 2nd order accuracy for nonlinear systems and 3rd order accuracy for linear systems (equivalent to Kalman Filter). This is because a Gaussian is fit to the posterior distribution at each measurement time [24]. The initial distribution of the sigma points is provided in Eq. 3.1,

$$\chi_k^{(i)} = \begin{cases} X_k & i = 0 \\ X_k + \gamma\sqrt{P_k} & 1 \geq i \geq N \\ X_k - \gamma\sqrt{P_k} & N > i \geq 2N \end{cases} \tag{3.1}$$

where γ is a tuning parameter (Eq. 3.2). Superscript (i) denotes the i^{th} sigma point and +/- denotes the posterior/a priori distribution. Subscript k indicates the k$^{\mathrm{th}}$ time step.

$$\gamma = \sqrt{N+\lambda} \tag{3.2}$$

$$\kappa = 3 - N \tag{3.3}$$

$$\lambda = \alpha^2 (N+\kappa) - N \tag{3.4}$$

α determines spread of sigma points and is selected so that $1e - 4 < \alpha < 1$. A scaled parameterization of the UKF was introduced by Julier [32] that avoids negative particle weights and is therefore less susceptible to numerical errors. These adjustments are reflected in the weight of the mean sigma point ($i = 0$) used in the calculation of the state mean ($\overline{x}$), the state in measurement space ($\overline{y}$), and the covariance (P) defined by Eq. 3.5, and Eq. 3.6 for covariance calculations. The weight superscripts m and c indicated the weighting parameters used for the weighted state mean and covariance calculations, respectively.

$$w^{(0),m} = \frac{\lambda}{N + \lambda} \tag{3.5}$$

$$w^{(0),c} = \frac{\lambda}{N + \lambda} + \left(1 - \alpha^2 + \beta\right), \ \beta = 2 \tag{3.6}$$

Likewise, the weights for the remaining sigma points are defined by

$$w^{(1 \geq i \geq 2N),m} = w^{(1 \geq i \geq 2N),c} = \frac{1}{2\left(N + \lambda\right)} \tag{3.7}$$

Weights are defined in such a way that the sum of the particle weights equals one. The sigma points are propagated using the full nonlinear dynamics,

$$\chi_k^{(i)-} = \mathcal{M}\left(\chi_{k-1}^{(i)+}\right) \tag{3.8}$$

and the resulting weighted mean state and covariance are calculated:

$$\overline{X}_k = \frac{1}{N} \sum_{i=1}^{N} w^{(i),m} \chi_k^{(i)-} \tag{3.9}$$

$$P_k^- = \sum_{i=1}^{N} w^{(i),c} (\chi_k^{(i)-} - \overline{X}_k)(\chi_k^{(i)-} - \overline{X}_k)^T + Q \tag{3.10}$$

where Q is the process noise covariance. Unlike the EnKF and the PF, the UKF involves an optional extra step in the time update, in which the sigma points are re-computed in order to incorporate the effects of process noise (Q) [33], [32]:

$$\chi_k^{(i)} = \begin{cases} X_k & i = 0 \\ X_k + \gamma\sqrt{P_k^-} & 1 \geq i \geq N \\ X_k - \gamma\sqrt{P_k^-} & N \geq i \geq 2N \end{cases} \tag{3.11}$$

The state in measurement space is defined by

$$\mathcal{Y}_k^{(i)} = \mathcal{G}\left(\chi_k^{(i)-}\right) \tag{3.12}$$

and the weighted average of the sigma points in measurement space is computed,

$$\overline{y}_k = \sum_{i=1}^{N} w^{(i),m}\mathcal{Y}_k^{(i)} \tag{3.13}$$

The measurement update is computed via the Unscented Transform:

$$P_{yy} = R + \sum_{i=1}^{N} w^{(i),c}(\mathcal{Y}^{(i))} - \overline{y}_k)(\mathcal{Y}^{(i))} - \overline{y}_k)^T \tag{3.14}$$

$$P_{xy} = \sum_{i=1}^{N} w^{(i),c}(\chi_k^{(i)-} - \overline{X}_k)(\mathcal{Y}^{(i))} - \overline{y}_k)^T \tag{3.15}$$

$$K = P_{xy}(P_{yy})^{-1} \tag{3.16}$$

$$X_k^+ = \overline{X}_k + K\left(\tilde{y}_k - \overline{y}_k\right) \tag{3.17}$$

$$P_k^+ = P_k^- - KP_{yy}K^T \tag{3.18}$$

where $\tilde{y}$ is the observation vector. The post measurement update sigma points are then computed using X_k^+ and P_k^+ in Eq. 3.1. The process repeats with each measurement.

3.3 Ensemble Kalman Filter

The EnKF was introduced to estimate high-dimensional systems in a computationally tractable manner [16]. It does so by combining Kalman filter theory and Monte Carlo estimation methods

[28]. It was formulated to accommodate high-dimensional atmospheric forecast modeling applications via a low-dimensional approximation in order to maintain computational feasibility. The EnKF applies a linear update to a prior, non-Gaussian distribution, making it a partially nonlinear filter [17]. The particles take on a non-Gaussian distribution in the case of a nonlinear problem and are not forced to form a Gaussian distribution at any point in the filter. However, the covariance is assumed to be Gaussian which leads to a linear measurement update. As a result, the solution of the EnKF is thought of as "something between a linear Gaussian update and a full Bayesian computation" [17]. This is because the uncertainty of the state is derived from a non-Gaussian distribution (non-Gaussian for a nonlinear system as a result of nonlinear dynamics) and an information loss occurs when the state covariance matrix is represented as a single Gaussian distribution. The measurement update would be a full Bayesian computation if the non-Gaussian nature of the state uncertainty was maintained.

The EnKF employs a MC method in the form of an ensemble representation of the estimated state probability distribution. The ensemble contains N members where each member is a sample realization of the state that is drawn from a normal distribution. Combined, the ensemble members represent the a priori distribution of the state and define the initial mean and variance of each state element. An ensemble representation of the initial state is generated using the a priori statistical information of the state X as shown in Eqs. 3.19 and 3.20.

$$\chi_0^{(i)} = X_0 + \eta^{(i)} \tag{3.19}$$

$$\eta^{(i)} \sim \mathcal{N}(0, P_0) \tag{3.20}$$

The EnKF time update is the nonlinear propagation of the state ensemble (Eq. 3.21 [28]), where $\mathcal{M}$ is the nonlinear operator. Superscript (i) denotes the i$^{\text{th}}$ ensemble member. The covariance time update is given by Eq. 3.24. The covariance calculation differs from the UKF due to the absence of particle weights in the EnKF.

$$\chi_k^{(i)-} = \mathcal{M}\left(\chi_{k-1}^{(i)+}\right) + \omega^{(i)} \tag{3.21}$$

$$\omega^{(i)} \sim \mathcal{N}(0, Q) \tag{3.22}$$

$$\overline{X}_k = \frac{1}{N}\sum_{i=1}^{N} \chi_k^{(i)-} \tag{3.23}$$

$$P_k^- = \frac{1}{N-1}\sum_{i=1}^{N} (\chi_k^{(i)-} - \overline{X}_k)(\chi_k^{(i)-} - \overline{X}_k)^T + Q \tag{3.24}$$

The measurement update equations can be written as Eq. 3.12 and Eqs. 3.25-3.31 [28]. The function, $\mathcal{G}$ (Eq. 3.12), is the measurement equation as a function of the state, Y is the predicted measurement, and R is the measurement covariance matrix. Note that the Kalman gain (Eq. 3.28) is equivalent to the Kalman gain calculation in the UKF (Eq. 3.16); the only difference being the definition of P_{xy} and P_{yy}. Unlike the UKF or PF, the measurement update for each EnKF ensemble member incorporates a uniquely perturbed observation, y^{o} (Eq. 3.29). This is a signature feature of the stochastic EnKF.

$$\overline{y}_k = \frac{1}{N}\sum_{i=1}^{N} \mathcal{Y}_k^{(i)} \tag{3.25}$$

$$P_{yy} = R + \frac{1}{N-1}\sum_{i=1}^{N} (\mathcal{Y}^{(i))} - \overline{y}_k)(\mathcal{Y}^{(i))} - \overline{y}_k)^T \tag{3.26}$$

$$P_{xy} = \frac{}{N-1}\sum_{i=1}^{N} (\chi_k^{(i)-} - \overline{X}_k)(\mathcal{Y}^{(i))} - \overline{y}_k)^T \tag{3.27}$$

$$K = P_{xy}(P_{yy})^{-1} \tag{3.28}$$

$$y_k^o = \tilde{y}_k + \epsilon \tag{3.29}$$

$$\epsilon \sim \mathcal{N}(0, R) \tag{3.30}$$

$$\chi_k^{(i)+} = \chi_k^{(i)-} + K\left(y_k^o - \mathcal{Y}^{(i))}\right) \tag{3.31}$$

The EnKF extracts the posterior state covariance (Eq. 3.32) from the posterior distribution represented by the ensemble, similar to the Unscented Kalman Filter or a MC approach. The inverse operation of $\mathcal{G}$ is implemented by linear regression using the prior ensemble, which transforms the posterior ensemble expressed in the measurement space to state space.

$$P_k^+ = (I - KH)P_k^-(I - KH)^T + KRK^T \tag{3.32}$$

$$H = \frac{\partial Y}{\partial X} \tag{3.33}$$

The overall flow of the EnKF algorithm is shown below in Figure 3.1. Each time step of the algorithm typically includes the propagation to the next time step (step 1) and the application of a correction to each ensemble member based on the measurement (step 2m and 2s). Step 2m represents the state ensemble and measurement in measurement space, and the correction applied to each ensemble member due to the measurement. Step 2s shows the state ensemble and each member's corrections in state space. Upon completion of steps 1 and 2, a new estimate is available that represents the filter's best estimate at time $\mathrm{t_{k+1}}$.

3.4 Particle Filter

The PF was introduced in 1993 [22] and is known as a truly non-linear filter because no Gaussian or linear assumptions are made. The PF is a MC filter, similar to the UKF and the EnKF. The PF approach is more similar to the UKF in that both employ a MC method within each measurement step. In the PF, resampling can occur at all measurement times, referred to as sampling importance *resampling*, or resampling can occur at select measurement times, termed

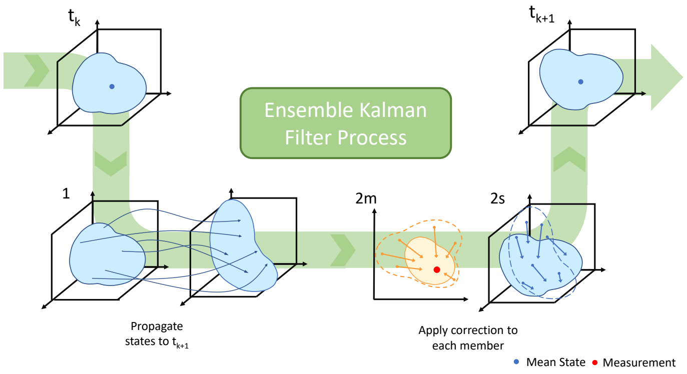
t_k
t_{k+1}
Ensemble Kalman Filter Process
1
2m
2s
Propagate states to t_{k+1}
Apply correction to each member
● Mean State ● Measurement

Figure 3.1: EnKF Algorithm Summary

sampling importance *sampling*. The filter effectively restarts, or bootstraps, the MC simulation with each resampling. This is in contrast to the EnKF which maintains the same particles throughout.

The PF begins with a distribution of particles generated from the initial state covariance (Eqs. 3.19 and 3.20). The particles are distributed in such a way as to capture the statistically probable states. The generation of the initial particle distribution is comparable to the EnKF initial ensemble generation; however, far more particles (10^n or greater) are generated for the PF than the EnKF. The PF requires a large quantity of particles because it is desired that each state element, or dimension, have at least 10 independent possible particle realizations [54]. The initial particles have a uniform weight of $1/10^n$.

The forecast step for the PF is the same as the UKF and EnKF. The particles are propagated with the full nonlinear dynamics (Eq. 3.8). Each particle weight is maintained through the propagation. However, variations of the PF exist in which resampling and particle weight updates can occur after the propagation within the time update.

The analysis step, or measurement update, of the PF is where the major differences between it and the UKF/EnKF occur. Instead of a measurement update operating on the state realization of each particle, as is done in the EnKF, the measurement update occurs via the particle weights alone. The new particle weights are calculated via the likelihood function; the higher the likelihood of a particle, the higher the particle weight. The likelihood function is provided below (Eq. 3.34).

$$\tilde{w}_k^i = p\left(y|\chi_k^i\right) \tag{3.34}$$

By setting the particle weights directly equal to the likelihood function the filter is simply asking “how close is the state in measurement space (Eq. 3.12) located to the measurement PDF, $\mathcal{N}(y, R)$?” This idea is demonstrated below, in Figure 3.4, where a handful of 1D particles are compared to a 1D measurement PDF.

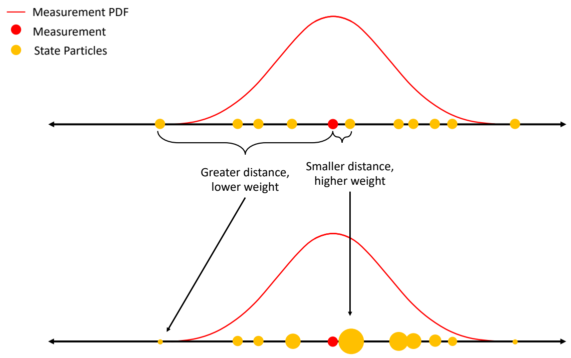
Measurement PDF
Measurement
State Particles
Greater distance, lower weight
Smaller distance, higher weight

Figure 3.2: PF Particle Weight Update

After finding the likelihood of each particle, the particle weights are normalized to sum to 1 (Eq. 3.35) and resampled.

$$w_k^i = \frac{1}{N}\sum_{i=0}^{N} \tilde{w}_k^i \tag{3.35}$$

The PF resamples the posterior distribution of particles based on their corresponding weights and a proposal function $q\left(\cdot\right)$ called the importance density [4]. There are several different importance densities that have been developed for the PF. For the purpose of this paper, we implement the most straightforward resampling strategy: systematic resampling [58]. Eq. 3.36 indicates the particular importance density used in this algorithm.

$$\chi_{k+1}^{(i)} \sim q\left(\chi_{k+1}|\chi_k, y_{k+1}\right) \tag{3.36}$$

The PF does not necessarily have to resample with every measurement; instead, a strategy can be selected for determining when resampling should be employed; as mentioned previously, this is termed sampling importance sampling (SIS) [24]. Otherwise, sampling occurs at every measurement time and this is called sampling importance resampling (SIR) [24]. If using an SIS scheme, one can determine if resampling is necessary based on the effective number of particles (N_{eff}) [24].

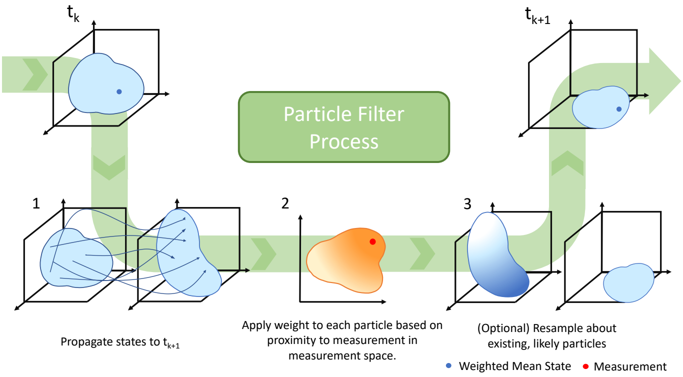
t_k
t_{k+1}
Particle Filter Process
1
2
3
Propagate states to t_{k+1}
Apply weight to each particle based on proximity to measurement in measurement space.
(Optional) Resample about existing, likely particles
● Weighted Mean State
● Measurement

Figure 3.3: PF Algorithm Summary

The overall flow of the PF algorithm is shown in Figure 3.3. Each time step of the algorithm includes the propagation to the next time step, adjustment of particle weights based on the measurement, and particle resampling. The filter begins with the best estimate at time t_k, the blue dot on the upper left, and the state covariance, represented by the blue cloud. The state at time t_k is either an estimate from the previous time step of the filter or *a priori* information. Particles that represent the probabilistic realizations of the state at time t_k are propagated forward to time t_{k+1} in Step 1. The uniform color of the blue cloud depicts the uniformly weighted particles. The particle weights are updated based on their proximity to the measurement, the red dot, in Step 2. This adjusts the particle distribution, as shown in measurement space by the orange cloud with the larger weighted particles represented by the dark orange region.

Next, the updated particles are shown in state space in Step 3 with larger weighted particles concentrated towards the bottom of the blue cloud (dark blue region). Step 3 is an optional step in which the particles are resampled and the outcome is a uniformly weighted particle distribution. The particles with larger weights are more likely to be resampled. Therefore, the resulting distribution is located where the dark blue region is before resampling. Based on particle position and weights, the best estimate is computed for time t_{k+1}, shown by the blue dot in the last figure of the PF process.

3.5 Summary and Comparison

As mentioned, a major difference between these three filters is the number of “particles”; the UKF has the smallest number of particles, while the PF has the greatest (orders of magnitude greater than the UKF). The initial distribution of particles is generated in the same way in the EnKF and PF: a cloud of particles distributed within the covariance ellipsoid. Whereas, the UKF has one particle equal to the mean and the others $(2n)$ are distributed by the square root of the covariance scaled by a tuning parameter.

In the time update, or forecast, all three filters utilize the fully nonlinear dynamics of the system in the time propagation. The computation time of this step is defined equally for each filter: O(N), but the computational cost varies greatly between the filters due to the vastly different number of particles.

The measurement update, or analysis step, varies greatly between the filters; the differences are driven by the assumptions in each. Similar to the PF, the UKF takes advantage of weighted particles, while the EnKF does not. All particles have uniform weight at all times in the EnKF. The correction applied to each particle is calculated almost identically in the UKF and EnKF. They both compute a Kalman gain using the innovation and cross-correlation matrices. The only difference is that the measurement is perturbed by ϵ (Eq. 3.29) in the stochastic formulation of the EnKF (formulation presented in Section 3.3). The PF takes a completely different approach to the measurement update. It updates the state estimate by updating particle weights and generating a new particle distribution based on the new weights. The UKF resamples the distribution after each measurement, whereas there are various resampling strategies for the PF. The PF may resample after the time update, or only resample at select measurement times [4]. An advantage of all three filters is that Jacobians are not needed for the measurement update, unlike the Extended Kalman Filter. This reduces the computation cost of each particle when the state is relatively large (N $\geq$ 6).

This work informed the decision of the filter type applied to each parametrization of my research estimation problem (Chapter 4, and Appendices A and B). Using satellites to estimate thermospheric density is a challenging estimation problem that requires special filtering methods and numerical techniques. Appendix B describes my initial approach to thermospheric density state estimation in which observations of 16 simulated debris objects are used to estimate a 3000-element spatial grid of thermospheric density. The true strength of an EnKF is estimating a high dimensional-state using a large number of measurements. Therefore, the POEnKF, a type of EnKF, was selected for this initial approach because radar measurements from a *large* number of debris objects are used to estimate a *high-dimension* spatial grid of density. The first version of SoleiTool, described in Appendix A, is when a two-step filtering process was implemented to use simulated orbital debris data in a UKF to estimate acceleration due to drag, and then a PF to estimate Kp and F10.7 TIE-GCM input parameters. The PF is the perfect selection for the second stage filter because it estimates a small, 2-element state; the PF actually becomes intractable when states have a large number of elements. In the current version of SoleiTool (Chapter 4), a two-step filtering process is still utilized, but the UKF has been replaced with a Batch filter because the state was re-parameterized to estimate a slowly-changing density coefficient of CubeSats with known ballistic coefficients, instead of acceleration due to drag of debris objects.

Chapter 4

Method: SoleiTool

As shown in Figure 4.1, SoleiTool is comprised of two filters: the Batch filter and Particle Filter. Separating the problem into two cascading filters provides an opportunity to take advantage of the unique strengths of each filter. The purpose of the Batch filter is to estimate thermospheric density in a localized spatial area using a single arc of CubeSat POD measurements. The Batch filter is run in a loop that steps through the available satellite data, advancing in time, and running one Batch per arc of data (single-arc Batch). This is done for each satellite, separately. Each run of the Batch filter ingests CubeSat POD measurements (described in detail in Section 6.1) and estimates a density coefficient (C_ρ) for a single arc of data. The coefficient measurements for all satellites are passed to a single PF to estimate a time-series of Kp with a fixed F10.7. The end product of SoleiTool is an estimated global density field time-series obtained using the estimated Kp time-series and fixed F10.7 as input parameters to TIE-GCM (Section 4.3). Details on each filter and the resulting global density field are provided in the following subsections. Features of the Batch and PF design and implementation decisions have been informed by Spire data, TIE-GCM characteristics, and the case study. The filter type, measurements, and estimated states of both filters are summarized as follows:

(1) **Batch**

- Measurements: CubeSat POD Position and Velocity
- Estimated State: CubeSat Position and Velocity, C_ρ per Arc

(2) **PF**

- Measurements: C_ρ
- Estimated State: Kp

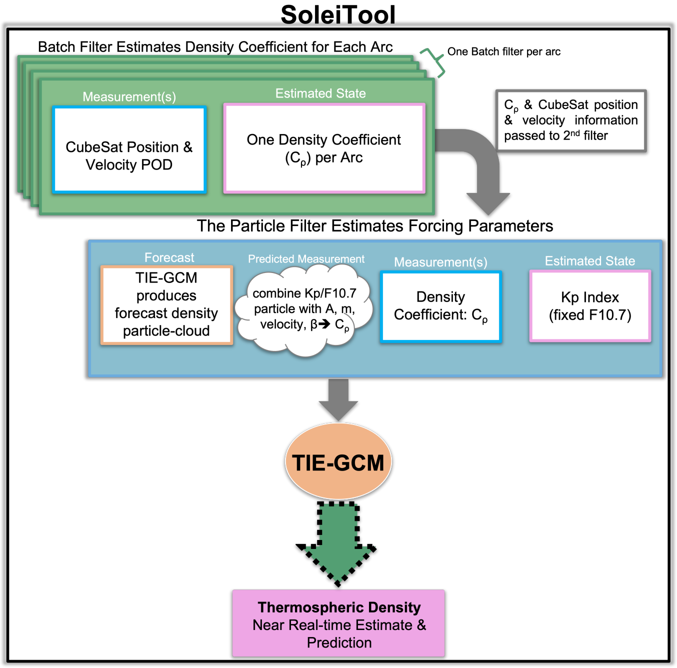
SoleiTool
Batch Filter Estimates Density Coefficient for Each Arc
One Batch filter per arc
Measurement(s)
CubeSat Position & Velocity POD
Estimated State
One Density Coefficient (C_ρ) per Arc
C_ρ & CubeSat position & velocity information passed to 2nd filter
The Particle Filter Estimates Forcing Parameters
Forecast
TIE-GCM produces forecast density particle-cloud
Predicted Measurement
combine Kp/F10.7 particle with A, m, velocity, β→ C_ρ
Measurement(s)
Density Coefficient: C_ρ
Estimated State
Kp Index (fixed F10.7)
TIE-GCM
Thermospheric Density
Near Real-time Estimate & Prediction

Figure 4.1: SoleiTool Flow

4.1 Batch Filter

The purpose of the Batch filter is to relate each observed satellite trajectory to an observed thermospheric density. The single-arc Batch architecture is designed to sense the density of a localized spatial area, while also considering a long enough arc of data such that the density-driven drag perturbations are observable (given the POD measurement error). This is accomplished by quantifying differences between the observed spacecraft trajectory (obtained from POD ephemeris) and a trajectory propagated using a background density model. The background density model is a fixed density field generated with TIE-GCM for relatively calm conditions (i.e., input parameters of Kp=3 and F10.7=80). SoleiTool loops through the data and runs an instance of the batch on each arc of data until the end of the data set is reached. An arc is defined as 20 to 40 minutes of continuous POD for a single CubeSat.

Orekit [39] is used to model the underlying orbit dynamics, including 140 degree/order Earth gravity; sun and moon third body perturbations; cannonball SRP; solid Earth and ocean tides; relativity; and atmospheric drag. We augmented the standard Orekit atmospheric drag model to utilize TIE-GCM density rather than one of the conventional underlying density models included with Orekit, such as Harris Priester.

The Batch filter ingests a single arc of CubeSat POD measurements in the form of a position and velocity time-series. At time k the measurement is,

$$\vec{y_k} = \left[\vec{r}_{k,POD}^{T}, \vec{v}_{k,POD}^{T}\right]^{T} \tag{4.1}$$

and estimates a density coefficient (C_ρ) based on the nominal acceleration due to drag equation (e.g., Vallado, 2013),

$$a_{drag} = -\frac{1}{2}\left(C_\rho * \rho_b\right)\frac{C_D A}{m} v_{rel}^2 \frac{\vec{v}_{rel}}{|\vec{v}_{rel}|} \tag{4.2}$$

with the addition of C_ρ multiplying the background density field,

$$C_\rho * \rho_b \tag{4.3}$$

The background density model is generated using Kp=3 and F10.7=80. C_D is computed using Sutton's force model [61] with an accommodation coefficient of $\alpha = 0.93$. The time-varying cross-sectional area, A, is computed using a spacecraft body model and attitude information.

A Batch filter estimated density coefficient greater than one ($C_\rho > 1$) indicates that the satellite experienced an enhanced density field on average throughout the arc compared to the background density model. On the other hand, an estimated density coefficient less than one ($C_\rho < 1$) indicates the satellite experienced less acceleration due to drag than that produced by the background density model, meaning that the true density field is likely generated using Kp $<$ 3 and/or F10.7 $<$ 80 s.f.u. On occasion, the Batch filter estimates a negative coefficient of density, which would signify a non-physical density field. These non-physical estimates are replaced by $C_\rho = 0.001$ to represent the most calm version of the density field. Once the Batch filter has processed all available CubeSat data, C_ρ estimates for each arc are passed to the PF.

4.2 Particle Filter

A single run of the PF assimilates information from all arcs and all CubeSats to reveal the time varying global density. More specifically, the PF uses the time-series of all C_ρ estimates resulting from all runs of the Batch, for all CubeSats, to estimate a time-series of TIE-GCM input parameters.

TIE-GCM is run in steady-state mode to populate the Kp and F10.7 state-space for the length of the case study before filter execution (described in more detail in Section 2.3.2). By precomputing, we avoid running TIE-GCM during filter execution.

Kp and F10.7 are the primary drivers of TIE-GCM. Historically, F10.7 is less dynamic and remains relatively constant for days to weeks because it is driven by the 11-year solar cycle and partially dependent on the solar rotation. The initial SoleiTool method was to estimate both Kp and F10.7 indices. However, when this was done, we found that the PF did not estimate a Kp and F10.7 time-series that is representative of the physical drivers of the density enhancement. The Kp and F10.7 retrievals in the PF attributed more to F10.7 and less to Kp than one would expect from the historical record, where F10.7 is flat and Kp is quite elevated during the storms. In order for TIE-GCM to be driven by the appropriate physics so that it can provide a meaningful density forecast, it is necessary to correctly attribute the physical drivers of a disturbed density field. Therefore, we estimate only a Kp time-series while holding F10.7 stationary. When SoleiTool is executed in near real-time, a fixed F10.7 can be chosen based on the most recent several days of measured F10.7. This forces the enhanced density field sensed via the in-situ satellite POD to be represented via changes only in the Kp index instead of both Kp and F10.7. This aligns with the true Kp and relatively constant F10.7 behavior shown in Figure 6.3 (relatively constant F10.7).

The PF combines 1 to 1.5 hours of C_ρ measurements from various satellites at their locations to execute a state update. The PF applies corrections in the form of weight adjustments to the cloud of Kp particles based on the difference between the C_ρ predicted measurement and the C_ρ measurements (Batch filter estimates). The predicted C_ρ measurement for a particular Kp particle is the ratio between the Kp particle density ($\rho_{particle}$) and the background model density (ρ_b) at the satellite location and time of the C_ρ measurement:

$$C_{\rho\ predicted} = \frac{\rho_{particle}}{\rho_b} \tag{4.4}$$

4.3 SoleiTool Density

The final step is to produce an estimated global density field that aims to capture the true underlying density field by using the estimated Kp time-series and fixed F10.7 as input to TIE-GCM. We do not aim to estimate the actual 3-hour global index of geomagnetic activity (Kp), but rather TIE-GCM input parameters that reproduce the observed CubeSat acceleration due to drag. The density field typically takes about a half day to fully respond to the energy input from an elevated Kp (geomagnetic storm). So, when there is an event, we expect ground measurements of Kp to increase first and then SoleiTool estimated Kp to rise later, once the enhanced density field affects the CubeSat orbit.

The assimilation of GPS data from individual CubeSats to provide a global thermospheric density model highlights the true power of SoleiTool. SoleiTool extends our understanding of the LEO space environment using limited localized CubeSat information.

Chapter 5

Simulation Scenario

This chapter evaluates the performance of SoleiTool using a simulation scenario modeled after the current Spire CubeSat constellation. This simulation scenario is created to study the ability of SoleiTool to detect changes in thermospheric density, therefore it takes place during a geomagnetic storm that is modeled after a historical storm in May 2019. A total of 119 satellites are propagated through the simulated density field and processed in SoleiTool to obtain the estimated global density field. The simulated satellite orbits and physical characteristics emulate the satellites in the Spire constellation. The simulation scenario provides a realistic case study in a controlled environment where the true density field and true satellite trajectories are known; this allows for more in depth performance analyses than is possible with Spire CubeSat flight data in Chapter 6.

This chapter begins with a description of the scenario, including the simulated geomagnetic storm and satellite characteristics. Next, we analyze the performance of SoleiTool when applied to the simulation scenario. The results are presented for each portion of the algorithm separately, as follows: the Batch Filter (Section 5.2), the PF (Section 5.3), the estimated density (Section 5.4), and a performance evaluation via satellite propagation (Section 5.5). The chapter concludes with a summary of findings.

Two example satellites, a Low-Altitude, Mid-Inclination satellite (LAMI) and a High-Altitude, Polar-Inclination satellite (HAPI), are selected for analysis. The LAMI and HAPI satellites are selected because they are modeled after satellite 62 and satellite 99 in the Spire CubeSat case study, respectively. In addition to the LAMI and HAPI satellites, the Batch filter section also includes an example of a satellite named NOOBNOOB[1] to demonstrate the effect of arc length, ballistic coefficient, and orbit altitude on Batch filter performance.

5.1 Scenario

The simulation scenario takes place during a geomagnetic storm that is modeled after the historical storm that occurred on May 10-14, 2019 (Figure 5.1); during which 119 satellites are simulated orbiting. Calm conditions (Kp=0.6) are simulated leading up to and following the storm to isolate and focus on the single storm. In order to study smaller-scale density features, the 3-hour historical Kp indices are interpolated to create 1-hour Kp indices, and additional artificial small-scale density features, in the form of three elevated 1 to 2-hour Kp values, are added in the first half of May 11th. A historically measured 3-hour elevated Kp feature occuring at the end of May 11th is also included. The historical F10.7 is used with no alterations as the simulated F10.7. The measured F10.7 remains in the calm solar activity range throughout this period of time (solar minimum). These profiles of Kp and F10.7 indices are input to TIE-GCM, run in nominal mode, to generate the simulated density field through which the CubeSats are propagated. Figure 5.2 provides the simulated true global mean density, and shows the small-scale density field enhancements caused by the small-scale Kp features on May 11th.

Only information from the week(s) leading up to the case study would be used to determine the PF fixed F10.7 in operations. For the simulation, a fixed F10.7 of 75 s.f.u. is chosen for the PF portion of SoleiTool in order to have an offset to the true F10.7 behavior, as would likely be the case in operational use.

[1] NOOBNOOB is the space object catalog name of the Spire constellation CubeSat with NORAD ID 47538

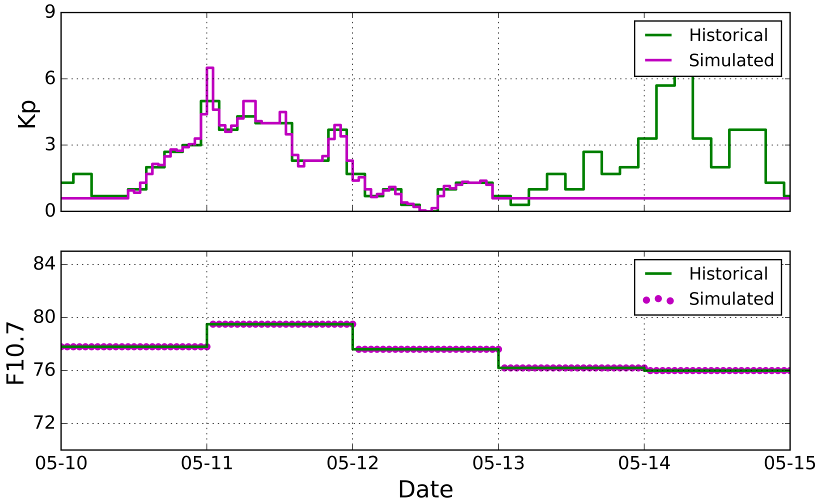

Figure 5.1: Simulated Geomagnetic Storm

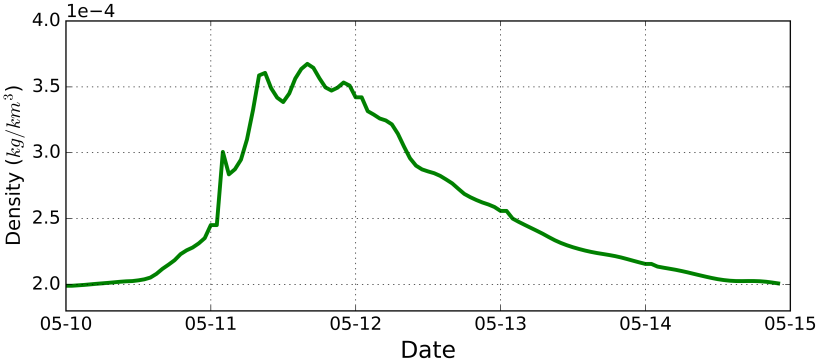

Figure 5.2: Mean true global density

Each of the simulated satellite orbits is modeled after one of the CubeSats in the Spire constellation. All Spire CubeSats have a near circular orbit (e $<$ 0.006). Satellites are simulated to have constant ballistic coefficients ($\beta = \frac{C_D A}{m}$) in the range of 0.04 to 0.07 m^2/kg. This is similar in magnitude to the ballistic coefficients of the actual Spire CubeSats, but omits the complexities of the changing attitude and effective cross section encountered in the flight data. Figures 5.3 - 5.4 provide the number of satellites with each type of altitude and inclination orbit characteristic. Most Spire CubeSats are in near-polar orbits with an altitude above 500 km.

The CubeSats are propagated with acceleration due to drag using the simulated density field, and the following dynamics modeled using Orekit [39]: 140 degree/order Earth gravity; sun and moon third body perturbations; cannonball solar radiation pressure; solid Earth and ocean tides; and relativity. POD measurements are simulated based on the propagated CubeSat trajectories. A time-series of white Gaussian errors with a standard deviation of 1 meter for position and 10^{-4} m/s for velocity are applied to the POD measurements. Various on and off duty-cycles are implemented based on actual Spire CubeSat GPS data characteristics.

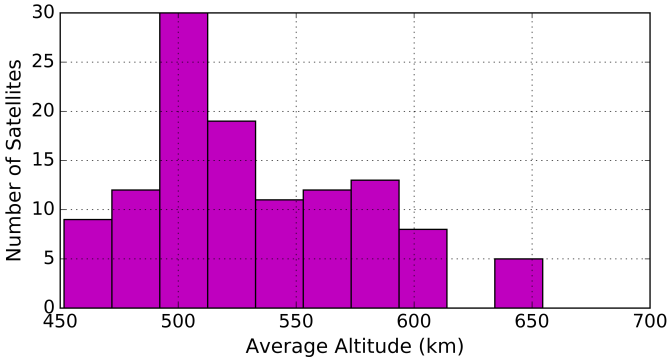

Figure 5.3: Spire Constellation Mean Altitude Histogram

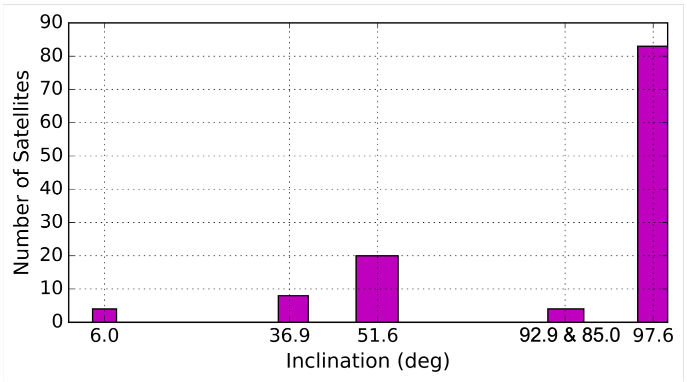

Figure 5.4: Spire Constellation Inclination Histogram

Table 5.1 provides a more detailed description of LAMI, HAPI, and NOOBNOOB, which are the focus for the results. LAMI is in the lowest altitude orbit with the longest arc length. HAPI and NOOBNOOB are in near-polar orbits with about half the arc length of LAMI, and orbit altitudes about 50 km and 80 km higher than LAMI, respectively. As mentioned earlier, LAMI and HAPI are selected for more in depth analysis because they are two of the 10 satellites in the Spire CubeSat flight data case study (Chapter 6); LAMI is satellite 62 and HAPI is satellite 99. Satellite 62 and 99 are the two satellites chosen for in depth analysis in the Spire CubeSat flight data case study because of their disparate orbital characteristics (orbit altitude and inclination). In the simulation scenario, the aim is to investigate the effect of orbit altitude, inclination, ballistic coefficient, and arc length on SoleiTool performance in a controlled environment so that we may apply this knowledge to understand the more complex flight data case study. Flight data from satellites with higher altitudes than HAPI (satellite 99) were not available for the Spire CubeSat flight data case study, so an analysis of simulated NOOBNOOB Batch results is used to further explore the effect of altitude on SoleiTool performance.

Table 5.1: LAMI, HAPI, and NOOBNOOB Satellite Properties

	LAMI Satellite	**HAPI Satellite**	**NOOBNOOB**
Altitude Range (km)	443-451	494-516	511-538
Inclination (deg.)	52	97.5	97.5
Ballistic Coef. (m^2/kg)	0.06	0.05	0.07
On Duty-Cycle (min)	40	20	25

5.1.1 Filter Initialization Details

This section provides the initial conditions applied in the Batch and Particle Filter portions of SoleiTool. The measurement variances for the POD position and velocity measurements are set to $\sigma^2 = (1m)^2$ and $\sigma^2 = (10^{-4}m/s)^2$, respectively. The Batch filter estimated state is

$$\vec{X} = \left[\vec{r}^{\,T},\ \vec{v}^{\,T},\ C_\rho\right]^T \tag{5.1}$$

The initial $\vec{r}$ and $\vec{v}$ estimates are set to the first POD measurements reported for each arc. The initial C_ρ state is set to 1, meaning that a density field generated by the reference values of Kp=3 and F10.7=80 is assumed. An estimated C_ρ with a value other than one implies a density field other than that generated by the reference values in TIE-GCM. The initial Batch state covariance is set to

$$P_0 = diag\left(\left[(10m)^2, (10m)^2, (10m)^2, (0.1m/s)^2, (0.1m/s)^2, (0.1m/s)^2, (1)^2\right]\right) \tag{5.2}$$

In the second filter, the PF, the state is just Kp initialized to

$$\vec{X}_0 = [Kp] = [1.5] \tag{5.3}$$

Note that the true Kp is unknown, so the initial state estimate is set to a Kp that represents calm conditions. Only information from the week(s) leading up to the available CubeSat POD would be used to determine the PF fixed F10.7 in operations. For this simulation scenario, a fixed F10.7 of 75 s.f.u. is chosen for the PF portion of SoleiTool. The initial Kp state covariance is

$$P_0 = \left[(1)^2\right] \tag{5.4}$$

An initial cloud of 50 Kp particles (χ_0) is generated via a uniform distribution defined by the initial state (Eq. 5.3) and its *a priori* statistical information (Eq. 5.4). The PF Kp state is estimated by using C_ρ measurements produced by the Batch filter (for all processed satellites).

5.2 Batch Filter

The Batch filter estimates a density coefficient (C_ρ) for each POD arc for each satellite. Figures 5.5 - 5.7 provide the Batch filter results for our three example satellites. The time-series of estimated density coefficients (top plot) are used to compute the single-satellite-solution estimated density, $C_\rho * \rho_b$, shown in the middle plot (magenta markers). The middle plot also provides the true density along the satellite trajectory (green markers). The C_ρ error and its 3-σ bounds are provided in the bottom plot. A trend in the estimated density coefficient is apparent for all three satellites; the density coefficient is higher during the period of density field enhancement caused by the geomagnetic storm.

The Batch filter estimated density aligns almost perfectly with the truth for LAMI, whereas HAPI has several arcs for which the estimated density does not align with the true density. This is because HAPI is in a higher orbit where the impact of drag on its orbit is lower compared to LAMI, and HAPI has a shorter arc length (20 minutes) than LAMI (40 minutes).

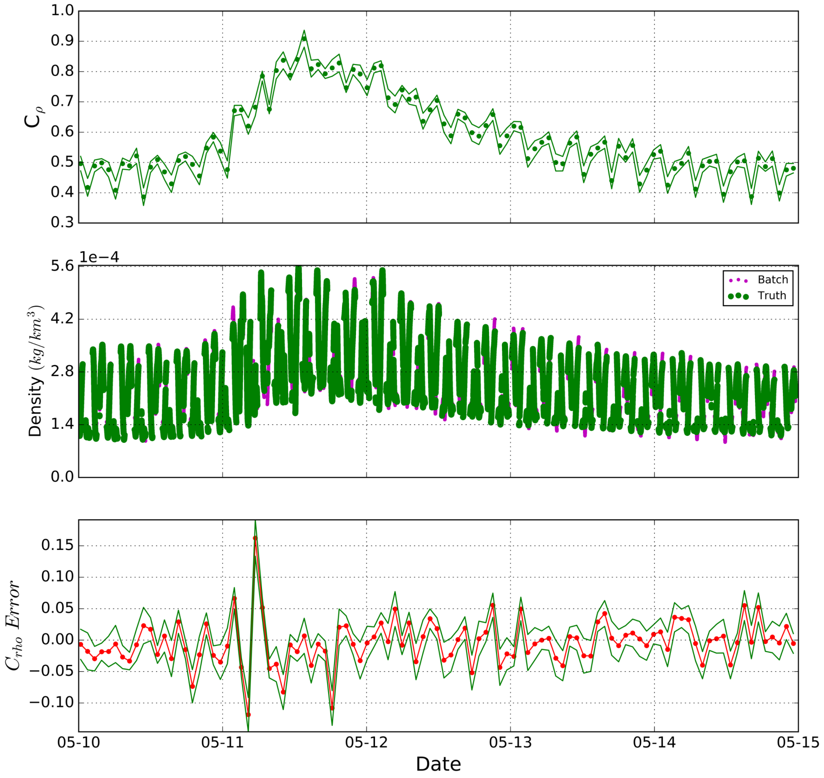

Figure 5.5: LAMI: a) Batch estimated density coefficient and 3-σ bounds; b) True density and Batch estimated density along arcs; c) Density coefficient error and 3-σ bounds

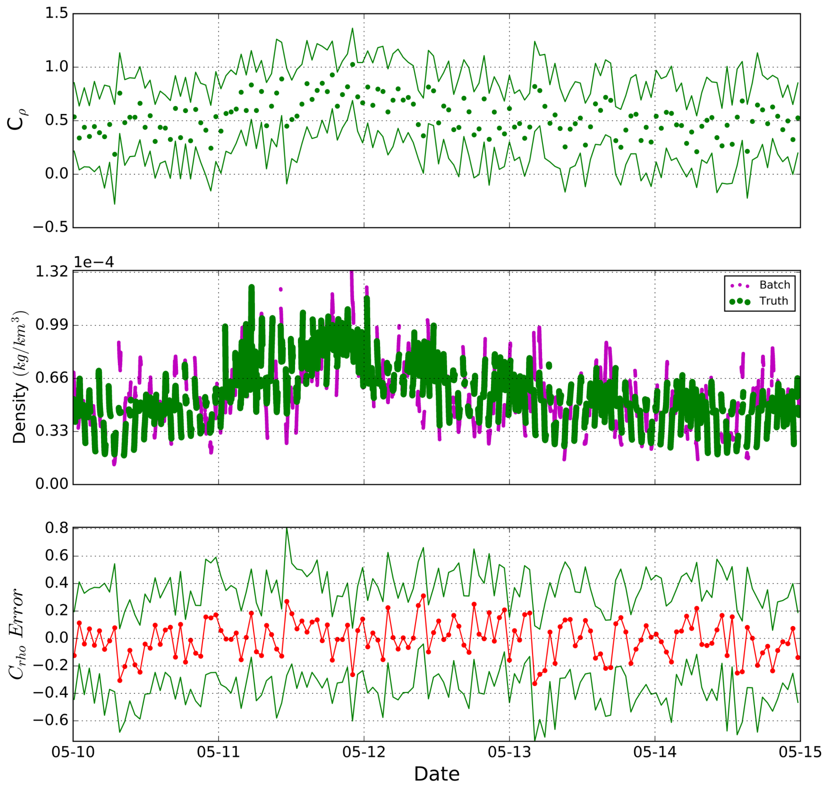

Figure 5.6: HAPI: a) Batch estimated density coefficient and 3-σ bounds; b) True density and Batch estimated density along arcs; c) Density coefficient error and 3-σ bounds

Interestingly, NOOBNOOB has better performance than HAPI even though it is in a higher altitude orbit than HAPI. This is because NOOBNOOB has a slightly longer arc length and a larger ballistic coefficient than HAPI. The drag effect is more likely to be obscured by the measurement noise for satellites with a high altitude, shorter arc length, and/or smaller ballistic coefficients; which leads to less accurate density coefficient estimates and larger 3-σ uncertainty bounds. NOOBNOOB's results show that higher altitude satellites can still provide density information given the combination of a larger ballistic coefficient and longer arc length.

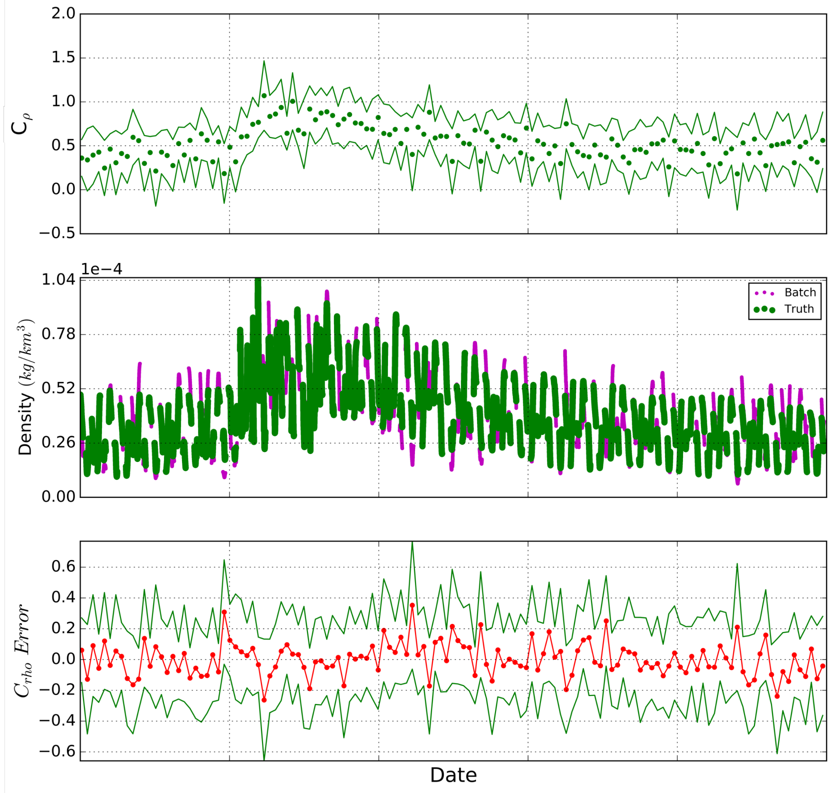

Figure 5.7: NOOBNOOB: a) Batch estimated density coefficient and 3-σ bounds; b) True density and Batch estimated density along arcs; c) Density coefficient error and 3-σ bounds

Table 5.2 provides the mean density coefficient error and the mean 3-σ uncertainty bounds for LAMI, HAPI, and NOOBNOOB. As expected, LAMI has the lowest density coefficient error and 3-σ uncertainty. The combination of a longer arc length and higher ballistic coefficient causes NOOBNOOB to have a slightly lower density coefficient error and 3-σ uncertainty than HAPI.

Table 5.2: LAMI, HAPI, and NOOBNOOB Batch C_ρ Error and 3-σ Uncertainty

	Mean C_ρ Error	**Mean 3-σ Uncertainty**
LAMI	0.03	0.02
HAPI	0.1	0.4
NOOBNOOB	0.08	0.3

Figure 5.8 shows the density coefficient RMS error as a function of its 3-σ uncertainty. Each data point represents the C_ρ RMS error and mean C_ρ 3-σ over the duration of the simulation for a particular satellite in the Spire constellation. As expected, the C_ρ uncertainty is a clear indication of the C_ρ error. Figure 5.9 provides a histogram of all the density coefficient 3-σ uncertainties for each arc of the full constellation; most of the C_ρ estimates have a 3-σ < 0.3.

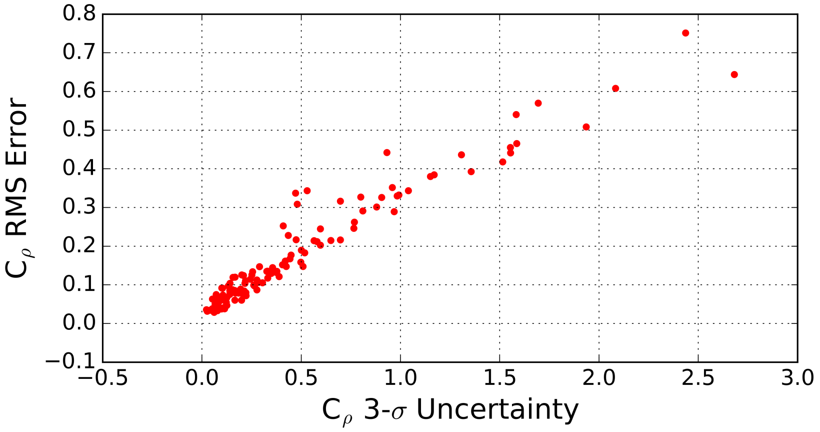

Figure 5.8: Density coefficient RMS error vs. Density coefficient 3-σ uncertainty

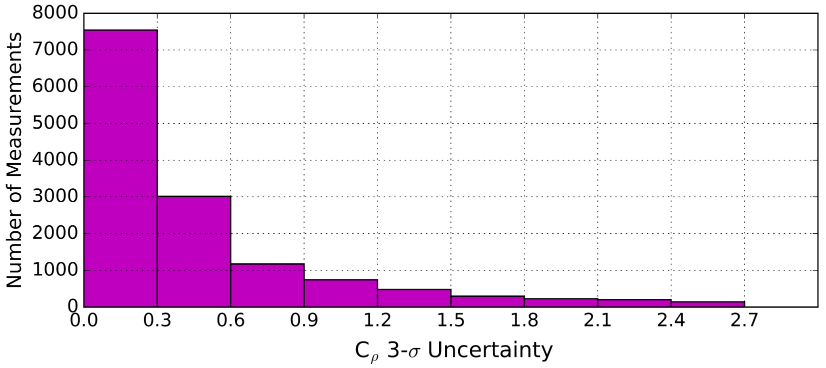

Figure 5.9: Histogram of the density coefficient 3-σ uncertainties for each arc of the full constellation

5.3 Particle Filter

The PF accepts the C_ρ estimates from the full CubeSat constellation processed in the Batch and uses them to estimate a Kp time-series with a 1-hour resolution. In order to filter out incorrect density coefficient estimates, the PF only uses density coefficient measurements with 3-σ < 0.3. This is not done in the Spire CubeSat flight data case study in Chapter 6 due to the limited amount of available satellite data (10 satellites). A finer Kp time-resolution of 1 hour is applied in the simulation scenario compared to the flight data case study, where we used a 1.5-hour resolution, due to the larger number of CubeSats and measurements available for processing.

Figure 5.10 presents the PF results incorporating all density coefficient measurements from the full constellation processed in the Batch. The top plot shows the estimated steady-state Kp and simulated nominal Kp, and the bottom plot shows the 3-σ uncertainty. The gray line shows the simulated Kp index that was used as input to TIE-GCM to generate the density field through which the satellites were simulated orbiting. The magenta line shows the estimated steady-state Kp index. Both the simulated and estimated Kp indices are initially small. As the storm builds, the simulated Kp index increases first, starting mid-way through May 10th, and the estimated Kp increases a little later and more gradually, as the density field responds to the geomagnetic field disturbance. There is a clear peak in the estimated Kp caused by the geomagnetic storm, indicating an enhanced density field. Then the simulated Kp decreases, followed by the estimated Kp slowly tapering off after the geomagnetic storm has ended.

The PF does not appear to have sensed the small-scale density enhancements caused by the simulated Kp small-scale features on May 11th. If the PF had sensed the small-scale density enhancements, we would expect to see an estimated Kp curve very similar to the shape of the true mean global density in Figure 5.2, with small secondary modulations during the storm on May 11th. Uncertainty in the Kp estimate occurs at the beginning of the simulation, and remains relatively low for the remainder of the simulation.

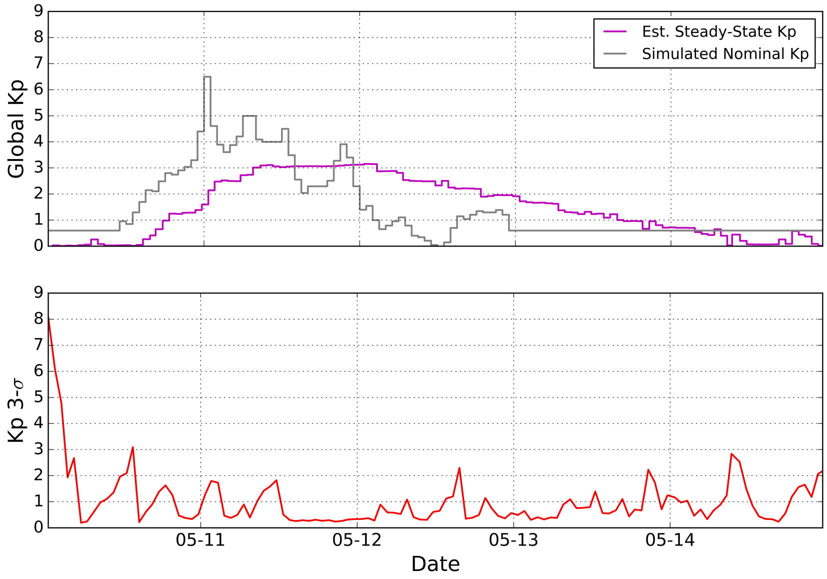

Figure 5.10: Estimated Kp time-series with simulated Kp indices for reference; Estimated Kp 3-σ uncertainty

5.4 SoleiTool Estimated Density

The estimated Kp time-series is used as input to TIE-GCM to obtain the estimated global density field. To evaluate the estimated density, it is directly compared to the truth along the trajectories of LAMI and HAPI. This section also includes an analysis of the estimated global density field.

Figures 5.11 and 5.12 present the estimated density along the trajectory of LAMI. The estimated density is obtained by using the estimated steady-state Kp time-series and fixed F10.7 as input to TIE-GCM in steady-state mode. Each figure also provides the Batch single-satellite-solution density (Eq. 4.3), computed using only that satellite's POD, and the true density through which the satellites were simulated orbiting. The Batch density is available only during the POD measurement arcs, whereas the estimated density and true density is available at all times. The density along the trajectory is initially small and then increases as the storm builds, before tapering off after the conclusion of the storm. The higher frequency sinusoidal density behavior is caused by the satellite moving through areas of low and high density corresponding to near local noon and near local midnight, respectively. The density along HAPI's trajectory is more dynamic because of its polar orbit. Overall, the estimated density aligns well with the true density along both LAMI and HAPI's trajectory. However, some small-scale features of higher and lower density during the storm are not captured in the estimated density.

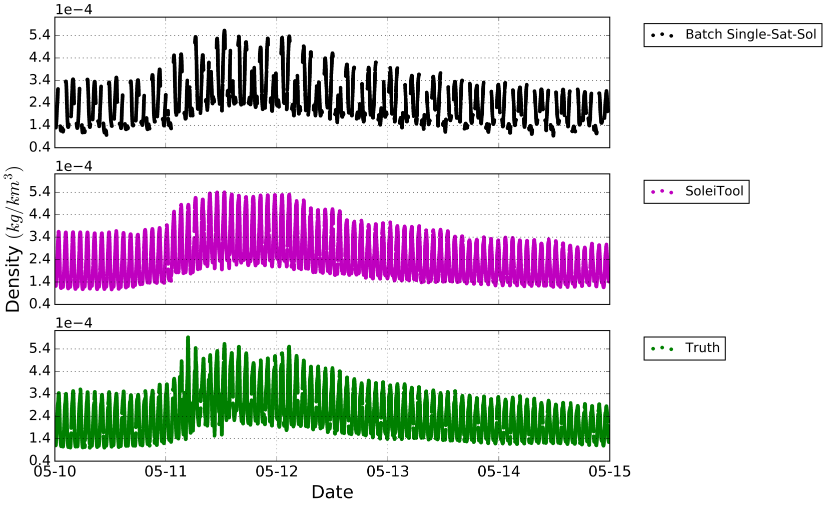

Figure 5.11: Estimated Density along LAMI's trajectory compared to Batch single-satellite solution and truth

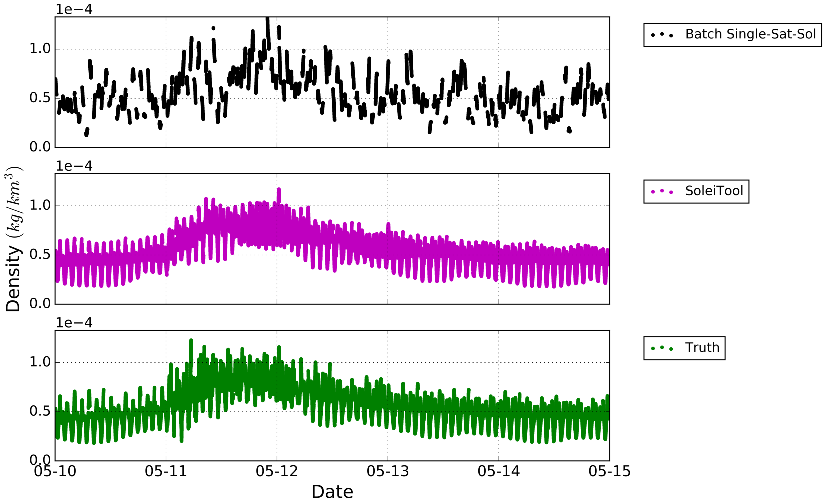

Figure 5.12: Estimated Density along HAPI's trajectory compared to Batch single-satellite solution and truth

Next, Figure 5.13 shows the density error along LAMI and HAPI's trajectories. As expected, LAMI has a larger density error magnitude compared to HAPI due to LAMI's lower altitude orbit. The positive and negative density error peaks occurring on May 11th are caused by SoleiTool underestimating and overestimating the density, respectively, as the storm builds.

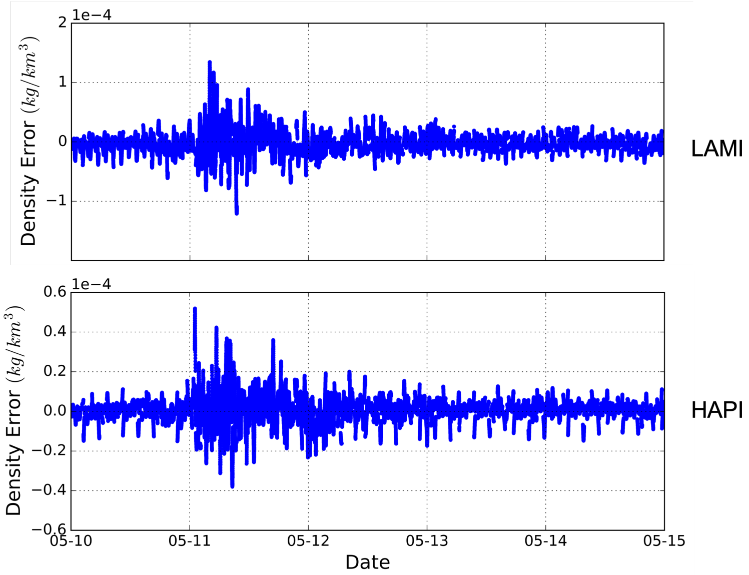

Figure 5.13: Density Error along LAMI and HAPI trajectories

Now we also quantify the performance of the estimated density by comparing it to the true density via the metrics presented in [64] and [62]. These metrics include $\mu\,(m/o)$, $\sigma\,(m/o)$, and $RMSe\,(m/o)$, all in log space:

$$\mu\,(m/o) = \exp\left(\frac{1}{N}\sum_{i=1}^{N}\ln\frac{\rho_{m,i}}{\rho_{o,i}}\right) \tag{5.5}$$

$$\sigma\,(m/o) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\ln\frac{\rho_{m,i}}{\rho_{o,i}} - \ln\mu\,(m/o)\right)^2} \tag{5.6}$$

$$RMSe\,(m/o) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\ln\frac{\rho_{m,i}}{\rho_{o,i}}\right)^2} \tag{5.7}$$

where “m” is the *model* density and “o” is the *observed* density, or truth density in this case. The $\sigma\,(m/o)$ and $RMSe\,(m/o)$ quantities are best interpreted as percentages, using:

$$\% = 100 \times (\exp\sigma\,(m/o) - 1) \tag{5.8}$$

$RMSe\,(m/o)$ is a good indicator of total model errors [64]. These metrics are provided for the estimated density along the trajectory of LAMI and HAPI in Table 5.3. LAMI has slightly lower error than HAPI, and it is clear that the error is due to the standard deviation of the estimated density compared to the truth, not a bias in the mean. Overall, both satellites have good agreement between the estimated and true density, indicated by $RMSe\,(m/o) \leq 11\%$.

Table 5.3: Performance Metrics of SoleiTool Density with respect to True Density along LAMI and HAPI's Trajectories

Model	$\mu\,(m/o)$	$\sigma\,(m/o)$	$RMSe\,(m/o)$
LAMI	1.0	7%	8%
HAPI	1.0	11%	11%

Next, we present a more comprehensive evaluation of estimated density through an analysis of the global density field. Figure 5.14 shows the estimated global density field at 450 km altitude for two date/times in the top plots and the bottom plots show the estimated density error (difference between true and estimated density). The dates presented are May 11th at 12:00, during the peak of the storm, and 00:00 on May 13th, during the calm conditions after the storm. The same density range, from 1.8e-4 to 5e-4 kg/km^3, is used for both top plots, and the density error magnitude in the bottom plots is several times smaller with a range of ± 1.4e-4 kg/km^3. As apparent in the top plots, the highest magnitude of thermospheric density typically occurs near local noon and the equator, while relatively lower density occurs near local midnight and the South Pole for this scenario. The North Pole has a higher density than the South Pole in this scenario because the northern hemisphere is experiencing the latter part of spring (near summer). The May 11th density field (top left) is more enhanced than May 13th (top right). This is expected because May 11th is during the peak of the storm, and May 13th is during the calm conditions. The density error follows the same trend, with a higher density error in the bottom left plot during storm conditions than the calm conditions in the bottom right plot.

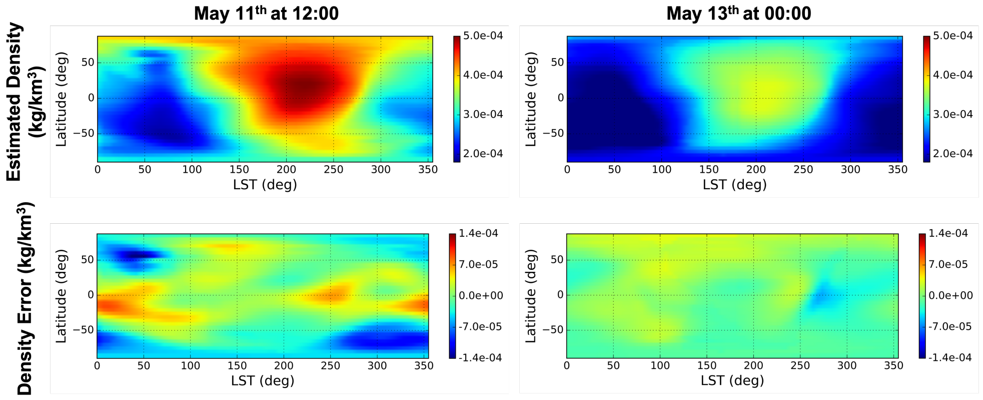

Figure 5.14: Estimated Global Density; Global Density Error

Figure 5.15 provides a quantification of the estimated *global* density performance. This is accomplished by computing the performance metrics (Eqs. 5.5 - 5.7) for the global density (instead of just along a satellite trajectory) with a time-step of one hour. The mean global density is also provided in green for reference to the storm. The density response to the storm begins early May 11th and peaks the latter half of May 11th. Interestingly, the largest error occurs during the beginning of the storm as the storm builds, and the error tapers down to 10% and below once the peak of the storm occurs. This shows that SoleiTool has more difficulty estimating the build of a storm compared to the storm peak, storm decline, and calm conditions; which is likely driven by the steady-state nature of the background density model and the fixed F10.7 applied in SoleiTool. Overall, the RMSe reaches a maximum of 25%, and remains below 10% during the calm conditions. It is apparent that the RMSe is dominated by $\sigma\%$, the distribution of the estimated density compared to the true distribution; whereas, μ remains within 10% of 1.0 (no mean bias), even during storm-time.

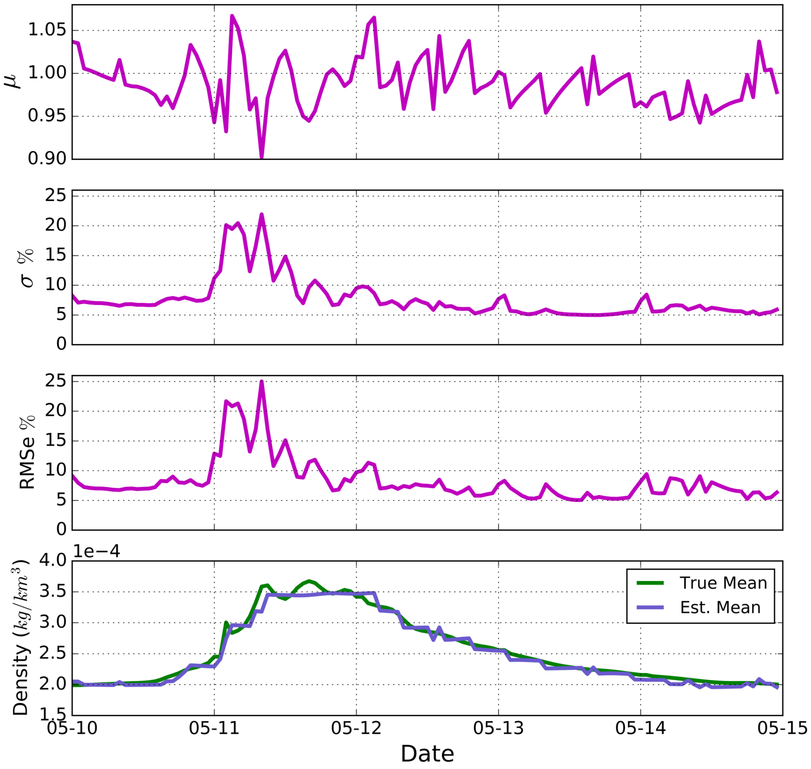

Figure 5.15: Global density error metrics; True and estimated mean global density

5.5 Satellite Propagation Analysis

The desired end result of SoleiTool is improved thermospheric density modeling to allow for more accurate LEO object trajectory prediction. To evaluate this, satellites are propagated with the estimated density field generated using the estimated Kp and fixed F10.7 in TIE-GCM. Each propagation is initialized with the true ephemeris initial conditions. This is done for each arc, separately. Then the position difference between the propagation and true ephemeris at the end of each arc is computed. A smaller position difference indicates a better-calibrated density model. Each trajectory is propagated using the same dynamical models in Orekit as the true trajectory; the only difference is the underlying density model.

To evaluate each model's performance during key times throughout the simulation scenario, we compute the average difference between model propagation and true ephemeris (i.e., error) throughout each time period indicated in Figure 5.16 (true density provides reference for time-region definitions). This demonstrates how well each model performs during the storm peak verses the calm conditions before and after, for example. Figure 5.17 shows the mean end of arc error across a time-region for each model for LAMI and HAPI.

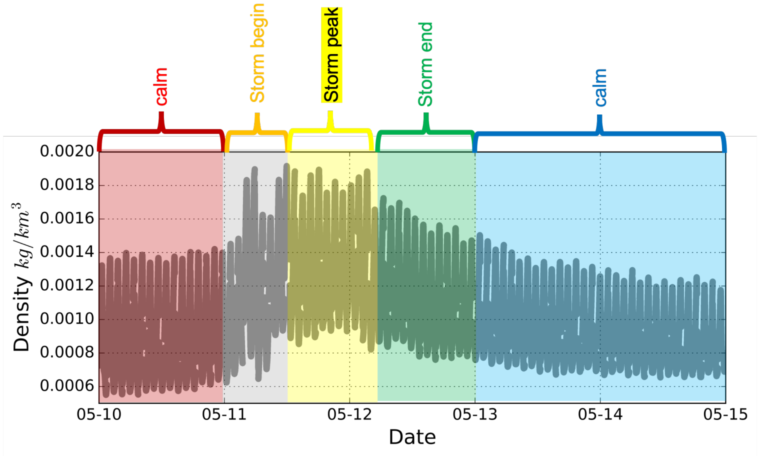

Figure 5.16: Satellite propagation analysis time region definitions; True Density density (gray) provides reference for time-region definitions

The propagation results for LAMI and HAPI show that the mean 20-minute arc propagation error with SoleiTool is always less than 3 centimeters. Overall, HAPI has a lower propagation error than LAMI. This is because HAPI is in a higher altitude orbit, and is therefore less affected by drag.

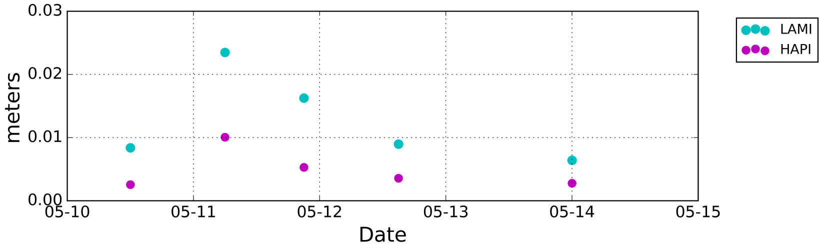

Figure 5.17: LAMI and HAPI Model Propagation Position Error

5.6 Chapter Conclusion

This Spire CubeSat constellation simulation scenario demonstrates SoleiTool's ability to extract local density information from as little as a 20-minute arc, and use several arcs from several satellites to estimate a global density field. POD position and velocity measurements are simulated by propagating satellites through a simulated density field during a geomagnetic storm. The Batch filter estimates a density coefficient per 20 to 40-minute arc of POD measurements. Then, these density coefficients for all arcs and all satellites are passed to the PF to estimate a Kp time-series. The estimated Kp time-series and fixed F10.7 are input to TIE-GCM in steady-state mode to obtain a global density field. To evaluate SoleiTool performance, the estimated density field is compared to the true density field. A satellite propagation analysis is also executed in which satellites are propagated using SoleiTool density, and the difference between the propagated trajectory is compared directly to the true satellite trajectory.

In the evaluation of the Batch filter results, the single-satellite-solution estimated density appears to align almost perfectly with the truth for the lower altitude satellite, but accuracy declines for satellites with higher altitudes, lower ballistic coefficients, and/or shorter arc-lengths. Satellites with higher altitudes require a longer arc of data and/or a higher ballistic coefficient to extract density information.

The PF estimates a Kp time-series that aligns well with the expected density field response to the simulated geomagnetic storm. Furthermore, the estimated density is very similar to the true density; it matches the overall shape of the true density, however, the estimated density is smoother and does not capture the small-scale density enhancements. The metrics that quantify the agreement between the estimated density and true density along LAMI and HAPI's trajectories reveal zero mean bias ($\mu\,(m/o) = 1.0$) and a reasonable overall error of $RMSe\,(m/o) \leq 11\%$. The *global* density performance metrics showed that the overall error, RMSe, is higher during the beginning of the storm and tapers off by the storm peak. Lastly, the propagation analysis shows that all 20-minute propagations during the simulation have an error less than 3 centimeters.

This simulation scenario was developed based on experience working with the 10-satellite Spire CubeSat flight data case study in Chapter 6. The aim of the simulation scenario is to investigate the effect of orbit altitude, inclination, ballistic coefficient, and arc length on SoleiTool performance in a controlled environment so that we may apply this knowledge to understand the more complex flight data case study in the next chapter, for which there is a time-varying ballistic coefficient and non-constant arc length, among other complexities (i.e., anomalous data).

Chapter 6

Case Study with Spire CubeSat Data

This chapter evaluates the performance of SoleiTool when applied to a 10-satellite Spire CubeSat flight data case study during two historical geomagnetic storms in May 2019. The aim is to demonstrate SoleiTool's ability to estimate the storm-time density field using imperfect and more complex CubeSat POD flight data (time-varying ballistic coefficients, non-constant arc lengths, etc.). We apply what was learned from the simulation scenario in order to understand the SoleiTool results when the true density field and CubeSat orbits are unknown.

6.1 Spire Data

Section 2.5 described prior methods of using GPS information to derive non-gravitational accelerations and density directly along satellite trajectories [42, 37, 12, 69]. SoleiTool aims to assimilate information from multiple satellites to estimate *global* density. The growing number and size of LEO CubeSat constellations offer a unique opportunity to utilize accurate GPS information from hundreds of CubeSats for improved thermospheric density modeling. In this work, Spire Global CubeSat data are used to demonstrate SoleiTool's capabilities using only ten satellites.

Spire Global has a growing 3U CubeSat constellation of 100+ CubeSats in the 400 to 600 km altitude range. Spire CubeSats carry dual-frequency GPS receivers that provide precise position information at 1Hz [50]. GPS information is only available for portions of the orbit corresponding to the active duty cycle of each satellite. Depending on the Spire CubeSat, the active duty-cycle spans 40 to 70 minutes. Ten minutes at the beginning and end of a continuous arc of POD are trimmed due to POD solution degradation near the edges of the fit. The remaining data are referred to hereafter as an arc of data. Arcs greater than 40 minutes are split in half to create two arcs. Anomalous arcs of data with unrealistic altitudes are removed, accounting for less than 10% of the original data. During the period of the 11-day case study, there are a total of 1,189 arcs for the ten CubeSats, combined; this results in 1,189 density coefficient (C_ρ) Batch estimates (one per arc) that are passed to the PF.

Table 6.1 provides a description of the satellite orbits, and Figure 6.1 shows the satellite orbits, generated with STK [2]. Spire's internal satellite ID numbers are used to reference the satellites throughout this chapter. Satellite orbit altitudes range from 426 to 520 km. Six of the ten satellites are in a Sun-Synchronous Orbit (SSO). All SSO satellites have a similar Local Time of Descending Node (LTDN), between 09:26-10:10 in LTDN. In fact, satellites 99, 100, 101, and 102 are in the same SSO, separated by 300-600 km. A total of seven satellites are in near-polar orbits (six SSO satellites and satellite 86). The three lower inclination orbits have lower altitudes compared to the near-polar orbiting satellites.

Table 6.1: Spire CubeSat Properties

Spire ID	**NORAD ID**	**Lemur Version**	**Altitude (km)**	**Inclination***	**Mass (kg)**
46	42779	3.0	480-510	SSO (LTDN 9:26)	4.45
62	43048	3.1	426-443	51.6 deg.	4.45
84	43559	3.3	472-484	51.6 deg.	4.93
85	43558	3.3	472-484	51.6 deg.	4.93
86	43695	3.3	490-520	83 deg.	4.93
90	43745	3.3	460-510	SSO (LTDN 10:10)	4.93
99	44084	3.4	483-520	SSO (LTDN 09:30)	5.10
100	44085	3.4	483-520	SSO (LTDN 09:30)	5.10
101	44086	3.4	483-520	SSO (LTDN 09:30)	5.10
102	44087	3.4	483-520	SSO (LTDN 09:30)	5.10

*SSO satellites are in a near polar orbit. LTDN provides information about the orbit geometry with respect to the Sun.

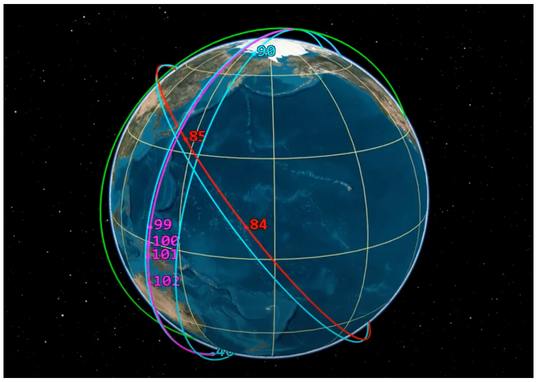

Figure 6.1: Spire CubeSat Orbits

In addition to Spire's POD ephemeris time-series, satellite attitude information and body models are also used to compute necessary time-varying physical characteristics that affect drag perturbations. Spire logs and downlinks GPS pseudorange and phase measurements, and produces ECEF POD using RTOrb [23], a software that uses a Kalman filter-based approach to estimate orbit ephemeris. We transform ECEF POD to ECI using a standard, high-precision transformation in Orekit [39]. Spacecraft attitude information is used to compute the CubeSat projected area facing into the velocity vector ("A" in Eq. 4.2). Attitude information is only available during active arcs of data collection, concurrent with GPS data.

Each satellite's mass and physical characteristics are known. The satellites used for this effort are one of four versions of the Lemur CubeSat; versions 3.0, 3.1, 3.3, and 3.4 of which 3.0 and 3.1 are physically equivalent. The physical characteristics, combined with attitude information, are required to properly model the ballistic coefficient portion of the acceleration due to drag. Table 6.1 identifies the Spire CubeSat version. Version 3.3 is bigger than version 3.1 by an inch (added to the total solar panel width on each side of spacecraft). Similarly, version 3.4 is bigger than version 3.3 with an additional solar panel added to each side of the spacecraft body, for a total of 6 solar panels instead 4 (Figure 6.2).

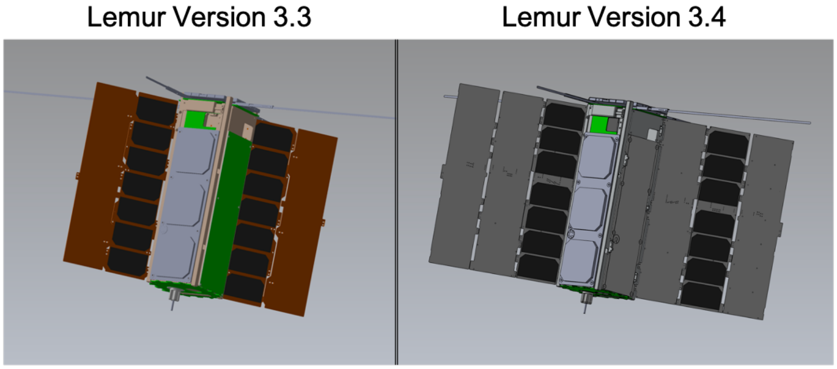

Figure 6.2: Spire CubeSat Lemur Versions 3.3 and 3.4; Courtesy of Spire Global

6.2 Case Study

We demonstrate SoleiTool's ability to estimate the dynamical behavior of the space environment by evaluating its performance in a case study using CubeSat position and velocity POD collected during an 11-day time period in which two geomagnetic storms occur. By estimating a series of Kp TIE-GCM drivers, a thermospheric density model that reflects the current state of the space environment is achieved.

The case study covers May 10 - 21, 2019. This date range is selected because minor (Kp = 5) and moderate (Kp = 6) geomagnetic storms occurred on May 10 - 11 and 14, respectively (Figure 6.3) [14]. Typically, a quiet or calm space environment is described as having both a low Kp (Kp $\leq$ 2) and F10.7 (F10.7 $\leq$ 80) [65]. The measured F10.7 remains in the calm solar activity range (70 - 80 s.f.u.) throughout this period of time (solar minimum).

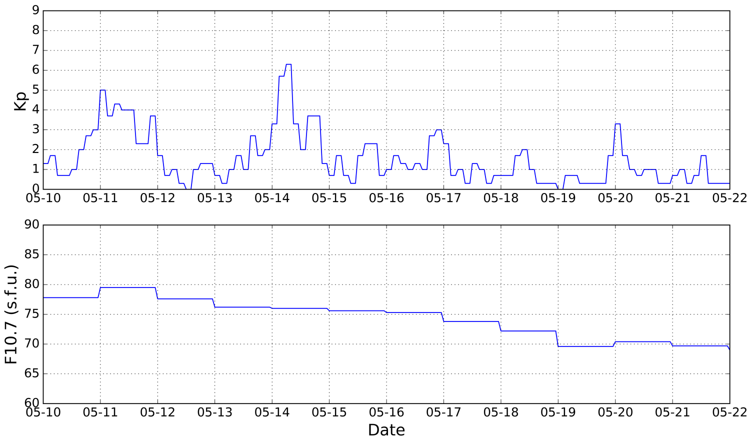

Figure 6.3: Historical Kp and F10.7 for May 10- 21, 2019

Only historical information from the week(s) leading up to the case study is used to determine the PF fixed F10.7, as could be done operationally (Figure 6.4). A fixed F10.7 of 80 s.f.u. is chosen for the PF portion of SoleiTool because it is the closest F10.7 to the historical data that is available in the TIE-GCM runs executed to populate the Kp-F10.7 state-space prior to filter execution.

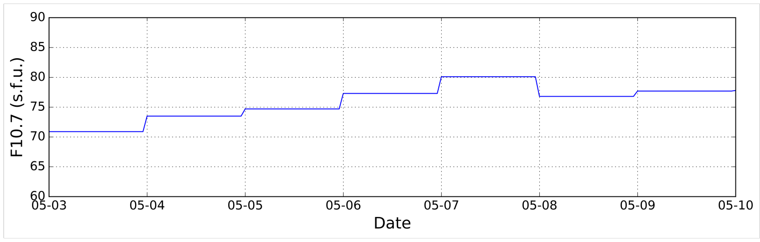

Figure 6.4: Historical F10.7 for May 3-9, 2019

6.2.1 Filter Initialization Details

This section provides the initial conditions applied in the Batch and Particle Filter portions of SoleiTool. The measurement variances for the POD position and velocity measurements are set to $\sigma^2 = (1m)^2$ and $\sigma^2 = (10^{-4} m/s)^2$, respectively. The Batch filter estimated state is

$$\vec{X} = \left[\vec{r}^{\,T},\ \vec{v}^{\,T},\ C_\rho\right]^T \tag{6.1}$$

The initial $\vec{r}$ and $\vec{v}$ estimates are set to the first POD measurements reported for each arc. The initial C_ρ state is set to 1, meaning that a density field generated by the reference values of Kp=3 and F10.7=80 is assumed. An estimated C_ρ with a value other than one implies a density field other than that generated by the reference values in TIE-GCM. The initial Batch state covariance is set to

$$P_0 = diag\left(\left[(10m)^2, (10m)^2, (10m)^2, (0.1m/s)^2, (0.1m/s)^2, (0.1m/s)^2, (1)^2\right]\right) \tag{6.2}$$

In the second filter, the PF, the state is just Kp initialized to

$$\vec{X}_0 = [Kp] = [1.5] \tag{6.3}$$

with a fixed F10.7 of 80 s.f.u. Note that the true Kp is unknown, so the initial state estimate is set to a Kp that represents calm conditions. The initial state covariance is

$$P_0 = \left[(1)^2\right] \tag{6.4}$$

An initial cloud of 50 Kp particles (χ_0) is generated via a uniform distribution defined by the initial state (Eq. 6.3) and its *a priori* statistical information (Eq. 6.4). The PF Kp state is estimated using C_ρ measurements produced by the Batch filter (for all processed satellites).

6.3 Results

This section provides the Batch, PF, and estimated density results, as well as a propagation analysis. First, the Batch results are discussed in the section below (Section 6.3.1) with two specific satellite examples. Then, the PF reults are analyzed via the estimated Kp time-series in plot form (Section 6.3.2). In Section 6.3.3, the method is further investigated via the estimated density (TIE-GCM density using estimated Kp and fixed F10.7 input parameters) along two Spire satellite trajectories. We then also evaluate SoleiTool's performance by comparing its estimated density to an independent data set of observed densities along a Swarm satellite trajectory. Lastly, a comparison between the satellite POD and the propagated trajectory of a particular CubeSat with several different density profiles is presented in Section 6.3.4.

6.3.1 Batch Filter

The Batch filter uses each arc of CubeSat POD measurements to estimate an independent density coefficient (Eq. 4.2). This section provides the Batch results for two of the 10 Spire satellites, satellite 62 and satellite 99. Satellite 62 is in a 51.6 deg inclination orbit and satellite 99 is in a polar orbit. The arc length for these satellites range from 25 to 39 minutes for satellite 62, and 30 to 39 minutes for satellite 99. Satellite 62 and 99 have the same orbital characteristics as the LAMI and HAPI satellites in the simulation scenario (Chapter 5).

Figure 6.5 provides the Batch results for satellite 62 along its trajectory. Satellite 62 is in the lowest altitude orbit of the 10 satellites (about 435 km altitude), with a 51.6 deg inclination. A total of 88 arcs are used to estimate 88 density coefficients. The time-series of estimated density coefficients (Figure 6.5a) are used to compute the single-satellite-solution estimated density, $C_\rho * \rho_b$, shown in Figure 6.5b. HASDM density is provided in gray alongside the single-satellite-solution density for comparison. The historical Kp index is provided in Figure 6.5c for reference to show the onset of the measured geomagnetic storms. The single-satellite-solution density indicates two geomagnetic storms and aligns well with HASDM density (Figure 6.5b). The estimated density coefficients in 6.5a) show a clear first peak; whereas, the second peak contains negative and low density coefficient estimates, and is therefore not as apparent as the first peak. There is also a large outlier that occurs between the two storms, on May 13th. The probable cause of this outlier is discussed below in the Satellite 99 results, for which there are several outliers.

As mentioned in Section 4.1, the Batch filter occasionally estimates small negative density coefficients, which are replaced by 0.001 before being passed to the PF. These negative estimates are shown in the estimated density coefficient plots (Figures 6.5 and 6.6). Any large negative C_{rho} estimates are considered outliers and are removed from the dataset. The estimated density plots provide the density computed using corrected density coefficients (small negative C_ρ estimates replaced with 0.001).

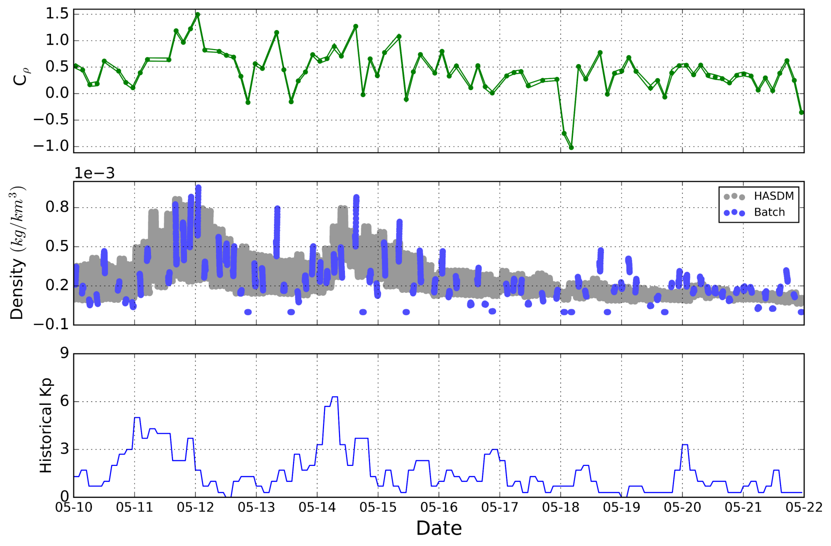

Figure 6.5: Satellite 62: a) Batch estimated density coefficient and 3-σ bounds; b) Batch estimated density along arcs and HASDM density; c) Historically Measured Kp index

The Batch results for satellite 99 are provided in Figure 6.6. Satellite 99 is in a near-polar orbit with the highest altitude of the 10 satellites (about 500 km altitude). Unlike satellite 62, there is no apparent trend in the estimated density coefficients. In this case, the single-satellite-solution density (Batch density, Figure 6.6b) has some outlier structures that do not align with HASDM, but has overall higher density during storm time compared to calm conditions. As discovered in the simulation scenario (Chapter 5), the larger number of Batch density outlier structures of satellite 99 compared to satellite 62 is primarily due to the higher altitude of satellite 99 (about 60 km higher). The effect of density on a satellite arc is less at higher altitudes, therefore making the density coefficient less observable. Other factors that may contribute to density outliers are the arc length, ballistic coefficient, and LST; all of which affect the magnitude of the drag perturbation on the satellite arc. The satellite 62 large outlier that occurs on May 13th is likely caused by a combination of these factors. A possible solution is to filter the arcs based on information content, as is done in the simulation scenario; however, due to the lack of available CubeSat data in the flight data case study, all Batch arcs are utilized in the PF.

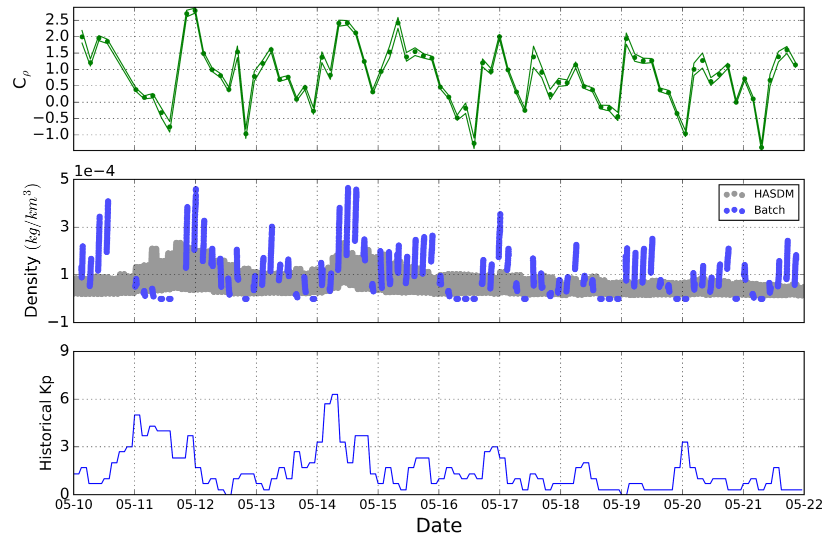

Figure 6.6: Satellite 99: a) Batch estimated density coefficient and 3-σ bounds; b) Batch estimated density along arcs and HASDM density; c) Historically Measured Kp index

Table 6.2 provides the mean 3-σ uncertainty bounds for satellite 62 and 99. As expected, satellite 62 has a smaller density coefficient 3-σ uncertainty due to its lower altitude than satellite 99. These results align with the simulation scenario results for LAMI and HAPI, modeled after satellite 62 and 99, respectively.

Table 6.2: Satellite 62 and 99 Batch C_ρ 3-σ Uncertainty

	Mean 3-σ Uncertainty
Satellite 62	0.03
Satellite 99	0.1

6.3.2 Particle Filter

The PF accepts the C_ρ estimates from each CubeSat arc processed in the Batch filter and uses them as measurements to estimate a Kp time-series. The case study applies a slightly larger PF resolution of 1.5 hours compared to the 1-hour resolution in the simulation scenario. A longer time window of C_ρ measurements is necessary in order to help compensate for the smaller number of satellites available in the Spire CubeSat case study. Nevertheless, an average of 6 density coefficient measurements are available for each state update, compared to 60 in the simulation scenario.

Figure 6.7 shows the PF estimated Kp and its 3-σ uncertainty. The historically measured Kp index for the days leading up to the case study is also included to provide context for the smaller estimated Kp peak the beginning of May 10. The estimated Kp shows one clear estimated Kp peak aligning with the second half of the first geomagnetic storm. However, the estimated Kp during the second geomagnetic storm is more noisy. One can observe the expected delay in the estimated Kp peak compared to the historical Kp due to its steady-state nature.

The highest Kp uncertainties occur during storm-time. It is evident that the uncertainties for the flight data case study are higher and more variable than the simulation scenario. This is likely because there are significantly fewer CubeSats in the flight data case study (10 satellites) compared to the simulation (119 satellites), and a large portion of the CubeSats have several density coefficient outliers due to their relatively high altitude. Another factor is that the PF measurement filtering based on density coefficient uncertainty that is done in the simulation scenario cannot be applied in the flight data case study due to the lack of available data.

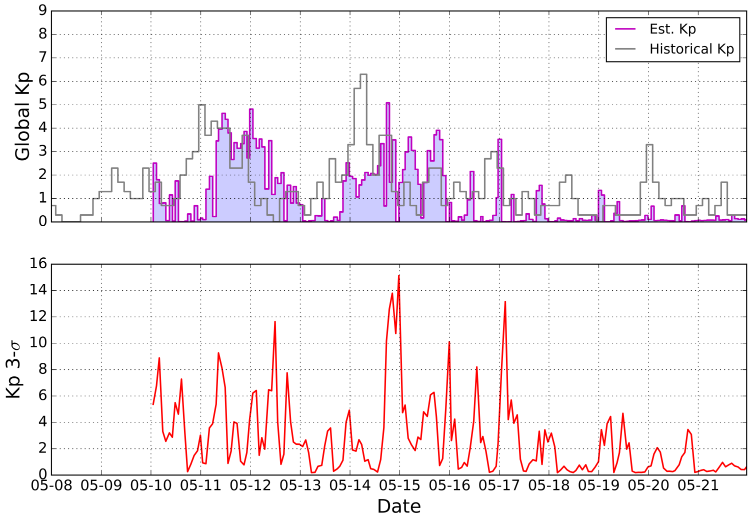

Figure 6.7: Estimated Kp time-series with historically measured indices for reference; Estimated Kp 3-σ uncertainty

6.3.3 SoleiTool Estimated Density

The final estimated density is obtained by using the estimated Kp and fixed F10.7 time-series as input to TIE-GCM. The *a priori* TIE-GCM runs described in Section 2.3.2 are utilized again here for consistency. This section first explores the estimated density along the trajectory of a Spire CubeSat by comparing the estimated density to other models (e.g., HASDM). Next, we evaluate the performance of the estimated density by comparing it to an independent data set: the Swarm-derived density along the trajectory of the Swarm-C satellite. Both cases are examined visually with density plots and quantified with error metrics.

The estimated density along the trajectory of satellite 62 is provided in Figure 6.8. This figure also shows the Batch single-satellite-solution density (Eq. 4.3, computed using only satellite 62's POD), HASDM [66], historical TIE-GCM, and MSIS [52] density for comparison. MSIS and historical TIE-GCM density is generated using the historically measured Kp and F10.7 indices.

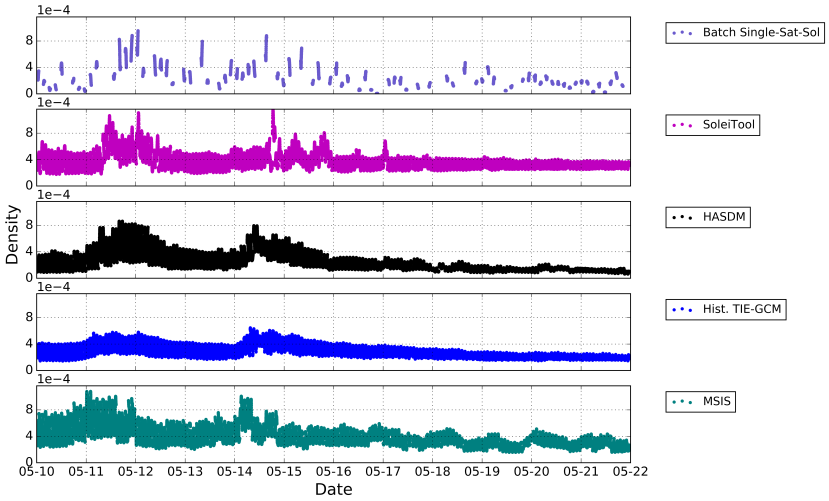

Figure 6.8: Estimated Density Along Satellite 62 Trajectory Compared to Batch Single-Satellite Solution, HASDM, Historical TIE-GCM, and MSIS

The density trend for all models is similar; all models have two periods of density enhancement that correlate with the two geomagnetic storms. The historical TIE-GCM density appears to be attenuated compared to other models; with the first storm peak of half the magnitude compared to other models. SoleiTool is able to sense some smaller-scale features demonstrated via the more dynamic behavior of the storm-time density field enhancements compared to other models. This is partly due to the finer time resolution of SoleiTool, but may also be driven by the density coefficient outliers present in the high altitude satellite Batch results.

Notice that the Batch Single-Satellite-Solution enhanced density occurring on May, 13 is not reflected in the SoleiTool Multi-Satellite-Solutions. This is either because of a measurement anomaly or that the region of enhancement is too small to be captured by our limited data set. Observations from more CubeSats with various orbital characteristics would allow the method to effectively sense the LEO density field with adequate temporal and spatial resolution to answer such questions.

Next, we turn our attention to comparing the estimated density to the density derived by the Swarm mission along the Swarm-C satellite trajectory. The purpose of this analysis is to demonstrate SoleiTool's performance at locations far away from the Spire CubeSats used for assimilation. As mentioned in Section 2.5, after arriving in orbit, the accelerometers onboard the three Swarm satellites were found to have erroneous performance. Therefore, a method was developed by van den IJssel, et al. [2020] to determine thermospheric density by applying corrections to the accelerometer data using POD information. These corrections have a temporal resolution of about 20 minutes. The Swarm-C satellite is in a near-polar orbit with an altitude ranging 424-445 km during our case study. Any negative, non-physical density values are removed from the Swarm-C data set before comparing it to SoleiTool and other models.

A visual comparison between the Swarm-C derived density and the estimated density, as well as other models, is shown in Figure 6.9. The model storm peaks align well with the Swarm-C derived density peaks, except for the SoleiTool second storm peak, which is delayed compared to the others. There are also a couple of interesting small-scale features present in the Swarm-C derived density on May 14, during the second storm, indicated by the green arrows. A zoomed in version of Figure 6.9 is provided in Figure 6.10 to allow for a more in-depth analysis of these particular features. The first small-scale Swarm-C feature (first green arrow, near 06:00) is not captured by SoleiTool, HASDM or historical TIE-GCM, and MSIS over-estimates density during that time. The second small-scale feature (second green arrow, near 18:00) is captured by HASDM and is slightly diluted in the historical TIE-GCM model. SoleiTool, on the other hand, senses an isolated increase in density but is larger than the Swarm-derived density peak. Lastly, the Swarm-derived density shows nocturnal peaks that occur over the equator, indicated by the orange arrows, caused by the Midnight Density Maximum (MDM) as discussed by Arduini et al. [3] and Akmaev et al. [1]. The MDM is not captured by HASDM, but is visible in TIE-GCM-based models, including SoleiTool.

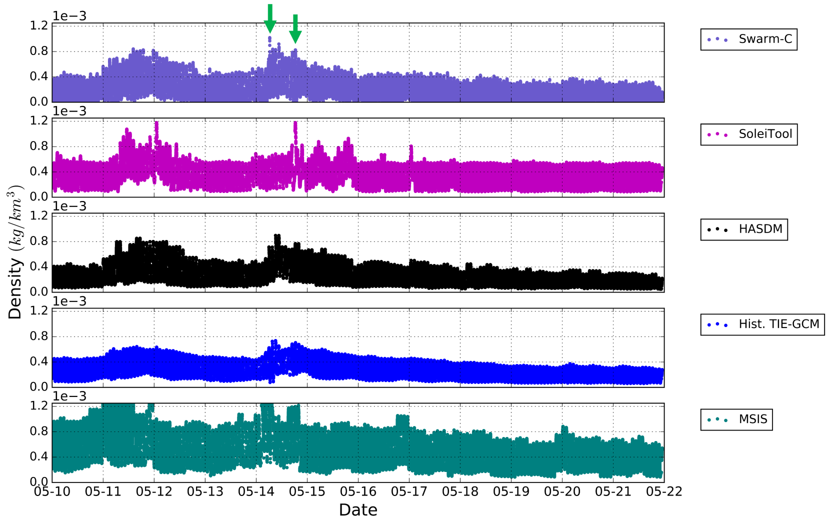

Figure 6.9: Comparison between Swarm-C derived density and SoleiTool, HASDM, Historical TIE-GCM, and MSIS Along Swarm-C Trajectory

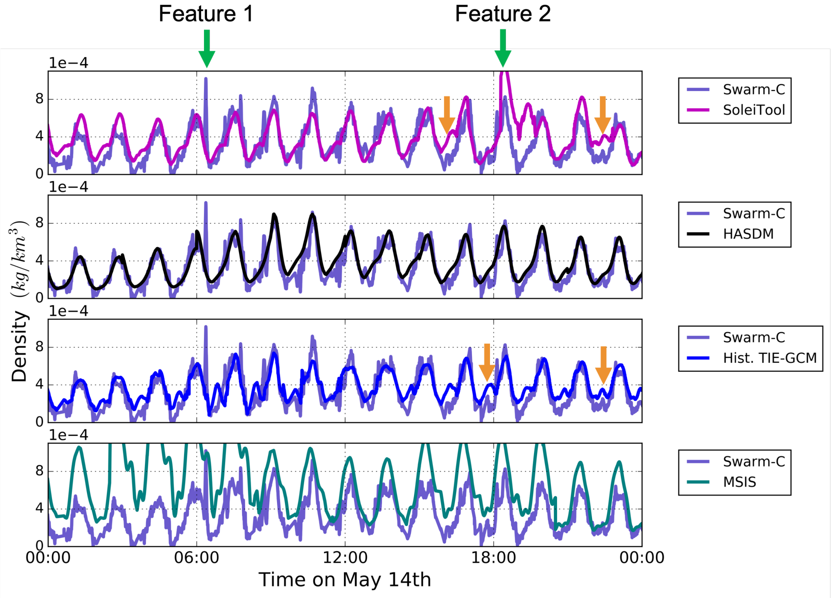

Figure 6.10: Second storm comparison between Swarm-C derived density and SoleiTool, HASDM, Historical TIE-GCM, and MSIS Along Swarm-C Trajectory

We quantify the performance of the estimated density by comparing it to other models via the metrics presented in [64] and [62]. These metrics include $\mu\left(m/o\right)$, $\sigma\left(m/o\right)$, and $RMSe\left(m/o\right)$, all in log space:

$$\mu\left(m/o\right) = \exp\left(\frac{1}{N}\sum_{i=1}^{N}\ln\frac{\rho_{m,i}}{\rho_{o,i}}\right) \tag{6.5}$$

$$\sigma\left(m/o\right) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\ln\frac{\rho_{m,i}}{\rho_{o,i}} - \ln\mu\left(m/o\right)\right)^{2}} \tag{6.6}$$

$$RMSe\left(m/o\right) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\ln\frac{\rho_{m,i}}{\rho_{o,i}}\right)^{2}} \tag{6.7}$$

where “m” is the *model* density and “o” is the *observed* density, or the closest to truth density that is available (e.g., HASDM). The $\sigma\left(m/o\right)$ and $RMSe\left(m/o\right)$ quantities are best interpreted as percentages, using:

$$\% = 100 \times \left(\exp\sigma\left(m/o\right) - 1\right) \tag{6.8}$$

$RMSe\left(m/o\right)$ is a good indicator of total model errors [64]. These metrics are provided for the estimated density along the trajectory of Spire satellite 62 in Table 6.3, and for the Swarm-C satellite in Table 6.4. This analysis focuses on the more interesting part of the case study: the two storms, spanning May 10 12:00 UT - May 16 00:00 UT 2019.

For the Spire satellite 62 metrics, each model density is directly compared to HASDM density. These metrics are a measurement of amplitude and phase, so an offset in either one of these components can impact the performance evaluation negatively. Historical TIE-GCM aligns best with HASDM. SoleiTool performs better than MSIS and has worse alignment with HASDM than historical TIE-GCM. This could be caused in part by the steady-state nature of SoleiTool’s Kp estimates and the fixed F10.7 applied in SoleiTool; both of which cause a loss in the thermospheric dynamics captured by SoleiTool. Another possible factor is the finer time-resolution of the estimated density solution. HASDM coefficients and historical TIE-GCM indices have a 3-hour resolution, compared to SoleiTool’s 1.5-hour resolution, and therefore may not be able to capture small scale features. Mis-alignment between SoleiTool and HASDM may also be driven by certain data arcs not containing enough information to estimate a density coefficient. This would result in noise or an offset in the density coefficient estimate, as indicated by the higher density coefficient 3-σ uncertainty for higher altitude satellites.

Table 6.3: Performance Metrics of Each Model Density with respect to HASDM Observed Density along Satellite 62 Trajectory

Model	$\mu\,(m/o)$	$\sigma\,(m/o)$	$RMSe\,(m/o)$
Batch	1.02	49%	49%
SoleiTool	1.35	30%	49%
Hist. TIE-GCM	1.15	25%	30%
MSIS	1.67	40%	86%

Table 6.4 provides the density comparison metrics along the Swarm-C trajectory. It is important to note that, similar to HASDM in Table 6.3, the Swarm-derived density is not actually "true" density. Swarm density is of a lower fidelity than accelerometer measurements made by, for instance, CHAMP or GRACE satellites, but Swarm density is what we think is the best available during this time period. Due to known errors in the Swarm-derived density, the orbit-average densities of the Swarm-C are compared to the corresponding orbit-average densities of each model, as done in [64, 20]. HASDM and historical TIE-GCM have the best alignment with Swarm-derived density, according to these metrics. It is important to note that HASDM uses information from 75+ satellites, while SoleiTool uses only 10 satellites for this work.

Table 6.4: Performance Metrics of Each Model Density with respect to Swarm-Derived Density along Swarm-C Trajectory

Model	$\mu\,(m/o)$	$\sigma\,(m/o)$	$RMSe\,(m/o)$
SoleiTool	1.44	25%	53%
HASDM	1.11	11%	16%
Hist. TIE-GCM	1.22	16%	28%
MSIS	2.19	32%	130%

Swarm-C density (derived using POD to correct accelerometer data) is the *observed* density.

6.3.4 Satellite Propagation Analysis

The desired end result of SoleiTool is improved thermospheric density modeling to allow for more accurate LEO object trajectory prediction. To evaluate this, Spire CubeSats are propagated with SoleiTool, HASDM, historical TIE-GCM, and MSIS. Each model propagation is initialized with the same ephemeris initial conditions: the POD position and velocity at the beginning of the arc. For SoleiTool, the satellites are propagated through the density field generated using the estimated Kp and fixed F10.7 in TIE-GCM. This is done for each arc, separately. Then the position difference between the propagation and POD ephemeris at the end of each arc is computed. We can infer that a smaller position difference indicates a better-calibrated density model. All propagated trajectories have inherent error that is not due to density, but another unknown dynamical effect (e.g., cannonball SRP assumption, bias in ballistic coefficient calculation). So, in all cases there is a baseline error; onto which, the error due to density accumulates. Each trajectory, for SoleiTool, HASDM, historical TIE-GCM, and MSIS, is propagated using the same dynamical models in Orekit; with the only difference being the specific density model.

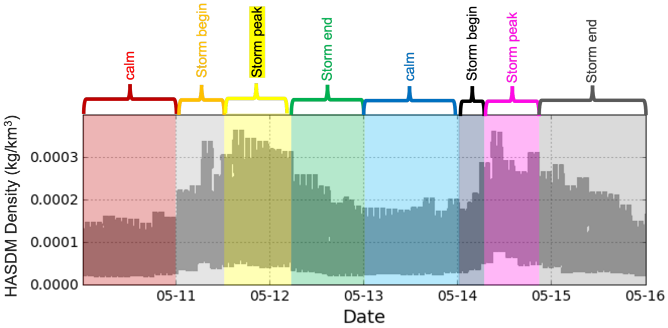

Figure 6.11: Satellite propagation analysis time region definitions; HASDM density (gray) provides reference for time-region definitions

To evaluate each model's performance during key times throughout the case study, we compute the average difference, or error, between model propagation and POD throughout each time period indicated in Figure 6.11 (HASDM density provides reference for time-region definitions). This demonstrates how well each model performs during the peak of a storm verses the calm conditions in between, for example. We continue to focus on May 10-16th for this analysis. Figure 6.12 shows the mean error magnitude across a time-region for each model for satellite 66 and 99. The dot markers represent estimated density model performance and the "x" markers show the performance of other models: HASDM, historical TIE-GCM, and MSIS. Satellite 62 propagation error has larger variation due to the larger effect of density on a lower altitude orbit.

Propagation of satellite 62 and 99 with the estimated density aligns well with HASDM and historical TIE-GCM, and outperforms MSIS in almost all cases. The lowest propagation errors occur during the peak of the second storm, indicating that all models provide a density field that aligns well with the true space environment during peak storm-time. It appears that all models perform slightly worse during the first storm peak compared to the second. The calm conditions before and after the first storm are modeled to a similar level of accuracy as the storm-time conditions. This is noteworthy because calm conditions can be more difficult to sense with methods like SoleiTool and HASDM due to the smaller affect of drag on observed satellite trajectories.

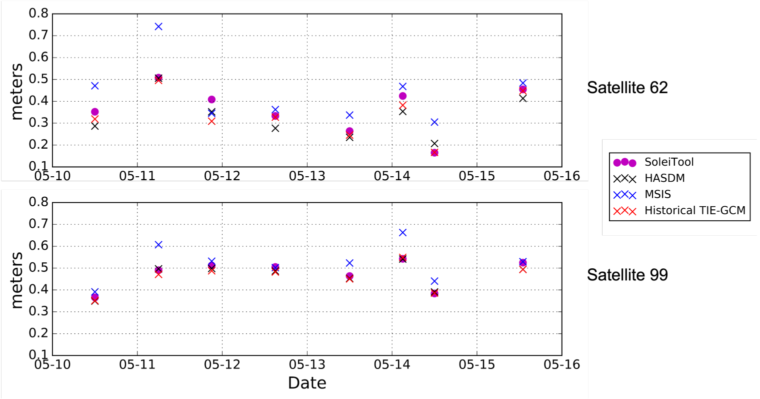

Figure 6.12: Satellite 62 and 99 Model Propagation Position Error

The results for all ten Spire CubeSats in the case study are averaged together in Figure 6.13 to show overall model propagation performance. The mean 20-minute arc propagation error for all models is always less than 0.6 meters. MSIS propagation is always worse than other models. Although SoleiTool uses only 10 satellites, its performance is very similar to HASDM in most cases, and outperforms HASDM in one case (second storm peak).

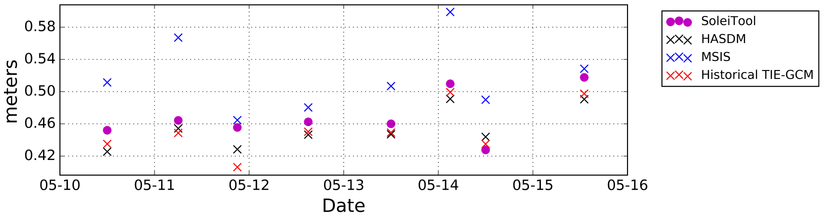

Figure 6.13: Average Model Propagation Performance Across all 10 Spire CubeSats

6.4 Chapter Conclusion

We have demonstrated SoleiTool's ability to utilize CubeSat POD information from ten CubeSats to estimate thermospheric density. First, we provided the Batch Single-Satellite-Solution in which it uses an arc of POD information from a single satellite to estimate a density coefficient that informs the space environment the satellite is traversing through. The Batch portion of SoleiTool processes each CubeSat POD data for the length of the case study to extract space environment information via density coefficient estimation for each arc. This information for all CubeSats is collected and passed to the PF to estimate the Kp index. The performance of SoleiTool is evaluated in the following ways: analysis of the estimated Kp time-series, comparison of the estimated density with Swarm-derived-density, HASDM, historical TIE-GCM, and MSIS, and CubeSat location difference with POD after model propagation with SoleiTool, HASDM, historical TIE-GCM, and MSIS density.

The Batch results showed that satellites with higher altitudes have a larger number of Single-Satellite-Solution density outliers; which is consistent with the findings of the simulation scenario. Evaluation of three error metrics that quantify the difference between observed density and model density shows that HASDM outperforms SoleiTool. However, it is important to note that HASDM and Swarm densities are not actual truth data, so performance evaluation should not rely too heavily on these metrics.

One of the strengths of SoleiTool when compared to what is used operationally, is the time resolution. SoleiTool extracts density information from a local spatial area with a time resolution as short as 20 minutes. SoleiTool also provides updated TIE-GCM input parameter(s) with a time resolution of 1.5 hours; whereas, HASDM has a 3-hour time resolution. Because of this, we are able to discover finer time-scale features as the ones discussed earlier. These features may be due to one or two satellites that pass through a region of high density, but this region of high density is not representative of what is occurring on a global scale. Another possible cause of the small-scale density features is the outlier density coefficients estimated in the Batch; these measurements are not filtered based on their uncertainty in the PF due to the lack of available data.

The end goal of SoleiTool is to improve thermospheric density estimates in order to allow for more accurate LEO object motion prediction. Therefore, we perform a density model propagation analysis in which Spire satellite trajectories propagated with each density model are compared to POD. This propagation analysis shows that SoleiTool outperforms HASDM in one case (second storm peak); while HASDM and historical TIE-GCM slightly outperform SoleiTool in the remaining cases.

Overall, this case study allows us to evaluate the method's performance using ten CubeSats (ranging from 420-520km altitude) for 11 days to estimate a dynamical space environment. Whereas, the operationally used HASDM effort uses 75+ satellites ranging from 200-800km altitude. The true strength of the method presented here is its potential to exploit data already collected on large LEO constellations.

Chapter 7

Conclusion

This dissertation presented SoleiTool, a novel method for estimating global thermospheric density using CubeSat GPS information and a physics-based space environment model. Method performance was evaluated via application to a simulation scenario modeled after the full Spire CubeSat constellation and a 10-satellite Spire CubeSat flight data case study. The simulation scenario provides a more comprehensive evaluation because true global density is known.

The Batch estimates a single density coefficient for a 20-40 minute arc of POD data. The estimated density coefficient represents the arc-averaged behavior of the sensed density conditions. Advances in GPS technology could improve POD accuracy to a point that would allow for finer density coefficient time-resolution by using shorter arcs (arcs $<$ 20 minutes). Meanwhile, some satellites with lower altitudes (altitudes $<$ 400 km) and relatively high ballistic coefficients may also be able to provide enough information to apply a shorter arc length in the Batch.

The simulation scenario revealed an apparent limit to SoleiTool performance; the global density RMSe never fell below 5% error. This threshold is likely caused by the fixed F10.7 applied in the PF and the steady-state nature of the TIE-GCM runs used in SoleiTool; both of which limit the full range of transient thermospheric behavior captured. This provides motivation for augmenting SoleiTool to utilize TIE-GCM in nominal mode.

7.1 Review of Research Contributions

The three primary contributions of this research are as follows:

1. This dissertation introduced a novel method for utilizing CubeSat GPS information to estimate global thermospheric density. SoleiTool is a two-step filtering process in which a Batch filter estimates the density of a localized spatial area, and a PF assimilates the Batch density information to estimate drivers to TIE-GCM. The end product is an estimated global density field.

2. SoleiTool's ability to estimate thermospheric density from as little as a 20-minute arc of CubeSat GPS information via a Batch filter was demonstrated.

3. The performance of SoleiTool was demonstrated via a simulation scenario of the full Spire CubeSat constellation (119 satellites) and a 10-satellite case study using Spire CubeSat flight data during two historical geomagnetic storms. SoleiTool performance in the simulation scenario was evaluated via a direct comparison to true density. Whereas, tool performance in the case study was evaluated via comparison to the operational density model, HASDM, and density from the Swarm mission.

SoleiTool was developed with a vision of improving LEO object motion prediction by providing more accurate global thermospheric density estimates. In doing so, I hope to decrease the number of unnecessary satellite maneuvers executed to avoid satellites and debris objects. This not only extends satellite lifetime, but also provides for a safer space environment by preserving fuel to correct for actual collision risks.

7.2 Future Research Considerations

The findings discussed in this dissertation suggest some improvements and directions for future research. The first is to replace TIE-GCM steady-state mode with nominal TIE-GCM. As discussed in Chapter 2, using TIE-GCM in steady-state mode reduces the fidelity of the dynamics captured by the model. In order to take full advantage of TIE-GCM, and its modeling of the full near-Earth space environment dynamics, it is necessary to run TIE-GCM in its nominal mode rather than steady-state mode, but this is not straightforward. It requires a restructure of SoleiTool's data assimilation scheme that involves several calls to TIE-GCM during PF execution. In addition, a Kp index estimate in the PF would not be solely dependent on the current 1-1.5 hours of density coefficient measurements. This is because the density field response to a Kp index takes place over a time period up to about 24 hours. So, future measurements will affect the current Kp estimate. Therefore, instead of estimating one 1-hour Kp with one hour of density coefficients, for example, all 24 1-hour Kp indices leading up to the measurement time will be adjusted with the latest hour of density coefficients. The PF 24-hour window would move forward one hour with each measurement step. A similar data assimilation scheme in which TIE-GCM is used as the background model to estimate a 24-hour window of Kp indices is described by Sutton et al. [64].

It would also be advantageous to explore applying a machine learning (ML) algorithm to estimate both Kp and F10.7, instead of applying a fixed F10.7. An ML algorithm could potentially allow for the effects of Kp and F10.7 to be distinguished from one another so that both could be estimated. Historical Kp and F10.7 indices and historical HASDM global density field estimates can be used as training data sets. Applying a time-varying Kp and F10.7 as inputs to TIE-GCM in SoleiTool would likely reduce the overall global density error floor of 5% discovered in the simulation scenario.

Another suggested SoleiTool update is to improve the background dynamical model applied in the Batch filter by replacing the simple cannonball SRP model with a high-fidelity SRP model. This involves incorporating the attitude and satellite geometry models, already used in the drag cross-sectional area calculation, to compute necessary time-varying physical characteristics that affect SRP perturbations.

The ultimate goal of this research is to support SDA applications by extracting thermospheric density information from CubeSats in near real-time. To do so requires deploying SoleiTool for operational use. This would require a number of software improvements and modifications in SoleiTool to allow for parallel computing and automated processing. Operational deployment of SoleiTool would provide an opportunity to sense near-Earth space weather events as they occur. Application of SoleiTool to additional CubeSat flight data during a range of thermospheric density conditions will allow for further research and development, and potential performance improvement of the tool, while also providing near real-time insight into the behavior of thermospheric density for the benefit of LEO operations.

Bibliography

[1] R. A. Akmaev, F. WU, T. J. Fuller-Rowell, H. Wang, and M. D. Iredell. Midnight density and temperature maxima, and thermospheric dynamics in whole atmosphere model simulations. Journal of Geophysical Research, 115, A08326, 1997.

[2] Analytical Graphics, Inc. Systems tool kit (STK). `https://www.agi.com/products/stk`, April 2, 2022. AGI.

[3] C. Arduini, G. Leneve, and F. A Herrero. Local time and altitude variation of equatorial thermosphere midnight density maximum (MDM): San marco drag balance measurements. Geophysical Research, 24, No. 4, 1997.

[4] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking. IEEE Transactions on signal processing, 50(2):174–188, 2002.

[5] T. E. Berger, M. J. Holzinger, E. K. Sutton, and J. P. Thayer. Flying through uncertainty. Space Weather, 18, 2020.

[6] B. Bowman and M.F Storz. Time series analysis of HASDM thermospheric temperature and density corrections. AIAA/AAS Astrodynamics Specialist Conference, 2002.

[7] B. Bowman and M.F Storz. High accuracy satellite drag model (HASDM) review. Advances in the Astronautical Sciences, 116:1943–1952, 2003.

[8] B. R. Bowman, W. K. Tobiska, F. A. Marcos, C. Y. Huang, C. S. Lin, and W. J Burke. A new empirical thermospheric density model JB2008 using new solar and geomagnetic indices. In AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, Hawaii, 2008.

[9] R.G. Brown and P.Y. Hwang. Introduction to random signals and applied Kalman filtering : with MATLAB exercises and solutions. Wiley, 4th edition, 2012.

[10] A. Calabia and S Jin. GPS-based non-gravitational accelerations and accelerometer calibration. In: Jin, S. (Ed.), Satellite positioning: methods, models and applications, InTech-Publisher, Rijeka, Croatia:47–72, 2015.

[11] A. Calabia and S Jin. Assessment of conservative force models from GRACE accelerometers and precise orbit determination. Aerospace Science and Technology, 49:80–87, 2016.

[12] A. Calabia and S Jin. Thermospheric density estimation and responses to the March 2013 geomagnetic storm from GRACE GPS-determined precise orbits. Journal of Atmospheric and Solar-Terrestrial Physics, 154:167–179, 2017.

[13] Space Weather Prediction Center. F10.7 cm radio emissions. `www.swpc.noaa.gov/noaa-scales-explanation`. November 18, 2021.

[14] Space Weather Prediction Center. NOAA space weather scales. `www.swpc.noaa.gov/noaa-scales-explanation`, 2011. June 20, 2020.

[15] Space Weather Prediction Center. Station K and A indices. `www.swpc.noaa.gov/products/station-k-and-indices`, 2021. November 18, 2021.

[16] Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. Journal of Geophysical Research, 99(C5):143–162, 1994.

[17] Geir Evensen. Data assimilation: the ensemble Kalman filter. Springer Science Business Media, 2009.

[18] R. M. Fitzgerald and K. L. Cahoy. Localized in-situ density measurement in low earth orbit via drag torque estimation. Journal of Spacecraft and Rockets, 56(5), 2019.

[19] M. Garcia. Space debris and human spacecraft, 2020.

[20] D. J Gondelach and R. Linares. Real-time thermospheric density estimation via two-line element data assimilation. Space Weather, 18(2), e2019SW002, 2020.

[21] D. J. Gondelach and R. Linares. Real-time thermospheric density estimation via radar and GPS tracking data assimilation. Space Weather, 19(4), 2021.

[22] N.J. Gordon, D.J. Salmond, and A.F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings F-radar and signal processing, 140(2):107–113, 1993.

[23] Inc. GPS Solutions. RTOrb. `https://gps-solutions.com/brochures/GPSS_Brochure_RTOrb_Nov_2011.pdf`. April 1, 2022.

[24] F. Gustafsson. Particle filter theory and practice with positioning applications. IEEE Aerospace and Electronic Systems Magazine, 25(7):53–82, 2010.

[25] Fredrik Gustafsson and Gustaf Hendeby. Some relations between extended and unscented Kalman filters. IEEE Transactions on Signal Processing, 60(2):545–555, 2012.

[26] C. Henry. Lacuna space aims to ride IoT wave with a 32-CubeSat constellation. `https://spacenews.com/lacuna-space-aims-to-ride-iot-wave-with-a-32-/......./cubesat-constellation/`, 2019. May 5, 2021.

[27] C. Henry. Kepler decides to build its 140-satellite CubeSat constellation in-house. `https://spacenews.com/kepler-decides-to-build-its-140-satellite-cubesat-constellation-in-house/`, 2020. May 27, 2021.

[28] Pieter Houtekamer and Fuqing Zhang. Review of the ensemble Kalman filter for atmospheric data assimilation. Monthly Weather Review, 144, 06 2016.

[29] P.L. Houtekamer and H.L. Mitchell. Ensemble Kalman filtering. Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography, 131(613):3269–3289, 2005.

[30] N.L. Johnson. Orbital debris: The growing threat to space operations. Advances in the Astronautical Sciences, 137:3–11, 01 2010.

[31] S. J. Julier and J. K. Ulhmann. A new extension of the Kalman filter to nonlinear systems. Signal processing, sensor fusion, and target recognition, 3068:182–193, 1997.

[32] Simon Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. Proceedings of the IEEE, 92:401–422, 2004.

[33] Simon Julier, J.K. Uhlmann, and Hugh Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. Automatic Control, IEEE Transactions on, 45:477 – 482, 2000.

[34] B. Kirov, S. Asenovski, K Georgieva, and V. N. Obridko. What causes geomagnetic activity during sunspot minimum? Geomagnetism and Aeronomy, 55:1033–1038, 2015.

[35] Delores J. Knipp. Understanding Space Weather and the Physics Behind It. Space Technology (McGraw-Hill), 2011.

[36] D.J. Knipp, W.K. Tobiska, and B.A. Emery. Direct and indirect thermosphere heating sources for solar cycle 21-23. Solar Physics, 224:495–505, 2004.

[37] D. Kuang, S. Desai, A. Sibthorpe, and X. Pi. Measuring atmospheric density using GPS LEO tracking data. Advances in Space Research, 53:243–256, 2014.

[38] R. J. Licata, P. M. Mehta, W. K. Tobiska, B. R. Bowman, and M. D. Pilinski. Qualitative and quantitative assessment of the SET HASDM database. Space Weather, 19, 2021.

[39] L. Maisonobe, I.-T. Lee, V. Pommier, and P. Parraud. Orekit: an open-source library for operational flight dynamics applications. In International conference on astrodynamics tool and techniques, pages 3–6, 2010.

[40] T. Matsuo. Upper atmosphere data assimilation with an ensemble Kalman filter. In Book: Modeling the Ionosphere-Thermosphere System, edited by J. D. Huba, pages 273–282. Geophysical Monograph Series, vol. 201, 03 2014.

[41] T. Matsuo, I.-T. Lee, and J.L. Anderson. Thermospheric mass density specification using an ensemble Kalman filter. Space Physics, 118:1339–1350, 2013.

[42] C.A. McLaughlin, T. Lechtenberg, E. Fattig, and D. Mysore Krishna. Estimating density using precision satellite orbits from multiple satellites. The Journal of the Astronautical Sciences, 59(1-2):84–100, 2013.

[43] P. Mehta and R. Linares. A methodology for reduced order modeling and calibration of the upper atmosphere. Space Weather, 15(10):1270–1287, 2017.

[44] P. Mehta, R. Linares, and E. 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.

[45] P. Mehta, R. Linares, and E. K. Sutton. Data-driven inference of thermosphere composition during solar minimum conditions. Space Weather, 17:1364–1379, 2019.

[46] A. Morozov, A. Ridley, D. Bernstein, N. Collins, T. Hoar, and J. Anderson. Data assimilation and driver estimation for the global ionosphere-thermosphere model using the ensemble adjustment Kalman filter. Journal of Atmospheric and Solar-Terrestrial Physics, 104:126–136, 2013.

[47] S. Mutschler, P. Axelrad, and T. Matsuo. A partially orthogonal EnKF approach to atmospheric density estimation using orbital debris. Advances in Space Research, 65(8):1965–1980, 2020.

[48] S. Mutschler, P. Axelrad, T. Matsuo, and E. Sutton. Physics-based approach to density estimation and prediction using orbital debris tracking data. Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, HI, 11 pages:September 17–20, 2019.

[49] S. Mutschler, P. Axelrad, and E. Sutton. Application of soleitool for density estimation using cubesat GPS data. Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, HI, 15 pages:September 14–17, 2021.

[50] V. A. Nguyen, O. NoguCorreig, T. Yuasa, D. Masters, and V. Irisov. Initial GNSS phase altimetry measurements from the Spire satellite constellation. Geophysical Research Letters, 47, 2020.

[51] Department of Defense. Fiscal year (FY) 2017 president's budget submission: Research, development, test evaluation, air force, 2017.

[52] 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, 107(A12):1468, 2002.

[53] M. Pilinski, G. Crowley, E. Sutton, and M. Codrescu. Physics-based assimilative atmospheric modeling for satellite drag specification and forecasts. Advances in the Astronautical Sciences, 156:p. 4405, 2016.

[54] M. L. Psiaki. The blind tricyclist problem and a comparative study of nonlinear filters: A challenging benchmark for evaluating nonlinear estimation methods. IEEE Control Syst. Mag., 33(3):40–54, 2013.

[55] L. Qian and S. Solomon. Thermospheric density: An overview of temporal and spatial variations. Space Science Reviews, 168:1–27, 2011.

[56] Liying Qian, Alan Burns, Barbara Emery, Benjamin Foster, Gang Lu, Astrid Maute, A. Richmond, Raymond Roble, Stanley Solomon, and Wenbin Wang. The NCAR TIE-GCM: A community model of the coupled thermosphere/ionosphere system. In Book: Modeling the Ionosphere-Thermosphere System, edited by J. D. Huba, pages 73–83. Geophysical Monograph Series, vol. 201, 03 2014.

[57] A.D. Richmond, E.C. Ridley, and R.G. Roble. A thermosphere/ionosphere general circulation model with coupled electrodynamics. Geophysical Research Letters, 19:601–604, 1992.

[58] Branko Ristic, S. Arulampalam, and N. Gordon. Beyond the Kalman filter: Particle Filters for Tracking Applications. Artech House, 2004.

[59] J. Rosenblatt, D. Garber, and D. McKnight. Examination of constellation deployments relative to debris mitigation in low earth orbit. Proceedings of the 68th IAC (International Astronautical Congress), Adelaide, Australia, IAC-17-A6,10-B4.10,3, 2017.

[60] M.F. Storz, B.R. Bowman, J.I. Branson, S.J. Casali, and W.K. Tobiska. High accuracy satellite drag model (HASDM). Advances in Space Research, 36(12):2497–2505, 2005.

[61] E. Sutton. Normalized force coefficients for satellites with elongated shapes. Journal of Spacecraft and Rockets, 46(1):112–116, 2009.

[62] E. Sutton. A new method of physics-based data assimilation for the quiet and disturbed thermosphere. Space Weather, 16(6):736–753, 2018.

[63] E. Sutton, S. Cable, C. Lin, L. Qian, and D.R. Weimer. Thermospheric basis functions for improved dynamic calibration of semi-empirical models. Space Weather, 10, 2012.

[64] E. Sutton, J. Thayer, M. Pilinski, S. Mutschler, T. Berger, V. Nguyen, and D Masters. Toward accurate physics-based specifications of neutral density using GNSS-enabled small satellites. Space Weather, 19, 2021.

[65] R. Thompson. Space weather indices, australian government bureau of meteorology, 2020.

[66] W. K. Tobiska, B. R. Bowman, S. D. Bouwer, A. Cruz, K. Wahl, M. D. Pilinski, P. Mehta, and R. J. Licata. The SET HASDM density database. Space Weather, 19, 2021.

[67] W. K. Tobiska, B. R. Bowman, and Bouwer. D. Solar and Geomagnetic Indices for Thermospheric Density Models. In Book: COSPAR International Reference Atmosphere: Models of the Earth's upper atmosphere, Ch. 4, edited by Rees, D. and Tobiska, W. K.., pages 273–282. COSPAR, January 2012.

[68] Vallado. Fundamentals of Astrodynamics and Applications. Space Technology Library, 2013.

[69] J. van den IJssel, E. Doornbos, E. Iorfida, G. March, C. Siemes, and O. Montenbruck. Thermosphere densities derived from swarm GPS observations. Advances in Space Research, 65(7):1758–1771, 2020.

[70] J. van den IJssel and P. Visser. Performance of GPS-based accelerometer: a simulation experiment. Advances in Space Research, 45:225–238, 2010.

[71] J. van den IJssel and P. Visser. Performance of GPS-based accelerometry: CHAMP and GRACE. Advances in Space Research, 39:1597–1603, 2020.

[72] R. Walker. The formation and evolution of CIRs, 2008.

[73] H. Zell. What is a coronal mass ejection or CME? `https://www.nasa.gov/content/goddard/what-is-a-coronal-mass-ejection`, 2017. Dec. 10, 2021.

Appendix A

Physics-based Approach to Density Estimation and Prediction using Orbital Debris Tracking Data

This is a 2019 AMOS Conference paper [48].

A.1 Introduction

Our research utilizes debris objects to sense their local space environment. We focus on the low Earth orbital regime where mismodeling of atmospheric drag is the largest contributor to orbit prediction error. Current methods that estimate or model atmospheric density can be categorized as either empirical or physics-based approaches. Empirical models are computationally fast, but only represent the climatological atmospheric conditions. Physics-based models incorporate current knowledge of atmospheric conditions to provide forecasts, but require substantially more computing power than empirical models. Here, a methodology is described that aims to estimate forcing parameters of a physics-based space environment model to allow for improved density estimates and predictions. By improving space object predictions we hope to decrease the number of unnecessary satellite maneuvers executed to avoid debris objects. Thereby, increasing satellite mission lifetime and improving overall space situational awareness to the benefit of all space operations.

A.1.1 Atmospheric Density

An object in Low Earth Orbit (LEO) experiences atmospheric drag caused by particles in the atmosphere colliding with the surface of the object. Drag acts primarily in the opposite direction of the velocity vector to decrease the speed of an object. The magnitude of the force due to drag is directly dependent on neutral density (number of particles in the atmosphere). This is generally modelled in the equation of acceleration due to drag,

$$a_{drag} = -\frac{1}{2}\rho\frac{C_D A}{m}v_{rel}^2\frac{\vec{v}_{rel}}{|\vec{v}_{rel}\,|} \tag{A.1}$$

where

$$\vec{v}_{rel} = \frac{d\vec{r}}{dt} - \vec{\omega}_{\oplus} \times \vec{r} \tag{A.2}$$

is the velocity vector relative to the rotating atmosphere; C_D is the coefficient of drag; A is the exposed cross-sectional area; m is the object's mass; $\vec{\omega}_{\oplus}$ is the mean motion of the Earth's rotation; and $\vec{r}$ is the Earth Centered Inertial object position vector [14]. The drag coefficient, mass, and area of the object are typically combined to form the the ballistic coefficient ($\beta = \frac{C_D A}{m}$).

A.1.2 Prior Work

The High Accuracy Satellite Drag Model (HASDM) project is an example of an empirical approach. This effort estimates a time-series of thirteen spherical harmonic global density and temperature correction coefficients in a batch fit by using observations of 75-80 carefully selected calibration satellites (payloads and debris objects) in a batch fit [1, 2, 10]. Intensive sensor tasking is made available for this effort, which allows for the collection of approximately 500 observations per day per calibration satellite. HASDM decouples the ballistic coefficient and the density parameter by first solving for the long-term averaged ballistic coefficient of each satellite, averaging almost 3200 previously estimated values for each calibration satellite.

The HASDM empirical model does not allow for the spatial and temporal resolution necessary to capture the dynamical behavior of density. As described above, this method estimates coefficients of density that are applied to a pre-defined model, not density directly. Furthermore, by combining information from objects in different regimes, or at different altitudes, a single batch fit limits the spatial resolution of density corrections that can be achieved.

More recently, density modeling and forecasting methods have been developed that take advantage of physics-based models. Matsuo [4] showed that ionospheric electron density data can be used to estimate driver parameters of the physics-based Thermosphere-Ionosphere-Electrodynamics General Circulation Model (TIE-GCM) [9] for the purpose of extending thermospheric mass density predictions. Mehta and Linares [5] developed an approach that aims to combine the strength of a physics based model in its predictive capability, with the speed of an empirical model. Their approach is demonstrated using an empirical model, MSIS, as the underlying upper atmosphere model with the intention that a physics-based model will replace MSIS in the future. A physics-based approach that estimates atmospheric drivers, such as F10.7 and Kp indices, as inputs to TIE-GCM is described by Sutton [12]. This effort estimates drivers through an iterative process that involves data assimilation of neutral density data from the Challenging Mini-Satellite Payload (CHAMP).

Our tool is named SoleiTool to emphasize its focus on the sun-driven nature of the near-Earth space environment. It takes advantage of a physics-based model, while leveraging routinely observed debris object data to estimate TIE-GCM forcing parameters, specifically Kp and F10.7 indices. The estimated Kp and F10.7 indices are then input to TIE-GCM to obtain a time-series of density, and can also be used in real-time or for atmospheric density predictions.

Our density estimation method is introduced in Section A.2. Section A.3 describes the scenario, including the simulated debris tracking data (debris objects, measurements, etc.) and filter initializations. The performance of SoleiTool when applied to this system is analyzed in Section A.4. We conclude with a summary of findings and plans for future work.

A.2 Method

As shown in Fig. A.1, SoleiTool comprises of two filters. The first filter utilizes debris object tracking data, in the form of measurements collected from ground sensors, to estimate acceleration due to atmospheric drag (a_{drag}), as well as the position and velocity (orbit determination state) of the debris objects. The total estimated state contains six orbit determination elements and one

a_{drag} element per debris object. Debris object data are assimilated and the resulting estimated accelerations, as well as time, position, and velocity, are passed to the second filter. The filter type, measurements, and estimated states of both filters are summarized in Table A.1.

Table A.1: UKF and PF Summary

Filter Type	**Measurements**	**Estimated State**
1. UKF	Debris Object az, el, range	Debris Object position, velocity, and a_{drag}
2. PF	a_{drag}	Kp, F10.7

The second filter, a Particle Filter (PF), processes accumulated estimates to produce Kp and F10.7 forcing parameters of a physics-based space environment model, TIE-GCM. Ensemble filters have typically been employed in high-dimensional non-linear geophysical applications, such as weather forecasting of atmosphere and ocean systems. However, in the framework proposed here, a PF provides a viable option for the second stage filter. Because our approach effectively reduces the state dimension, n, to only 2 forcing parameters, it becomes feasible to implement the necessary particle space of 10^n particles [7]. It is noted that our SoleiTool aims to estimate a time-series of Kp and F10.7 forcing parameters to TIE-GCM that will produce a density field that captures the true underlying density field; not necessarily to estimate the actual 3-hour global index of geomagnetic activity (Kp) and solar flux at a wavelength of 10.7 cm (F10.7) [13, 6].

TIE-GCM is used to generate a cloud of forecast density particles in the PF. The density particles, combined with the debris object ballistic coefficient and velocity, form predicted a_{drag} measurements. The PF applies corrections in the form of weight adjustments to the cloud of Kp and F10.7 particles based on the difference between these a_{drag} predictions and the first filter's a_{drag} estimates.

For the purpose of this study, the altitude of the debris objects are limited to 400 km altitude for simplification. In fact, the resulting estimated forcing parameters to TIE-GCM can be used to model density values for objects in LEO up to 700 km during solar max.

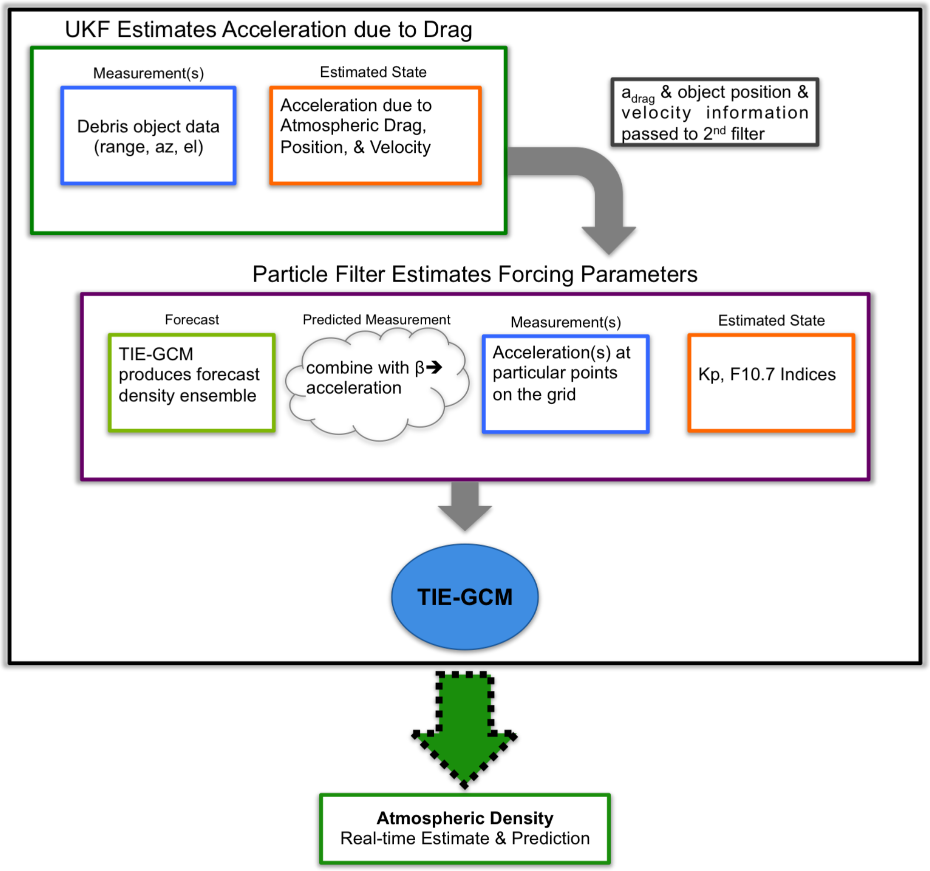
UKF Estimates Acceleration due to Drag
Measurement(s)
Debris object data (range, az, el)
Estimated State
Acceleration due to Atmospheric Drag, Position, & Velocity
a_drag & object position & velocity information passed to 2nd filter
Particle Filter Estimates Forcing Parameters
Forecast
TIE-GCM produces forecast density ensemble
Predicted Measurement
combine with β➔ acceleration
Measurement(s)
Acceleration(s) at particular points on the grid
Estimated State
Kp, F10.7 Indices
TIE-GCM
Atmospheric Density
Real-time Estimate & Prediction

Figure A.1: SoleiTool Flow

Our current approach assumes that the ballistic coefficients of debris objects are known to a reasonable accuracy. This assumption is based on the idea of initializing the ballistic coefficients with information from the HASDM, which is regularly used in operations for updated density estimates. To accomplish this, density information from HASDM, when available, will be combined with debris object measurements to infer debris object ballistic coefficients.

A.2.1 TIE-GCM

TIE-GCM is an open source general circulation model developed by the National Center for Atmospheric Research (NCAR) [8]. It is a three-dimensional, nonlinear representation of the coupled thermosphere and ionosphere system [11]. By adjusting model input parameters of the F10.7 index and Kp indices, TIE-GCM can simulate global thermospheric mass density changes under different solar extreme ultraviolet radiation and geomagnetic activity levels. For the purpose of this work, the model output time step is set to 1 hour and a default horizontal resolution of 5 degrees latitude and 5 degrees longitude is used with a vertical pressure level size of a half-scale height, extending from about 97 to 500 or 700 km (depending on solar cycles) [3]. Vertical interpolation is done during post-processing of the model output to obtain atmospheric density at 400 km altitude using the geopotential height. Interpolation is also executed in horizontal space and time to allow for a finer resolution of latitude, longitude, and time. In this work, Kp and F10.7 indices are estimated via assimilation of debris objects data in the PF.

A.3 Scenario

To develop the SoleiTool, we implemented a flexible simulation framework. Simulation details, such as debris objects, density field through which objects are propagated, and synthetic measurements are provided in Section A.3.1. Initializations applied in both filters are described in Section A.3.2.

A.3.1 Simulation Details

Fifteen debris objects are simulated for 650 minutes (roughly 11 hours). The objects are put in circular orbits with an initial altitude of 400 km. The remaining orbital parameters are distributed over the range provided in Table A.2.

Table A.2: Debris Object Orbital Elements

	Inclination	Initial True Anomaly	Right Ascension of Ascending Node
Minimum Value	10°	0°	0°
Maximum Value	45°	359 °	46°

The objects are propagated using two-body Keplerian dynamics plus atmospheric drag perturbations,

$$\ddot{\vec{r}} = \frac{\mu}{\vec{r}^3}\vec{r} - \frac{1}{2}\rho\frac{C_D A}{m}v_{rel}^2\frac{\vec{v}_{rel}}{|\vec{v}_{rel}\,|} \tag{A.3}$$

where $\vec{v}_{rel}$ is defined in Eq. A.2. The underlying density field through which the objects travel is generated via TIE-GCM. Calm conditions (low geomagnetic activity) are simulated using Kp and F10.7 index values of 3.166 and 127, respectively. The resulting density is shown in Fig. A.2.

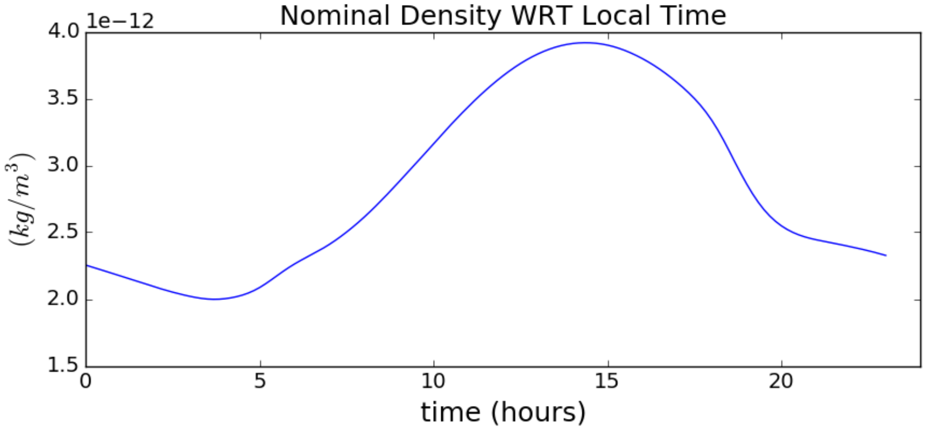

Figure A.2: Nominal Density Field Generated by TIE-GCM for a Particular Location on the Equator

Each object is simulated with an area to mass ratio (AMR) of 0.0955 m^2/kg. Nominally, the filter also assumes this correct AMR value of each object. Azimuth (az), elevation (el), and range (R) measurements of the objects are generated from the ground stations located in Canberra, Madrid, Diego Garcia, California, and Maui (red stars in Fig. A.3). Measurements are collected at a 3-minute cadence when objects are visible from a ground station. An azimuth and elevation sensor error standard deviation of 0.25 arc-seconds and range sensor error standard deviation of 0.25 meters is applied to the measurements. The ground track of one orbital period for each debris object is shown in Fig. A.3.

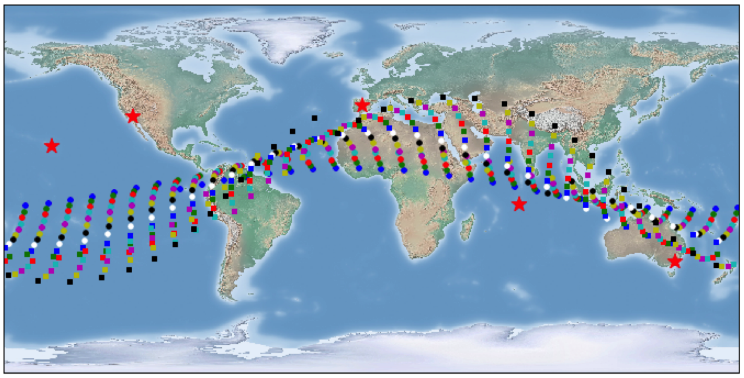

Figure A.3: One Orbital Period Groundtrack of Debris Objects

A.3.2 Initialization

In the UKF, the orbit determination (OD) state is defined as the position and velocity of each object:

$$\vec{X} = [\vec{r}, \vec{v}]^T \tag{A.4}$$

A perturbation is applied to the true initial a_{drag} and OD state so that the *a priori* state in the filter is not exactly equal to the true initial state,

$$\vec{X}_0' = \vec{X}_0 + \vec{p} \tag{A.5}$$

where

$$\vec{p}_{OD} = [100m, 100m, 100m, 0.1m/s, 0.1m/s, 0.1m/s]^T \tag{A.6}$$

and

$$\vec{p}_{a_{drag}} = \left[5e - 5cm/s^2\right]^T \tag{A.7}$$

It is noted that the true a_{drag} is on the order of 0.001 cm/s^2. Sigma points are generated using the perturbed initial state ($\vec{X}_0'$) and its *a priori* statistical information (P_0):

$$P_{0\ OD} = diag\left(\left[(150m)^2, (150m)^2, (150m)^2, (0.15m/s)^2, (0.15m/s)^2, (0.15m/s)^2\right]\right) \tag{A.8}$$

and

$$P_{0\ a_{drag}} = diag\left(\left[12e - 5cm/s^2\right]\right) \tag{A.9}$$

In the second filter, the PF, a similar approach is taken. A perturbation is applied in the PF to the true initial Kp and F10.7 index state (Eq. A.5), where

$$\vec{p} = [0.5, 15]^T \tag{A.10}$$

An initial cloud of Kp and F10.7 particles (χ_0) is generated using the perturbed initial state ($\vec{X}_0'$) and its *a priori* statistical information (P_0):

$$\chi_0^{(i)} = \vec{X}_0' + \eta^{(i)} \tag{A.11}$$

where,

$$\eta^{(i)} \sim \mathcal{N}(0, P_0) \tag{A.12}$$

and

$$P_0 = diag\left(\left[(0.85)^2, (30)^2\right]\right) \tag{A.13}$$

Fig. A.4 is the density generated via TIE-GCM using the initial 100 Kp and F10.7 particles. It shows the density for a single location on the equator as a function of local time. As expected, the maximum density occurs just after local noon. The maximum density achieved, around 9e-12 kg/m^3, is about 5e-12 kg/m^3 larger than the maximum density for nominal conditions (Fig. A.2). This range of densities from the initial Kp and F10.7 particles is necessary in order to capture the possible state space.

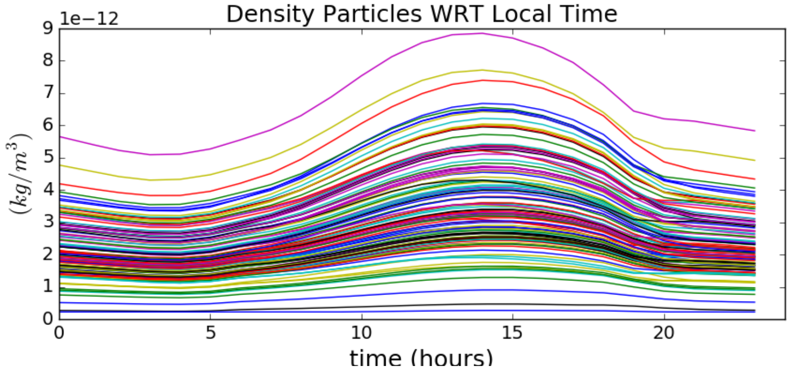

Figure A.4: TIE-GCM Density as Generated from 100 Particles

A.4 Results

The results of the UKF and PF are presented and analyzed separately in Sections A.4.1 and A.4.2.

A.4.1 First Filter: UKF

The UKF ingests debris object tracking data to estimate a_{drag}, position, and velocity of the debris objects. Fig. A.5 shows the true and estimated along-track acceleration due to drag for one debris object at the times that measurements are available. The true a_{drag} is cyclical because the debris object experiences areas of low (near local midnight) and high (near local noon) atmospheric density throughout one orbital period of 90 minutes. The estimated a_{drag} lags the true a_{drag} during the first 90 minutes. After about one orbital period, the UKF is able to better track the changing a_{drag} and follow its cyclic nature. It is apparent that the filter has more difficulty tracking the time-varying a_{drag} when the measurements are more sparse. This demonstrates the advantage of utilizing multiple debris objects. Some objects will provide more accurate a_{drag} estimates at times that other objects a_{drag} estimates have larger errors. When combined, they provide enough information to estimate TIE-GCM forcing parameters in the second filter, the PF.

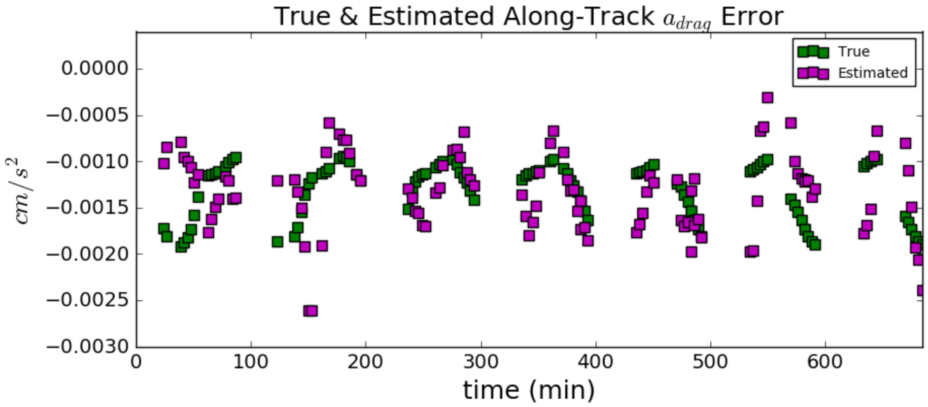

Figure A.5: Estimated v. True Along-track a_{drag} for One Object

Fig. A.6 is the estimated along-track a_{drag} for all 15 objects in the scenario. The first 90 minutes are not shown because only measurements beginning after filter convergence are ingested by the PF. These UKF estimated a_{drag} and corresponding variance time-series are then input as measurements and measurement uncertainty in the PF.

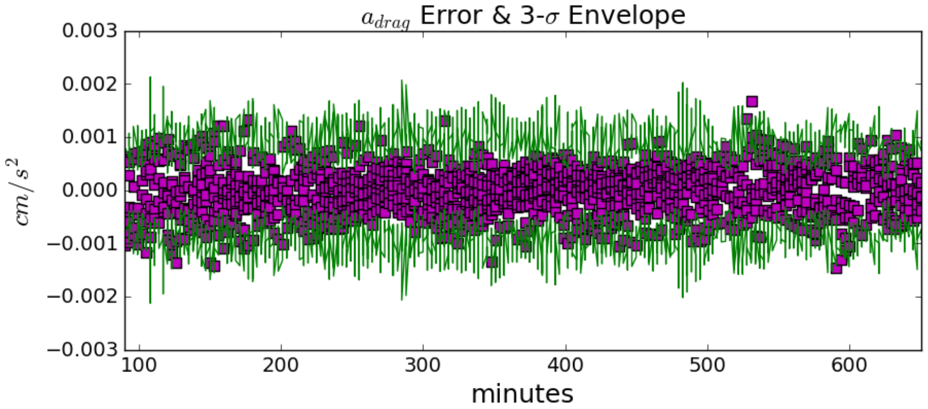

Figure A.6: Multi-Object Along-track Acceleration Error and 3-σ Envelope

A.4.2 Second Filter: PF

The PF uses the accumulated a_{drag} estimates from the UKF to produce Kp and F10.7 forcing parameters of TIE-GCM. Figs. A.7 and A.8 show the time evolution of the estimated Kp and F10.7 particles. The initial distribution collapses after the first few measurements. We inject a continual spreading of particles after the initial collapse by adding process noise to the particles in the time update. The Kp and F10.7 particle distribution is offset from the truth initially due to the perturbation applied to the *a priori* (described in Section A.3.2). Both particle distributions trend slightly away from the true indices (3.166 Kp and 127 F10.7) within the first 100 minutes of measurements; the estimated F10.7 trends low while the estimated Kp trends high. This may be indicative that the two inputs (F10.7 and Kp) are strongly correlated with respect to the synthetic data used. In other words, within TIEGCM, the two effects are compensating; the effect of a low F10.7 on density can be somewhat canceled by a high Kp. The time-series of estimated Kp and F10.7 indices are used as input to TIE-GCM to generate what we'll call the *estimated* density. Because the goal of our tool is to estimate forcing parameters that produce the true underlying density in TIE-GCM, the tool performance is evaluated by comparing the true and estimated density.

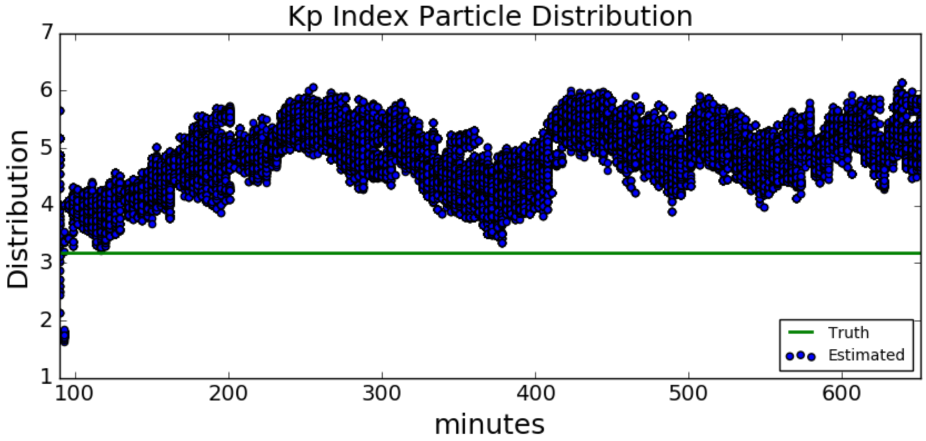

Figure A.7: Time Evolution of Kp Index Particles

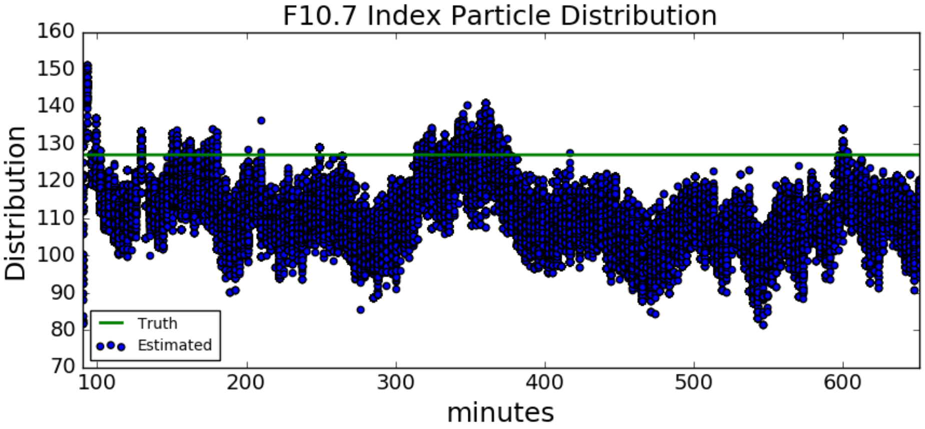

Figure A.8: Time Evolution of F10.7 Index Particles

The estimated and true density along one object's trajectory are shown in Fig. A.9. The magenta markers represent measurement times of this debris object. We see that the estimated density has relatively large error compared to the true density in the beginning of the simulation, but then starts to follow the same smooth cyclic nature as the true density for the remainder of the simulation. Fig. A.10 shows the estimated density error and its 3-σ bounds for this object.

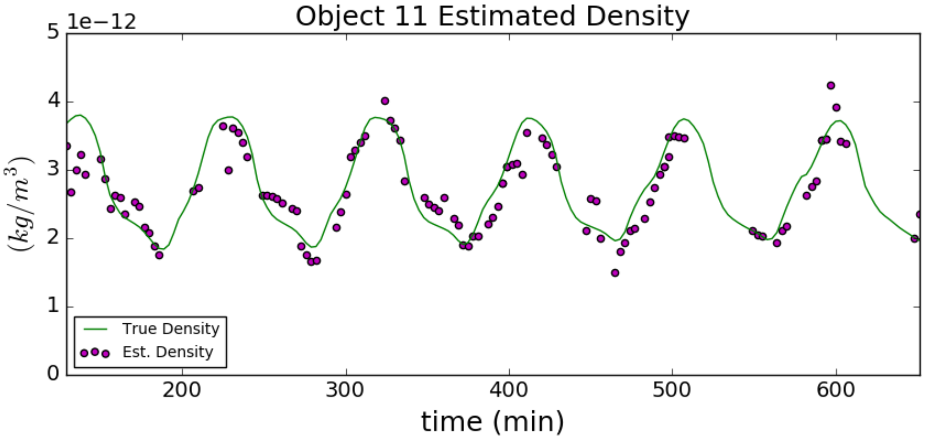

Figure A.9: Density Generated from Estimated TIE-GCM Kp and F10.7 Indices v. True Density for a Single Object

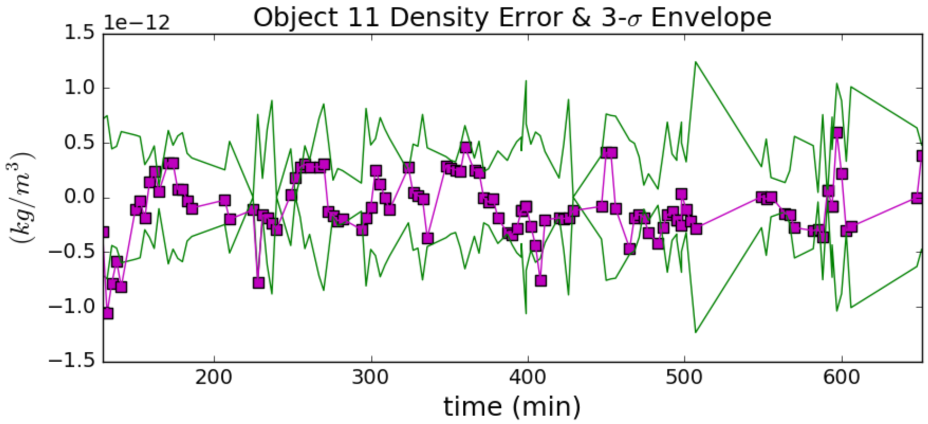

Figure A.10: Density Error and 3-σ Envelope for a Single Object

All the object results are combined to show the overall PF performance in Fig. A.11. This plot shows the time-series of the density error and 3-σ envelope at all measurement times (regardless of object). The 3-σ bounds are computed using the spread of density values produced in TIE-GCM via the cloud of state particles (Kp and F10.7) at each measurement time.

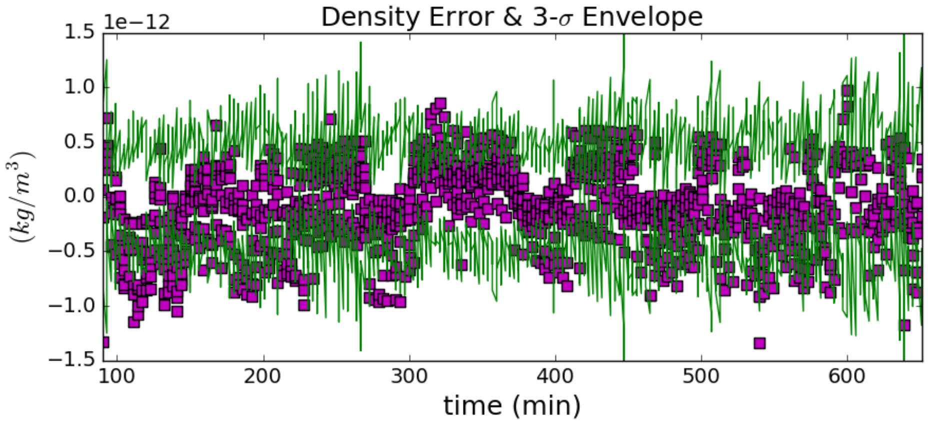

Figure A.11: Density Error and 3-σ Envelope

A.5 Conclusion and Future Work

Our work shows the potential for operationally collected debris object tracking data to be utilized to derive atmospheric density in LEO. The proposed two-filter process ingests debris object tracking data and first estimates a time-series of acceleration due to drag, before passing these estimates to the second filter to be used as measurements. The second filter, the PF, estimates Kp and F10.7 indices. These forcing parameters are input to TIE-GCM to produce an informed estimate of atmospheric density. SoleiTool also provides a predictive capability; forcing parameters can be utilized in TIE-GCM to forecast the near Earth space environment.

Although simulated tracking data from only 15 debris objects were used for this paper, we plan to demonstrate the performance of SoleiTool with hundreds to thousands of debris objects. By adding substantially more debris objects with greater variation in orbital elements, we plan to get an improved global picture of densities. A geomagnetic storm scenario will also be investigated in order to determine how well the filter is able to track time-varying Kp and F10.7 indices that increase to values as high as 9 and 273, respectively, for an extreme geomagnetic storm [13].

In order to make a more realistic scenario, various object AMRs will be simulated and imperfect knowledge of object AMRs assumed in the filter. Spinning objects will also be incorporated so that the object area considered in the equation for acceleration due to drag is not constant. Currently, only Kp and F10.7 indices are estimated in the PF; however, there are several other available input parameters to TIE-GCM. We plan to investigate estimation of additional TIE-GCM forcing parameters. Closed-loop feedback from the PF to UKF will also be considered in order to allow for an improved initialization of along-track a_{drag}.

A.6 Acknowledgments

Support for S. Mutschler has been provided by the National Defense Science and Engineering Graduate (NDSEG) Fellowship. We thank Chih-Ting Hsu, a postdoctoral fellow at University of Colorado Boulder, for her assistance with TIE-GCM.

A.7 References

1. Bowman, B., Storz, M.F., 2002. Time Series Analysis of HASDM Thermospheric Temperature and Density Corrections. In: AIAA/AAS Astrodynamics Specialist Conference.

2. Bowman, B., Storz, M.F., 2003. High accuracy satellite drag model (HASDM) review. *Advances in the Astronautical Sciences.* 116. 1943-1952.

3. Matsuo, T., Lee, I.-T., Anderson, J.L., 2013. Thermospheric mass density specification using an ensemble Kalman filter. *Journal of Geophysical Research.* Space Physics. 118. 1339-1350. 10.1002/jgra.50162.

4. Matsuo, T., 2014. Upper atmosphere data assimilation with an ensemble Kalman filter. In: Qian, L., Burns, A.G., Emery, B.A., Foster, B., Lu, G., Maute, A., Richmond, A., Roble, R.G., Solomon, S., Wang, W., 2014. *Modeling the Ionosphere-Thermosphere System.* 10.1002/9781118704417.ch7.

5. Mehta, P., Linares, R., 2017. A methodology for reduced order modeling and calibration of the upper atmosphere. *Space Weather.* 15 (10),1270-1287.

6. NOAA Space Weather Scales, 2011. Space Weather Prediction Center
www.swpc.noaa.gov/noaa-scales-explanation (accessed 10.1.18).

7. Psiaki, Mark. 2013. The blind tricyclist problem and a comparative study of nonlinear filters: A challenging benchmark for evaluating nonlinear estimation methods. *Control Systems, IEEE.* 33. 40-54.
10.1109/MCS.2013.2249422.

8. Qian, L., et al. (2014), The NCAR TIE-GCM: A community model of the coupled thermosphere/ionosphere system. In: *Modeling the Ionosphere-Thermosphere.* Geophysical Monograph Series, vol. 201, edited by J. D. Huba, pages 73-83., AGU, Washington, D. C, doi: 10.1002/9781118704417.ch7.

9. Richmond, A.D., Ridley, E.C., Roble, R.G., 1992. A thermosphere/ionosphere general circulation model with coupled electrodynamics. *Geophysical Research Letters.* 19, 601-604, doi:10.1029/92GL0040.

10. Storz, M.F., Bowman, B.R., Branson, J.I., Casali, S.J., Tobiska, W.K., 2005. High Accuracy Satellite Drag Model (HASDM). *Advances in Space Research*, 36 (12), 2497-2505.

11. Sutton, E., Cable, S., Lin, C., Qian, L., Weimer, D.R., 2012. Thermospheric basis functions for improved dynamic calibration of semi-empirical models. *Space Weather*. 10. 10.1029/2012SW000827.

12. Sutton, E., 2018. A New Method of Physics-Based Data Assimilation for the Quiet and Disturbed Thermosphere. *Space Weather*. 16 (6), 736-753.

13. Thompson, R., Space Weather Indices, Australian Government Bureau of Meteorology www.sws.bom.gov.au/
Educational/1/2/4 (accessed 10.1.18)

14. Vallado, D.A., 2013. *Fundamentals of Astrodynamics and Applications*. Space Technology Library.

Appendix B

A Partially Orthogonal EnKF Approach to Atmospheric Density Estimation using Orbital Debris

This was published in Advances in Space Research in 2020 [47].

B.1 Introduction

Our research proposes to utilize debris objects to sense their local space environment. A multi-object filter is used to extract the underlying non-conservative forces without a detailed physical description of the individual objects. The end goal is a data assimilation framework capable of handling high dimensional systems with nonlinear dynamics and leveraging all available observations. We focus on the low Earth orbital regime where mismodeling of atmospheric drag is the largest contributor to orbit prediction error. However, it is expected that the methods developed could be extended to additional orbital regimes for refinement of magnetic field and solar radiation models.

This work explores a problem in which we use multiple debris objects to estimate atmospheric density parameters, as well as the position and velocity of each debris object. The purpose of this article is to explore the combined orbit and density estimation problem to determine the system requirements, such as the number of ground sensors, measurement cadence, and ballistic coefficient knowledge. We also analyze the performance of the tool during a simulated geomagnetic storm in which the atmospheric density magnitude rapidly increases.

B.1.1 *Atmospheric Density*

An object in Low Earth Orbit (LEO) experiences atmospheric drag caused by particles in the atmosphere colliding with the surface of the object. Drag acts primarily in the opposite direction of the velocity vector to decrease the speed of an object. The magnitude of the force due to drag is directly dependent on neutral density (number of particles in the atmosphere). This is illustrated in the equation of acceleration due to drag,

$$a_{drag} = -\frac{1}{2}\rho\frac{C_D A}{m}v_{rel}^2\frac{\vec{v}_{rel}}{|\vec{v}_{rel}\ |} \tag{B.1}$$

where

$$\vec{v}_{rel} = \frac{d\vec{r}}{dt} - \vec{\omega}_{\oplus} \times \vec{r} \tag{B.2}$$

is the velocity vector relative to the rotating atmosphere, C_D is the coefficient of drag, A is the exposed cross-sectional area, m is the object's mass, $\vec{\omega}_{\oplus}$ is the mean motion of the Earth's rotation, and $\vec{r}$ is the Earth Centered Inertial object position vector [Vallado, 2013]. The drag coefficient, mass, and area of the object are typically combined to form the the ballistic coefficient ($\beta = \frac{C_D A}{m}$).

Atmospheric density is highly dynamic and depends on a number of factors, including solar cycle, diurnal cycle, geomagnetic activity, altitude, and latitude. Atmospheric models are used in orbital object prediction. A potential discrepancy between model and reality may lead to relatively large errors in orbit prediction. If a model assumes calm conditions, i.e. little to no geomagnetic activity, it produces an atmospheric density of about 0.0015 g/km^3 at 400 km altitude (using Ap and F10.7 indices of 17 and 127, respectively). On the other hand, when a geomagnetic storm hits, the density can rise more than an order of magnitude to 0.037 g/km^3 within minutes. This density value corresponds to input parameters, Ap and F10.7 indices, being set to 400 and 273, respectively in the MSIS (Mass Spectrometer and Incoherent Scatter) Atmospheric Model [Picone, et al., 2002]. Our research aims to estimate the atmospheric density in real time, specifically considering potential variability due to the diurnal cycle, latitude, and geomagnetic activity.

B.1.2 *Previous Work*

Current methods that estimate or model atmospheric density can be categorized as either empirical or physics-based approaches. Empirical models are computationally fast, but only represent the climatological atmospheric conditions. Physics-based models incorporate current knowledge of atmospheric conditions to provide forecasts, but require substantially more computing power than empirical models.

An example of an empirical approach is the High Accuracy Satellite Drag Model (HASDM) project. This effort estimates a time-series of thirteen spherical harmonic global density correction coefficients by using observations of 75-80 carefully selected calibration satellites (payloads and debris) in a batch fit [Bowman, Storz, 2002, 2003; Storz, 2005]. Intensive sensor tasking is made available for this effort, which allows for the collection of approximately 500 observations per day per calibration satellite. A batch fit solves for both temperature and density correction coefficients. HASDM decouples the ballistic coefficient and the density parameter by first solving for the long-term averaged ballistic coefficient of each satellite. This ballistic coefficient is computed by averaging almost 3200 previously estimated ballistic coefficients of each calibration satellite.

The Direct Density Correction Method (DDCM) project takes a slightly different approach by using Two-Line Element sets (TLE) of sixteen well-known objects [Yurasov, 2005, 2006]. DDCM empirically estimates two time-series density correction coefficients for both the MSIS Atmospheric Model [Picone, et al., 2002] and the Russian "Upper Atmosphere Model for Ballistic Calculations," known as GOST [Volkov, 2004], density models. The DDCM method includes secondary data processing in which smoothed orbits of each object and smoothed ballistic coefficients are estimated. This smoothing approach enables only long-period variations in density to be observable, which is a drawback of the DDCM effort.

The HASDM and DDCM empirical models do not allow for the spatial and temporal resolution necessary to capture the dynamical behavior of density. As described above, these methods estimate coefficients of density that are applied to a pre-defined model, not density directly. Furthermore, by combining information from objects in different regimes, or at different altitudes, a single batch fit limits the spatial resolution of density corrections that can be achieved.

More recently, density modeling and forecasting methods have been developed that take advantage of physics-based models. Matsuo [2014] showed that ionospheric electron density data can be used with a physics-based model to estimate and predict thermospheric mass density. Electron density data are assimilated with an Ensemble Kalman Filter (EnKF) to estimate input, or driver parameters of the physics-based Thermosphere-Ionosphere-Electrodynamics General circulation model (TIE-GCM). These drivers include thermospheric temperature and composition. Her work demonstrates that assimilation of COSMIC/FORMOSAT-3 electron density data can extend thermospheric mass density predictions produced by TIE-GCM by more than three days.

Mehta and Linares [2017] developed an approach that aims to combine the strength of a physics based model in its predictive capability, with the speed of an empirical model. In their approach, a reduced order model captures the upper atmospheric dynamics by using proper orthogonal decomposition to project a high dimensional system onto a set of low-order basis functions. This approach is demonstrated using an empirical model, MSIS, as the underlying upper atmosphere model with the intention that a physics-based model will replace MSIS in the future.

A physics-based approach that estimates atmospheric drivers, such as F10.7 and Kp indices, as inputs to TIE-GCM is described by Sutton [2018]. This effort estimates drivers through an iterative process that involves data assimilation of neutral density data from Challenging Mini-Satellite Payload (CHAMP). Each iteration initializes several instances of an Ionosphere-Thermosphere (I-T) model with an ensemble of drivers. An updated driver estimate is formed via the measurement update that assesses the agreement between each I-T model instance and the neutral density data. Once convergence between the I-T model and density data is reached, the driver estimates can be used with TIE-GCM to provide atmospheric forecasts.

Our proposed density estimation method is introduced in Section B.2. A comparison is presented between two nonlinear estimation filters, the Particle Filter (PF) and Ensemble Kalman Filter (EnKF), and the decision to use the Partially Orthogonal EnKF (POEnKF) as the chosen data assimilation tool is discussed. A simple example of a density estimation problem is introduced in Section B.3 to compare the performance of the PF and POEnKF. Section B.4 describes the full density estimation scenario and the performance of the PoEnKF when applied to this system is analyzed in Section B.5.

B.2 Method

Unlike current methods, our research estimates density from ground-based observations of object trajectories. This work also aims to take advantage of all debris objects, instead of a small, handpicked portion of the resident space object (RSO) population. This approach allows for a higher resolution density estimate. The resolution of the density estimate is dependent on its spatial parameterization and the available debris object distribution; details of the current parameterization scheme are given in Section B.4.1.

When limited to metric data, the ballistic coefficient and density terms cannot be decoupled if debris object physical characteristics are unknown, precluding the explicit estimation of density. (Interestingly, it has been shown that the ballistic coefficient and density terms can be decoupled if photometric data are utilized [Linares, et al., 2014].) The work presented here aims to utilize all debris objects, objects that are typically not characterized. To resolve this, a density calibration tool, such as HASDM, is leveraged to provide initial density estimates. These estimates are combined with debris object tracking information to solve for initial object ballistic coefficients. Once this “bootstrapping” process is complete, our density estimation method begins using the recently solved for ballistic coefficients.

B.2.1 *Particle Filter v. Ensemble Kalman Filter*

Nonlinear systems pose challenges in estimation that require special filtering methods and numerical techniques to handle non-Gaussian distributions that result from nonlinear dynamics and/or measurements. Two filters commonly used for nonlinear systems are the PF and the EnKF. The Particle Filter was introduced in 1993 and is known as a truly nonlinear filter because no Gaussian or linear assumptions are made in its formulation [Gordon, et al., 1993]. The EnKF was introduced to estimate high-dimensional systems in a computationally tractable manner [Evensen, 1994]. It does so by combining Kalman filter theory and Monte Carlo estimation methods [Houtekamer, Zhang, 2016].

An obvious commonality between the two filters is that they take advantage of a particle representation of the state PDF. The number of particles (N) required is an obvious difference between the filters. The PF requires at least 10^n (n is the number of state elements) weighted particles to capture the full non-Gaussian PDF, while the EnKF uses an ensemble of state realizations, or “particles,” of size greater than $2n+1$ (number of particles used in the Unscented Kalman Filter) and less than 10^n. In the EnKF, N is determined via a Monte Carlo study; there is no exact definition for N in the EnKF as there is in the PF and Unscented Kalman Filter. The number of particles in each filter is a direct indication of the level of nonlinearity the filter maintains. Although the PF maintains the full nonlinearity behavior of a system, it has a major obvious pitfall associated with the requirement of 10^n particles which becomes intractable for states with more than 5 elements (dimensionality curse). On the other hand, a well known strength of the EnKF is its ability to handle high-dimensional systems. Section B.3 introduces an example of a nonlinear high-dimension system, a density estimation problem, in which a comparison between the performance of a PF and EnKF will be discussed.

B.2.2 *The EnKF Approach*

The Ensemble Kalman Filter (EnKF) is typically used in high-dimensional nonlinear geophysical applications, such as forecasting of atmosphere and ocean systems, where there is an abundance of observations [Houtekamer, Zhang, 2016]. Our research applies the EnKF to a combination of a large-scale problem and a smaller scale problem. The latter is a typical orbit determination (OD) problem in which the position and velocity of the tracked objects are estimated. The large-scale portion is a high-dimensional geospatial estimation problem of the density field. Each of the N ensemble members are represented by $\chi_k^{(i)}$, where $1 < i < N$ and k denotes time-step. For this work, we use a variant of the Ensemble Kalman Filter, the POEnKF [Heemink, et al., 2000]; as described in the following.

B.2.3 Partially Orthogonal EnKF

Due to a lack of observability, the EnKF might suffer from a nonpositive definite covariance. A common solution to this issue is localization, which effectively localizes the measurement update to state elements close to the measurement. A smaller portion of the state correction is applied to elements farther from the measurement [Houtekamer, Zhang, 2016]. One example of a localization function is applying a ratio of the correction; the ratio is defined by an exponential function ($\mathcal{C}$), dependent on distance from the measurement:

$$\chi_k^{(i)+} = \chi_k^{(i)-} + \mathcal{C} * K \left(y_k^o - \mathcal{Y}^{(i)} \right) \tag{B.3}$$

where K is the Kalman gain, y^{o} is a uniquely perturbed observation, and $\mathcal{Y}$ is the predicted measurement [Gaspari, Cohn, 1999].

We use the MSIS Atmospheric Model [Picone, et al., 2002] to generate the density field through which we simulate debris object trajectories. Due to the globally coherent dynamical behavior of density represented by MSIS, localization is not an option. The POEnKF provides an alternate approach to localization that maintains the full-rank covariance. It approximates the initial full covariance matrix with a reduced rank matrix by selecting its q leading eigenvectors [Heemink et al., 2000]. The POEnKF is an effective solution to a weakly ranked covariance in a system like ours where the underlying model does not allow for localization. It does so by combining the EnKF and Reduced Rank EnKF (RREnKF) algorithms.

Following Heemink et al. (2000), the first step of the POEnKF is to select q, the number of leading eigenvectors to be used in the covariance approximation. This step helps form the L matrix, which is part of the RREnKF portion of the POEnKF. First, the initial covariance matrix is computed:

$$P_0^- = \frac{1}{N-1} E_0 E_0^T \tag{B.4}$$

where N is the number of ensemble members,

$$E_0^- = \left[\chi_0^{(1)-} - \overline{X}_0, \chi_0^{(2)-} - \overline{X}_0, ..., \chi_0^{(N)-} - \overline{X}_0\right] \tag{B.5}$$

and

$$\overline{X}_0 = \frac{1}{N} \sum_{i=1}^{N} \chi_0^{(i)} \tag{B.6}$$

The eigenvectors ($l_0^{(j)}$) and eigenvalues ($\lambda^{(j)}$) of the initial covariance matrix, P_0^-, are calculated and analyzed to select an appropriate number of leading eigenvectors for the reduced rank approximation.

$$L_0 = [l_0^{(1)}, l_0^{(2)}, ..., l_0^{(q)}\ ] \tag{B.7}$$

where

$$P_0^- \vec{v}^{(j)} = \lambda^{(j)} \vec{v}^{(j)} \tag{B.8}$$

and

$$l_0^{(j)} = \sqrt{\lambda^{(j)}}\ \vec{v}^{(j)} \tag{B.9}$$

L is a matrix of the q leading eigenvectors ($l_0^{(j)}$), normalized by the square root of the eigenvalues (Eqs. B.7 - B.9) [Heemink, et al., 2000]. The superscript j is used to denote indices of q leading eigenvectors; whereas, the superscript i is used to denote the indices of N ensemble members. The q selection for our particular system is described in Section B.4.5.

The POEnKF applies a linear update to a prior, non-Gaussian distribution, making it a partially nonlinear filter [Evensen, 2009]. The particles take on a non-Gaussian distribution in the case of a nonlinear problem and are not forced to form a Gaussian distribution at any point in the filter. However, the measurement update is based only on the covariance, making the update linear. As a result, the solution of the EnKF is described by Evensen as “something between a linear Gaussian update and a full Bayesian computation” [Evensen, 2009].

A Monte Carlo method is employed in the form of an ensemble representation of the estimated state probability distribution. The ensemble contains N members where each member is a sample realization of the state that is drawn from a normal distribution. Combined, the ensemble members represent the *a priori* distribution of the state and define the initial mean and variance of each state element. An ensemble representation of the initial state is generated using the *a priori* statistical information of the state X as shown in Eq. B.10.

The POEnKF ensemble initialization and forecast step are similar to the EnKF with the addition of a matrix L, the reduced rank, square root of the forecast covariance given in Eq. B.7 [Heemink, et al., 2000] as follows

$$\chi_0^{(i)} = \vec{X}_0 + \eta^{(i)} \tag{B.10}$$

where $\eta^{(i)}$ is a realization of the process noise.

$$\chi_k^{(i)-} = \mathcal{M}\left(\chi_{k-1}^{(i)+}\right) + w^{(i)} \tag{B.11}$$

$$w^{(i)} \sim \mathcal{N}(0, Q) \tag{B.12}$$

The POEnKF time update is the nonlinear propagation of the state ensemble (Eq. B.11) [Houtekamer, Mitchell, 2005], where $\mathcal{M}$ is the nonlinear forecast operator. Superscript (i) denotes the i$^{\text{th}}$ ensemble member. The L matrix is updated in the forecast step:

$$\overline{X}_{k-1} = \frac{1}{N}\sum_{i=1}^{N} \chi_{k-1}^{(i)+} \tag{B.13}$$

$$l_k^{(j)-} = \frac{1}{\epsilon}\left[\mathcal{M}\left(\overline{X}_{k-1} + \epsilon l_{k-1}^{(j)+}\right) - \mathcal{M}\left(\overline{X}_{k-1}\right)\right] \tag{B.14}$$

where ϵ is a perturbation that is set to equal 0.95. The L matrix represents the effect that a relatively small perturbation has on an ensemble member during the forecast step and is computed by Eq. B.14.

The prior covariance is computed as

$$P_k^- = \gamma L_k^- L_k^{-T} + \frac{1-\gamma}{N-1} E_k^{''-} E_k^{''-T} \tag{B.15}$$

where $0 \leq \gamma \leq 1$ is a weighting coefficient and a smaller γ is better for nonlinear systems ($\gamma = 0.4$ is used). Information from the POEnKF q leading eigenvectors is provided by the first term, $L_k^- L_k^{-T}$ (RREnKF portion of the POEnKF), and information from the full ensemble is incorporated via the second term, $E_k^{''-} E_k^{''-T}$. $E_k^{''-}$ is the projection of the ensemble members onto the q leading eigenvectors and is computed as follows:

$$E_k^{-T} E_k^- = X \Omega X^T \tag{B.16}$$

$$E_k^{''-} = \left[E_k^- X \right]_{:,1:q} \tag{B.17}$$

It should be noted that in the implementation of Eq. B.15 shown in Heemink et al. [2000], there is an additional contribution of the covariance projected in directions orthogonal to the space spanned by the ensemble $l_k^{(j)-}$. This term is intended to rectify the problem of rank-deficiency of the ensemble-based covariance matrix by modifying the covariance matrix in a way to increase its rank [Evensen, 2009]. Our filtering experiments have performed well without an inclusion of this additional covariance contribution. If issues of rank-deficiency in the context of this specific application arise, an effective way to remedy them will be investigated in the future.

The state measurement update is analogous to the EnKF with the addition of the L matrix and modified covariance update:

$$y_k^o = \tilde{y}_k + \epsilon \tag{B.18}$$

$$\epsilon \sim \mathcal{N}(0, R) \tag{B.19}$$

$$\mathcal{Y}_k^{(i)} = \mathcal{G}\left(\chi_k^{(i)-}\right) \tag{B.20}$$

$$\chi_k^{(i)+} = \chi_k^{(i)-} + K\left(y_k^o - \mathcal{Y}^{(i))}\right) \tag{B.21}$$

$$\vec{X}_k = \frac{1}{N}\sum_{i=1}^{N}\chi_k^{(i)+} \tag{B.22}$$

where $\mathcal{G}$ is the measurement equation as a function of the state, $\mathcal{Y}$ is the predicted measurement, and R is the measurement covariance matrix. The measurement update for each POEnKF member (Eq. B.18) incorporates a uniquely perturbed observation, $\mathrm{y^o}$ for each ensemble member. This is a signature feature of the stochastic EnKF not used in other nonlinear filters. The Kalman gain is calculated in a conventional manner as follows

$$K = P_k^- H^T \left[HP_k^- H^T + R\right]^{-1} \tag{B.23}$$

where H is the Jacobian matrix, i.e. the partial derivative of the measurement models ($\vec{Y}$) with respect to the state

$$H = \left[\frac{\partial\vec{Y}}{\partial\vec{X}}\right] \tag{B.24}$$

and the covariance update is computed as follows:

$$\tilde{L}_k^+ = [\,(I - KH)\,L_k^-,\ KR^{\frac{1}{2}}\,] \tag{B.25}$$

$$\tilde{L}_k^{+T}\tilde{L}_k^+ = X\Omega X^T \tag{B.26}$$

$$L_k^+ = \left[\tilde{L}_k^+ X\right]_{:,1:q} \tag{B.27}$$

$$E_k^{+T}E_k^+ = X\Omega X^T \tag{B.28}$$

$$E_k^{''+} = \left[E_k^+ X\right]_{:,1:q} \tag{B.29}$$

$$P_k^+ = \gamma L_k^+ L_k^{+T} + \frac{1-\gamma}{N-1} E_k''^+ E_k''^{+T} \tag{B.30}$$

B.3 Simplified Example of Density Estimation Problem

A 2D simplified version of an OD and atmospheric density estimation problem is created to illustrate the POEnKF and the PF for high dimensional systems. The full version of the density estimation problem is introduced in Section B.4.

B.3.1 *System Details and Initialization*

A scenario is created in which a single orbital debris object is utilized to estimate atmospheric density along its trajectory. The debris object trajectory is propagated using two-body Keplerian dynamics with the addition of acceleration due to atmospheric drag,

$$\ddot{\vec{X}} = \frac{\mu}{\vec{r}^3}\vec{r} - \frac{1}{2}\rho\frac{C_D A}{m} v_{rel}^2 \frac{\vec{v}_{rel}}{|\vec{v}_{rel}\,|} \tag{B.31}$$

where $\vec{v}_{rel}$ is defined in Eq. B.2. The debris object is in a near circular, equatorial orbit at 400 km altitude; range and range-rate measurements of the object are collected from an inertially fixed sensor located at [0, R_{Earth}, 0].

For this simplified system, the atmospheric density is modeled as a partial sinusoid that varies as a function of Local Sidereal Time (LST) with a peak in density at local noon (LST 12),

$$\rho = A * sin(B * LST) + C \tag{B.32}$$

Fig. B.1 shows the nominal density state as a function of LST. The debris object is simulated for a duration of 400 minutes. The true density through which the object passes is shown in Figure B.2.

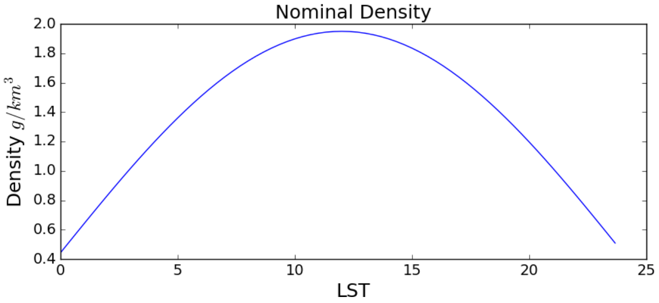

Figure B.1: True Density vs. LST Generated from Eq B.32, where A = 1.5 g/km^3, B = $\frac{2\pi}{48}$ hours, and C = 0.4 g/km^3

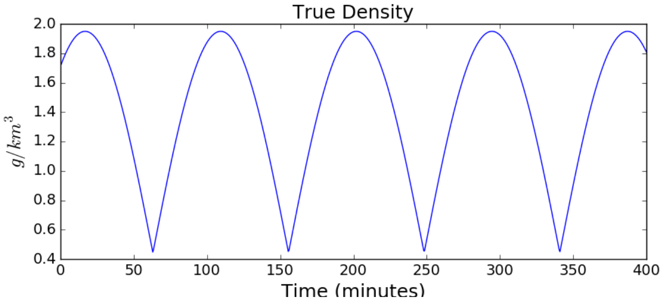

Figure B.2: Underlying True Density vs. Time

The goal of this problem is to estimate the density field that the debris object is orbiting through, along with the X and Y position and X and Y velocity of a debris object (OD state). The density field portion of the state is located at the equator at the altitude corresponding to the debris object orbit, and is parameterized by LST with a 5 degree resolution (72 elements).

A perturbation is applied to the true initial OD state so that the *a priori* state in the filter is not equal to the true initial state,

$$\vec{X}'_{OD\ 0} = \vec{X}_{OD\ 0} + p_{OD} \tag{B.33}$$

where the perturbation (p_{OD}) has a nonzero mean ($\mu_{pert\ OD}$):

$$p_{OD} \sim \mathcal{N}(\mu_{pert\ OD}, P_{pert\ OD}) \tag{B.34}$$

$$\mu_{pert\ OD} = [10m, 10m, 0.1m/s, 0.1m/s]^T \tag{B.35}$$

$$P_{pert\ OD} = diag\left(\left[(1m)^2, (1m)^2, (.01m/s)^2, (.01m/s)^2\right]\right) \tag{B.36}$$

The OD state ensemble is generated using the perturbed initial state ($\vec{X}'_{OD\ 0}$) and its *a priori* statistical information (P_0):

$$\chi^{(i)}_{OD\ 0} = \vec{X}'_{OD\ 0} + \eta^{(i)} \tag{B.37}$$

where,

$$\eta^{(i)} \sim \mathcal{N}(0, P_0) \tag{B.38}$$

and

$$P_0 = diag\left(\left[(10m)^2, (10m)^2, (.1m/s)^2, (.1m/s)^2\right]\right) \tag{B.39}$$

Similar to the OD state initial estimate, the initial filter estimate is perturbed from the true initial density:

$$\vec{X}'_{\rho\ 0} = \vec{X}_{\rho\ 0} + p_\rho \tag{B.40}$$

where,

$$p_\rho \sim \mathcal{N}(\mu_{pert\ \rho}, P_{pert\ \rho}) \tag{B.41}$$

$$\mu_{pert\ \rho} = [0.1, 0.1, ..., 0.1]\, g/km^3 \tag{B.42}$$

$$P_{pert\ \rho} = diag\left(\left[(5e^{-2})^2, (5e^{-2})^2, ..., (5e^{-2})^2\right]\right)(g/km^3)^2 \tag{B.43}$$

Next, the initial density state ensemble is generated by varying the vertical shift of the perturbed initial state ($\vec{X}'_{\rho\ 0}$) sinusoid. This approach reflects the behavior of a density ensemble generated via the MSIS model [Picone, et al., 2002], where input parameters simply control the magnitude of the density state, but do not affect its spatial variability. The resulting density ensemble is shown in Fig. B.3; each line represents the density state of one ensemble member.

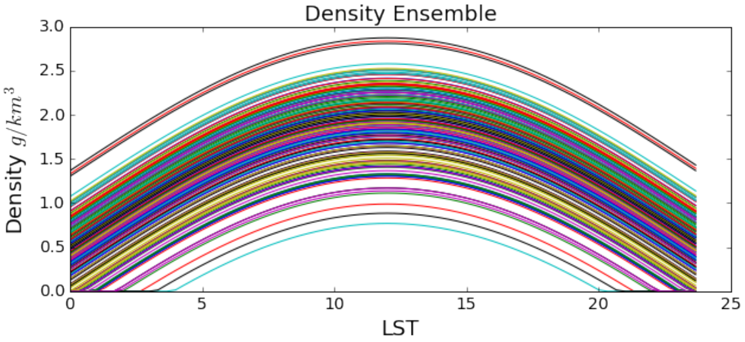

Figure B.3: Density Ensemble Generated from Perturbed Initial State

With 76 state elements required, an ensemble size (N) of 150 members is used in the POEnKF. The theoretically required number of particles for the PF (10^{76} particles) causes the filter to be computationally intractable for this high dimensional system, so 10^4 particles are used in the PF implementation. This number of particles is chosen because the filter can run in about double the amount of time as the POEnKF.

B.3.2 *Results and Analysis*

Both the PF and POEnKF are applied to this simplified density estimation system and their performance is compared. Figure B.4 shows the PF density estimate percent error. The estimate is more accurate in the beginning of the simulation but then quickly veers off track. This is due to the insufficient number of particles used (10^4 instead of 10^{76} particles). The evolution of the PF particles is shown in B.6. It is apparent that the particle distribution quickly collapses and, therefore underrepresents the full state space probability (Fig. B.5). This causes the covariance to collapse and the filter to discard information from the measurements.

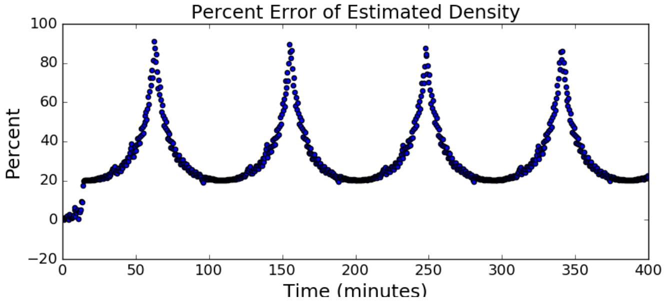

Figure B.4: PF Density Estimate Percent Error

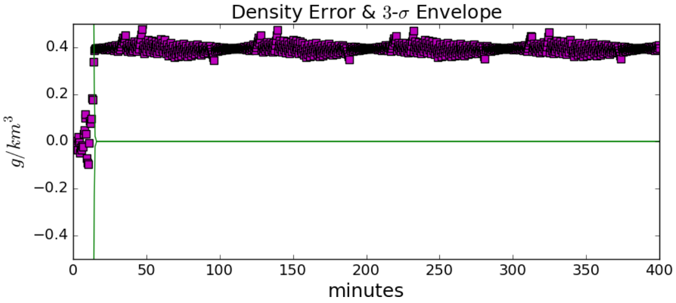

Figure B.5: PF Density Error and 3-σ Envelope

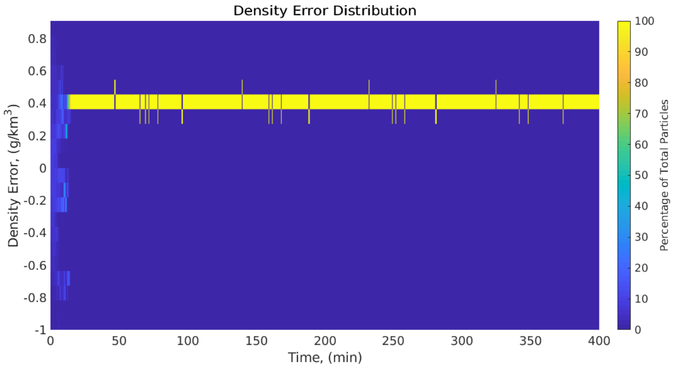

Figure B.6: PF Particle Evolution

Unlike the PF, the POEnKF is able to recover the true density to within 10% error after 100 minutes, as shown in Figure B.7. The POEnKF's particle, or ensemble member distribution accurately captures the probable state space and the estimate error magnitude.

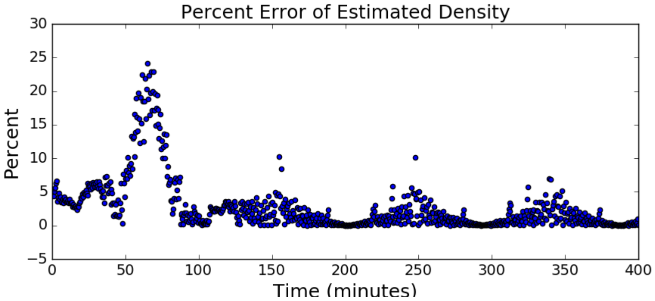

Figure B.7: POEnKF Density Estimate Percent Error

The evolution of ensemble members is shown in Figure B.8 with the same axis bounds as the PF (Fig. B.6). After estimate convergence at 100 minutes (shown in Fig. B.7), the distribution of particles still reveal a relatively large error that occurs cyclicly. The cyclic behavior of the error shown in Figure B.8.b (the zoomed in version of Fig. B.8.a) occurs when the true density has the smallest magnitude (Fig. B.2). This occurs on the night-side near 0 degrees LST. The filter performance is degraded in regions of low density because a small density has relatively little effect on the object orbit and is therefore less observable.

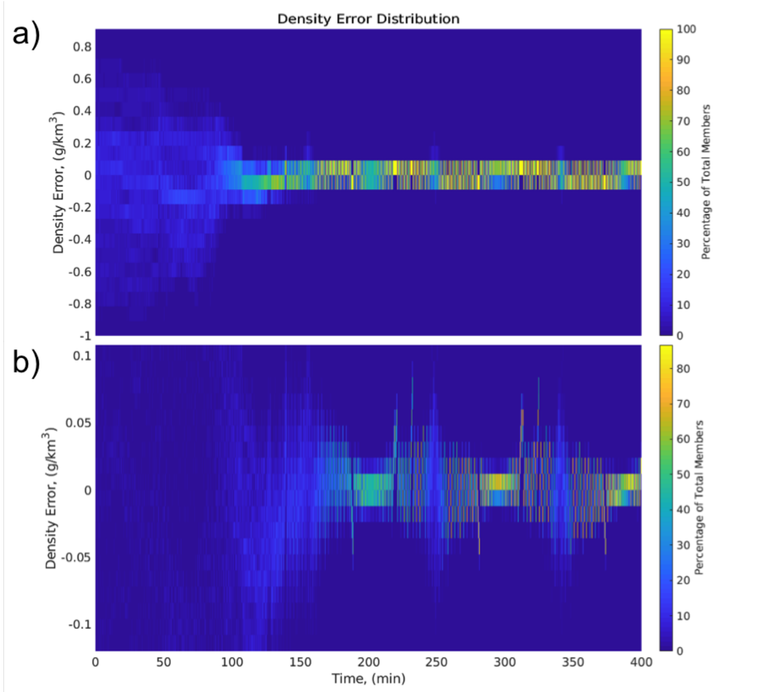

Figure B.8: POEnKF Particle Evolution Zoomed

A direct comparison between the performance of the PF and POEnKF is provided in Figure B.9, which shows their density estimation errors as a function of time. The black and red dotted lines represent the final errors of the PF and POEnKF, respectively. As seen in the earlier figures, the PF estimate diverges about 15 minutes into the run with the collapse of its covariance, and thereafter ignores the measurements and never recovers. The POEnKF performs properly, accumulating information from the measurements throughout the run, and settling near the truth with errors that are well described by the 3-sigma bounds.

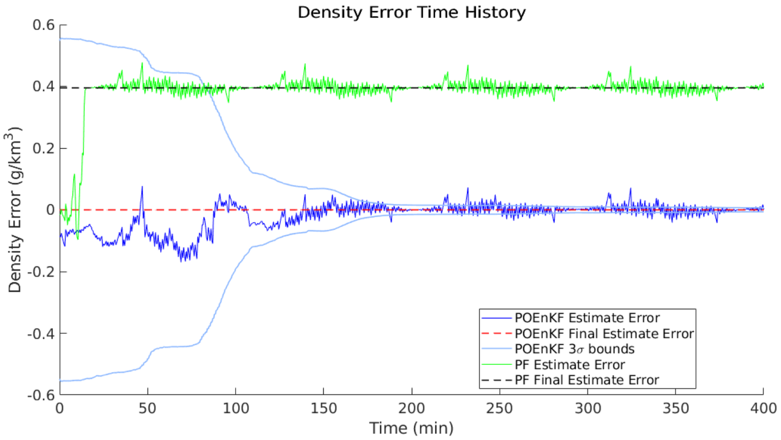

Figure B.9: PF and POEnKF Performance Comparison

This simplified version of the extended state density estimation problem illustrates the advantage of utilizing the POEnKF over the PF for high dimensional systems.

B.4 Full Density Estimation System

This section presents a more complete version of the density estimation approach. The scenario includes multiple debris objects orbiting at a fixed LEO altitude with various inclinations, right ascension of ascending nodes, etc., and therefore, expands the problem space from a single equatorial orbital path to a spherical shell encompassing Earth. Similar to the simplified example, the debris objects are propagated using two-body Keplerian dynamics with the addition of acceleration due to atmospheric drag (Eq. B.31). The partial sinusoid as the underlying atmospheric density model is replaced by the MSIS Atmospheric Model [Picone, et al., 2002].

B.4.1 *System Details*

The estimated states in this scenario are: the position and velocity of multiple orbital debris objects and a 2D grid of atmospheric density. We model the spherical shell of neutral density at 400 km altitude with a spatial grid parameterized by LST and latitude. An altitude of 400 km is used here, similar to Emmert [2008], Morozov [2013], and Pilinski [2016]. These parameters define a spatial grid in the sun-fixed coordinate frame. Both LST and latitude have a spatial resolution of five degrees. The resulting vector of densities, one at each grid point, contains 2701 elements, yielding a total of $2701 + 6D$ estimated elements when combined with the orbit states of each object. The grid of density elements is defined with respect to the sun in order to account for one of the major contributors of density variability, the diurnal cycle. Due to this design, the density state is modeled as stationary, requiring no forecast step for the density portion of the state, i.e. the density state portion of $\mathcal{M}$ in Eq. B.11 is identity.

The debris objects simulated for this project are inspired by the objects used in the HASDM effort. Orbital characteristics (inclination, right ascension of ascending node, etc.) of 16 HASDM objects [Bowman, Storz, 2002, 2003] are simulated; however, their eccentricity is set to zero and altitude is 400 km in order to fall within the defined problem space. For example, Vanguard 2 (satellite number 00011) has a 32.8 deg inclination, 45.2 deg right ascension of ascending node, 167.0 deg argument of perigee, and is set to have a 400 km altitude and zero eccentricity. Each object is simulated with an area to mass ratio (AMR) of 0.0955 m^2/kg. Nominally, the filter also assumes this AMR value for the mass of each object (assumes the true AMR); however, an evaluation of the effect of errors in the assumed AMR is described in Section B.5.6.

Azimuth (az), elevation (el), and range (R) measurements of debris objects are generated from three ground stations. The HASDM effort is able to execute a dedicated sensor tasking campaign in which 500 measurements per object per day are collected. This equates to a measurement of each object every three minutes. Therefore, we choose a three minute measurement cadence for these simulations.

B.4.2 *Scenario Use-Cases*

Six scenarios are explored in Sections B.5.1-B.5.7. A brief description and the resulting root mean squared (RMS) density error is provided for each case in Table B.1.

Table B.1: Case Study Comparison

Case	RMS Density Error (g/km^3)
Single Object	1.2
- one object	
- sensor error: 0.15 arc-seconds for az & el,	
0.1m for range	
- measurement cadence: 3 minutes	
- AMR: 0.0955 m^2/kg	
Multi-Object Baseline	0.14
- 16 objects	
- all objects AMR: 0.0955 m^2/kg	
Larger Sensor Error	0.42
- sensor error: 1 arc-second for az & el,	
0.5m for range	
Reduced Measurement Cadence	0.28
- measurement cadence: 6 minutes	
Unknown Ballistic Coefficient	0.21
- each object AMR randomly generated	
with particular mean & variance	
Geomagnetic Storm	3.47
- extreme geomagnetic storm	

The ground track of the orbital debris in the single object scenario is shown in Figure B.10. Red stars signify three ground stations for which measurements are generated. The stations located in Canberra, Madrid, and Diego Garcia are referred to as station 1, station 2, and station 3, respectively. The ground-tracks of the 16-object scenario are shown in Figure B.11. The object trajectory is simulated for roughly one orbital period. This scenario is used for the majority of our analysis (Sections B.5.3 - B.5.7).

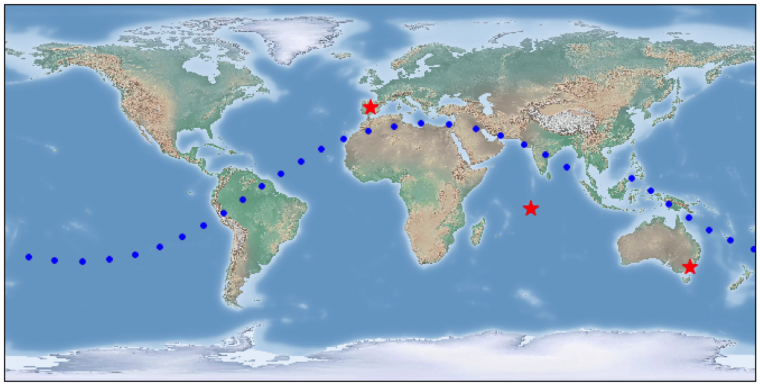

Figure B.10: Groundtrack of 90 minute trajectory for Single Debris Object Scenario

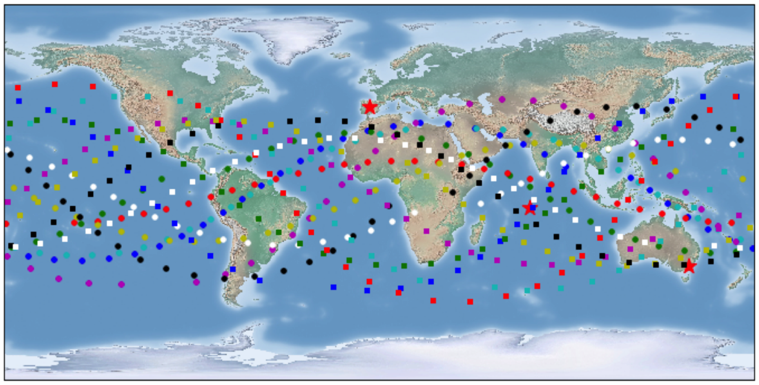

Figure B.11: Groundtrack of 90 minute trajectory for 16 HASDM inspired objects

Figure B.12 shows the LST of each object at the times when a measurement is collected of that object. It is apparent that there is a gap in the LST of the measurements, which shifts throughout the 24 hour simulation. This measurement gap is caused by the absence of sensor stations on the western side of the globe (Fig. B.11). As the local time for each ground station progresses throughout the day, so does the LST of the measurements collected by each ground station. This affects the estimation error for density state elements located at local sidereal times for which there are gaps. For example, during the first 250 minutes of the simulation there are no measurements taken of objects located between 0 and 150 deg LST. The effect of the measurement gap in LST is discussed further in Section B.5.3.

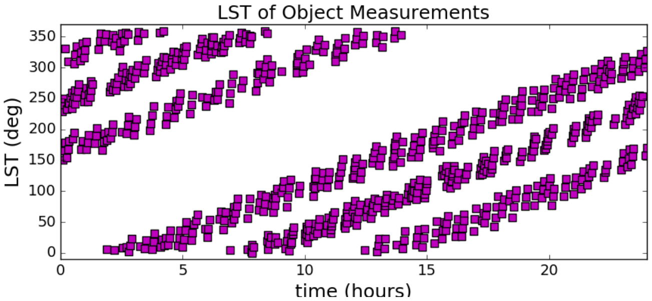

Figure B.12: Local Sidereal Time of all Object Measurements during 24 hour Multi-Object Scenario

B.4.3 *System Initialization*

As in the simple model, a perturbation is applied to the true initial OD state so that the *a priori* state in the filter is not equal to the true initial state,

$$\vec{X}'_{OD\ 0} = \vec{X}_{OD\ 0} + p_{OD} \tag{B.44}$$

where the perturbation has a nonzero mean that is drawn from a normal distribution:

$$p_{OD} \sim \mathcal{N}(\mu_{pert\ OD}, P_{pert\ OD}) \tag{B.45}$$

$$\mu_{pert\ OD} = [100m, 100m, 100m, 0.1m/s, 0.1m/s, 0.1m/s]^T \tag{B.46}$$

$$P_{pert\ OD} = diag\left(\left[(1m)^2, (1m)^2, (1m)^2, (0.01m/s)^2, (0.01m/s)^2, (0.01m/s)^2\right]\right) \tag{B.47}$$

The MSIS model generates a density value when provided with various input parameters. These parameters include latitude, LST, altitude, day of year (DOY), F10.7 index, and the Ap index. Latitude and LST are inherently defined for each value of the density state in the parameterization scheme.

Table B.2: MSIS Parameters for Nominal Density

Parameter	DOY	Ap Index	F10.7 Index	Altitude
Value	175	17	127	400 km

June 24th is the date chosen for the simulation (DOY = 175). The true density field can be generated with the nominal inputs provided in Table B.3. An equatorial slice of the true underlying density field generated using these nominal input parameters is shown in Figure B.2.

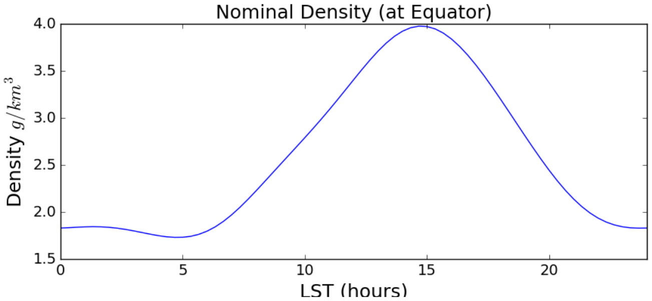

Figure B.13: Nominal Density State at Equator

Similar to the OD state, a perturbation is applied to the true initial density state to produce the filter's *a priori* estimate:

$$\vec{X}'_{\rho\ 0} = \vec{X}_{\rho\ 0} + p_\rho \tag{B.48}$$

$$p_\rho \sim \mathcal{N}(\mu_{pert\ \rho}, P_{pert\ \rho}) \tag{B.49}$$

$$\mu_{pert\ \rho} = \left[(0.1)^2, (0.1)^2, ..., (0.1)^2\right] g/km^3 \tag{B.50}$$

$$P_{pert\ \rho} = diag\left(\left[(0.5)^2, (0.5)^2, ..., (0.5)^2\right]\right)(g/km^3)^2 \tag{B.51}$$

B.4.4 Ensemble Generation

The number of ensemble members, N, was selected via a Monte Carlo study of the density state error distribution as a function of N. It was determined that a 100-member ensemble is sufficient to represent the full state of 2701 + 6D elements, where D is the number of debris objects. The method for generating a randomly distributed ensemble is different for the two portions of our estimated state.

To generate the initial ensemble ($\chi^{(i)}_{OD\ 0}$) for the position and velocity portion of the state, 100 samples are drawn from a distribution with $\vec{X}'_{0,OD}$ mean and P_0 variance using Eq. B.10. The standard deviations in P_0 correspond to the perturbations applied to the initial ensemble ($\mu_{pert\ OD}$),

$$P_0 = diag\left(\left[(100m)^2, (100m)^2, (100m)^2, (.1m/s)^2, (.1m/s)^2, (.1m/s)^2\right]\right) \tag{B.52}$$

The density ensemble is generated with a different approach because an appropriate covariance of the density random variable is unknown. In this case, MSIS is used to generate density values that represent the probable distribution statistics. To do so, we begin by defining an array of values for the latter four variables (DOY, F10.7 index, Ap index, and altitude).

$$\begin{aligned}
&\text{DOY} &&= [169, 171, 173, 175, 177, 179, 181]\\
&\text{F10.7} &&= [67, 83, 102, 124, 148, 172, 196]\\
&\text{Ap index} &&= [0, 4, 7, 14, 27, 48]\\
&\text{altitude} &&= [395, 396, 397, 398, 399]
\end{aligned}$$

Each parameter array is randomly sampled 100 times to generate 100 combinations of input to MSIS. Each combination produces a density ensemble member (Fig. B.14). Varying these parameters produces a range of density realizations encompassing the likely true densities. A comparison between the mean of the density ensemble and the true nominal density is provided in Figure B.15.

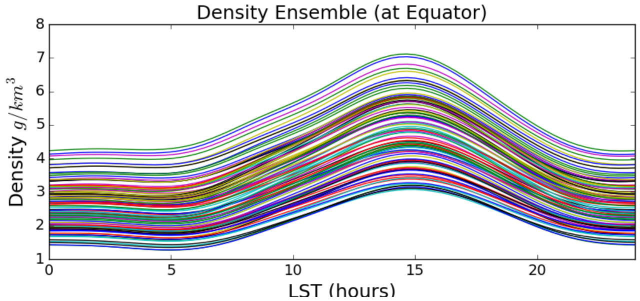

Figure B.14: Density Ensemble

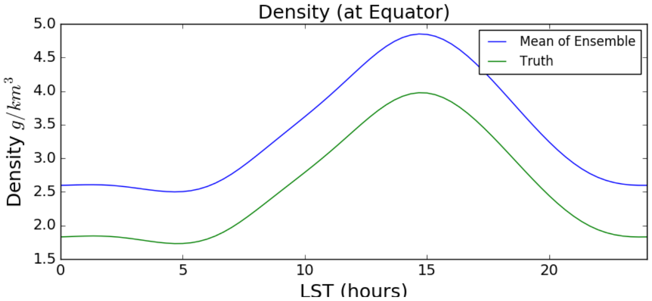

Figure B.15: Density Ensemble Mean and Density Truth

B.4.5 q Eigenvectors Selection

The number of leading eigenvectors used for the reduced rank representation of the full ensemble in the POEnKF is determined via an analysis of the initial ensemble covariance eigenvalues. It was found that including components with eigenvalues of 1e-6 and higher is sufficient. This suggests that $q = 26$ is appropriate for the state covariance reduced rank approximation in the single object scenario and 60 leading eigenvectors for the 16 object scenario.

Proper selection of q is important because selecting a relatively small q greatly reduces computation time. However, too small of a q can also cause filter divergence because the reduced rank approximation will not adequately represent the full state behavior [Heemink, et al., 2000].

B.5 Analysis and Results

This section presents a series of scenarios in which various use cases for our tool are analyzed. These cases demonstrate the sensitivity of the POEnKF to various conditions such as constant verses nominal surveillance, single verses multi-object, sensor error, measurement cadence, knowledge of the objects' AMR, and a geomagnetic storm.

B.5.1 Constant Surveillance: Single Object

This simulation includes a single debris object and constant surveillance in which there is a measurement every three minutes regardless of physical constraints or observability conditions. It is noted that due to the behavior of MSIS, the information from a single debris object at a particular location on the density grid inherently provides information for all density grid elements. The debris object completes three orbital periods throughout the simulation. There is a larger error in the density estimate during the first orbital period (first 90 minutes), but subsequently the solution converges and does not exceed 20% error nor the 3-σ bounds. Figure B.16 shows the percent error of the density estimate with each estimate's corresponding measurement station identified. Figure B.17 is the error of the density estimate with the same labels used to identify the station from which that estimate's measurement originated. This figure provides the estimate after the time update (black dot markers), as well as the estimate after the measurement update (square, triangle, and diamond markers indicating the ground station).

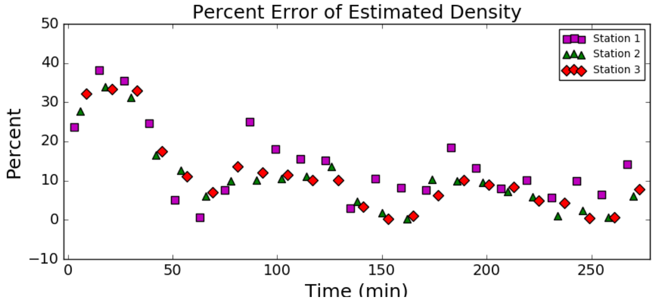

Figure B.16: Single Object Constant Surveillance Density Estimate Percent Error

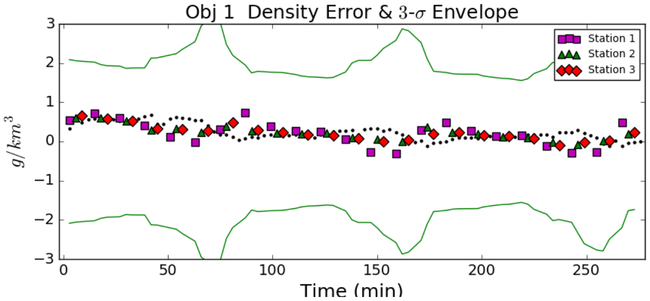

Figure B.17: Single Object Constant Surveillance Density Error and 3-σ Envelope

The overall RMS density error for this simulation is 0.32 g/km^3 out of an average 2.68 g/km^3 density. The RMS density error post-convergence (after 90 minutes) is 0.22 g/km^3. This post-convergence RMS density error is more indicative of the performance of the filter and will therefore be the error reported and used for comparison in the remaining scenarios (Sections B.5.2 - B.5.7). The performance of the OD portion of the state is illustrated below via the X position and X velocity error and covariance (Fig. B.18). The remainder of the state, the Y and Z position and Y and Z velocity, have similar performance with small RMS errors of 2 m, 2 m, 3 cm/s, and 4 cm/s, respectively.

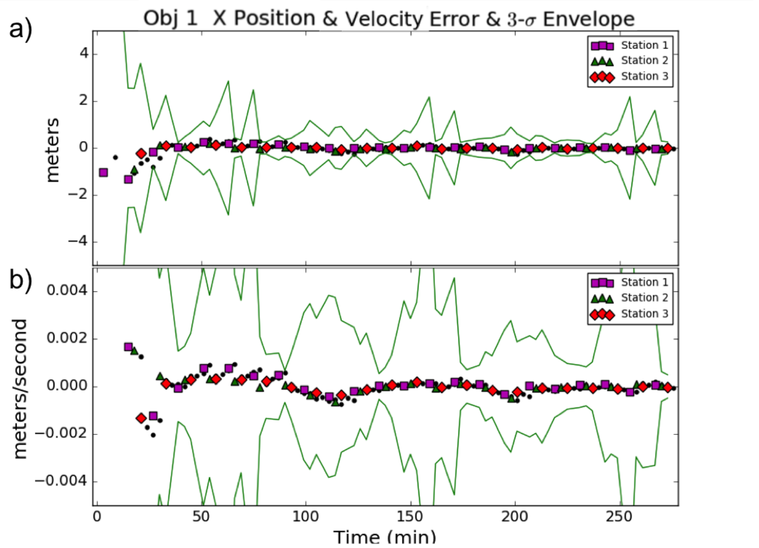

Figure B.18: Single Object Constant Surveillance X Position and Velocity Error and 3-σ Envelope

B.5.2 Nominal Surveillance: Single Object

For this simulation the debris object measurements are generated only when the debris object is visible from a ground station. Therefore, the measurements do not occur every three minutes consistently throughout the simulation. This simulation is run for the same duration as the simulation in Section B.5.1. As expected, the performance is worse than that with constant surveillance. The estimated RMS density error is 1.2 g/km^3 for nominal surveillance compared to 0.22 g/km^3 for constant surveillance. The estimated density error is shown in Figures B.19-B.20. Figure B.21 shows the ground track of the debris object measurements as seen by the three stations indicated by red stars.

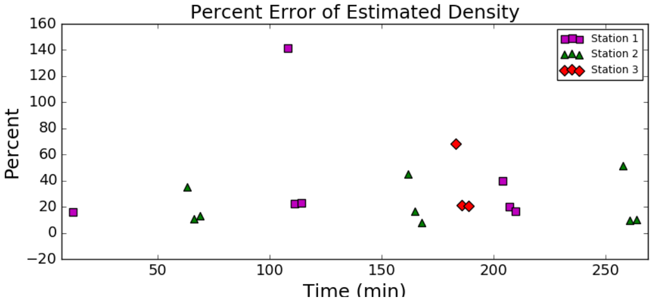

Figure B.19: Single Object Nominal Surveillance Density Estimate Percent Error

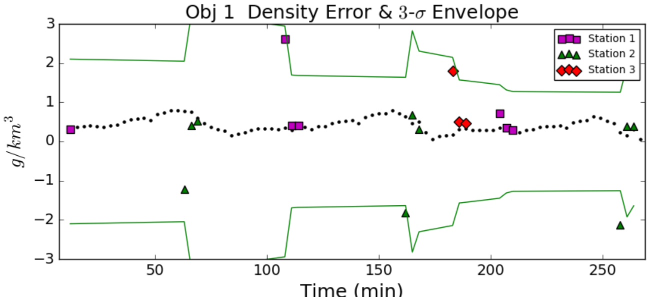

Figure B.20: Single Object Nominal Surveillance Density Error and 3-σ Envelope

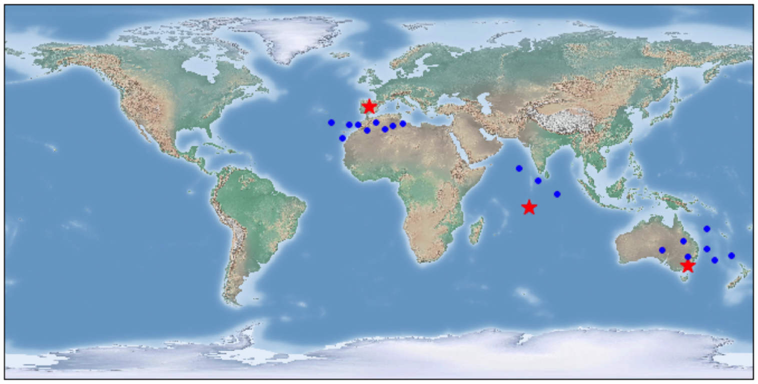

Figure B.21: Single Object Nominal Surveillance Ground Track of Debris Object Measurements

This scenario demonstrates the need for measurements of multiple debris objects in order to estimate density to the desired level of accuracy. It is not possible to consistently task sensors to "stare" at debris objects and collect hundreds of measurements of each object every day, as is done for the HASDM effort. We aim to develop a tool capable of using sparse debris data to estimate density; our advantage being that there are thousands of debris objects to utilize for this purpose. Section B.5.3 demonstrates the substantial improvement in the density estimate when data from 16 debris objects are utilized.

B.5.3 Nominal Surveillance: Multiple Objects

This scenario is similar to that of Section B.5.2, but with 16 HASDM-inspired objects instead of one. These objects vary in inclination, right ascension of ascending node (RAAN), and argument of perigee. This scenario is referred to as the baseline case. The overall RMS error of the density estimate is greatly reduced when using 16 debris objects compared to one object (Table B.1).

Figure B.22 shows that the density estimate percent error is relatively large in the beginning of the simulation, but the filter is able to converge to a relatively low error for the remainder of the simulation. Figure B.23 shows the updated density error for the element closest to the measurement (square, diamond, and triangle symbols) and the density error of the density elements located along the Object 1 orbital path (black dots).

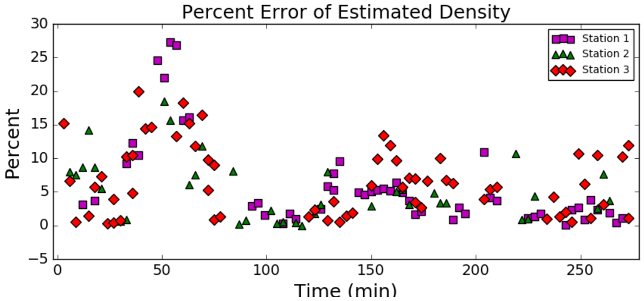

Figure B.22: Multi-Object Nominal Surveillance Density Estimate Percent Error

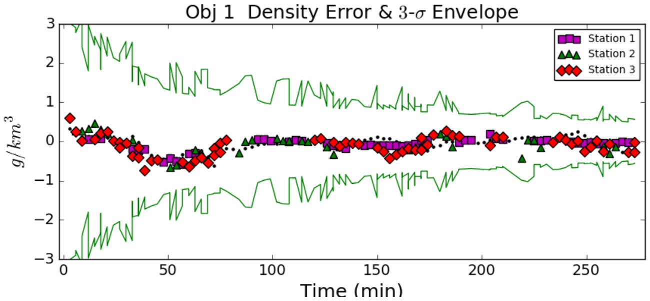

Figure B.23: Multi-Object Nominal Surveillance Density Error and 3-σ Envelope

Figure B.24 shows the percent error of the density estimate as a function of LST. It was shown in Section B.3.2 via the Simplified Problem that the filter has the smallest percent error in the density estimate when going through higher density regions, because the effect of density on the object trajectories is more observable in higher density regions. Therefore, it was expected that the filter would have the lowest percent error at local sidereal times near noon (highest density) for this multi-object scenario. However, we find that the lowest percent error in the density estimate actually occurs when measurements are taken of debris objects in *low density regions* (near zero deg LST). This is contradictory to what is expected, but is explained by the gap in measurements between 40 and 150 deg LST. The measurements taken directly after the gap have a high percent error when the lowest percent error is what is expected. If there were consistent measurements throughout all LST values, then the lowest percent error in the density estimate would occur around noon (180 deg LST) as seen in the Simplified Example. The cause of this measurement gap in LST was described in Section B.4.2. The ground track of the measurements for each of the sixteen objects is shown in Figure B.25.

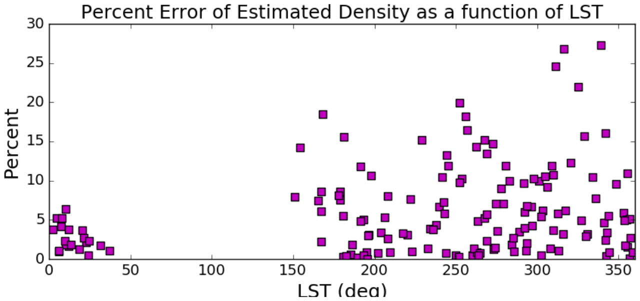

Figure B.24: Density Percent Error as a function of LST

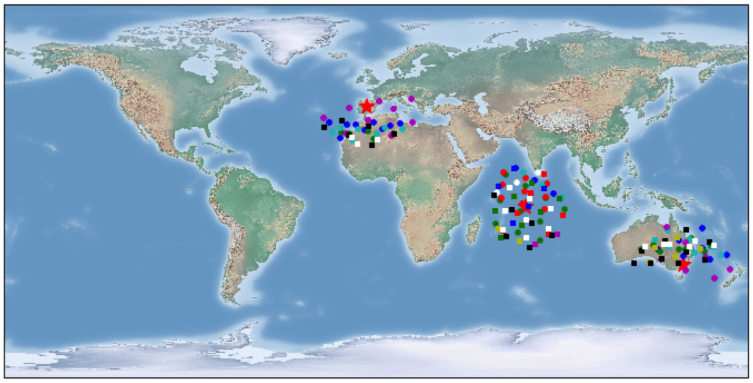

Figure B.25: Multi-Object Nominal Surveillance Ground Track of Debris Objects Measurements

The RMS density error over the entire grid, for each measurement time is shown in Figure B.26. This plot is indicative of the overall performance of the filter because it captures the filter's performance over the entire density state-space. Previous figures focused on the RMS error for a particular density element, or location on the spatial grid, as a function of time. It is apparent that the full density state error follows the same trend shown in Figure B.22, converging after 90 minutes, or one orbital period.

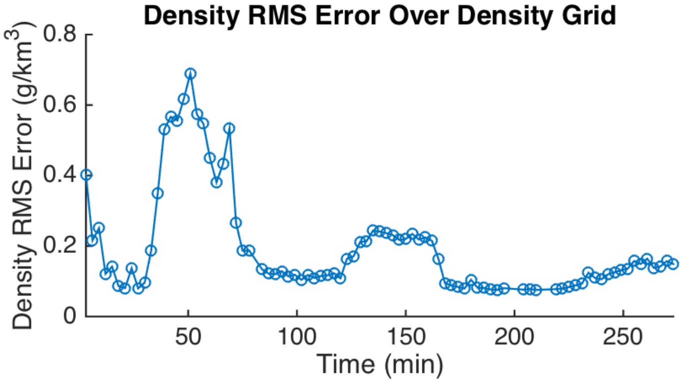

Figure B.26: The RMS density error over the entire spatial grid as a function of time for the Baseline Case.

B.5.4 Sensor Error Study

This scenario introduces a larger sensor error into the baseline multi-object scenario, demonstrating the sensitivity of the density estimate to measurement error and noise. The azimuth and elevation sensor error standard deviation is increased from 0.15 arc-seconds to 1 arc-second. Likewise, the range sensor error is increased from a standard deviation of 0.1 meters to 0.5 meters.

It is expected that the density estimate error increases with the same magnitude of the sensor error. Because density has a relatively small contribution to the debris object dynamics, any additional sensor error will impact the accuracy of the density estimate. This is shown in Figure B.27. The error in the density has increased throughout the simulation compared to the baseline case.

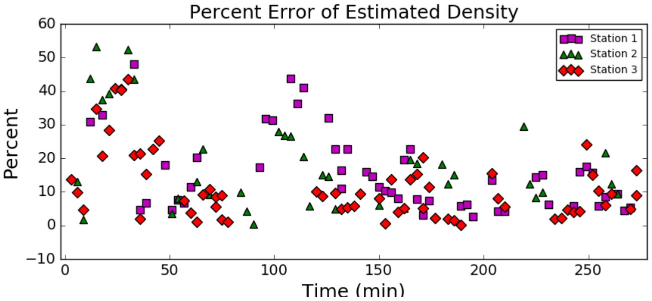

Figure B.27: Sensor Error Study Density Estimate Percent Error

B.5.5 Measurement Cadence Study

To study the effect of measurement cadence, we increased the time between measurements from three minutes to six minutes. Even though there are measurements from sixteen objects, the measurements are more sporadic. The overall RMS density error increases from 0.14 g/km^3 for the baseline multi-object scenario to 0.28 g/km^3.

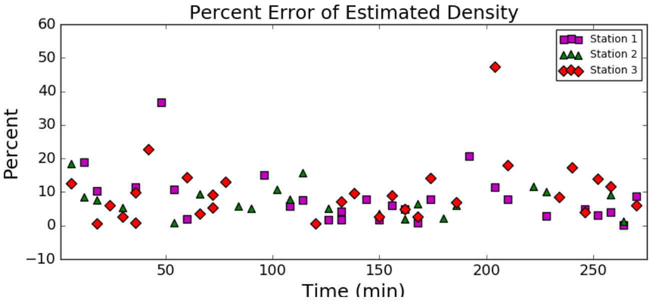

Figure B.28: Measurement Cadence Study Density Estimate Percent Error

B.5.6 Unknown Ballistic Coefficient

In each of the prior scenarios, the filter assumes it knows the true AMR of each object (0.0955 m^2/kg). This scenario simulates objects of different AMRs while the filter still assumes an AMR of 0.0955 m^2/kg for each debris object. Each AMR is drawn from a normal distribution with 0.0955 m^2/kg mean (mean AMR error of zero) and 0.015 m^2/kg standard deviation. The standard deviation is equal to roughly 15% of the total AMR. Figure B.29 shows that the filter has the highest error within the first 40 minutes and then is able to resolve the density estimate to within 15-20% error. The magnitude of error is about 5% higher than the baseline multi-object scenario throughout the simulation (Table B.1).

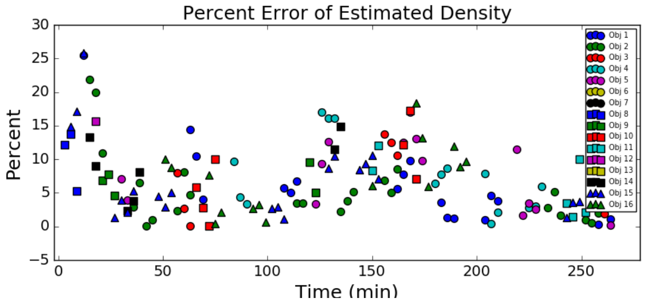

Figure B.29: Unknown Ballistic Coefficient Study Density Estimate Percent Error

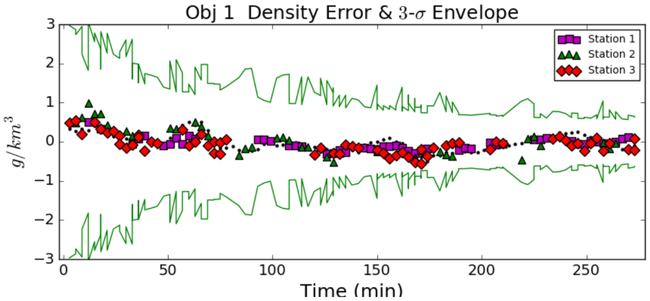

Figure B.30: Unknown Ballistic Coefficient Study Density Error and 3-σ Envelope

This scenario shows that the filter should still be able to estimate the density to a reasonable level of accuracy for a real-world case where the debris object ballistic coefficients are not known precisely. A best guess of the ballistic coefficients would be used in the filter. These ballistic coefficients are from the bootstrap process described in Section B.2.

B.5.7 Geomagnetic Storm

This scenario simulates an extreme geomagnetic storm that ramps up to Ap and F10.7 indices of 400 and 273, respectively [NOAA Space Weather Scales; Thompson, Space Weather Indices].

Table B.3: MSIS Parameters for an Extreme Geomagnetic Storm

Parameters	Ap Index	F10.7 Index
Nominal Conditions	17	127
Extreme Geomagnetic Storm	400	273

The geomagnetic storm begins building at approximately 200 minutes into the simulation and continues for the remainder of the simulation (Fig. B.31). Once the geomagnetic storm begins, the density estimation error increases rapidly as the filter attempts to compile enough information to sense the true density. After the storm reaches a steady-state, around 500 minutes into the simulation, the filter takes about one orbital period, 90 minutes, to correct the density estimates (Fig. B.32). From this point forward, the density estimate stays within 30%-40% error and to within 20% error by the end of the simulation (2225 minutes). The RMS density error over the full density state for each measurement time is shown in Figure B.33.

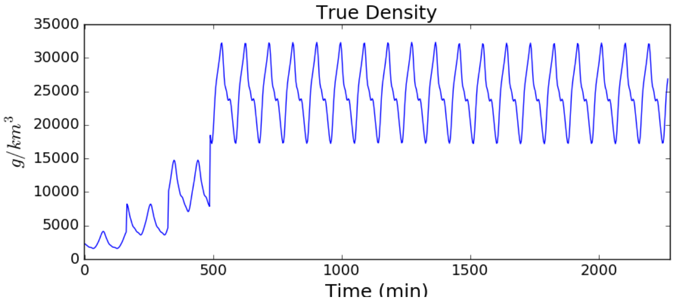

Figure B.31: True Density during Extreme Geomagnetic Storm

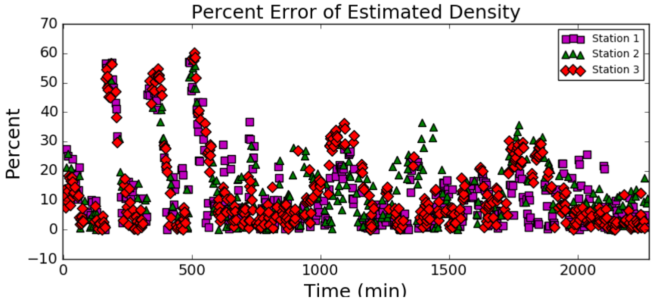

Figure B.32: Geomagnetic Storm Density Estimate Percent Error

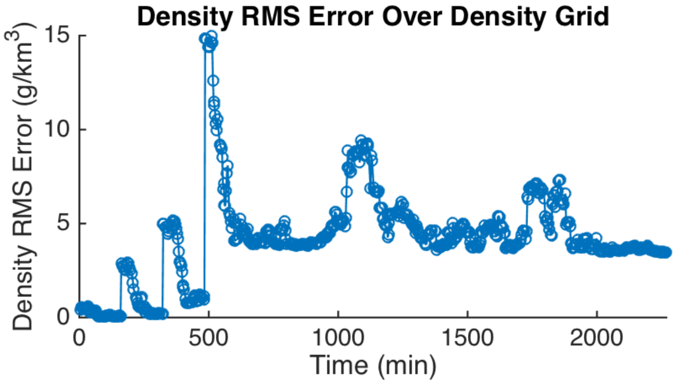

Figure B.33: The RMS density error over the entire spatial grid as a function of time for the Geomagnetic Storm Case.

B.5.8 Results Summary

Table B.1 summarizes the results for case studies exploring the filter's dependence on sensor error, measurement cadence, ballistic coefficient information, and a geomagnetic storm. It is found that increasing the sensor error by an order of magnitude also increases the RMS density error by roughly half an order of magnitude. When the measurement cadence is reduced by 50%, the RMS density error doubles. The density error increases by 50% when the ballistic coefficient is known within 15% of its true value. By far, the filter had the most difficulty in the geomagnetic storm case, but this is to be expected when the density is increasing by an order of magnitude within just a couple hours. Figure B.34 shows the RMS error over the entire spatial grid of density elements as a function of time for comparable cases (Sections B.5.3 - B.5.6).

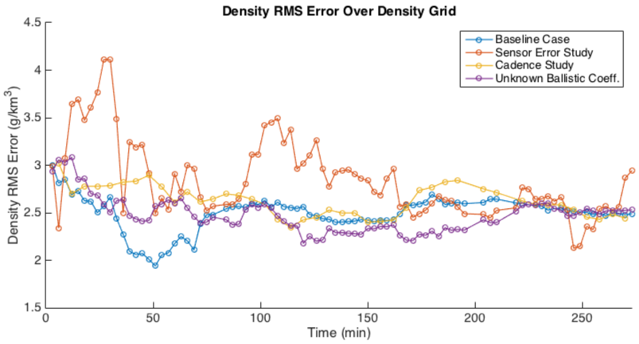

Figure B.34: The RMS density error over the entire spatial grid as a function of time for various case studies.

B.6 Conclusion

The POEnKF proves to be a suitable method for estimating a large state comprised of debris object states (position and velocity) and a grid of atmospheric density. This particular filter avoids a nonpositive definite matrix while still maintaining enough information, or covariance eigenvectors, to allow for the regions of different density throughout the grid to be effectively estimated. Not only is the POEnKF able to estimate disparate regions of density, but it is also capable of tracking rapid changes in global density due to an extreme geomagnetic storm.

Overall, the filter use cases demonstrate the tool's effectiveness in determining what criteria (sensor error, measurement cadence, ballistic coefficient knowledge) are required to achieve a certain level of accuracy in the density estimate. In Section B.5.4, it is shown that if a larger measurement error, is applied the density estimate error increases. Likewise, if the ballistic coefficient is not known precisely, the error grows proportionally. This is due to the direct correlation between atmospheric drag and both the ballistic coefficient and atmospheric density. These parameters are coupled (Eq. B.1).

In conclusion, it has been shown that using debris objects tracking data can greatly improve our real-time estimate of atmospheric density compared to assuming a particular density model which will likely be incorrect when the atmosphere is disturbed. The dispersion of multiple debris objects provides global information about the atmospheric density at any given time, instead of at a single debris object location. If a geomagnetic storm occurs, this tool can provide more accurate density estimates than that of a simple exponential atmospheric density model or MSIS (when calm conditions are assumed). The work presented here explores the POEnKF performance when using information from only 16 debris objects, the research vision is to utilize information from all debris objects. Although observations of particular debris objects are sporadic, observations of the overall debris population are collected regularly. Our tool can utilize this influx of debris object information to provide improved density estimates consistently. Enhanced knowledge of near real-time atmospheric density behavior will allow for improved predictions of all objects, not just orbital debris.

The next phase of planned research is to incorporate a physics-based atmospheric model in the forecast step. This version of the tool will involve estimating accelerations from debris object measurements and using these accelerations to infer forcing parameters, such as Kp and F10.7 indices, of the physics-based model, TIE-GCM. Similar to the current tool, the product of our new tool will be an improved time-series estimate of atmospheric density, with the addition of a physics driven forecasting ability.

B.7 References

(1) Bowman, B., Storz, M.F., 2002. Time Series Analysis of HASDM Thermospheric Temperature and Density Corrections. In: AIAA/AAS Astrodynamics Specialist Conference.

(2) Bowman, B., Storz, M.F., 2003. High accuracy satellite drag model (HASDM) review. Advances in the Astronautical Sciences. 116. 1943-1952.

(3) Emmert, J., Picone, J., Meier, R. 2008. Thermospheric global average density trends, 1967-2007, derived from orbits of 5000 near-Earth objects. Geophysical Research Letters.

35, L05101.

(4) Evensen, G., 1994. Sequential Data Assimilation with a Nonlinear Quasi-geostrophic Model using Monte Carlo Methods to Forecast Error Statistics. Journal of Geophysical Research. 99 (10), 143-162.

(5) Evensen, G., 2009. Data Assimilation - The Ensemble Kalman Filter. Springer.

(6) Gordon, N., Salmond, D., Smith, A., 1993. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. lEE Proceedings on Radar and Signal Processing. 140 (2), 107-113.

(7) Gaspari, G., Cohn, S.E., 1999. Construction of Correlation Functions in Two and Three Dimensions. Quarterly Journal of the Royal Meteorological Society. 125, 723-757.

(8) Houtekamer, P., Mitchell, H.L., 2005. Ensemble Kalman Filtering. Quarterly Journal of the Royal Meteorological Society. 131, 3269-3289.

(9) Houtekamer, P., Zhang, F., 2016. Review of the Ensemble Kalman Filter for Atmospheric Data Assimilation. Monthly Weather Review. 144, 4489-4532.

(10) Heemink, A., Verlaan, M., Segers, A., 2000. Variance Reduced Ensemble Kalman Filtering. Monthly Weather Review. 129, 1718-1728.

(11) Linares, R. Jah, M., Crassidis, J., Leve, F.A., Kelecy, T., 2014. Astrometric and photometric data fusion for inactive space object feature estimation. Acta Astronautica. 99, 1-15.

(12) Matsuo, T., 2014. Upper atmosphere data assimilation with an ensemble Kalman filter. In: Qian, L., Burns, A.G., Emery, B.A., Foster, B., Lu, G., Maute, A., Richmond, A., Roble, R.G., Solomon, S., Wang, W., 2014. Modeling the Ionosphere-Thermosphere System. 10.1002/9781118704417.ch7.

(13) Mehta, P., Linares, R., 2017. A methodology for reduced order modeling and calibration of the upper atmosphere. Space Weather. 15 (10),1270-1287.

(14) Morozov, A., Ridley, A., Bernstein, D., Collins, N., Hoar, T., Anderson, J. 2013. Data assimilation and driver estimation for the Global Ionosphere-Thermosphere Model using the Ensemble Adjustment Kalman Filter. Journal of Atmospheric and Solar-Terrestrial Physics. 104, 126-136.

(15) NOAA Space Weather Scales, 2011. Space Weather Prediction Center www.swpc.noaa.gov/noaa-scales-explanation (accessed 10.1.18).

(16) Picone, J. M., Hedin, A. E., Drob, D. P., Aikin, A. C., 2002. NRLMSISE-00 Empirical Model of the Atmosphere: Statistical Comparisons and Scientific Issues. Journal of Geophysical Research. 107 (A12), 1468.

(17) Pilinski, M., Crowley, G., Sutton, E., Codrescu, M., 2016. Physics-based assimilative atmospheric modeling for satellite drag specification and forecasts. Advances in the Astronautical Sciences. 156, 4405-4423.

(18) Storz, M.F., Bowman, B.R., Branson, J.I., Casali, S.J., Tobiska, W.K., 2005. High Accuracy Satellite Drag Model (HASDM). Advances in Space Research, 36 (12), 2497-2505.

(19) Sutton, E., 2018. A New Method of Physics-Based Data Assimilation for the Quiet and Disturbed Thermosphere. Space Weather. 16 (6), 736-753.

(20) Thompson, R., Space Weather Indices, Australian Government Bureau of Meteorology www.sws.bom.gov.au/Educational/1/2/4 (accessed 10.1.18)

(21) Vallado, D.A., 2013. Fundamentals of Astrodynamics and Applications. Space Technology Library.

(22) Volkov, I., 2004. Earth Upper Atmosphere Density Model for Ballistic Support of Flights of Artificial Earth Satellites (GOST R 25645.166.2004). Moscow: Publishing House of the

Standards.

(23) Yurasov, V., Nazarenko, A., Cefola, P., Alfriend, K., 2005. Density Corrections for the NRLMSISE-00 Atmosphere Model. In: AIAA/AAS Space Flight Mechanics Conference. AAS 05-168.

(24) Yurasov, V., Nazarenko, A., Alfriend, K., Cefola, P., 2006. Direct Density Correction Method: Review of Results. In: 57th International Astronautical Congress Conference.