FABIAN SCHIEMENZ

COVARIANCE AND UNCERTAINTY REALISM FOR LOW EARTH ORBITING SATELLITES VIA QUANTIFICATION OF DOMINANT FORCE MODEL UNCERTAINTIES

COVARIANCE AND UNCERTAINTY REALISM FOR LOW EARTH ORBITING SATELLITES VIA QUANTIFICATION OF DOMINANT FORCE MODEL UNCERTAINTIES

FABIAN SCHIEMENZ, MASTER OF SCIENCE (M.SC.)

DISSERTATION

Presented to the Faculty of Mathematics and Computer Science
of the University of Würzburg
in partial fulfillment of the requirements
for the degree of

DOCTOR RERUM NATURALIUM

Prepared at
Airbus Defence and Space GmbH
88039 Friedrichshafen, Germany

The University of Würzburg
October 2020

Covariance and uncertainty realism for low Earth orbiting satellites via quantification of dominant force model uncertainties
A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor Rerum Naturalium (Dr. rer. nat.)

SUPERVISOR: Dr. Jens Utzmann (Airbus Defence and Space)
REVIEWERS: Prof. Dr. Hakan Kayal (The University of Würzburg),
Prof. Dr. Moriba Jah (The University of Texas at Austin)

To my wife and my parents

ABSTRACT

The safety of future spaceflight depends on space surveillance and space traffic management, as the density of objects in Earth orbit has reached a level that requires collision avoidance maneuvers to be performed on a regular basis to avoid a mission or, in the context of human space flight, life-endangering threat. Driven by enhanced sensor systems capable of detecting centimeter-sized debris, mega-constellations and satellite miniaturization, the space debris problem has revealed many parallels to the plastic waste in our oceans, however with much less visibility to the eye. Future catalog sizes are expected to increase drastically, making it even more important to detect potentially dangerous encounters as early as possible.

Due to the limited number of monitoring sensors, continuous observation of all objects is impossible, resulting in the need to predict the orbital paths and their uncertainty via models to perform collision risk assessment and space object catalog maintenance. For many years the uncertainty models used for orbit determination neglected any uncertainty in the astrodynamic force models, thereby implicitly assuming them to be flawless descriptions of the true space environment. This assumption is known to result in overly optimistic uncertainty estimates, which in turn complicate collision risk analysis.

The keynote of this doctoral thesis is to establish uncertainty realism for low Earth orbiting satellites via a physically connected quantification of the dominant force model uncertainties, particularly multiple sources of atmospheric density uncertainty and orbital gravity uncertainty. The resulting process noise models are subsequently integrated into classical and state of the art orbit determination algorithms. Their positive impact is demonstrated via numerical orbit determination simulations and a collision risk assessment study using all non-restricted objects in the official United States space catalogs. It is shown that the consideration of atmospheric density uncertainty and gravity uncertainty significantly improves the quality of the orbit determination and thus makes a contribution to future spaceflight safety by increasing the reliability of the uncertainty estimates used for collision risk assessment.

ZUSAMMENFASSUNG

Die Sicherheit der künftigen Raumfahrt hängt von der Weltraumüberwachung und dem Weltraumobjektmanagement ab, da inzwischen die Dichte an Objekten im Erdorbit ein Niveau erreicht hat, welches regelmäßige Kollisionsvermeidungsmanöver erfordert um eine missions- oder, im Kontext der bemannten Raumfahrt, lebensgefährdende Situation zu vermeiden. Durch verbesserte Sensorsysteme, die in der Lage sind, zentimetergroße Objekte zu erkennen, Megakonstellationen und die Satellitenminiaturisierung hat das Weltraummüllproblem viele Parallelen zu den Plastikabfällen in unseren Weltmeeren offenbart, jedoch mit deutlich geringerer Sichtbarkeit für das Auge. Es ist zu erwarten, dass die Größe der Weltraumobjektkataloge in Zukunft drastisch ansteigen wird, was es umso wichtiger macht, potenziell gefährliche Begegnungen so früh wie möglich zu erkennen.

Durch die begrenzte Anzahl an Überwachungssensoren ist eine kontinuierliche Beobachtung aller Objekte unmöglich, sodass die Umlaufbahnen und deren Unsicherheiten über Modelle vorausberechnet werden müssen, um die Bewertung von Kollisionsrisiken vorzunehmen und die Wartung der Objektkataloge sicherzustellen. Viele Jahre haben die zur Bahnbestimmung verwendeten Unsicherheitsmodelle jegliche Unsicherheit in den astrodynamischen Kräftemodellen vernachlässigt und somit implizit angenommen, dass diese fehlerfreie Beschreibungen der wahren Weltraumumgebung darstellen. Diese Annahme ist jedoch dafür bekannt, zu übermäßig optimistischen Unsicherheitsabschätzungen zu führen, was die Kollisionsrisikobewertung erschwert.

Das Leitthema dieser Doktorarbeit ist die Berechnung realistischer Unsicherheiten von Objekten in einer niedrigen Erdumlaufbahn anhand einer Unsicherheitsquantifizierung mit physikalischem Bezug zu den Kräftemodellen, welche die größten Anteile an der Propagationsunsicherheit aufweisen. Dies sind insbesondere mehrere Quellen von atmosphärischer Dichteunsicherheit, sowie Gravitationsunsicherheit. Die resultierenden Prozessrauschmodelle werden anschließend in klassische und moderne Algorithmen zur Umlaufbahnbestimmung integriert. Die positiven Auswirkungen dieser Technik werden durch numerische Simulationen zur Orbitbestimmung, sowie durch eine Risikobewertungsstudie anhand aller nicht-sensitiven Objekte in den amerikanischen Weltraumkatalogen belegt. Es wird gezeigt, dass die Berücksichtigung von Unsicherheiten in der atmosphärischen Dichte und dem Gravitationsmodell die Qualität der Umlaufbahnbestimmung signifikant verbessert und somit durch zuverlässigere Unsicherheitsschätzungen bei der Kollisionsrisikobewertung einen Beitrag zur künftigen Sicherheit im Weltraum leistet.

PUBLICATIONS

The research carried out as part of this doctoral thesis was divided into building blocks to structure the overall topic and identify purposeful areas of investigation. The work packages ranged from a classic state of the art analysis up to the final development of a collision assessment risk analysis software suite that incorporates all research contributions. Significant findings were published as peer-reviewed journal papers to validate their novelty and confirm the scientific contribution. The following list contains this work in chronological order:

Peer-reviewed research contributions

Schiemenz, Fabian, Utzmann, Jens, and Kayal, Hakan (2019a). "Survey of the operational state of the art in conjunction analysis." In: *CEAS Space Journal* 11.3, pp. 255–268. ISSN: 1868-2510.

Schiemenz, Fabian, Utzmann, Jens, and Kayal, Hakan (2019b). "Propagating EUV solar flux uncertainty to atmospheric density uncertainty." In: *Advances in Space Research* 63.12, pp. 3936–3952. ISSN: 0273-1177.

Schiemenz, Fabian, Utzmann, Jens, and Kayal, Hakan (2019c). "Least squares orbit estimation including atmospheric density uncertainty consideration." In: *Advances in Space Research* 63.12, pp. 3916–3935. ISSN: 0273-1177.

Schiemenz, Fabian, Utzmann, Jens, and Kayal, Hakan (2020a). "Accurate estimation of relative atmospheric density error on the example of uncertain geomagnetic activity information." In: *Advances in Space Research* 65.1, pp. 251–270. ISSN: 0273-1177.

Schiemenz, Fabian, Utzmann, Jens, and Kayal, Hakan (2020b). "Propagation of grid-scale density model uncertainty to orbital uncertainties." In: *Advances in Space Research* 65.1, pp. 407–418. ISSN: 0273-1177.

Schiemenz, Fabian, Woodburn, James, Coppola, Vincent, Utzmann, Jens, and Kayal, Hakan (2020c). "Improved Orbit Gravity Error Covariance." In: *Journal of Guidance, Control and Dynamics* 43.9, pp. 1671–1687. ISSN: 1533-3884.

Schiemenz, Fabian, Utzmann, Jens, and Kayal, Hakan (2020d). "Adaptive Gaussian Mixture based Orbit Determination with combined atmospheric density uncertainty consideration." In: *Advances in Space Research* 66.7, pp. 1609–1634. ISSN: 0273-1177.

Nothing in life is to be feared,
it is only to be understood.
Now is the time to understand more,
so that we may fear less.

— Marie Curie

ACKNOWLEDGMENTS

Quite a number of people have helped me to reach this milestone in my life. Their support not only made this thesis possible, but also enabled me to dig deeper in times where the way forward was hidden to me.

First and foremost I would like to thank my wife. It is hard to find words that can express her unconditioned, multifaceted and unbreakable support. She sacrificed countless evenings and weekends together, so that I could pursue my research studies in parallel to a regular job. I thank her from the bottom of my heart for her incredible patience to listen to all of my issues, thoughts and ideas. Also her care of my well-being was beyond comparison (being gifted an ergonomic chair and desk to better work from home means a lot!).

I feel the same for my parents, who raised me in unconditioned and unlimited love. This unburdened environment not only enabled me to grow and become strong, but also to pursue my dreams, enjoy education and develop a deep interest in nature and its mechanisms. I will be eternally grateful to them for everything they have given me and pass on these values whenever I can.

Professionally, I want to express my gratitude to those people that have made this doctoral research at Airbus possible. I am indebted to Dieter Ulrich, who not only always supported me in pursuing my PhD project ever since he hired me, but also contributed fundamentally to developing a framework that enabled conducting research in parallel to work. Likewise, I want to thank my supervisor Jens Utzmann who had just the right topic to work on at a time when there were no official PhD positions within Airbus. Many thanks also to my professor at the University of Würzburg, Hakan Kayal, for agreeing to supervise this external doctoral research study and giving me the freedom to conduct this work according to my wishes.

Further people at Airbus have helped me numerous times, e.g. by acquiring literature, providing access to high performance computers, installing software or managing the dual-nature of my work and research student identities. Thank you Angela Price, Simone Günthör, Uwe Brehmer, Lutz Koop, Gaby Brielmeier, Michaela Spieß and many others!

I have received lots of professional support from the amazing people at Analytical Graphics Inc. My deepest thanks go to David Vallado, who always answered all of my questions within hours and shared his incredibly broad knowledge with me. Similarly, I want to thank Jim Woodburn for his warmhearted help, fruitful discussions and insight on the topic of orbit gravity error covariance estimation. His support made the improvements in this area and the resulting joint paper possible. I also want to thank T. S. Kelso for providing me a conjunction analysis dataset for algorithm validation and Salvatore Alfano for the discussions on his long-term encounter collision probability algorithms. I wish everyone would respond to their fellow colleagues in such a supportive and responsive manner as the people at AGI!

Finally, I also want to credit Sean Bruinsma from CNES for providing the Fortran source code of DTM-2012/DTM-2013, Diego Escobar from GMV for his explanations on the smart sieve conjunction screener and Vitali Braun from the ESA Space Debris Office for providing a dataset on the metric dimensions of all public resident space objects.

Fabian Schiemenz in October 2020

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ALGORITHMS AND PROGRAM CODE

ACRONYMS

ADR	Active Debris Removal
AGI	Analytical Graphics Inc.
AGMUKF	Adaptive Gaussian Mixture Unscented Kalman Filter
AI	Artificial Intelligence
ANCAS	Alfano Negron Close Approach Software
ART	Airbus Robotic Telescope
ASAT	Anti Satellite
ASW	Astrodynamics Support Workstation
BLWN	Band-Limited White Noise
CA	Conjunction Assessment
CAM	Collision Avoidance Maneuver
CARA	Conjunction Assessment and Risk Analysis
CDF	Cumulative Distribution Function
CDM	Conjunction Data Message
CNES	Centre National d'Études Spatiales
COE	Classical Orbital Elements
COLA	Collision Avoidance
CP	Covariance Propagation
CPU	Central Processing Unit
CSPOC	Combined Space Operations Center
CSSI	Center for Space Standards & Innovation
CTWN	Continuous-time White Noise
DCA	Distance at Closest Approach
DE	Development Ephemerides
DISCOS	Database and Information System Characterising Objects in Space
DOD	Department of Defense
DST	Disturbance Storm Time
DTM	Drag Temperature Model
EGM	Earth Gravitational Model

EKF	Extended Kalman Filter
EME2000	Earth Mean Equator and Equinox of J2000
EOE	Equinoctial Orbital Elements
EOP	Earth Orientation Parameter
ESA	European Space Agency
EUV	Extreme Ultraviolet
FOV	Field Of View
FPKE	Fokker-Planck-Kolmogorov Equation
FUV	Far Ultraviolet
GCRF	Geocentric Celestial Reference Frame
GCRS	Geocentric Celestial Reference System
GEO	Geostationary Earth Orbit
GFZ	Geoforschungszentrum Potsdam
GGM	Grace Gravity Model
GH	Gauss-Hermite
GMM	Gaussian Mixture Model
GMV	Grupo Tecnològico e Industrial GMV
GNSS	Global Navigation Satellite System
GOF	Goodness Of Fit
GP	General Perturbations
GPS	Global Positioning System
GPU	Graphics Processing Unit
HAMR	High Area To Mass Ratio
HASDM	High Accuracy Satellite Drag Model
HBR	Hard Body Radius
HIE	High Interest Event
HWM	Horizontal Wind Model
ICRS	International Celestial Reference System
IERS	International Earth Rotation and Reference Systems Service
IMF	Interplanetary Magnetic Field
IOD	Initial Orbit Determination
IR	Infrared
ISD	Integral Squared Difference

ISS	International Space Station
JB	Jacchia-Bowman
JBH	Jacchia Bowman HASDM
JD	Julian Date
JPL	Jet Propulsion Laboratory
KL	Kullback-Leibler
LAM	Likelihood Agreement Measure
LEO	Low Earth Orbit
LL	Log Likelihood
LTE	Long-Term Encounters
LUT	Lookup Table
MC	Monte-Carlo
MEO	Medium Earth Orbit
MHT	Multiple Hypothesis Tracker
MSIS	Mass Spectrometer Incoherent Scatter
MSISE	Mass Spectrometer Incoherent Scatter Extended
NASA	National Aeronautics and Space Administration
NLNM	Normal Lognormal Mixture
NOAA	National Oceanic and Atmospheric Administration
NORAD	North American Aerospace Defense Command
NRLMSISE	Naval Research Laboratory Mass Spectrometer Incoherent Scatter Radar Extended
O/O	Owner/Operator
OD	Orbit Determination
ODE	Ordinary Differential Equation
ODTK	Orbit Determination Tool Kit
OE	Orbital Elements
OOM	Order Of Magnitude
OPENMP	Open Multi-Processing
OPM	Outer Probability Measure
OUP	Ornstein-Uhlenbeck Process

PDF	Probability Density Function
PPT3	Position and Partials as functions of Time Model 3
PROOF	Program for Radar and Optical Observation Forecasting
PSD	Positive Semidefinite
PSD	Power Spectral Density
PVONMS	Pioneer Venus Orbiter Neutral Gas Mass Spectrometer
PVT	Position Velocity Time
QQ	Quantile Quantile
RCS	Radar Cross-Section
RIC	Radial/In-track/Cross-Track
RMSE	Root Mean Square Error
RSO	Resident Space Object
RTN	Radial/Transverse/Normal
SBSS	Space-Based Space Surveillance
SBWLS	Sequential Batch Weighted Least Squares
SDA	Space Data Association
SDE	Stochastic Differential Equation
SDO	Space Debris Office
SDP4	Simplified Deep Space Perturbations Model 4
SFU	Solar Flux Unit
SGP4	Simplified General Perturbations Model 4
SKF	Schmidt-Kalman filter
SLR	Satellite Laser Ranging
SOCRATES	Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space
SP	Special Perturbations
SPCS	Space Control Squadron
SPOOK	Special Perturbations Orbit determination and Orbit analysis toolKit
s_q	Solar Quiet
s_r	Solar Regular
SRP	Solar Radiation Pressure
SSA	Space Situational Awareness
SSN	Space Surveillance Network

SSR	Satellite Situation Report
SST	Space Surveillance and Tracking
STE	Short-Term Encounters
STK	Systems Tool Kit
STM	State error Transition Matrix
SWPC	Space Weather Prediction Center
TCA	Time of Closest Approach
TEG	Texas Earth Gravity model
TEME	True Equator Mean Equinox
TLE	Two-Line Elements
UC	Uncertainty Characterization
UCT	Uncorrelated Track
UKF	Unscented Kalman Filter
UQ	Uncertainty Quantification
US	United States
USKF	Unscented Schmidt-Kalman filter
USSF	US Space Force
UT	Unscented Transformation
UTC	Coordinated Universal Time
UV	Ultraviolet
VCM	Vector Covariance Message
VNC	Velocity Normal Co-Normal
WLS	Weighted Least Squares
WRMS	Weighted Root Mean Square

The following is a summary of conventions followed in this work and frequently encountered, exclusively allocated symbols.

Convention	**Example**
Scalars are denoted by plain characters	ρ, M
Vectors are denoted by small-type bold characters	$\boldsymbol{x}$
Matrices are denoted by capitalized bold characters	$\boldsymbol{\Phi}$
Estimates carry a hat on top of the variable	$\hat{\boldsymbol{x}}$
An absolute error carries a prefixed delta. Error definition: error $=$ truth $-$ estimate	$\delta\boldsymbol{x} = \boldsymbol{x} - \hat{\boldsymbol{x}}$
A relative error carries a prefixed epsilon. Error definition: absolute error/estimate	$\epsilon_\rho = \frac{\rho - \hat{\rho}}{\hat{\rho}}$
From-to relationships (e.g. vectors or transformation matrices) are typeset using subscript letters in the order *destination, origin*	$\mathbf{T}_{b,a}$ (from a to b)
Matrix determinants are expressed by vertical bars	$\lvert\mathbf{P}\rvert$
A superscript T denotes the matrix transpose	$\mathbf{P}^T$
A subscript 0 is used to indicate initial (e.g. epoch or baseline) quantities	$\mathbf{P}_0$, T_0

Symbol	**Description**
BC	Ballistic coefficient
ϵ_ρ	Relative mass density error
$\boldsymbol{\Phi}$	State error transition matrix
$\boldsymbol{\mu}$	Mean vector of a distribution
$\mathbf{P}$	State covariance matrix
P_c	Collision probability
θ	Rate of mean reversion (Ornstein-Uhlenbeck process)
$\mathbf{Q}$	Process noise covariance matrix
$\boldsymbol{r}$	Position vector
t	Time
$\boldsymbol{v}$	Velocity vector
$\boldsymbol{x}$	Kinematic state vector (position and velocity)
$\boldsymbol{z}$	Evidence (observation) vector

Part I

THE IMPORTANCE OF ACTIVE SPACE SURVEILLANCE AND IMPROVED UNCERTAINTY REALISM

1

INTRODUCTION

Corresponding publication: Schiemenz et al. (2019a)

Ever has the view into our sky fascinated people all around the world. Be it the desire to find an answer to the question of what other than humankind is out there, the apparently infinite open spaces, the quest of understanding the fundamental physics of the universe or simply the desire to reach the "unreachable" - space has always been and will always be a source of inspiration and the ultimate benchmark for our technological capabilities.

Our understanding of the Earth, the solar system and the universe has increased rapidly since the launch of the first artificial satellites in the middle of the twentieth century. With the political interest in space and the advent of increasingly powerful digital computers, the conditions for the development of a commercial market centered around our near-Earth environment were quickly met. In 2017, 60 years after the first man-made satellite was launched, a million people were employed in the space economy, which grew to 414.75 billion US dollar per fiscal year in 2018. As of January 2020, 81 nations are operating in space (The Space Foundation, 2020). Today, humanity has become dependent on satellite technology without most people noticing. Space-based services, such as global telecommunications, Earth observation or global positioning are the building blocks that, besides the numerous scientific and military applications, drive what many of us take as granted: precise weather forecasts, Google Maps, satellite TV, banking or even full shelves of fruits and vegetables in our supermarkets[1]. Also the future trend is clear: technologies currently being developed, including safety-critical ones like autonomous driving or urban air mobility, are impossible without constellations of satellites in Earth orbit.

Despite this crucial role of our space assets, satellite operators have long trusted in the “big sky theory”, which states that the space around Earth is too large for collisions between active satellites and resident space objects[2] (RSOs) to take place (Newman et al., 2009). While this assumption was certainly valid for the early days of spaceflight, a first warning about the accumulation of space debris[3] was published already two decades after Sputnik 1 was brought into orbit, at a time where the total number of tracked objects was about 4000 (Kessler and Cour-Palais, 1978). Furthermore Kessler and Cour-Palais also predicted the first collision between cataloged RSOs to take place

1 Satellite data is used in many places to increase farming efficiency

2 A resident space object is defined as any natural or artificial object in a permanent orbit around another body

3 Space debris is defined as any anthropogenic space object which no longer serves a useful purpose

between 1989 and 1997. In 1990 Kessler continued the study on cascading collisions and postulated the existence of a critical population density, which once reached, causes the rate of accidental collisions to exceed the object removal rate due to atmospheric drag, such that the overall debris population increases without further satellites being placed into orbit (Kessler, 1991). Such a self-sustained growth could potentially render certain altitude shells unusable for many years due to the debris environment being too hostile for operational use. This assessment later became known as "Kessler Syndrome".

1.1 POPULATION OF RESIDENT SPACE OBJECTS IN EARTH ORBIT

Today, collisions in space are a reality and their effect can be anything from negligible to catastrophic. The first recorded unintentional collision of an operational spacecraft with a cataloged object in orbit around Earth happened on July 24th, 1996, when the gravity gradient boom of CERISE, a French reconnaissance spacecraft, was severed by a piece of Ariane 1 rocket stage debris (Johnson, 1996; Anz-Meador et al., 2018). If collisions of non-functional RSOs are included, the first event even dates back to December 23rd, 1991, when a piece of debris from Cosmos 926 collided with the non-functional Cosmos 1934 satellite (Liou, 2005; Anz-Meador et al., 2018). Both incidents fall into the timeframe predicted by Kessler and Cour-Palais.

A list of known and suspected unintentional collisions is given in table 1.1. Other events that are well-known for their creation of large amounts of debris are the intentional destructions of Fengyun-1C and Microsat-R during the 2007 Chinese and 2019 Indian anti satellite tests.

Known and suspected unintentional collisions involving at least one artificial space object

1991	Inactive Cosmos 1934 satellite hit by cataloged Cosmos 296 debris
1996	Active Cerise satellite hit by cataloged Ariane 1 rocket stage debris
1997	Inactive NOAA 7 satellite hit by uncataloged debris
2002	Inactive Cosmos 539 satellite hit by uncataloged debris
2005	US rocket body hit by cataloged Chinese rocket stage debris
2007	Active Meteosat 8 satellite hit by uncataloged debris
2007	Inactive NASA UARS satellite believed hit by uncataloged debris
2009	Active Iridium satellite hit by inactive Cosmos 2251
2013	Active BLITS satellite hit by cataloged debris belonging to the Chinese anti satellite weapon test
2013	CubeSats NEE-01 Pegaso and CubeBug-1 hit by debris belonging to a Tsyklon-3 upper stage
2016	Solar array of Sentinel 1A hit by micrometeoroid

Table 1.1: Known and suspected unintentional collisions involving at least one artificial space object. Collisions involving active satellites are colored gray. Sources: Wright (2009), Anz-Meador et al. (2018), Kelso et al. (2013) and Nader and Kelso (2014)

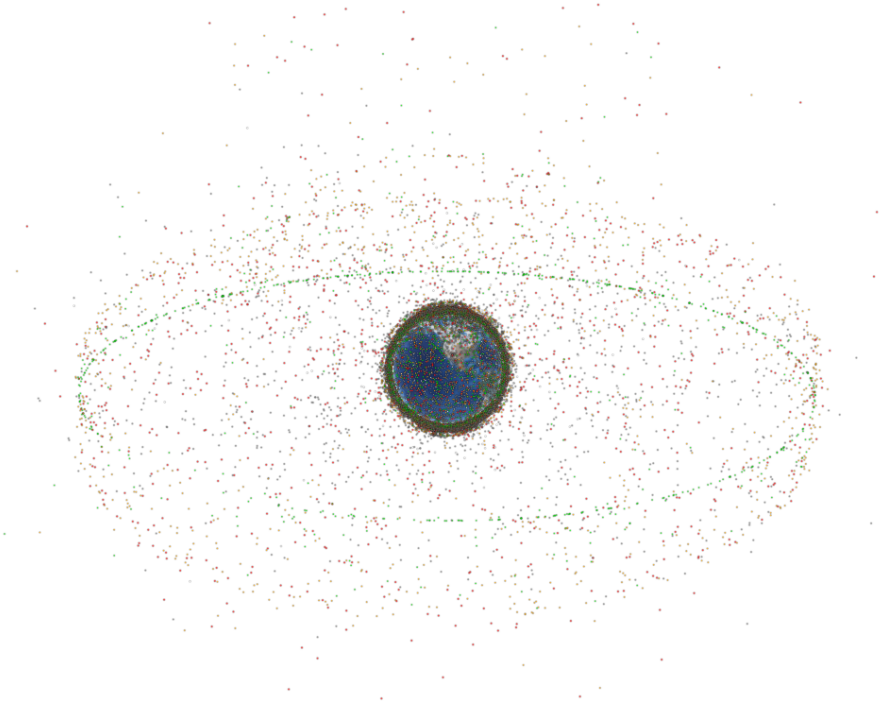


Figure 1.1: Visualization of public two-line elements catalog from May 4th, 2020. 19094 objects are contained in the source file. The heavily populated low Earth orbit shell, as well as the geostationary belt can be clearly depicted

On May 4th, 2020 the public catalog available through www.space-track.org contained 19094 RSOs. In total the US Combined Space Operations Center (CSpOC) currently tracks ~23 000 objects, of which about 8% (~1800) are operational spacecraft (Liou, 2020). A visualization of this dataset is depicted in figure 1.1. The historical evolution of the population is shown in figure 1.2. Most of the cataloged objects are located in a low Earth orbit (LEO), which may be defined as the altitude shell between 200 and 2000 km altitude. Within LEO altitudes, two peaks can be identified at about 780 km and 850 km. The increased spatial density at these altitudes can be attributed to the destruction of Fengyun-1C and the collision between Cosmos 2251 and Iridium 33.

Active debris removal (ADR) is the subject of current research projects (e.g. ESA's ADRIOS program), but not yet practically feasible. It is furthermore questionable if this technology is the genuine solution to the space debris problem, or whether ADR will essentially be used to remove only critical RSOs. The focus must therefore continue to be placed on preventing further debris in orbit. Stringent requirements to limit the generation of mission-related debris and enforce post-mission disposal have been put in place by all major space agencies in the past 25 years, however non-compliances are regularly observed (Esteva et al., 2020). While these requirements have helped to slow down the growth of debris since their implementation, single events, such as the Chinese anti satellite (ASAT) test in 2007, have destroyed the efforts of decades.

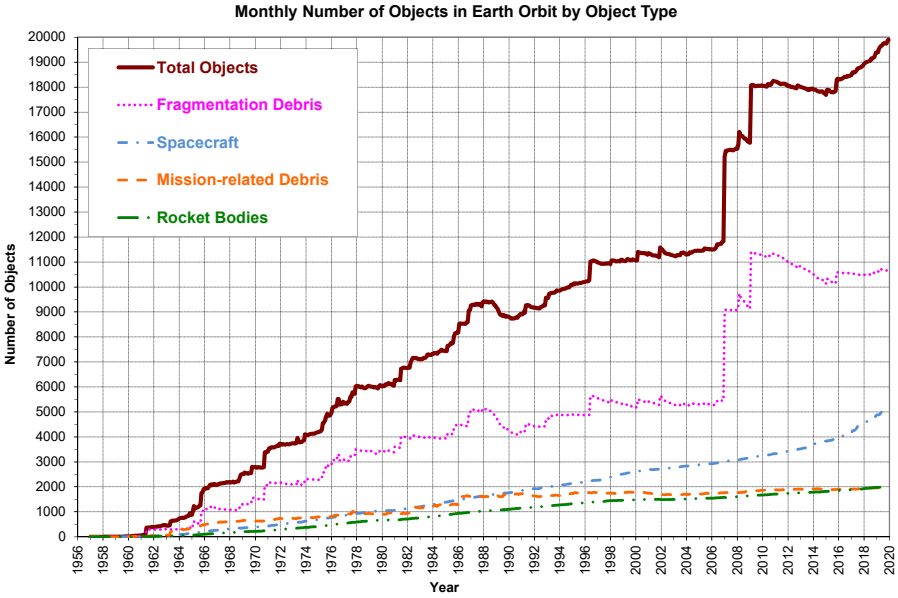


Figure 1.2: Evolution of cataloged population in Earth orbit (Liou et al., 2020). The jumps in 2007 and 2009 relate to the Chinese ASAT test and the Cosmos/Iridium collision. Currently about 1800 of the spacecraft are operational, the remaining ones derelict

Limiting mission-related debris does not only refer to spacecraft engineering (e.g. the minimization of pyro mechanisms), but also implies the avoidance of accidental collisions. Spacecraft operators are therefore required to establish operational conjunction assessment (CA) and, if necessary, to execute collision avoidance maneuvers (CAMs).

The execution of a CAM is a decision to be well-considered. Maneuvers cost valuable fuel, reduce the operational lifetime and usually require a mission interruption. Hence, unnecessary maneuvers have to be avoided. The answer to the question of when to maneuver implies the application of CA, followed by a thorough risk analysis. Only if the risk metric (commonly a collision probability) is (ideally multiple times) above the owner/operator (O/O) threshold, a CAM is executed. The combined process is known as conjunction assessment and risk analysis (CARA)[4] and requires the approaching RSO to be cataloged.

The current US space catalogs contain objects of ~10 cm diameter and larger. Due to the high average impact speeds of ~10 km/s all particles larger than 1 cm can be considered to pose a mission-ending threat in LEO. So far there was no means to cover the blind spot between the centimeter-level and 10 cm. The gap is however expected to be narrowed in the near future, as the next generation US space surveillance system, the Space Fence radar, has been declared operational on March 27th, 2020 by the US Space Force (USSF, 2020). The reduction of the minimum detectable size to ~5 cm is expected to grow the catalog dimension by a factor of three to ten (Hejduk et al., 2019).

4 The distinction between CA and CARA is fuzzy in the literature. In this work CA is defined as the identification of close encounters, whereas CARA is used to designate CA followed by a conjunction risk evaluation

A second driver of future catalog growth is the construction of mega-constellations with thousands of satellites, such as Starlink.

This heavy increase of the catalog size has direct consequences for future satellite operations. First, the number of conjunction warnings is going to increase drastically, which is likely to lead to a higher number of CAMs. Second, the expected increase of conjunction warnings underlines the need for automatization in the CARA process.

1.2 THE SPACE SURVEILLANCE AND TRACKING SEGMENT

Successful collision avoidance (COLA) requires knowledge of the kinematic state (i.e. position and velocity coordinates) and its uncertainty for all potential conjunction partners in space. The establishment and maintenance of this knowledge in the form of a space object catalog is realized by a system of systems which forms the space surveillance and tracking (SST) segment of a space situational awareness (SSA) program.

For a resident space object to be cataloged, it first needs to be observed and measurements of its orbital path need to be taken. Measurements are acquired by radars or telescopes either from ground or via space-based satellites. In some cases also high-precision laser-based measurements are possible, however these typically[5] require the target to be equipped with retro reflectors, which is only the case for some of the active satellites. Once measurements have been acquired, the measurement data needs to undergo data association. The purpose of this stage is to determine if the observed tracks belong to an object which is already known and cataloged, or if a new RSO has been detected.

Following the data association, the next step is to perform an initial orbit determination (IOD) in case of a new object or orbit determination (OD) in case of a cataloged object. The purpose of IOD is to derive an initial estimate of the object's state[6] and its uncertainty without any prior knowledge, using only the measurement data. In case of a cataloged object, an estimate of the object state and its uncertainty are already available, hence the task of OD is to update the prior estimate using the new measurement data. Both IOD and OD require the abilities of orbit and uncertainty propagation. The task of orbit propagation is to predict an object's state at a future (or past) time, given its last known state and possibly further parameters required for propagation. Similarly, the purpose of uncertainty propagation is to propagate the object's uncertainty to a future (or past) epoch. Orbit propagation can be subdivided into two main classes[7]: analytical propagation, also known as General Perturbations (GP) theory, which

5 Efforts are also underway to establish operational satellite laser ranging (SLR) for non-cooperative (and thus reflector-less) targets (Flohrer et al., 2016)

6 The term *state* is henceforth used to refer to the state vector of a system, which typically contains the kinematic state errors and optionally additional parameters

7 Also semi-analytical theories exist (e.g. the Draper Semi-analytical Satellite Theory), but they are uncommon for CARA, as official catalogs are only available for the GP and SP theories

is based on Two-Line Elements (TLE) data and numerical propagation, also known as Special Perturbations (SP) theory, which numerically integrates the equations of motion including the accelerations of all perturbations that shall be considered in the propagation. The choice of the propagation method essentially depends on three factors: data availability, required accuracy and computing speed. Common data sources have been studied in Schiemenz et al. (2019a). The analytical theory has an accuracy limit in the order of 1 km, whereas the numerical theory is capable of meter-level accuracy (Vallado, 2013). If the 1 km accuracy is sufficient for the requirements of the user, the GP theory has some great advantages over other propagation methods: first, due to its analytical nature, the computational load of the orbit propagation is independent of the propagation duration and only requires a few floating point operations. Second, no further data other than the TLE is required for propagation[8]. A third advantage is the public availability of the TLE catalog through `https://space-track.org`, resulting in easily repeatable and fast computations.

Besides these positive aspects of the GP theory, the increase in accuracy promised by using SP theory is desirable for CARA. SP-based propagation is not linked to a specific file format or perturbation model. Any combination of perturbing accelerations can be integrated to obtain the desired solution. Therefore the exact setup consisting of the satellite properties (ballistic coefficient, solar radiation pressure coefficient), perturbation models, space weather input files, Earth orientation parameter (EOP) data, coordinate frame realizations, numerical integrator, integration accuracy, etc. needs to be specified to obtain repeatable results. The overall runtime is significantly influenced by the fidelity of the models (e.g. the degree and order with which the aspherical gravity field of the Earth is computed) and the numerical integration accuracy. Furthermore, the SP catalog is not publicly available, but access needs to be requested and approved (also at SpaceTrack). For SP-based propagation it is important to maintain the highest degree of consistency possible with respect to the system that is used to maintain the catalog. An example that serves as a warning by demonstrating a higher fidelity model resulting in lower accuracy due to model inconsistency is given in Kaya et al. (2004).

Uncertainty propagation is a complex field of its own. Its algorithms can be broadly divided into four categories: linear methods, nonlinear methods, Monte-Carlo (MC) propagation and others (e.g. hybrids). Among these four categories linear uncertainty propagation is the most common in catalog maintenance, OD and CARA. Modern sigma-point approaches on the other hand have shown superior accuracy and are enjoying increasing popularity. An excellent overview of probabilistic uncertainty propagation techniques is given in Luo and Yang (2017).

8 Only the conversion of the GP-propagated state to an inertial frame requires additional files containing information about the Earth orientation

Nearly all of the existing theories for uncertainty propagation are restricted to the SP theory and incompatible with the GP theory. Satellite Os/Os have therefore started to mix theories by using GP for orbit propagation and SP for uncertainty propagation. This inconsistency between the state and uncertainty propagation is discouraged and leads to similar effects as those described in Kaya et al. (2004). In defense of the Os/Os one may discern that the GP theory never was invented for uncertainty propagation. A workaround to this problem has been presented in Escobar et al. (2011) and was also successfully applied to the work presented in this thesis.

Once orbit and uncertainty propagation have been established, OD and IOD can be performed. For both tasks multiple algorithms are available. Common choices for IOD are the methods of Laplace, Gauss, Gooding and Escobal's "Double-r" integration. The techniques are described in Vallado (2013). Also modern Gaussian sum algorithms have been developed (DeMars and Jah, 2013). The OD algorithms can be grouped into two categories: batch algorithms that perform "differential correction" and process multiple measurements from different times as a batch to derive an updated estimate of the state and its uncertainty, as well as filters, which operate sequentially in a predictor-corrector fashion. Examples of batch algorithms are Weighted Least Squares (WLS) and Sequential Batch Weighted Least Squares (SBWLS). Also a myriad of Kalman filters have been developed. Well-known realizations are the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), Schmidt-Kalman filter (SKF) and Unscented Schmidt-Kalman filter (USKF).

All of these OD algorithms assume the state uncertainty to be properly described by a Gaussian probability distribution. If this assumption is deemed inadequate, also modern probabilistic approaches exist that do not assume the overall probability density function (pdf) to be Gaussian. Examples of this category are Gaussian sum filters (which also exist in multiple variants), the Gauss von Mises filter and particle filters. If computations are not required in real-time, filters may be followed by a smoother which uses future data to improve the filtered solutions. After the orbits and uncertainties of all objects in the catalog have been updated, the data may be used for CARA or other subtasks of catalog maintenance, such as advanced sensor management. The building blocks of space surveillance and therefore prerequisites for conjunction analysis are, together with a number of common and modern approaches, summarized in figure 1.3.

Due to the huge amount of objects in Earth orbit and the different orbital characteristics, a single sensor is not enough to maintain a comprehensive and up to date catalog. A space surveillance system like the US Space Surveillance Network (SSN), consisting of multiple sensors all over the world, is required to guarantee timely updates of all objects. Catalog establishment and maintenance is therefore a very costly task which traditionally has been limited to company, agency

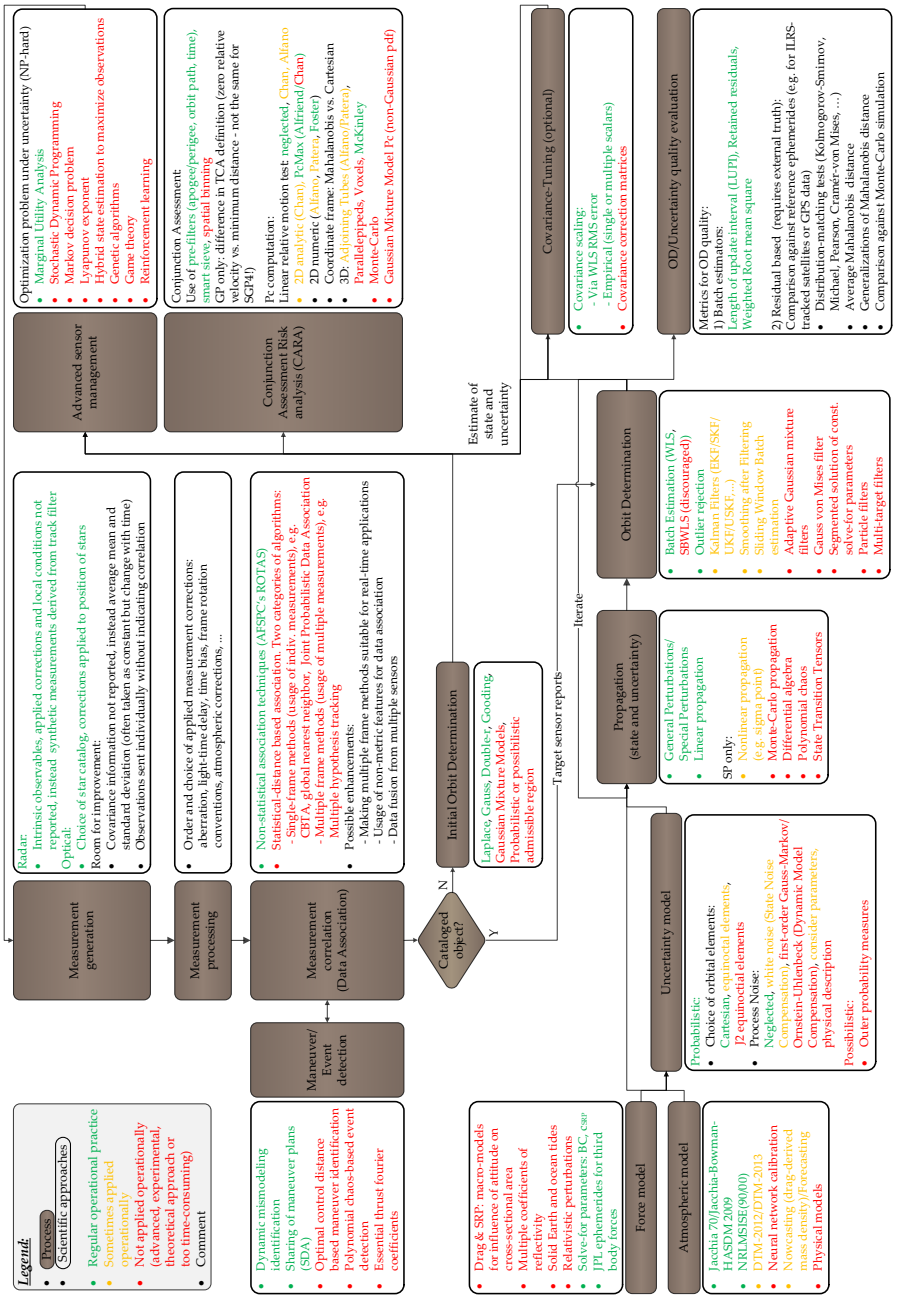


Figure 1.3: Building blocks of space surveillance (prerequisites for conjunction analysis). Each block is accompanied with a list of methods to complete the task. Operational state of the art approaches are colored green

or governmental scales[9]. Currently, the US-led CSpOC maintains the most comprehensive and up to date single source catalog. Additionally it performs routine conjunction screening and automatically sends standardized SP-based conjunction data messages (CDMs) to all registered satellite Os/Os. Further details about the duties of CSpOC, including applied algorithms and methodologies, may be found in Schiemenz et al. (2019a).

9 Recently, AstriaGraph, a crowdsourced and open source project which aims to break this legacy has been launched (Esteva et al., 2020)

The maintenance of a space catalog and the provisioning of the data to the spacecraft Os/Os allows to decouple most of the building blocks listed in figure 1.3 from the CARA process. Besides a catalog snapshot, only orbit and uncertainty propagation is required at the O/O-side to facilitate collision risk analysis, given the initial state and uncertainty information provided in a CDM. Also without a CDM the Os/Os can use their own information about the operated spacecraft and screen the orbital path(s) against the catalog. The only missing piece of information is then an estimate of the uncertainty of the catalog information, as the publicly available GP catalog contains no information about its uncertainty. The uncertainty information is also removed from the restricted SP catalog, which is accessible by approved users, to prevent possible conclusions about the sensor accuracy of the SSN.

1.3 UNCERTAINTY QUANTIFICATION IN SPACE SURVEILLANCE

Please note:
Section 1.3 gives a comprehensive overview on the importance of realistic uncertainty quantification for space security, which has substantially influenced the scope of this dissertation.
Readers who would like to get a direct overview of the scientific objectives pursued within the thesis may first read section 1.4 before returning to this point

Space situational awareness (SSA) refers to acquiring as much information as possible about our space environment. Obviously it is impossible to be aware of everything at all times, but this is also not necessary. Instead, events that require to take action on ground or in space are of particular interest and should not be missed. Space surveillance and space weather monitoring are therefore integral components of SSA. The success of preventive measures depends on how early knowledge about an event is obtained. Neither the orbit observations nor the force models used for propagation are perfect. For this reason, estimates of the kinematic state vector are always connected to a measure of uncertainty. Consequently, uncertainty quantification (UQ)[10] and propagation are critical steps to evaluate the risk of close encounters.

When discussing uncertainty, it is important to properly define the terminology upfront, especially since the terms are often mixed in common language (which is not a problem if the context is clear). In the most general setting, *uncertainty* refers to incomplete information. Randomness on the other hand relates to an intrinsic property of a system. It cannot be reduced, as the variability is of physical origin. Probability theory models random events using random variables. Their outcome cannot be predicted, however the frequency of the possible outcomes can (cf. section 1.3.2). *Variability* is often considered synonymous to randomness, however also other uses are frequently encountered (e.g. as an alternative expression for the variance or simply as a way of indicating that the state of a system changes, without specifying the cause more precisely). In the spoken language,

10 For uncertainties caused by randomness, UQ and uncertainty characterization (UC) are treated as synonyms. In the more general case of random and nonrandom uncertainties, UC is typically not possible due to missing information, however a more general quantification of the overall uncertainty is feasible using an uncertainty measure capable of describing both components

Remember:
Randomness ⇒ Uncertainty
but
Uncertainty ⇏ Randomness
(the equality only holds for purely random uncertainty)

possibility indicates that a certain state cannot be excluded from happening. Similarly, in a statistical context the term relates to possibility theory, which also aims at describing uncertainty, however does not restrict itself to randomness. It is always correct to infer uncertainty from randomness, however the opposite is generally false and only applies under certain assumptions (uncertainty does not always imply randomness).

1.3.1 *Classification of uncertainty*

Uncertainty can be categorized based on its nature. If there is an element of chance associated with its origin, it is said to be *aleatoric* or *aleatory*. Without a random component it is of systematic nature and therefore classified as *epistemic*. Also mixed contributions are possible (and in fact typical for real world systems).

Epistemic uncertainty contains all types of uncertainty where in principle it is possible to know better, however further data/research or effort would be required to obtain this improved knowledge. Epistemic uncertainty is therefore sometimes also referred to as "reducible uncertainty". A simple example of this category is the precision imposed by TLE-files. The TLE file format limits the accuracy of the mean orbital elements to a few decimal places. Consequently the uncertainty of each orbit propagation has an epistemic component caused by a limited knowledge of the initial state. Since in principle it is possible to obtain better information (although this would require changing the well-established file format), the resulting uncertainty component is epistemic.

Aleatoric uncertainty on the other hand refers to the natural variability in a system, i.e. its randomness. Mathematically it can therefore be treated with all tools available through probability theory.

Operational SST algorithms require statistical inference tools that enable sequential data processing. Since these have not been available for non-probabilistic approaches, alternative theories such as fuzzy logic, imprecise probabilities, possibility theory, fuzzy random sets, plausibility measures or the Dempster-Shafer theory have not gained operational relevance (Delande et al., 2019). For this reason, the operational state of the art in SST is to treat uncertainty as pure randomness. Probabilistic formulations however require to treat *all* uncertainty as randomness, which is modeled via random variables and probability density functions (pdfs). In many places it is therefore common to approximate epistemic uncertainty using the tools for aleatoric uncertainty. The treatment of gravity uncertainty due to errors of omission (potential field truncation), which is discussed in chapter 3, is an example of this approach.

While in theory randomness is an intrinsic property, in practice, randomness is incomplete information

— Nassim Nicholas Taleb

Very recently, however, a new generalization of probability measures has emerged, which may change this operational practice, since it

allows the joint modeling of epistemic and aleatoric uncertainties while supporting sequential data processing (i.e. filtering). Based on the available information, outer probability measures (OPMs) formulate probability bounds which tighten if more knowledge about a system is acquired and loosen if less information is available. The bounds reflect the level of available information to support or reject certain probability distributions as candidates to describe the aleatoric uncertainty component of a system. The smaller the epistemic uncertainty in a system, the narrower the bounds. OPMs quantify a system's uncertainty but do not characterize it, since in general arbitrarily many probability distributions fit between the bounds. However, if all epistemic uncertainty is removed, the bounds collapse to a single probability function, which then fully quantifies and characterizes the aleatoric uncertainty of the system. OPMs use possibility functions to establish upper bounds on residual errors, which allows to incorporate negative evidence by excluding certain areas of the state space as possible solutions.

Each hypothesis which is not known to be impossible is considered as potentially true, which leads to an expansion of the overall OPM probability bounds if less is known about a system. In contrast, probability theory is founded on the Kolmogorov axioms, of which the third (σ-additivity) is responsible for excluding any hypothesis for which the pdf has no support as possible state (i.e. for which no evidence has been collected). OPMs replace σ-additivity for the looser concept of subadditivity and random variables by uncertain variables to achieve the less descriptive (i.e. more general) uncertainty formulation. What distinguishes OPMs from the other nonprobabilistic uncertainty models is that they are compatible with the statistical inference rules of probability theory, which supports the formulation of sequential filters. In fact, operations for OPMs are similar to pdfs, as essentially only integrals are replaced by suprema. The full mathematical details may be found in Houssineau and Bishop (2018).

Since their introduction, the application of OPMs has been demonstrated for a number of SST tasks, like state updates with TLE-files (Delande et al., 2018a,b), multi-fidelity credibilistic filtering (Jones et al., 2019), admissible region initial orbit determination (Cai et al., 2020) or two-dimensional collision probability computations in Delande et al. (2019). However, there are still many open questions, such as the modeling and definition of possibility functions (single functions vs. distributions, collection of particles vs. Gaussian max mixtures, Gaussian vs. box shape, etc.), which can have a significant impact on the outcomes. First results, for example, suggest that Gaussian possibility functions are not an optimal choice for the determination of OPM collision probability limits (Delande et al., 2019). Further research in this field is therefore necessary, but in the long run OPMs have the necessary properties to change the established state of the art, provided that they find broad acceptance in the community.

A secondary option to categorize uncertainty is based on context. Slightly different categories exist throughout literature. The most common ones, including one class specific to SSA, are:

- *Structural uncertainty*. The estimation process requires models for state and uncertainty propagation. Models are always approximations of the truth. Structural uncertainty hence refers to model discrepancy. Mathematically this may be written as $\frac{\partial}{\partial t}\mathbf{x}(t) = f(\mathbf{x}(t), \mathbf{p}, t) = \hat{f}(\mathbf{x}(t), \mathbf{p}, t) + \delta f(\mathbf{x}(t), \mathbf{p}, t)$, where the true evolution of the state $\mathbf{x}$ over time t is described via $f(\mathbf{x}(t), \mathbf{p}, t)$ using the parameter set $\mathbf{p}$. Since the true form of f is generally unknown, it is approximated by a model of the form $\hat{f}(\mathbf{x}(t), \mathbf{p}, t)$. The resulting difference, $\delta f(\mathbf{x}(t), \mathbf{p}, t)$, is the model deficiency that results in structural uncertainty

- *Model parameter uncertainty*. Despite the uncertainty of a model itself, also parameters that are used by the models are only known to a certain extent (e.g. the value of the solar constant)

- *Input parameter uncertainty*. Not only the parameters that are used by the models, but also its inputs are affected by uncertainty. This is true especially in case of real-time SSA applications, where fully processed space weather data is not yet available and forecasted data needs to be used

- *Observation uncertainty*. This type of uncertainty relates to all errors introduced by the measurement system, such as measurement bias, measurement noise, scale factor errors and asymmetries. The observation uncertainty in space surveillance systems is commonly assumed to be Gaussian and white. Often also biases are considered negligible, as the sensors are believed to be correctly calibrated. Under these (debatable) assumptions, observation uncertainty may be quantified using only a measurement covariance matrix

- *Algorithmic uncertainty*. It is not always possible to exactly reproduce a model in computer code. Often approximations need to be applied, such as the truncation of infinite sums or numerical integration of a differential equation. A reduction of algorithmic uncertainty is often possible via increased runtime, e.g. by raising the number of summation terms, lowering numerical integration tolerances or increasing the floating-point accuracy (e.g. using quadruple precision instead of double precision). Algorithmic uncertainty should always be kept well below all other sources of uncertainty

- *Uncertainty due to misassociation*. This type of uncertainty is also known as cross-tagging uncertainty and very specific to SSA. It describes the uncertainty that is introduced into the space surveillance loop due to erroneous data association, i.e. attributing information of one object to another

Also other origins, such as hardware faults (e.g. a malfunctioning sensor) or further application-specific points may be added to the list. Not all types of uncertainty are always included in the estimation process. Hardware faults for example are expected to be detected and taken care of on site.

1.3.2 *Uncertainty characterization via probability distributions*

Probabilistic uncertainty is fully characterized/quantified by probability distributions, which can be defined either via a cumulative distribution function (cdf), or its derivative, the probability density function (pdf). Every cdf is a real-valued, non-decreasing, right-continuous function which assigns a value between 0 and 1 to the entire space of possible outcomes ("probability space"). The sum/integral of the pdf over the entire probability space equals unity, indicating that each possible outcome is part of it.

Additionally, a definition (or in some cases approximation) is possible using parameters that describe the location, scale and shape of a distribution. Two families of parameters are common for this purpose: moments and cumulants. Distributions with equal moments are also guaranteed to possess identical cumulants. Different variants of these parameters may be defined using centralization and normalization. The purpose of centralization is to formulate location-invariant parameters, whereas normalization results in parameters that are independent of the scale. Standardization refers to both operations.

Each defining parameter is always connected to an order. The first order parameter, the first (raw) moment and cumulant, refers to location and is commonly known as the mean μ of a distribution. If the location is taken with respect to the mean, then the centralized location may be defined as zero. Second order parameters describe the scale of a distribution. The most common scale-defining parameter is expressed in the centralized form and known as the variance of a distribution. Due to its second order dependency, it is defined in a quadratic manner as the square of the standard deviation (σ^2). The variance equals the second central moment and the second cumulant. If standardization is applied, then the scale parameter becomes unity due to division by σ^2.

Any higher order parameter defines a part of the shape and is also commonly given in centralized and normalized form. The standardized third order parameter is the skewness of a distribution, which can be formalized using the third standardized moment and the third cumulant. Typical choices of the fourth order shape parameter are the fourth standardized moment, better known as kurtosis and the fourth cumulant, known as excess kurtosis. Excess kurtosis relates to the kurtosis above a normal distribution and is hence obtained via $\mathrm{excess\ kurtosis} = \mathrm{kurtosis} - 3$. Any parameter above the 4^{th} order is

called a higher-order parameter. A moment or cumulant generating function can be defined in most cases to easily obtain any order of interest using differentiation and evaluation at zero. The cumulant generating function is specified as the natural logarithm of the moment generating function. Depending on the type of probability distribution, only some orders may be nonzero.

APPEALING PROPERTIES OF A NORMAL DISTRIBUTION

The pdf of a univariate normal (or Gaussian) distribution in the scalar variable x is defined by equation 1.1, where μ is the mean and σ^2 the variance.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1.1}$$

The normal distribution is fully characterized by its mean and standard deviation (or variance)

The moment generating function of the univariate Gaussian is given by $M(t) = e^{\mu t + \frac{1}{2}\sigma^2 t^2}$ and the cumulant generating function by $g(t) = \ln M(t) = \mu t + \frac{1}{2}\sigma^2 t^2$. As the cumulant generating function is a second order polynomial in t, any cumulant above second order is zero. Similarly, all moments above order two are only combinations of the mean and standard deviation (see also table A.1 in appendix A). Consequently no shape parameters exist (or they are merely combinations of the location and scale parameters), such that the normal distribution is fully characterized via its mean and standard deviation (or variance). This property of the normal distribution is the reason for the assumption of Gaussianity being found in lots of places throughout the space surveillance loop depicted in figure 1.3.

In the general case of a multivariate normal distribution the mean becomes a vector and the variance a matrix, which is known as variance-covariance matrix or simply covariance matrix. The mean and the covariance suffice to fully characterize the pdf, which in its multivariate form, is given by equation 1.2 with $\boldsymbol{\mu}$ denoting the mean vector, $\mathbf{P}$ the covariance matrix, $(\cdot)^T$ the matrix transpose, $|\cdot|$ the matrix determinant and $(\cdot)^{-1}$ the matrix inverse.

$$f(x_1, \ldots, x_n) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{P}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^n |\mathbf{P}|}} \tag{1.2}$$

Customary abbreviations of the multivariate normal pdf are $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{P})$, $\mathcal{N}(\mathbf{x}-\boldsymbol{\mu}, \mathbf{P})$ or simply $\mathcal{N}(\boldsymbol{\mu}, \mathbf{P})$. Given a purely probabilistic description of the uncertainty, the assumption of Gaussianity allows to treat the terms uncertainty and covariance as synonyms (it should however be kept in mind that despite the popularity of these assumptions, this formally implies epistemic uncertainties to be negligible and that it is appropriate to express the randomness via a Gaussian pdf).

The covariance matrix of an n-dimensional multivariate normal distribution is best thought of as a hyper-ellipsoid in n-dimensional space. Geometrically this is easiest depicted for two and three dimensions,

where the representations are ellipses and ellipsoids. The ellipsoid is of special importance to any space surveillance task that works with Cartesian coordinates, as then the upper and lower 3×3 sub-matrices of the six-dimensional covariance matrix correspond to the position and velocity covariance ellipsoids. The main axes of an n-dimensional hyper-ellipsoid are given by the eigenvectors of its covariance matrix. As the square roots of the eigenvalues that correspond to the eigenvectors equal the standard deviation along that direction, the axes lengths of an m-sigma, n-dimensional hyper-ellipsoid are computed via $m\sqrt{EV(i)}$, with $i = 1 \dots n$. The eigenvectors and eigenvalues may be conveniently found using spectral decomposition.

For a one-dimensional Gaussian pdf 99.7% of the probability mass is located within three sigma of its mean. For higher dimensions the corresponding m-sigma rules are computed as the cdf of the chi-squared distribution with n degrees of freedom, evaluated at $x = m^2$. This may be compactly written using the generalized regularized incomplete gamma function Q, via $Q(\frac{n}{2}, 0, \frac{m^2}{2})$. Using $n = 2$ and $m = 3$ one finds that 98.9% of the probability mass of a two-dimensional Gaussian is contained within a three-sigma ellipse around the mean. Similarly for $n = 3$ and $m = 3$ the ellipsoidal three-sigma limit is computed as 97.1%.

COVARIANCE VS. UNCERTAINTY REALISM

Whenever the statistical uncertainty in a system is assumed to be aleatoric and Gaussian, it can be described using a mean and covariance. In this light, *covariance realism* requires the mean and covariance obtained by an estimator to equal the true mean and covariance, such that the overall uncertainty (the location and the scale of the pdf in the Gaussian case) is properly characterized.

Covariance realism is a necessary, but not a sufficient condition for uncertainty realism

If the uncertainty cannot be described by a Gaussian pdf, covariance realism still only refers to the consistency of the first two statistical moments and therefore does not imply that the overall uncertainty is correctly estimated. Instead, *uncertainty realism* is required, which for aleatoric uncertainties implies the correct estimation of the pdf. Covariance realism is therefore a necessary, but not a sufficient condition for the more general uncertainty realism. If the uncertainty is in fact Gaussian, both properties coincide.

To achieve uncertainty realism in a system, the prevalent origins of uncertainty must be identified and properly quantified for the estimator to consider the uncertainty in the estimation process. It must also be decided if an estimator is used that assumes Gaussianity or estimates the entire pdf. While the first type is simpler to implement and executes faster, it cannot maintain uncertainty realism if the pdf becomes non-Gaussian. If epistemic uncertainties cannot be neglected, uncertainty realism cannot be achieved using probability density functions and more generalized concepts of uncertainty, which also support sequential data processing, need to be applied.

1.3.3 *Principal error sources and origins of uncertainty*

Before origins of uncertainty may be discussed, it is worthwhile to provide an overview of prevailing perturbations considered for orbit propagation and related models. Perturbations are all accelerations other than the gravitational two-body acceleration. Dominant near-Earth forces and associated models are collected in table 1.2.

Perturbation	Modern/well-established models
Aspherical gravity field	Analytic (low precision, e.g. GP), EGM-96, GGM02(S/C), TEG-4, GGM05(S/C), EGM-08
Third body perturbations	Analytic. Numerical instability resolved via Taylor expansion or Legendre polynomials. Position of third body estimated analytically or obtained via Jet Propulsion Laboratory (JPL) development ephemerides (DE)
Solar Radiation Pressure	Analytic. Constant or piecewise constant SRP coefficient. Macro-models for high precision applications. Different shadow models (cylindrical vs. umbra/penumbra). Most important third bodies to be considered: sun and moon
Atmospheric drag	Analytic (low precision, e.g. GP), Jacchia-70, Jacchia-Bowman-2008, JBH09†, MSISE-90, NRLMSISE-00, DTM-2012, DTM-2013

Table 1.2: Overview of dominant disturbing forces and modern/well-established models used for space surveillance

The perturbations listed in table 1.2 are sufficient for accurate OD and form a good compromise between precision and runtime. Their relative influence depends on the orbital altitude. A comparison is given in Montenbruck and Gill (2000, Section 3.1). In case of high-precision requirements also further perturbations, such as additional third bodies (Jupiter, Venus), Earth radiation pressure (albedo and Earth infrared radiation), tides (solid Earth, ocean and polar), relativistic effects and thermospheric winds may be accounted for. Models of these perturbations are also explained in Montenbruck and Gill (2000, Section 3.7).

In contrast to the aforementioned perturbations, satellite thrusting is a planned perturbation. If maneuver plans are shared (e.g. sent from a spacecraft O/O to CSpOC or via a coordinating institution like the Space Data Association (SDA)), maneuvers can be included in the orbit propagation. Without knowledge of a maneuver, only posterior maneuver-detection is possible, which gives rise to increased uncertainties in the orbital paths of the objects capable of thrusting. Maneuver detection is especially challenging for satellites that use electric propulsion due to very low thrust levels.

† JBH09 = Jacchia-Bowman-2008 + High Accuracy Satellite Drag Model (HASDM) + Anemomilos solar storm prediction model

Referring to figure 1.3, different types of uncertainty (structural, parameter, input, algorithmic, observation or misassociation uncertainty) may be associated to each space surveillance task. In the following, the origins of uncertainty in these blocks are introduced and summarized in figure 1.7.

ASTRODYNAMIC FORCE MODEL UNCERTAINTIES

Earth Gravity Model

Earth's aspherical gravity field is by far the most dominant perturbing force up to geostationary altitudes. Modern Earth gravity models on the other hand are highly precise and range up to degree and order 2159 in case of EGM-08. This is well above the level required for orbit determination, such that gravity uncertainty can in principle be kept small. However, this level of accuracy has to be paid with significant computing time, such that greater uncertainty is often accepted for performance reasons. The International Earth Rotation and Reference Systems Service (IERS) 2010 conventions recommend a potential field truncation at degree/order 90 for LEO satellites, degree and order 20 at an orbit radius of 12 270 km and degree/order 12 for medium Earth orbit (MEO) GPS altitudes, based on expected accuracy levels of better than 0.5 mm (Petit and Luzum, 2010). At geostationary Earth orbit (GEO) altitudes even smaller fields are sufficient.

The computational complexity of the aspherical gravity acceleration is quadratic. For each degree/order combination the sums of the spherical harmonics components need to be computed, which also requires the associated Legendre function of each degree/order pair (see section 8.6.1 in Vallado, 2013 for the details). For a 90×90 field this requires approximately 8000 iterations to compute the spherical harmonics and, depending on the implementation, another ~8000 for the associated Legendre functions – at every single time-step of the orbit propagation. In LEO, gravity fields are therefore often truncated to a lower degree/order than recommended by the IERS 2010 conventions (e.g. 36×36), in favor of greater computing speed. Gravity uncertainty is therefore of dominant importance for low Earth orbits.

The mismodeling due to potential field truncation is of epistemic nature and called error of omission. All coefficients considered in the computation of the perturbation also possess an associated aleatoric uncertainty. Errors due to this model parameter uncertainty are referred to as errors of commission.

Work on gravity field uncertainty quantification dates back to Kaula (1959). Other researchers, such as Gersten, Heiskanen, Moritz or Pechenick have addressed further parts of the topic. James Wright worked on it for ~30 years and authored multiple papers on different aspects of gravity uncertainty, including an algorithm suitable for sequential filtering. An improved and extended version of his algorithm was derived as part of this thesis. The topic is discussed in chapter 3.

Third body perturbations

Third body perturbations are computed analytically by applying Newton's second law to determine the total gravitational acceleration experienced by a satellite and Earth. Subsequent subtraction then yields the overall acceleration, which in case of n additional bodies is given by equation 1.3,

$$\mathbf{a}_{\oplus,\text{🛰}} = \underbrace{-\frac{G(m_{\text{🛰}} + m_{\oplus})}{r_{\text{🛰},\oplus}^3}\mathbf{r}_{\text{🛰},\oplus}}_{\text{Two-body}} + \underbrace{G\left(\sum_{j=1}^{n} m_j \left(\frac{\mathbf{r}_{j,\text{🛰}}}{r_{j,\text{🛰}}^3} - \frac{\mathbf{r}_{j,\oplus}}{r_{j,\oplus}^3}\right)\right)}_{\text{Perturbation}} \tag{1.3}$$

where G denotes Earth's gravitational constant, $m_{\text{🛰}}$ the mass of the satellite, $m_{\oplus}$ the mass of Earth, $\mathbf{r}_{\text{🛰},\oplus}$ the vector from Earth to the satellite, $\mathbf{r}_{j,\text{🛰}}$ the vector from the satellite to the j^{th} body and $\mathbf{r}_{j,\oplus}$ the vector from Earth to the j^{th} body. The first term is the two-body acceleration. The second represents the perturbation due to additional bodies.

Error sources which introduce uncertainty into the acceleration are dominantly caused by inaccurate knowledge of the inertial position vectors. These errors are therefore of epistemic nature. A recommendation is hence to use the precise JPL planetary ephemerides instead of analytic approximations to reduce uncertainty.

Equation 1.3 is not always numerically stable, since the cubed distances between the satellite and the third body, as well as the Earth and the third body are very similar, such that both fractions in the perturbing term are very small and highly alike. The subtraction of these similar values isn't robust and introduces additional numerical errors. Computer implementations therefore sometimes make use of alternate representations which approximate the perturbing term using Taylor series expansion or Legendre polynomials. In both cases algorithmic uncertainty is introduced due to infinite sum truncation.

Additional comparatively small sources of uncertainties are model parameter uncertainty for Earth's gravitational constant G and the neglect of the satellite mass in the two-body component of the acceleration.

Solar radiation pressure

The SRP perturbation is caused by photons ejected from the Sun striking the RSO, which leads to a slight acceleration or deceleration depending on the current direction of travel. Also a torque is generated, especially in case of asymmetric shapes. The resulting effects on the orbital state vector are periodic variations in all elements, as well as secular changes of the ascending node, argument of perigee and mean anomaly (Vallado, 2013).

Since the perturbation is caused by photons, its altitude dependence is coupled to the photon density and therefore to the solar flux, which has a quadratic dependence on the distance from the Sun. As for

Earth orbiting objects the distance is mostly influenced by the orbit of Earth around the Sun and not satellite altitude, it is often set to the semi-major axis of Earth with respect to the Sun, such that the average solar flux constant of $1367\,W/m^2$ may be used to estimate the incident solar pressure via $p_{SRP} \cong \frac{1367\,W/m^2}{3 \cdot 10^{-8}\,m/s}$. From LEO to GEO its magnitude is therefore nearly constant.

The incoming solar radiation pressure may be translated to acceleration via equation 1.4, where c_R is an effective constant of reflectivity, $\mathbf{r}_{\odot,\text{🛰}}$ the vector from the satellite to the sun and $A_\odot$ the exposed surface perpendicular to $\mathbf{r}_{\odot,\text{🛰}}$. The multiplicatively coupled term $c_{SRP} := \frac{c_R A_\odot}{m}$ is also known as solar radiation pressure coefficient.

$$\mathbf{a}_{SRP} = -p_{SRP} \frac{c_R A_\odot}{m} \frac{\mathbf{r}_{\odot,\text{🛰}}}{r_{\odot,\text{🛰}}} \tag{1.4}$$

The overall effect of solar radiation pressure depends strongly on c_{SRP}. For most objects it is not a dominant perturbation, however for GEO objects and those with a high area to mass ratio (HAMR) it may be of greatest influence. Also it is a significant contributor to propagation uncertainty.

Equation 1.4, while being the baseline in many OD applications, is only a very simplified model and therefore suffers from structural uncertainty. A very general model may be found in Wetterer et al. (2014). Precise models need to evaluate the acceleration on every part of the object's surface, instead of forming an average over the entire body, such that both macro-models of the object shape and attitude information become necessary. Similarly, a single coefficient of reflectivity also represents a simplification. A more precise model requires a reflectance distribution function that takes both specular and diffuse reflection into account. This in turn necessitates knowledge about the material properties of an object.

This structural uncertainty is directly followed by input model uncertainty, considering that also the constituents of the solar radiation pressure coefficient in equation 1.4 are very difficult to obtain, since they can neither be measured from ground nor are they constant. As the necessary degree of information required for high precision computations is unavailable for almost all objects and due to the entire input uncertainty being multiplicatively coupled into a single parameter c_{SRP}, the solar radiation pressure coefficient is often estimated along with the satellite state as part of the orbit determination.

This approach *works*, but it also introduces additional difficulties. Loosely constrained solve-for parameters are likely to experience observability issues, which causes the uncertainties to remain unrealistically large. Solve-for parameters are only sensitive to the models used. Any error in the force models will be compensated to the maximum possible extent by the solve-for parameters. This effect is known as "error aliasing". The accuracy of the solve-for parameters can thus be

increased by improving their models, as well as more accurate modeling of other forces. Consequently, estimates of a *constant* solar radiation pressure coefficient and its uncertainty, even though they often minimize the residuals with respect to the observations, are often suffering from considerable error aliasing. Hence, if a solar radiation pressure coefficient is estimated, it should be modeled at least piecewise constant, such that the variability of the parameter can be partially captured. This is the current practice applied at CSpOC (Schiemenz et al., 2019a).

A second difficulty due to multiple solve-for parameters with different units is an increase of the matrix conditioning numbers, which complicates the matrix inversion required for OD. Advanced inversion algorithms, such as presented in Rump (2009), are recommended to avoid additional algorithmic uncertainty.

Despite circumventing some of the uncertainty in c_{SRP} via estimation, additional sources of epistemic and aleatoric uncertainty exist. The use of a solar *constant* results in structural uncertainty. Its adopted value furthermore suffers from model parameter uncertainty, which has not been properly quantified (Cook, 2001). Another source of epistemic input uncertainty is the vector from the satellite to the sun. A standing recommendation is therefore to use the JPL DE ephemerides to obtain this direction.

Last but not least, solar radiation pressure can only exert a force on a resident space object if it is exposed to sunlight. Earth shadow models are therefore required to determine the exact times of shadow entry and exit. Typical shadow models are the simple cylindrical model and the accurate conic umbra/penumbra model. The cylindrical model introduces structural uncertainty due to its simplifying shadow assumptions, whereas the conic model gives rise to structural uncertainties during penumbral force modeling. Both models are furthermore susceptible to algorithmic uncertainty, if the orbit propagation is not integrated to the exact times of eclipse entry and exit. Precise shadow models also need to check the position of the Moon (again best via JPL ephemerides) to determine if it is causing partial or full solar eclipsing.

Atmospheric drag

Atmospheric drag is a well known phenomenon, since it is not restricted to space, but can be experienced on a daily basis. Four factors influence the personal experience of air resistance: velocity, winds, the area exposed to the environment and the object mass. When stationary, atmospheric drag is only perceived in case of strong winds. This is also when the effect of mass can be experienced, as lightweight people are more affected by gusts than heavy ones. Once an object starts moving, winds still have an impact, but personal experience indicates that the air resistance increases at higher speed. Therefore racing cyclists make sure to ride in a crouched posture to present as little area to the environment as possible.

The same concepts also apply in space. Atmospheric drag is caused by momentum transfer from molecules and atoms in the atmosphere to the RSO. On ground the air density is about $1.225\,kg/m^3$ according to the international standard atmosphere. For space surveillance the upper part of our atmosphere, which starts at ~90 km, covers the thermosphere (90-600 km), exosphere (> 600 km) and extends until upper LEO altitudes, is of greatest interest. Since atmospheric density decreases exponentially with altitude, the total mass density at 400 km is only $\sim 2\,g/m^3$ (Emmert, 2015). However, due to the extremely fast motion of the RSOs ($\gtrsim 7\,km/s$ in LEO), this residual atmosphere is sufficient to significantly influence the orbital paths of objects in low Earth orbit – especially at altitudes below ~1200 km.

The basic aerodynamic formula is given in equation 1.5. A certain analogy to the solar radiation pressure equation may be identified as a result of the similar concepts.

$$\mathbf{a}_{drag} = -\frac{1}{2}\frac{c_D A}{m}\rho v_{rel}^2 \frac{\boldsymbol{v}_{rel}}{v_{rel}} \tag{1.5}$$

The effect of thermospheric winds is contained in $\boldsymbol{v}_{rel}$, the velocity vector relative to the atmosphere. Thermospheric winds typically range from a few meters per second up to 2-3% of LEO velocities (Liu et al., 2016), however also momentarily stronger gusts may occur. Thermospheric winds can be estimated and included into the drag acceleration computation using empirical models, such as HWM-14 (Drob et al., 2015).

The susceptibility to drag is described via the "ballistic coefficient"[11], commonly denoted BC or B and defined as $BC := \frac{c_D A}{m}$. The same kinds of uncertainties apply to the ballistic and the solar radiation pressure coefficient. Due to its input uncertainty, the ballistic coefficient is also often appended to the solve-for state vector during orbit determination, thereby causing error aliasing and increasing susceptibility to a lack of observability.

The most critical parameter however is not the ballistic coefficient, but the thermospheric[12] mass density at the location of the RSO, which is denoted by ρ (Schiemenz et al., 2020a). To understand the challenges in obtaining accurate density estimates and the corresponding uncertainties, an overview of the fundamental processes that influence local thermospheric density is required. These are temperature, species composition and horizontal variations (dynamics).

The *composition* of the thermosphere is defined by the concepts of hydrostatic balance (canceling gravitational force and vertical pressure gradient) and "molecular diffusion", which describes the separation

11 Some authors also define BC as the *inverse* ballistic coefficient. The exact definition should therefore always be checked to avoid confusion

12 In this work the terms "atmospheric", "thermospheric" and "exospheric" are treated as synonyms when discussing density uncertainty

of the well-mixed fluid found at its bottom into individual gases according to the molecular masses of its constituents. Heavier gases fall off more quickly than lighter ones and hence do not reach as high altitudes. The dominant gas between ~200 km and 600 km (solar minimum)/900 km (solar maximum) altitude is atomic oxygen, which is created from the decomposition of molecular oxygen upon exposure to EUV and FUV radiation. It is hence causing most of the drag for satellites orbiting in this regime. Above 600-900 km the mass densities of the light species, especially helium, start to dominate over atomic oxygen (Emmert, 2015). The energy of the incoming radiation determines how deep it enters the atmosphere. The shorter the wavelength, the higher its energy and the earlier it gets absorbed. Consequently, the portion of the overall mass density which may be attributed to ionized gases (plasma) increases with altitude.

Temperature variations cause the atmosphere to expand and contract, thereby changing its density. The alternation of expansion and contraction is sometimes referred to as "atmospheric breathing". The thermosphere is a highly driven environment for which the forcing processes that determine its temperature need to be captured within a model to derive density estimates. Thermospheric heating is largely caused by radiative heat transfer, i.e. near, middle, far and extreme ultraviolet (UV) radiation. The dominant wavelengths are within the EUV and FUV bands. Secondary sources of heating include molecular thermal conduction (collisional heat exchange) from highly energetic particles (e.g. within the magnetosphere), chemical reactions, frictional heating between ions and neutral species, the dissipation of electric currents and upward propagating waves (Emmert, 2015). Cooling is achieved via infrared radiation. Since the dominant species in the upper thermosphere (atomic oxygen, molecular oxygen and molecular nitrogen) do not radiate effectively in infrared wavelengths, any heat absorbed in the upper thermospheric altitudes needs to be transported downwards via collisions until sufficient concentrations of carbon dioxide and nitric oxide are encountered (at ~120 to 150 km altitude), which are capable of effective IR radiation (Roble, 1995). It has also been found that breaking atmospheric gravity waves generate turbulence that transports heat downwards to the mesosphere. Therefore these so-called "buoyancy waves" represent an additional source of thermospheric cooling (Apruzese et al., 1984). As cooling is only effective in the lower thermosphere, its temperature increases quickly with altitude (hence its name) and reaches an asymptotic boundary value of 600 K to 1400 K near the exosphere. The temperature gradient at the lower boundary, the current local temperature and the asymptotic exospheric temperature are key parameters of any semi-empirical model in the computation of a density estimate.

A graphical summary of the atmospheric temperature (including heating/cooling mechanisms) and composition is given in figure 1.4.

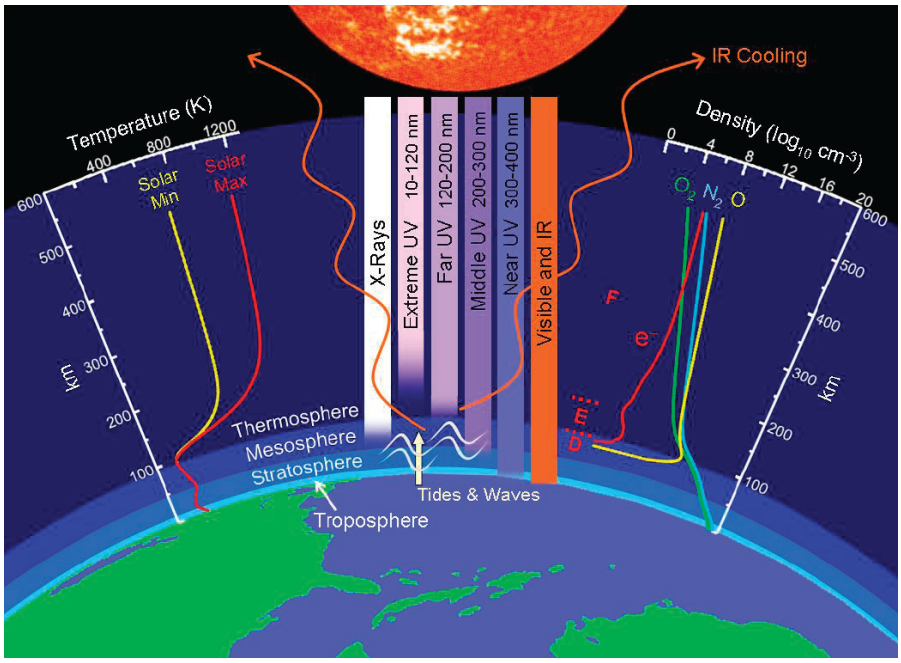


Figure 1.4: Atmospheric structure, dominant species number density, temperature ranges and heating/cooling processes. Source: Emmert (2008). Reprinted with permission

The *dynamics* of the thermosphere, i.e. horizontal (latitude/longitude) changes in density, is predominantly influenced by the spatial heat distribution. Local density variations are therefore closely linked to solar periodicities and the Earth's rotation. Atmospheric density models need to account for diurnal variations caused by Earth's rotation, the 27-day solar rotation cycle, seasonal variations caused by the eccentric orbit of Earth around the Sun and the ~11 year solar cycle.

Temperature gradients are accompanied by density gradients, which furthermore lead to wind systems that interact with the natural circulation of the atmosphere caused by the rotation of Earth. Winds also change the distribution of energetic particles, thereby giving rise to current systems (Emmert, 2008).

A second source of horizontal thermospheric density variability is the Earth's magnetosphere. The magnetosphere acts as a shield against the influence of the solar wind, deflecting its charged particles along the magnetic field lines, thereby causing atmospheric compression on the dayside and expansion on the nightside. A fraction of the solar wind reaches Earth's polar cusp, where the charged particles precipitate along the magnetic field lines into the high latitude region of the thermosphere, causing much of their energy to be absorbed near the magnetic poles. Some reference changes of density due to the different dynamic drivers are found in Emmert (2015) and listed hereafter. At an altitude of 400 km, atmospheric density:

1. varies by a factor of 10 between solar minimum and maximum
2. is about 3.5 times larger at the dayside than the nightside

3. is 50% larger in the summer season than in the winter season
4. increases by 40% during a geomagnetic storm classified via a K_p index of 5
5. varies by 19% due to seasonal oscillations
6. varies by 13% due to annual oscillations

These variations give an impression of the importance to faithfully account for the solar and geomagnetic energy input, as well as the uncertainties associated with imperfect knowledge. Information about solar (especially EUV) and geomagnetic activity is hence vital to any atmospheric model, as otherwise no meaningful estimates of density can be computed.

Atmospheric models differ in their nature. Some are empirically dominated, meaning that only fundamental physical relationships are implemented in a model and that the critical model parameters are constructed from optimal averaging of measurements taken by various sensors over different periods of time. These semi-empirical models hence represent the average behavior during the measurement intervals and do not contain features that were not observable in the underlying measurement database. The static nature of these models is one of their main drawbacks. Methods to improve density "nowcasting", i.e. adapting semi-empirical model parameters to better reflect the current situation, have been developed. A prominent example is the High Accuracy Satellite Drag Model (HASDM) (Storz et al., 2005). The method makes use of drag-derived density data from calibration-satellites with known area to mass ratio in order to debias the underlying atmospheric model. These techniques however have so far only been deployed against selected density models. Furthermore, the need of the calibration satellite measurements restricts the application of this technique to a very limited community that has access to the required data.

Another class of models is physically dominated and aims to solve the fluid and photo-chemical equations that govern the atmosphere. Operational satellite orbit determination and propagation exclusively make use of semi-empirical models for their fast and pointwise generation of density estimates, however there is a possibility of a paradigm shift in the distant future. Semi-empirical models currently feature similar accuracies as physical ones and are far less computationally demanding (National Research Council, 2012).

Applying this information to density *uncertainty*, different types may be identified. The semi-empirical models suffer from structural uncertainty due to their static nature and limited physics, both at at spatial scales within and below the model resolution. This uncertainty is known as grid-scale and subgrid-scale density uncertainty. Grid-scale density uncertainty is also referred to as average density uncertainty and a significant contributor to the overall uncertainty realism, since it is applicable at all times.

Another source of structural uncertainty is caused by the choice of the parameters that are used to represent the solar and geomagnetic forcing of the atmosphere. Since these parameters are only partially capable of describing the true solar and geomagnetic energy input, they are referred to as "proxies". A very common choice for the solar forcing is the F10.7 proxy, which is a measure of the solar radio flux at 10.7 cm wavelength and correlates well with the EUV irradiance. The index is given in solar flux units (sfu), where $1\,\mathrm{sfu} = 10^{-22}\,\mathrm{Wm^{-2}Hz^{-1}}$.

Models typically use two F10.7 inputs: a daily flux input of the previous day and some form of 81-day average which corresponds to three solar rotations. The reason for the two parameters is a separation of faster and slower variations in the solar flux input. Both types have different impacts on the thermosphere. The slow variations are represented by the running 81-day average and fast variations via the difference between the daily value and the 81-day average (Dudok de Wit and Bruinsma, 2017). Caution is required for the long-term average, because some models use the average of the past 81 days, whereas others were constructed using a centered 81 day average value.

F10.7 is not the only proxy available and it has been recognized that the choice of the proxy/proxies itself has a profound impact on the overall density uncertainty (Marcos, 2006; Dudok de Wit and Bruinsma, 2011). Hence, in the development of the JB2008 model, significant focus has been put on the inclusion of heating-processes at other wavelengths. It therefore requires multiple solar proxies (F10.7, S10 (He-2), M10 (MG-2) and Y10 (X-ray flux)) (Bowman et al., 2008). While the incorporation of multiple proxies generally improves data fitting residuals, Bruinsma and Dudok de Wit mention that this approach has its limit in terms of statistical significance (Dudok de Wit and Bruinsma, 2017). Also Emmert reports that it is "virtually guaranteed" that a combination of proxies is able to reduce the fit errors during model construction, however that "the statistical significance of the additional terms has not yet been demonstrated" (Emmert, 2015).

The proxies used to model the solar and geomagnetic forcing are further subject to input parameter uncertainty. This is true not only for real-time operations and orbit forecasting, where predicted proxy values are used due to lack of final proxy data, but also for published proxy information, due to the accuracy of the measurements themselves, the spatial distribution of the measurement stations and temporal undersampling (Xu, 2008; Tapping, 2013). Furthermore different data sources often report slightly different values, which literally also classifies as input uncertainty.

An excellent overview of the F10.7 solar proxy generation is collected in Tapping (2013). A comprehensive survey of geomagnetic indices may be found in Menvielle et al. (2011).

In addition to input parameter uncertainty, further algorithmic uncertainties result from improper proxy handling. Examples are proxy interpolation (some models require the proxies to be interpolated with respect to time, whereas others like NRLMSISE-00 do not document the expected procedure at all), improper selection of proxy data (adjusted F10.7 vs. observed F10.7, last vs. central 81-day average), proxy conversion (e.g. between geomagnetic Ap and Kp) and different algorithms used for these operations (interpolation/conversion). Also the choice of a model itself represents a significant source of variability to the orbit propagation and calls for proper uncertainty quantification. A critical assessment of these factors, including experimental tests, is compiled in Vallado and Finkleman (2014).

Overall, atmospheric drag is the perturbation with the largest associated uncertainty (Poore et al., 2016). Due to its exponential altitude dependence, only LEO RSOs are affected, however this still accounts for 78.2% of the TLE catalog at the time of writing (May, 2020). Since drag is a non-conservative force, its errors (and therefore the associated uncertainty) accumulate over time, such that the entire evolution of the density errors determines the effect on the orbit and uncertainty propagation, giving rise to overly optimistic uncertainty estimates if density uncertainty is neglected during OD of LEO RSOs.

In this thesis for the first time complete physics-based analytic covariance matrices are derived, which are able to quantify the propagation uncertainty due to three origins of density uncertainty: solar flux input errors, magnetic index input errors and structural model errors. These contributions utilize the seminal work of Emmert et al. (2017) to translate the relative density uncertainty to mean motion and mean anomaly uncertainty and extend it towards a combined density uncertainty quantification framework for orbit determination. The developments are derived and presented in chapter 2.

UNCERTAINTY MODEL, UNCERTAINTY PROPAGATION, ORBIT DETERMINATION

Given a proper force model uncertainty quantification, the remaining building blocks of the space surveillance loop need to make sure that this uncertainty is faithfully translated into state vector uncertainty.

The *uncertainty model* is a critical factor for the reliability and meaningfulness of the SST computations. The first choice to be made is the mathematical treatment of uncertainties. As explained in section 1.3.1, currently a probabilistic uncertainty model is the state of the art (often in combination with the assumption of Gaussianity in operational environments), although this might change in future with the advent of OPMs. A disadvantage of probabilistic uncertainty descriptions is that they are only well suited for aleatoric uncertainties. Consequently, epistemic uncertainties need to be neglected or approximated with pdfs, which can be problematic at times, since epistemic uncertain-

ties cannot be characterized by stochastic moments, resulting in their ambiguous interpretation as error bounds.

A fundamental requirement of any operationally relevant uncertainty model is the ability to perform sequential updates once new evidence[13] is available. All algorithms relevant for SST rely on Bayesian inference, which is based on Bayes' theorem, for this purpose. The sequential updates lead to the classical recursive predictor-corrector nature of the Bayes estimator. The working principle for a probabilistic uncertainty modeling is depicted in figure 1.5, where the random variables x_k and z_k denote the state vector and evidence provided at time t_k, $p(x_0|z_0)$ is the initial probability density, $p(x_k|z_{k-1})$ the predicted pdf and $p(x_k|z_k)$ the updated posterior pdf. The prediction error $p(z_k|z_{k-1})$ normalizes the corrected probability density.

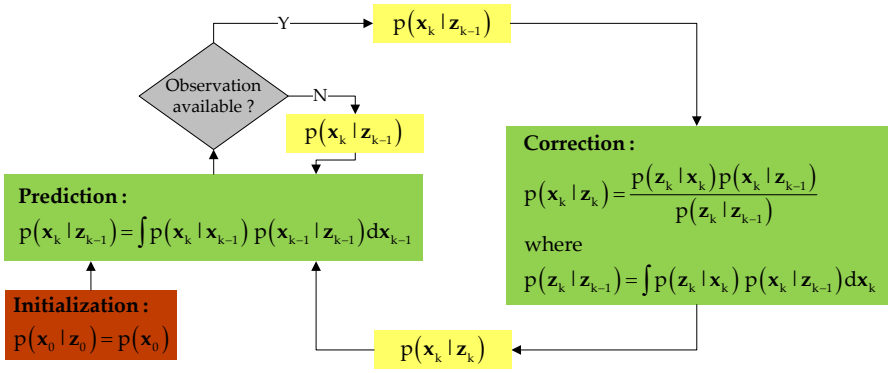


Figure 1.5: Recursive predictor-corrector nature of the Bayes estimator (adapted from Horwood et al., 2011)

The prediction step, i.e. the time evolution of the pdf, is governed either by the Chapman-Kolmogorov (discrete time steps, shown in figure 1.5) or the Fokker-Planck equation[14] (continuous time evolution). Whenever new evidence is available, Bayes' theorem is used to obtain the corrected (a posteriori) pdf from the predicted (a priori) probability density, the measurement model and the measurements.

Within a probabilistic setup, the uncertainty model further determines if the assumption of Gaussianity is enforced in the uncertainty propagation, such that either covariance propagation/estimation is performed or the full pdf is propagated/estimated. Any initially Gaussian pdf will eventually become non-Gaussian under nonlinear transformations, such as long-term orbit propagation. This on the other hand does not imply that advanced estimation techniques like Gaussian Mixture Model (GMM)[15] filtering are necessary for practical operations. If measurement update times can be kept sufficiently short,

13 In most cases *evidence* refers to measurements/observations, however in principle state and uncertainty updates can also be performed using other data sources, e.g. TLE files (Delande et al., 2018b)

14 The Fokker-Planck equation is also known as Kolmogorov forward equation, which is why it is often collectively referred to as Fokker-Planck-Kolmogorov equation (FPKE)

15 GMM is a synonym to Gaussian sum

the assumption of purely Gaussian uncertainties can be maintained. Furthermore, it has been recognized that the choice of the coordinate frame for the state vector (Cartesian position and velocity vs. polar coordinates or equinoctial orbital elements) has a significant impact on covariance realism. Orbital element space (e.g. equinoctial coordinates) often allows for covariance consistency up to three times longer than Cartesian state vectors. Relevant publications on this topic include Junkins et al. (1996), Sabol et al. (2010), Alfriend and Park (2016), Woodburn and Coppola (2013) and Ghrist and Plakalovic (2012).

Once an uncertainty model sufficient for uncertainty realism has been selected, algorithms must be picked that meet these requirements. This applies to *uncertainty propagation*, where a common choice is the use of linear or nonlinear (e.g. sigma point) uncertainty propagators and orbit determination (filters, batch estimators, mixture models). In some cases also linear covariance propagation techniques can be used for pdf propagation (e.g. for GMM filtering via classical linearized Kalman filters). If it cannot be guaranteed upfront that the pdf remains Gaussian during all encountered propagation periods, a nonlinearity check must be implemented to treat nonlinearities when they occur. Alternatively, one may also always use filters that estimate the entire pdf, however this is somewhat like using a sledgehammer to crack a nut.

Despite the ability to meet the uncertainty model, the algorithm used for *orbit determination* should also be capable of handling process noise to properly account for the astrodynamic force model uncertainty. This precludes some of the classic implementations of the batch estimators, where process noise is commonly neglected. A process noise extension of the classical batch estimators has been worked out as part of this thesis and is presented in chapter 4.

An improper choice of the uncertainty model and/or the algorithms used can lead to corruption of uncertainty realism or waste valuable computing time. Some applications however may also tolerate a certain degree of unrealistic uncertainties, such that a sensitivity study with respect to covariance/uncertainty realism is advised.

MEASUREMENT GENERATION AND DATA ASSOCIATION

Measurement data is used in the orbit determination process to reduce the orbital uncertainty that accumulates during the propagation phases. An optimal weighting of the measurement uncertainty and the propagation uncertainty allows to compute minimum variance estimates of the chosen state vector. If one of these two sources of information suffers from degraded uncertainty quantification, this results in the OD algorithm to carry out an inferior weighting, leading to suboptimal estimates of the state and its uncertainty. Correct characterization of sensor-level uncertainties is therefore an important cornerstone to achieving overall uncertainty realism.

Measurement uncertainty quantification requires a deep understanding of the sensor functional principle, intrinsic observables ("non-metric" measurements), temporal behavior and conversions performed to generate the reported quantities ("metric" measurements). Typical sensors used for space surveillance applications include optical telescopes (ground-based or space-based) and radars (phased-array, dish or hybrids, currently all ground-based). Satellite laser ranging and GPS are common sources of high-precision ephemerides in the sensor calibration process.

Radars

Several types of radar sensors are used for space surveillance. Depending on the actual sensor, different quantities may be reported. Among the typical ones are range, azimuth, elevation and range-rate. Some radars only report range, others may report all four quantities or even the rates of the angles. Range-rate (if available) is derived either via usage of Doppler waveforms or from differentiating the range measurements. In the latter case the measurements are not independent observables but correlated with the range measurements.

The intrinsic observables depend on the type of radar. Dish radars measure range, bearing (azimuth) and elevation (RAE). Phased array radars also measure range, however azimuth and elevation are replaced by the direction cosines U and V. The RUV coordinate system is non-orthogonal and referenced to the transmit-receive modules of the antenna (Li and Jilkov, 2001b).

Radars send out electromagnetic waves and evaluate the signal turnaround time and returned energy. For space surveillance applications they are usually operated in tracking mode, which requires knowledge of the object trajectory for proper pointing. Consequently, the orbital path must be estimated in advance to generate a pointing profile or a tracking filter must be used.

Tracking filters essentially perform a simplified orbit determination to facilitate pointing and noise reduction. Since tracking takes place in real-time, simplified dynamics models are used. An overview of ballistic target dynamics models for tracking filters can be found in Li and Jilkov (2001a). Tracking filters usually assume that their inputs have been corrected for atmospheric effects and correspond to zero-mean Gaussian random variables. Therefore, corrections to compensate the effects of ionospheric refraction and detection level (electronic) biases are performed prior to supplying measurements to a tracking filter (Poore et al., 2016). Besides being required for proper instrument pointing, tracking filters are also important to reduce measurement noise. In fact, some sensors of the US SSN would not meet the angular accuracy requirements without the noise reduction provided by a tracking filter (Poore et al., 2016).

Despite these positive effects, tracking filters also have a significant impact on data errors and their associated statistical properties. The computed state vectors and covariances highly depend on the tracking filter implementation and models used. In practice however, radar sensor noise is driven by factors such as wavelength, waveform, signal to noise ratio, beamwidth, aperture size, temperature and waveform processing (Poore et al., 2016). Tracking filters on the other hand introduce correlation into the reported measurements, as the tracking filter solution at time t_n depends on all prior states and non-metric measurements. At each time-step the tracking filter computes the reported measurements based on the state vector solution. This conversion is in many cases highly nonlinear. While covariance information may be transformed from state space to the metric observable space via linearization or nonlinear methods, it is rarely reported with the measurements (CSpOC, for example, is not receiving any covariance information from the radar sites), thus forcing the user to conduct additional calibration campaigns based only on the reported measurements, which degrades the realism of the measurement uncertainty quantification.

Radar biases are driven by environmental effects (e.g. ionospheric refraction or temperature gradients), electronic components (e.g. phase errors or scale factor errors) and pointing. Electronic biases originate from limited knowledge and stability of radar performance parameters and may be constant or vary between tracks. Correlation times are long in comparison to target observation timescales, such that they can be treated as constant during data acquisition. Nevertheless, electronic errors exist that cannot be effectively treated as biases. An example is temperature-induced expansion/contradiction, which causes changes to the transmit/receive module grid of phased array radars and thereby errors that scale with the magnitude of the direction cosines. If only biases are modeled in the calibration process, the extent to which these scale factor errors can be compensated by a bias depends on the variation of the intrinsic observables during target tracking and leads to a reporting of covariances with varying realism.

Other electronic biases are for example constant phase errors (small oscillator frequency offsets with respect to the nominal value). Generally this type of error is accounted for in the commissioning phase of a sensor via extensive calibration campaigns. The degree to which they are resolved depends on how much effort has been spent on doing so. In practice, many sensors are only calibrated close to their boresight (in favor of high signal to noise ratios) and not over the entire field of view (FOV). Nevertheless, the obtained calibration parameters are applied for the full FOV, creating pointing-dependent residual electronic biases due to the boresight-dominated characteristics. Additionally also drifts in electronic biases, caused by aging of components, represent a source of uncertainty that is not explicitly taken into account and leads to the need for repeated calibration campaigns.

Optical measurements

Optical sensors report angles only measurements, commonly expressed in right ascension and declination. Their angular accuracy is by far better than the azimuth/elevation information of radars, however range information is not provided. Telescopes and space-based space surveillance (SBSS) satellites collect incident light and report the amount of accumulated electrical charge for each pixel of the sensor array. The images formed from the pixel charges represent the intrinsic observables. Optical sensors do not need to actively radiate energy and may be operated in two different modes: surveillance and tracking. In surveillance mode the sensor is pointing into a constant direction. Hence the stars appear as bright points and the RSOs as streaks. This is a simple operation mode, however it only allows for objects to be detected that cross the field of view. In case of SBSS sensors the surveillance mode may require active attitude control or gimbal bearings.

In tracking mode the sensor follows the object either via a precomputed pointing profile or via autotracking based on the received visual magnitude. If the information about the expected orbital motion is sufficiently accurate, tracking turns the path of an object into a bright dot and the stars into streaks. Velocity errors may result in object smearing, giving rise to inaccurate results and degraded statistical properties. The purpose of the tracking mode is to extend measurement tracks and enable longer light integration times, such that also fainter objects can be tracked.

Optical measurements are processed differently than radar measurements to obtain the raw metric observables, which are then corrected for environmental effects to generate the reported measurements. Required processing-steps and corresponding sources of uncertainty are:

- Star/background identification $\rightarrow$ usage of different star catalogs
- Track identification via removal of background, end-point determination and pixel-centering $\rightarrow$ algorithmic uncertainties, uncertainty in centroid determination, presence of stars close to tracks
- Generation of raw metric observables using relative geometry of identified scenery, ground station location and tracks $\rightarrow$ accuracy of sensor location (lat./lon./alt. on ground or state vector in space), performed corrections

Once the raw observables have been computed, corrections are applied to compensate for the finite speed of light. These include light-time delay corrections that account for the motion of the observation target and aberration corrections that account for the motion of the observer. The exact aberration corrections which need to be performed depend on the coordinate system of the final reported measurements (the International Celestial Reference System (ICRS) does not account for annual aberration, whereas the Geocentric Celestial Reference System (GCRS)

does). Once light-time delay and aberration have been corrected, the last step is to perform a system delay compensation, also called "time bias" correction. In principle the time bias should be an actual bias, as it is meant to compensate for the constant delay between the light reception time and the image registration time. In practice, however, it is often computed as an overall optimization to minimize the residuals generated from the external precision ephemerides, thereby causing it to compensate for all kinds of remaining errors. Strong variations in the time "bias" are therefore often a sign of sub-optimal or erroneous prior corrections. A practical difficulty in the process is to make sure that each required correction is only applied once, as some corrections are already applied in the image processing software, whereas others are left to the user. Consequently missing or double corrections need to be avoided. A small side-task of the present thesis was to derive a relativistic aberration correction algorithm capable of adding and removing the effect of aberration to/from measurements. The algorithm may be found in appendix B.6.

The uncertainty in the fully processed metric observables is commonly determined in a calibration campaign against high precision (GPS or laser-based) ephemerides. If it is assumed to be Gaussian, then each calibration run results in a bias/standard deviation estimate for each metric observable. The results obtained for right ascension and declination are commonly considered independent, such that the final covariance has diagonal shape. Each of these bias/standard deviation tuples represents an uncertainty estimate for the time during which the respective observations were taken. The biases and standard deviations are therefore themselves subject to variation and uncertainty. For this reason, multiple calibration runs are performed under different conditions during calibration campaigns, resulting in multiple means and variances. The realism of the finally reported measurement uncertainty strongly depends on how the large number of bias/standard deviation pairs is used to generate the final uncertainty estimate. Often a simple averaging over the means and sigmas is carried out, thereby losing any information about the variability in the uncertainty with respect to the underlying dependencies. A better option is to categorize the uncertainty parameters based on their dependencies and to report the parameter set that most closely matches the current conditions. Both approaches however cannot avoid further recalibration campaigns, as typically no model is available to predict the drift in the calibration based on a change in the environmental parameters.

Once the calibration runs have been finished, a final test must show if the computed residuals in the metric observables actually follow a Gaussian pdf of the reported mean and standard deviation. This is true for both optical and radar based sensors. Possible test criteria for the evaluation of Gaussianity are the QQ-Plot with Royston-Michael or Kolmogorov-Smirnov acceptance boundaries (Michael, 1983; Royston,

1993; Wright, 2011; Johnson, 2015) or some other goodness of fit (GOF) test, such as the Cramér-von Mises criterion (Darling, 1957) or the Anderson-Darling metric (Anderson and Darling, 1954).

Summarizing measurement uncertainty quantification, it can be noted that there is still much room for improvement. Many assumptions are made (e.g. with regard to the independence of measurements and measurement variables), whose justification seems to be based more on simplicity than physics. The empirical calibrations are affected by epistemic and aleatoric uncertainty. Nevertheless, only aleatoric uncertainty is reported alongside the measurement data, which in many cases only represents an average value, such that the actual uncertainty can sometimes differ considerably from the reported one. In practice, this structural uncertainty is attempted to be compensated via regular calibration campaigns, however the reported results are always only as good as the assumptions and models they are based on. A circumstance that complicates advancements in this area and limits them to larger institutions is the actual availability of a sensor, such that calibration procedures can be tested against real world data. Some recommendations for future high-precision sensor systems may be found in Poore et al. (2016).

Measurement correlation

Once measurement data has been acquired, it needs to be matched to an object in the catalog. This task is in many cases not obvious due to the dense population of resident space objects. Incorrect association of measurement data leads to cross-tagging uncertainty. This type of uncertainty is very specific to space surveillance but easily understood. If measurements are attributed to a different object, then the (initial or regular) orbit determination of this object is performed with measurements that do not describe the true path of the object under consideration. Instead of reducing uncertainty, uncertainty realism is corrupted by trusting into false information. Once cross-tagging uncertainty has been introduced into a system, it can only be removed by further, correctly associated measurement data or by restoring the catalog state and uncertainty of the object prior to the misassociation and repeating the OD with new information regarding the correlation.

A standing recommendation is hence to use algorithms for measurement correlation that maintain multiple hypothesis at the same time and only decide upon association once enough evidence has been obtained. These types of algorithms are called multiple hypothesis trackers (MHTs). A description of the theory, including an application to real world data, may be found in Rodriguez Fernandez et al. (2019a).

As cross-tagging uncertainty is of epistemic nature and OPM-based MHTs have not yet been presented, there is currently no proper means for its quantification. Instead, since most correlation algorithms are

based on probability distributions or covariance information, the state of the art is to minimize cross-tagging by achieving uncertainty realism in the other parts of the space-surveillance loop and using MHTs for track correlation. The principle of cross-tagging is sketched in figure 1.6.

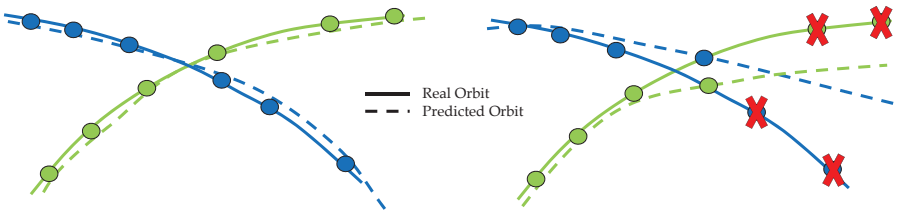


Figure 1.6: Principle of measurement cross-tagging. Left: successful correlation, right: erroneous association of measurements and uncorrelated observations due to bad OD

SUMMARY

Perfect uncertainty quantification would require all origins of uncertainty to be quantified. Obviously this is not possible in practice, however performing uncertainty quantification for the dominant sources of uncertainty would already be a substantial improvement with respect to the current operational state of the art.

A further source of uncertainty that is common to all blocks and has not yet been mentioned is related to coordinate frame differences. Examples of this kind are deviations in the definition of inertial frames (EME2000 vs. GCRF) and differences in the inertial to Earth fixed conversions caused by EOP data, which may vary depending on its source, processing (interpolation vs. nearest neighbor) and age (in case of predicted Earth orientation parameters).

Also the GP theory, which uses its own analytical models, is defined in the True Equator Mean Equinox (TEME)-frame, which has no clear scientific realization. The probably closest public implementation with respect to the USSF codes used to generate the TLEs is presented in Vallado et al. (2006).

A graphical summary of the major uncertainty contributors is given in figure 1.7, which follows the building block structure introduced in flowchart 1.3.

1.3.4 *Impact of uncertainty realism on space surveillance*

Besides aiming for maximum uncertainty realism, also an understanding of its relevance for the various tasks that make use of the OD results is required. This allows to identify where simplifications may be justified. Four main subjects make use of the estimated state and its uncertainty: maneuver detection, data association, sensor management and CARA.

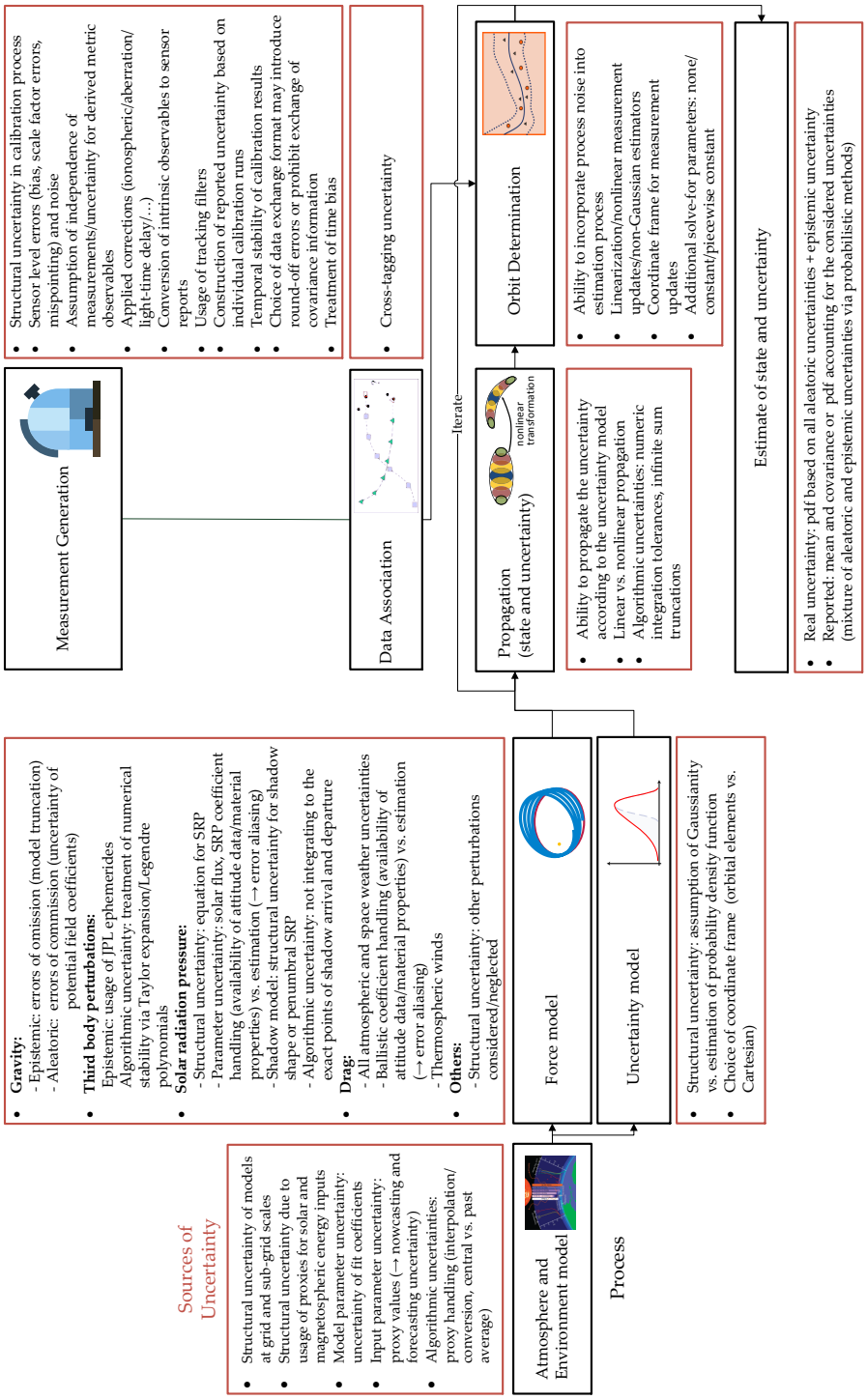


Figure 1.7: Dominant sources of uncertainty in the building blocks of space surveillance

MANEUVER DETECTION AND DATA ASSOCIATION

Maneuvers are part of orbit maintenance, orbit raising and deorbiting. Maneuver detection is closely linked with data association, as observations of objects that have maneuvered can only be correctly correlated if the maneuver has been detected. Two kinds of approaches exist:

those that aim to determine that a maneuver has occurred (maneuver identification) and those that also attempt maneuver characterization. Obviously the results of the latter type depend on assumptions regarding maneuver properties (e.g. maximum/minimum thrust), strategies (fuel-optimal/minimum duration) and type (impulsive/continuous).

Different algorithms have been proposed in the literature (cf. figure 1.3). Very promising results have been reported for the optimal control distance metric approach in Singh et al. (2012), which builds upon Holzinger et al. (2012). Common to all approaches is that a maneuver is considered as some sort of statistically rare event. Hence the success of measurement correlation is directly tied to the uncertainty information of an object, as reported by the OD and stored in the space object catalog.

Overly optimistic pdfs may result in false alarms as they increase the likelihood of reporting mismodeled dynamics as maneuvers. Correlation can also be obstructed by considering the maneuver required for the association of the observations with the object under consideration as too unlikely. Overly pessimistic pdfs on the other hand can lead to the opposite effects, such that maneuvers are not detected because they are still within the realm of the dynamics or incorrect correlations are still considered sufficiently likely. Large initial uncertainties can also lead to pdf overlap between closely spaced RSOs, thereby obstructing data association in that no clear decision is possible based on single measurements. Further observations and multi-measurement ("multiple frame") algorithms are required in these cases.

SENSOR MANAGEMENT

The generation of the required observations (sensor tasking) and the distribution of these tasks to the available sensors (sensor scheduling) is collectively referred to as sensor management. It is treated as an optimization problem under uncertainty with the goal of obtaining as much information as possible about the space environment. Algorithms applied operationally and modern solutions that have shown promising improvements on a smaller scale are collected in figure 1.3.

Sensor task generation needs to make sure that fundamental constraints, such as sensor operability, line of sight between the sensor and the object and sufficient sensor performance (tracking capability) are fulfilled. For sensor tasking, uncertainty realism is key to finding an optimum between the average catalog accuracy and the number of cataloged objects.

Overly optimistic pdfs will result in less observation requests to be generated, thereby risking objects to "get lost" and result in future uncorrelated tracks (UCTs). Similarly, overly pessimistic pdfs result in an overall degradation of the catalog accuracy, since either more observation requests are generated than can be fulfilled by the available sensors or too much sensor time is spent to track objects with pessimistic

uncertainty information, thereby leading to less frequent updates of the other objects in the catalog. Further details on the coupling between nonlinear estimation and the performance of sensor tasking can be found in Williams (2012). The tasking algorithms available at CSpOC are outlined in Miller (2004, 2007). A possible future genetic algorithm solution has been compared to CSpOC's SP Tasker solution in Greve et al. (2018). If multiple sensors are available to execute an observation, sensor scheduling should maximize the overall information about the object via intelligent distribution of the sensor tasks.

Consider figure 1.8 as an example. Both available angles-only sensors, S1 and S2, are of similar accuracy, meaning that they result in a comparable reduction of the uncertainty volume. However, due to its head-on location, sensor S1 lacks observability of the dominant in-track component of the uncertainty volume, whereas sensor S2 has observability of this direction and is therefore able to significantly reduce the maximum encountered uncertainty, which is a desirable outcome since it minimizes the aspect ratio of the covariance. Uncertainty realism hence also aids sensor scheduling and therefore the quality of the subsequent OD.

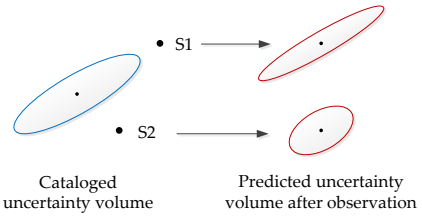


Figure 1.8: Impact of covariance shape on sensor scheduling. Adapted from Zhao et al. (2002, figure 3)

COLLISION PROBABILITY

Last but not least, the sensitivity of the collision probability to changes in the pdf is of great interest to CARA.

Frequentists vs. Bayesians or collision probability = collision risk ?

Before going into the details, a review of the centuries old frequentists vs. Bayesians debate from the perspective of RSO collision risk is in order. Both schools are concerned with the definition and interpretation of probabilities. Frequentists only assign probabilities to repeatable events and consider them as limiting case of long-term frequencies. Bayesians on the other hand see probabilities as a more general tool which also covers degrees of certainty. Recommended references on the discussion include: Barnett (1999) and Vallverdú (2015). A quick introduction is found in VanderPlas (2014).

Frequentists only consider probabilities as meaningful to describe inherent randomness, i.e. aleatoric uncertainty. For many years the differences between both methodologies have been restricted to philosophical aspects or the justification of (unavailable) prior probability functions (Balch et al., 2019), however it just happens to be the topic of collision risk mitigation, which pours new oil into the fire. Orthodox frequentists would refrain from assigning collision probabilities to close encounters, since they see neither randomness nor repeatability in them. In fact, also less stringent frequentists would raise concerns

about collision probabilities as the main risk assessment metric, since they consider the contribution of aleatoric errors to be below that of systematic errors[16].

While it cannot be determined with certainty if encounters in space are deterministic or not (there is no randomness in purely deterministic processes), it is unquestionable that essentially the trajectory estimates suffer from uncertainties and not, or to a much lesser extent, the true trajectories. For Bayesians it makes no difference whether the uncertainty is attributed to the event itself, or if other circumstances introduce it into the system. For them, the definition of a collision probability is correct and natural. This is also the predominant position within the space community, which over the last two decades has adopted the use of collision probabilities over miss distances as the primary risk metric.

A critical work which examines the topic of collision probabilities from a frequentists' perspective is Balch et al. (2019). In practice, neither school can avoid using ideas of the other to construct a comprehensive sequential estimation theory. This dualism is particularly evident in case of OPMs, which are using Bayesian inference to construct a sequential filtering algorithm, but only assign probability densities to purely aleatoric uncertainties, whereas the overall uncertainty is described using the more general outer probability measures.

Returning to the problem of collision *probability* sensitivity, fundamental relationships can be obtained by computing the collision probability of short-term encounters within parameter limits that represent operationally relevant bounds. Assuming a Gaussian pdf and using Chan's analytic equation for the collision probability (Chan, 2008, chapter 5), a miss distance x_m of 1 km, symmetric standard deviations of 100 m to 100 km and combined hard body radii between 10 cm (debris) and 50 m (space station), figure 1.9 can be obtained. The combined hard body radius (HBR) determines the domain of integration, such that the collision probabilities increase with greater objects.

Unlikely vs. uncertain close encounters

To avoid drawing false conclusions from the collision probability risk metric, it is prudent to distinguish between *unlikely* and *uncertain* encounters, where unlikely expresses confidence, i.e. that given the available knowledge about the system a collision can be excluded with a high degree of certainty. Uncertain collision probabilities on the other hand simply do not allow to infer if the risk is high or not[17]. They do not allow decision-making with confidence. Unlikely collision probabilities are therefore desirable, whereas uncertain ones are not.

16 The proportion of aleatoric and epistemic uncertainties within the kinematic state vector uncertainty is unknown and varies from conjunction to conjunction

17 A low P_c results, since for probabilistic risk metrics the probability is always low, except if evidence has been collected which allows to infer that it is high

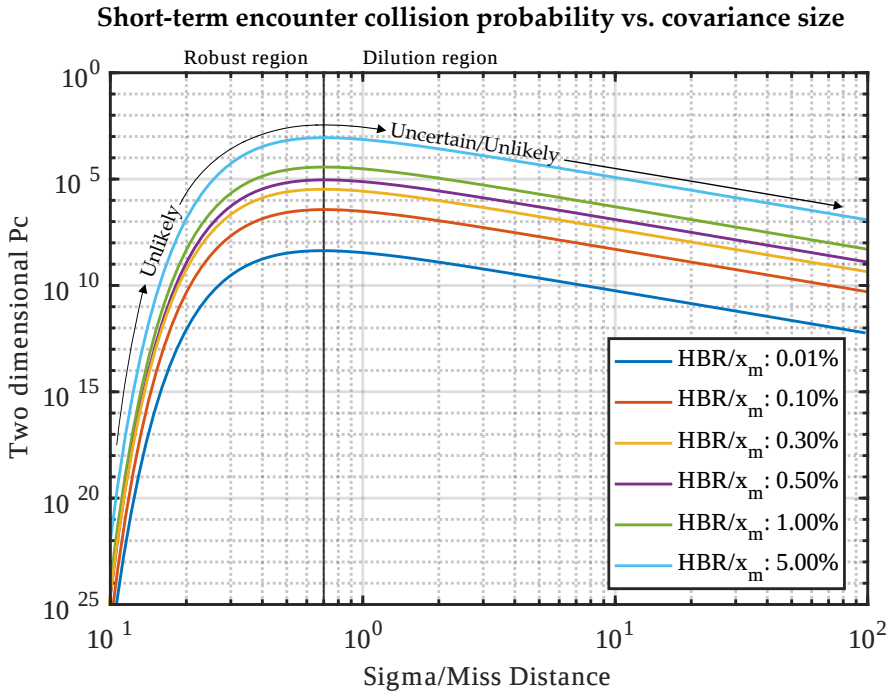


Figure 1.9: Log-log plot of collision probability sensitivity with respect to covariance size

Two domains, known respectively as the as the *robust* and *dilution* region, can be identified in figure 1.9. If the miss distance is much larger than the standard deviation (robust region), the collision probability is low, because the nominal point of closest approach lies in the tails of the combined uncertainty pdf. Low P_c computations in the robust region are therefore unlikely. However, collision probabilities in this domain are strongly affected by the size of the covariance ellipsoid, such that small changes in its length can result in multiple orders of magnitude (OOM) increases or decreases of the risk metric. Covariance realism is therefore of extreme importance for decision-making in the robust region. The maximum P_c separates the robust and dilution regions. It is attained when Sigma/Miss Distance $= 1/\sqrt{2}$, i.e. when the standard deviation approximately equals the miss distance.

The dilution region has first been analyzed by Alfano (2003) and follows the domain of near-maximum collision probabilities when further increasing the covariance size. It is characterized by declining collision probabilities, which can generate a false sense of confidence if low P_c results are blanketly considered unlikely. The dilution region has been discussed in various research papers, however recently it has become the central point of a scientific discourse that not only questions the current operational practice of probabilistic uncertainty modeling, but also reopens the Bayesians vs. frequentists debate.

From a purely mathematical perspective, the presence of a dilution region is a logical consequence of a probabilistic risk measure. Since

probabilities must integrate to unity over the state space, large covariances result in less probability density within the state space region that bounds the close encounter. Therefore, collision probabilities have to decrease after reaching a maximum.

For frequentists this is an absurd property of the collision probability risk metric, since the introduction of greater uncertainties into a system cannot reduce the true collision risk[18]. Strict frequentists therefore consider the sole existence of the dilution region as an argument for their belief that it makes no sense to associate a probability with non-repeatable events. Dilution region encounters can be anything from uncertain (no data is available to infer a high risk) to unlikely (the major part of the uncertainty is caused by irreducible aleatoric uncertainties, cf. note 18).

Real encounters typically follow figure 1.9 from the right to the left. When encounters are first discovered, the screening duration is in the order of multiple days up to a week, which results in diluted collision probabilities due to extended covariance propagation. With advancing time, the covariance propagation duration is reduced and further observations are acquired, which results in the encounters to progress towards the maximum collision probability. Most encounters then evolve in such a way that the collision probability decreases rapidly, as a consequence of the conjunction entering the robust region. However, in 40-45% of the cases requiring mitigation, the encounters are still within the dilution region when a maneuvering decision is to be made (Hejduk et al., 2019). In practice, dilution region encounters are treated no differently from those ranging in the robust region. The residual risk is simply accepted as part of the satellite operations background risk.

The debate about the "correct" operational treatment of dilution region encounters is highly charged, as can be realized when comparing Hejduk et al. (2019) and Balch et al. (2019) who arrive at fundamentally different conclusions and comment on each other's points of view. While the former authors reason for the current operational practice, the latter support the position of frequentists. Both works provide convincing arguments for their positions and recognize that the results of collision probability calculations are linked to the quality of the observations. The implications of future sensor developments and launch frequencies are however not fully elaborated.

18 Actually, this reasoning of frequentists is somewhat short-sighted. While it is certainly true that the recalculation of a collision probability based on poorer sensors has no influence on the true collision risk, an increase of the covariances can also be caused by aleatoric environmental uncertainties (e.g. certain types of atmospheric density uncertainty), which are not reducible and thus can lead to dilution region conjunctions even in the case of highly accurate sensors and nearly perfect force models. Under these circumstances it is correct to infer that conjunctions in the dilution region are unlikely and not uncertain, since then it is the irreducible natural randomness of the trajectories which makes the evolution of the relative position unlikely to result in a collision

In the long term, there will be a growing number of sophisticated sensors capable of tracking also cm-level objects with small covariances. This will minimize the accepted background risk criticized in Balch et al. (2019) and also ensure that many conjunctions are going to fall within the robust region, where the use of collision probabilities is far less disputed. On the other hand, current megaconstellation-driven launch rates are expected to put many tens of thousands of satellites into orbit over the course of the next few years. In the absence of binding political laws, it is therefore conceivable that the background risk, which currently appears to be acceptable, will increase swiftly. It is hence debatable whether the reasoning applied in Hejduk et al. (2019) can also be sustained under these circumstances.

The waves of collision risk assessment could be smoothed in the long term through the application of more generalized uncertainty models like OPMs. Based on the proportion of epistemic and aleatoric uncertainties, OPMs allow for probability dilution as a consequence of dominating irreducible aleatoric uncertainties, or for the probability bounds to increase as a consequence of dominating epistemic uncertainties. Although larger probability bounds do not imply that these conjunctions should operationally be treated differently than in case of diluted collision probabilities, they are likely easier to interpret for space analysts. Examples of these concepts can be found in (Delande et al., 2019).

At present, the operational success of CARA justifies risk assessment on the basis of collision probabilities, including the treatment of dilution region encounters. This however does not imply that the current state of the art is sustainable in the long term. Further research is necessary to quantify the impact of aleatoric and epistemic uncertainties and thereby improve operational risk assessment.

Collision probability sensitivity with respect to covariance orientation

Following the discussion about the criticism regarding collision probabilities and their sensitivity to the covariance size, also the sensitivity with respect to covariance orientation is required to be analyzed. Using the same ratios of the hard body radii versus the miss distance as in figure 1.9 (except for the 0.3% case which has been removed for better readability), the collision probability can be studied by either rotating the covariance and holding the miss vector constant or vice versa (fixing the covariance and rotating the miss vector). Different ratios of the semi-major to semi-minor axis are required to demonstrate that the sensitivity with respect to covariance orientation depends on covariance eccentricity. The results obtained using Chan's analytical short-term encounter collision probability equation are compiled in figure 1.10 for a covariance semi-major axis standard-deviation of 1 km and varying semi-minor axis standard deviations. The curves demonstrate that there is negligible effect of the collision probability on the orientation until an eccentricity of approximately 0.8. Very pronounced effects on the other hand come to light for eccentricities

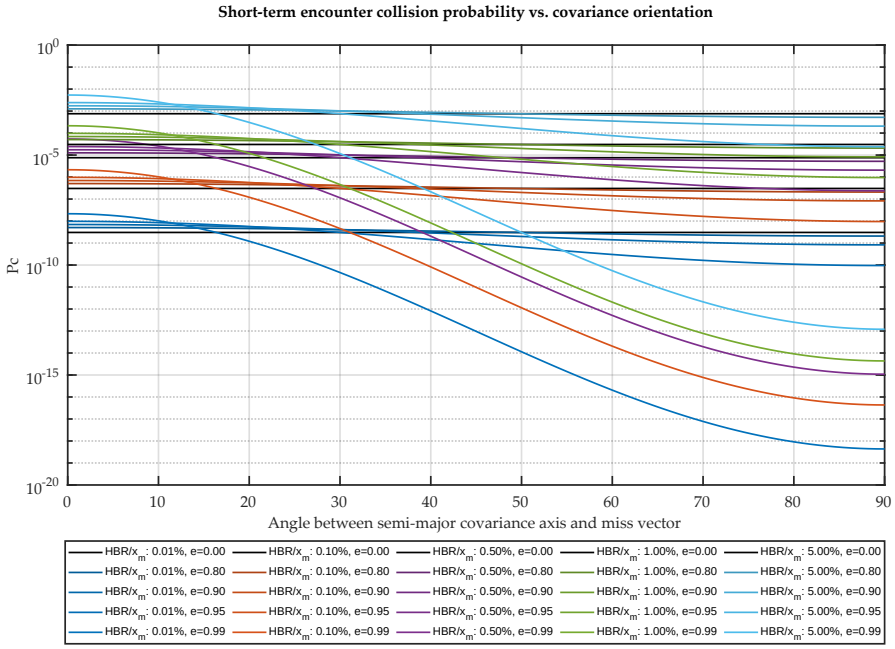


Figure 1.10: Collision probability sensitivity regarding covariance orientation

larger than 0.95. For such elongated ellipses small orientation changes result in multiple OOM difference. Orientation covariance realism is therefore important once the in-track component of the covariance significantly dominates the overall covariance shape. To suppress this sensitivity, it is therefore important to keep the covariance aspect ratios reasonably small, which can be achieved via advanced sensor management capable of optimum sensor selection (cf. figure 1.8) and propagation duration minimization. Smaller aspect ratios are also beneficial for algorithms that compute the collision probability via numerical integration[19].

Another interesting property is the sensitivity of Chan's maximum collision probability, which depends only on the covariance orientation and HBR/x_m. Figure 1.11 shows that the maximum collision probability has its largest sensitivity for a small neighborhood around the alignment of the semi-major covariance axis with the miss vector. The overall orientation related sensitivity is about one to two OOM.

Finally, also the sensitivity with respect to the shape of the pdf, i.e. covariance realism vs. uncertainty realism is of significant interest, as any pdf which is initially Gaussian will eventually become non-Gaussian during extended propagation under nonlinear dynamics. An excellent study on this topic, including multiple simulation results and figures, is Ghrist and Plakalovic (2012). Their finding is that non-Gaussian error volumes have negligible impact on the collision

19 Chan's analytical formulation is not reliable in case of very large aspect ratios, cf. section 6.1.2, such that alternative algorithms based on numerical integration are used in practice

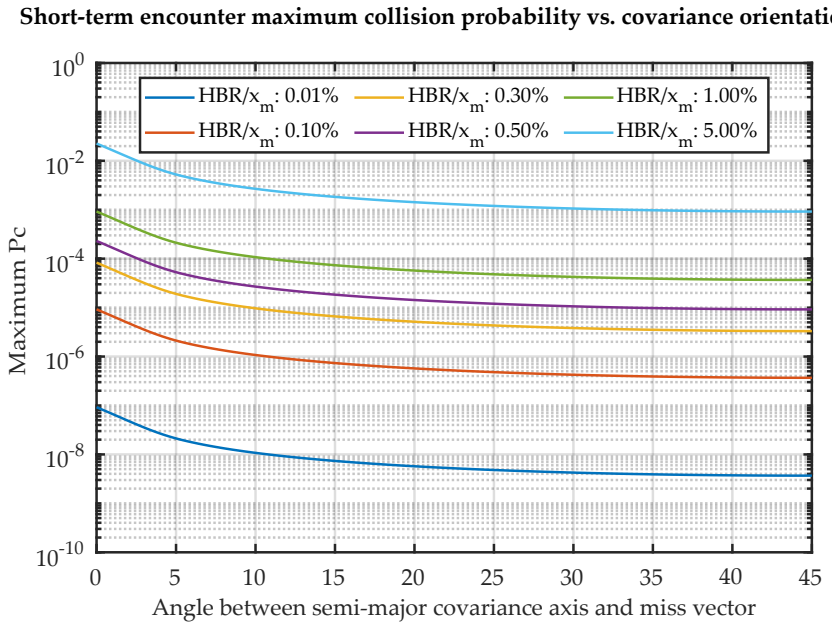


Figure 1.11: Sensitivity of Chan's maximum collision probability regarding covariance orientation

probability (less than 15%, i.e. significantly less than one OOM) in case of high interest events (HIEs), which are defined by collision probabilities larger than 1 in 10 000. These findings can be explained by the overlap of the probability distributions in the area of greatest density (cf. figure 1.12), thereby resulting in similar collision probabilities despite the disagreement in the overall pdf shape (Poore et al., 2016; Hejduk, 2017). The study also notes that non-HIEs were out of scope, such that for lower collision probabilities "a possibility exists that a conjunction event that is considered to have a low collision risk might have a significantly higher P_c than expected because of non-Gaussian deviations in the error volume" (Ghrist and Plakalovic, 2012).

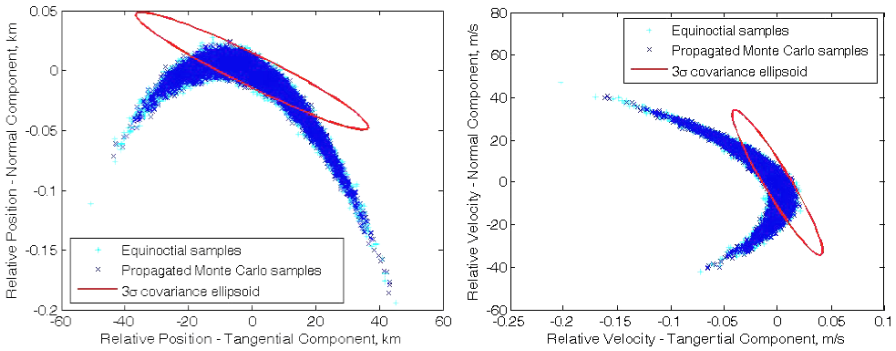


Figure 1.12: Example case of violated uncertainty realism from Ghrist and Plakalovic (2012), demonstrating the difference between the true probability density (banana/crescent-shaped) and the predicted covariance under the assumption of Gaussianity (red ellipse)

1.4 SCIENTIFIC OBJECTIVES

The accuracy of the space catalog determines the success of CARA and therefore also the contribution to spaceflight safety. This link was elaborated thoroughly in section 1.3. The topic of "covariance realism", or the more general "uncertainty realism", has hence gained considerable momentum in recent years and became the keynote for the thesis at hand.

1.4.1 *Scope*

Space is a data-starved environment. Due to the limited number of space surveillance sensors and the vast population of resident space objects (RSOs), propagation times of multiple days are not uncommon for cataloged objects. Although further work on measurement uncertainty quantification was found to be beneficial for increasing overall uncertainty realism, advances in this field have their strongest contribution for short tracks with few measurements and to a lesser extent for long-term propagation phases after measurement updates with well-conditioned tracks. A likely substantial improvement for uncertainty realism is hence to tackle the uncertainty quantification during extended propagation phases. This is of special importance for all LEO satellites, as the near Earth environment is the altitude shell with the most complex dynamics, greatest force model uncertainties and largest number of close conjunctions due to the dense population of RSOs.

As outlined in section 1.3.3, atmospheric density uncertainty and gravity uncertainty are the dominant drivers of force model uncertainty in LEO altitudes. A significant improvement in uncertainty realism is therefore expected from the development of a density and gravity error uncertainty quantification framework that is based on the same physical models used for orbit propagation and compatible with classical and modern orbit estimation methods. The derivation, validation and evaluation of this framework hence became the scope of this doctoral thesis. The developments are carried out based on the uncertainty classification introduced in section 1.3.1 and the fundamental stochastic principles outlined in section 1.3.2. Numerical Monte-Carlo simulations serve as state of the art validation technique. The operational impact of the research contributions is evaluated using an extensive study based on real world catalog data.

Potential users and beneficiaries of the developments are space industry companies developing SST software, satellite owners/operators, as well as space monitoring centers such as CSpOC or the German space situational awareness center ("Weltraumlagezentrum"). The developments enable a significant increase in confidence for probabilistic LEO uncertainty volumes, which in turn entails the positive effects of increased uncertainty realism discussed in section 1.3.4.

Concrete examples are calculations of collision probabilities with improved reliability, such that the rate of unnecessary maneuvers can be lowered. This in turn not only contributes to space security, but also allows to save fuel and reduce mission interruptions. Furthermore, the uncertainty quantification framework also enhances the accuracy of the cataloged orbits, since covariance matrices with increased realism improve the weighting process of the propagation and measurement uncertainties during orbit determination, which in turn decreases the risks of divergence and filter smugness. Since SST is conducted in a cyclic loop, the improved OD further results in benefits for sensor tasking, measurement correlation and maneuver detection/identification.

1.4.2 *State of the art*

Uncertainty growth during orbit propagation is dominated by two factors: the initial semi-major axis (radial) uncertainty at the start of a propagation phase and the uncertainties in the dynamics errors (Schiemenz et al., 2020d). While in a probabilistic[20] uncertainty model the first source is naturally modeled via covariance or pdf propagation, it has been customary to neglect force model uncertainties for many years (Horwood et al., 2011, p. 1841), resulting in degraded uncertainty realism. This has been noted especially in the context of CARA, where it is a known fact that the covariance information of the secondary objects ("chasers") which come close to the protected spacecraft ("targets") is overly optimistic, i.e. too small (Laporte, 2014; Poore et al., 2016, section 3.1.3).

Different approaches have been established by spacecraft owners and operators as workarounds. A simple method is using a single scaling parameter for the combined (target plus chaser) covariance, until it passes a χ^2 goodness of fit (GOF) test (Newman et al., 2016; Alfano and Oltrogge, 2018). CNES however decided to base CARA on a maximum collision probability obtained by individually scaling the target and chaser covariances within empirical bounds (Laporte, 2014). CSpOC's Astrodynamics Support Workstation (ASW) on the other hand uses the differential correction weighted root mean square (WRMS) error to scale the covariance information prior propagation, if the WRMS is greater than unity (Poore et al., 2016, section 3.1.3). Other references promote the use of individual scaling factors for each covariance axis (Hejduk et al., 2013) or correction matrices instead of scale factors (Cerven, 2011, 2013). A further approach that has been widely employed are ballistic and solar radiation pressure coefficient solve-for parameters. In addition to the WRMS-scaling and the usage of solve-for parameters, CSpOC also includes a consider parameter for

20 A probabilistic uncertainty model is the current state of the art in SST, however also more general uncertainty descriptions, such as OPMs (cf. section 1.3.1 for further details and references), are emerging

atmospheric density, which is implemented as an additive adjustment to the ballistic coefficient covariance (Poore et al., 2016).

While covariance tuning/scaling can mitigate the problem of overly optimistic covariances, it is not an optimal solution, since it neither tackles the underlying cause (the neglect of force model uncertainties) nor is there any statistical foundation in the number of scaling parameters and their applied bounds. Similarly, the consider parameter solution applied at CSpOC suffers from structural uncertainty, since its design as an additive adjustment of the ballistic coefficient covariance cannot reproduce the correct accumulation of atmospheric density uncertainty (Emmert et al., 2017).

The treatment of force model uncertainty quantification has received little attention in the literature prior to the efforts carried out in the context of this thesis. In recent years, the derivation of orbital gravity covariance estimation has solely been discussed in Wright et al. (2008a,b,c). Likewise, only two studies make use of physical relationships to propagate density errors caused by solar flux input uncertainty to satellite orbits (Emmert et al., 2014; Emmert et al., 2017).

Orbital variability due to density errors, however, has been analyzed at various levels, mostly at grid and subgrid spatial scales. Different strategies are used for these studies, but with a few exceptions almost all papers can be grouped into three categories. The first group is based on the extension of the state vector by one or more density parameters, which are then estimated together with the kinematic state (position and velocity vectors), as well as optionally also the ballistic and solar radiation pressure coefficients. Representatives of this approach are Rauch (1965), Wilkins and Alfriend (2000), as well as Gondelach and Linares (2020), of which the latter is to be highlighted for its additional assessment regarding the impact on collision probabilities. While the expansion of the state vector can often lead to promising results, this technique is subject to the concerns raised in section 1.3.3 (observability, model quality, error aliasing, matrix conditioning) and usually lacks a physical foundation. The realism of the orbit and density uncertainty estimates thus depends heavily on the model adopted for the density parameter propagation.

The second group of studies analyzes the effect of density variability predominantly by means of numerical orbital propagations. Popular references are Forbes (1972), Anderson et al. (2009), Leonard et al. (2012), Anderson et al. (2013), Vallado and Finkleman (2014), Sagnières and Sharf (2017) and Bussy-Virat et al. (2018). Although this technique allows the identification of important relationships, it cannot be used to establish a real-time capable uncertainty quantification framework, since a significant amount of computing time is required to calculate the results, which are always tied to the simulated scenario, such that a different scenario requires an individual, time-consuming analysis.

The third category is linked to the first in that the state vector is also extended by density parameters. However, the emphasis in this group lies on obtaining improved density estimates by using so-called "calibration satellites", which often have special properties, such as well known ballistic and solar radiation pressure coefficients or a spherical design. High precision observations of these objects can then be used to improve key parameters of the semi-empirical density models. These "nowcasted" model parametrizations can then also be used for all other RSOs of interest, leading to significantly reduced orbit prediction errors by means of classical orbit determination methods. This technique, on the other hand, is always tied to a specific semi-empirical density model and only available to a very small community with access to the calibration satellite data. Some researchers have therefore also attempted to apply the atmospheric calibration approach to the publicly accessible TLE catalog, albeit with limited practical acceptance. Examples that fall into this third category are: Marcos (1998), Nazarenko et al. (1998), Doornbos et al. (2008), Hinks and Psiaki (2010), as well as Storz et al. (2005), which is particularly well known for its long-term successful application at CSpOC.

A recent study also analyzes the effect of neutral density estimation errors on satellite conjunction serious event rates using CSpOC's JBH09/WRMS/consider parameter approach (Hejduk and Snow, 2018). The paper is of special interest, since information on the CSpOC system performance is rarely made publicly available.

1.4.3 *Outline and research contributions*

Following the literary and operational state of the art analysis (see also section 1.3, figure 1.3 and Schiemenz et al., 2019a), the objective of this doctoral thesis became the quantification of dominant LEO force model uncertainties using process noise matrices that derive from the physics of the astrodynamic models used for orbit propagation.

This approach requires no expansion of the state space, has a direct physical relationship with respect to the underlying uncertainty drivers, is interleaved with the models used for the orbit propagation, compatible with all uncertainty descriptions capable of including pdfs for modeling randomness and supports real-time computations.

A particular goal is the development of an uncertainty quantification framework for atmospheric density, compatible with classical and modern OD algorithms, which comprises the following research tasks:

- Quantification of relative density uncertainty due to solar flux input uncertainty
- Quantification of relative density uncertainty due to magnetic index input uncertainty
- Extension of the work presented in Emmert et al. (2017) concerning the cross-correlation coefficients of the covariance elements

- Quantification of structural grid-scale density model uncertainty
- Formulation of a holistic density uncertainty covariance matrix
- Analysis of force model uncertainty propagation alongside traditional covariance propagation techniques

The results of these topics are the content of chapter 2 and led to the publication of the following papers: Schiemenz et al. (2019b), Schiemenz et al. (2019c), Schiemenz et al. (2020a) and Schiemenz et al. (2020b). A secondary task, motivated by the common practice of early potential field truncation, is the derivation of improved gravity uncertainty covariance matrices based on Wright et al. (2008a). The findings are reported in chapter 3 and have led to the research article Schiemenz et al. (2020c).

Following the dominant force model uncertainty quantification (UQ), the incorporation of the resulting covariance matrices into classical and modern orbit determination (OD) algorithms is analyzed. This includes the extension of the classical WLS and SBWLS algorithms to incorporate process noise as part of chapter 4 and Schiemenz et al. (2019c), as well as an experimental application of the force model uncertainty framework towards the General Perturbations (GP) theory.

The Gaussian uncertainty assumption is relaxed in chapter 5, which addresses the development of a novel Gaussian mixture filter capable of achieving uncertainty realism by considering the nonlinear orbital dynamics, as well as force model uncertainty during the propagation phases. The corresponding paper is Schiemenz et al. (2020d).

Another research contribution of this doctoral research project is a study on the impact of covariance propagation including force model uncertainty regarding the risk assessment of close encounters identified in real world space object catalogs. The work package comprises the development of a complete conjunction assessment and risk analysis (CARA) software bundle, including the force model uncertainty quantification framework, a conjunction screener, the implementation of state of the art collision probability algorithms and a conjunction risk analysis tool. The developments and the impact assessment are presented in chapter 6. Further contributions that originate from this work package are an extension of Alfano's adjoining tube long-term encounter collision probability algorithm that does not require covariance symmetrization, as well as an algorithm for GP-based Monte-Carlo computations of the collision probability.

Topics that are not addressed but represent interesting candidates for future research are collected in chapter 7. The thesis ends with concluding remarks in chapter 8.

Part II

QUANTIFICATION OF DOMINANT FORCE MODEL UNCERTAINTIES IN LOW EARTH ORBITS

2

ATMOSPHERIC DENSITY UNCERTAINTY

Corresponding publications: Schiemenz et al. (2019b,c, 2020a,b)

Atmospheric drag is the main contributor to orbital uncertainty in low Earth orbits (Emmert et al., 2014). Knowledge of the acceleration due to drag is limited mainly by two factors: knowledge of the ballistic coefficient and knowledge of atmospheric density. Both parameters vary with time, however the uncertainty in the atmospheric density is generally greater than the uncertainty in the ballistic coefficient (Hejduk, 2017).

When performing orbit estimation, the atmospheric density estimate is obtained from semi-empirical atmospheric models, such as NRLMSISE-00 (Picone et al., 2002), JB2008 (Bowman et al., 2008), DTM-2012 (Bruinsma, 2013) or DTM-2013 (Bruinsma, 2015). If data of calibration satellites is available, also atmospheric calibration techniques, such as HASDM can be used to debias the underlying atmospheric model[1], which removes its static nature and leads to a significant improvement in the accuracy of the density estimate (Storz et al., 2005).

Density uncertainty arises at spatial scales below and within the model algorithm resolution, which is in the order of ~4000 km for NRLMSISE-00, DTM-2012 and DTM-2013. Submodel-scale spatial density uncertainty ("subgrid-scale density uncertainty") has been analyzed for example by Bruinsma and Forbes (2008) and Anderson et al. (2009). Grid-scale density uncertainty is also known as average model uncertainty. Average model accuracy is commonly discussed in the respective model reports (e.g. Picone et al., 2002 or Bruinsma, 2015). The corresponding grid-scale density uncertainty quantification is the topic of section 2.6 and Schiemenz et al. (2020b).

Despite model-uncertainty at grid and subgrid-scales, also errors in the inputs to the atmospheric models cause an additional input-uncertainty component. Prior to the publications associated with this thesis, this type of uncertainty has only received little treatment in the literature. Semi-empirical models typically make use of a spherical harmonics function to model periodic and non-periodic variations in the species number densities, exospheric temperature and, in the case of NRLMSISE-00, also in the baseline altitude temperature (Bruinsma, 2015). This function is commonly denoted $G(L)$, where L is the vector of the environmental parameters which are supplied to the model and contain the required information about latitude, longitude, local solar time, solar activity and geomagnetic activity. For further information on this model-specific function, see for example Bruinsma (2015).

1 Currently HASDM is only deployed against JB2008

Among the environmental parameters that are required to compute a density estimate with a semi-empirical atmospheric model, the solar flux and geomagnetic activity information are worthy of an input-sensitivity analysis. Any uncertainty in these parameters will propagate to an uncertainty in atmospheric density, which subsequently propagates to an uncertainty in the orbital mean motion and mean anomaly. Atmospheric density input uncertainty quantification is based on the fundamental physical relationship that models the evolution of the gas density with respect to altitude. This relationship is derived in section 2.1 and extends the presentation in Schiemenz et al. (2019b).

Subsequently, section 2.2 outlines how to use the equation governing the species number density to derive an expression that relates absolute input errors to relative errors in atmospheric density. By making use of the function $G(L)$, section 2.2 presents a rigorous derivation of the covariance propagation that relates Gaussian input errors ($\delta G(L)$) to Gaussian relative atmospheric mass density errors (ϵ_ρ).

The sequence in which absolute errors in the solar flux or geomagnetic index proxies propagate to relative density errors and its uncertainty is summarized in figure 2.1. Essentially, any input error propagates into a corresponding error in the function of the spherical harmonic variations. This error is different for every gas constituent and influences the species number density and hence also the overall gas number density and the gas mean mass. The errors and uncertainties in these quantities then determine the overall error/uncertainty propagation to the atmospheric mass density.

Figure 2.1: Propagation of absolute input errors to relative density errors

After the generic derivation of relative density errors and their uncertainties in section 2.2, the subsequent sections 2.3 and 2.4 are devoted to the explicit cases of solar flux input uncertainty and geomagnetic index uncertainty. Each of these sections starts with an overview of origins responsible for the respective type of input uncertainty, followed by a discussion on aleatoric uncertainty models and their practical application. After the derivations, the sections end with Monte-Carlo simulation results used for validation of the developments.

2.1 SEMI-EMPIRICAL DERIVATION OF ATMOSPHERIC DENSITY

This section is devoted to the derivation of the fundamental equation that allows to obtain the relative number density of a gas constituent (n_i), given the fundamental properties listed in section 1.3.3, i.e. the atmospheric temperature near the lower boundary of the

thermosphere ("baseline altitude", T_0), the temperature at the altitude of interest (T), the asymptotic exospheric temperature (T_{ex}) and the temperature/altitude gradient at the baseline altitude (T_0'). The derivation is valid for all models assuming hydrostatic balance, i.e. that the pressure gradient and gravitational forces cancel out (e.g. MSISE-90/NRLMSISE-00, DTM-2012 and DTM-2013, but not JB2008/JBH2009). As a starting point, consider the column of infinitesimal height dz shown in figure 2.2. Hydrostatic balance is expressed as:

$$\frac{dP}{dz} = -\rho g \tag{2.1}$$

where g is the local gravitational acceleration, z represents geometric height and P is atmospheric pressure. It makes sense to transform equation 2.1 into a form in which the gravitational acceleration does not decrease with height. This is achieved by transitioning to geopotential, which is defined by:

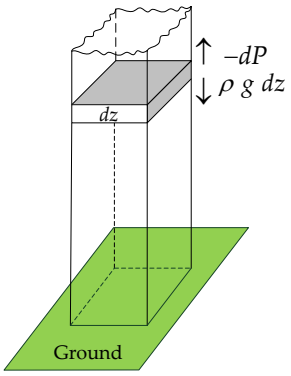


Figure 2.2: Hydrostatic balance: pressure gradient and gravity cancel

$$\phi(z) = \int_0^z g\, dh \Rightarrow d\phi = g\, dz \tag{2.2}$$

Substituting eq. 2.2 into equation 2.1 yields:

$$dP = -\rho\, d\phi \tag{2.3}$$

Geopotential height is the geopotential divided by a reference gravity. The geopotential distance above reference altitude z_0 is thus given by:

$$\zeta(z) = \frac{\phi(z) - \phi(z_0)}{g(z_0)} \Rightarrow\ d\phi = g_0\, d\zeta \tag{2.4}$$

Inserting equation 2.4 into eq. 2.3 results in the desired form of the hydrostatic balance equation:

$$dP = -\rho g_0\, d\zeta \tag{2.5}$$

A typical baseline altitude that is used by all models considered in this thesis is $z_0 = 120\,km$. At this height, $g_0 = 9.807\,\frac{m}{s^2} \cdot (6371/(6371 + 120))^2 \cong 9.45\,\frac{m}{s^2}$. Next, the ideal gas law allows to replace the dependency on pressure with one on temperature:

$$P = nkT = \rho\frac{kT}{M} \tag{2.6}$$

where $n = \frac{\rho}{M}$ is the mean particle density, M the mean particle mass, k the Boltzmann constant and T denotes temperature. Equation 2.6

represents an expression for P, whereas eq. 2.5 is written in terms of dP. The next step is hence to determine dP from equation 2.6. Also it should not be forgotten that the atmosphere consists of a mixture of different gases, each with different ρ, T and M. Consequently, atmospheric pressure varies with density, temperature and mean mass, where the latter is computed according to equation 2.7.

$$M = \frac{\sum_{i=1}^{n_{species}} n_i M_i}{\sum_{i=1}^{n_{species}} n_i} = \frac{\sum_{i=1}^{n_{species}} n_i M_i}{n} \tag{2.7}$$

To determine dP from P, it is necessary to apply the total differential, which using eq. 2.5, results in:

$$\begin{aligned} dP &= \frac{\partial P}{\partial n}\,dn + \frac{\partial P}{\partial T}\,dT = \frac{\partial P}{\partial \rho}\,d\rho + \frac{\partial P}{\partial T}\,dT + \frac{\partial P}{\partial M}\,dM \\ &= kT\,dn + nk\,dT = \frac{kT}{M}\,d\rho + \frac{\rho k}{M}\,dT - \frac{\rho kT}{M^2}\,dM = -\rho g_0\,d\zeta \end{aligned} \tag{2.8}$$

Division by P furthermore yields:

$$-\frac{g_0 M\,d\zeta}{kT} = \frac{dn}{n} + \frac{dT}{T} = \frac{d\rho}{\rho} - \frac{dM}{M} + \frac{dT}{T} \tag{2.9}$$

Realizing that $\frac{dx}{x} = d\int\frac{dx}{x} = d\ln(x)$, allows to write equation 2.9 either in terms of the average mass density ρ (eq. 2.10), or the average number density n (eq. 2.11).

$$d\ln(\rho) = -\frac{Mg_0}{kT}\,d\zeta - d\ln(\frac{T}{M}) \tag{2.10}$$

$$d\ln(n) = -\frac{Mg_0}{kT}\,d\zeta - d\ln(T) \tag{2.11}$$

Integrating formulae 2.10 and 2.11 with respect to geopotential height from ζ_0 to ζ yields:

$$\ln(\rho(\zeta)) = \ln(\rho(\zeta_0)) - \ln\frac{T(\zeta)}{T(\zeta_0)} + \ln\frac{M(\zeta)}{M(\zeta_0)} - \frac{g_0}{k}\int_{\zeta_0}^{\zeta}\frac{M}{T}\,d\zeta' \tag{2.12}$$

$$\ln(n(\zeta)) = \ln(n(\zeta_0)) - \ln\frac{T(\zeta)}{T(\zeta_0)} - \frac{g_0}{k}\int_{\zeta_0}^{\zeta}\frac{M}{T}\,d\zeta' \tag{2.13}$$

Equations 2.12 and 2.13 describe the profile of a fully mixed atmosphere. This is true up to approximately 100 km altitude. Between 100 km and 120 km the mixture of the gases transitions into an atmosphere that becomes dominated by molecular diffusion (Emmert, 2015). In the LEO-altitudes of interest (> 200 km) molecular diffusion

and gravity cause the constituent species to separate according to their mass and follow individual hydrostatic balance constraints. Therefore the concept of "diffusive equilibrium" is typically applied to equation 2.12 when working with the upper atmosphere.

Diffusive equilibrium is introduced into the mixed profile equations by adding a thermal diffusion coefficient and evaluating the equation of interest (mass density or number density) separately for each species. The species number density equation will be the baseline for the following considerations. Written per species, it holds that $M = M_i = \text{constant}$. Defining the thermal diffusion coefficient of species i as α_i and realizing that $\zeta_0 = \zeta(z_0) = 0$, equation 2.13 becomes:

$$\ln(n_i) = \ln(n_{i_0}) - \underbrace{(1+\alpha_i)}_{\text{thermal diffusion}} \ln\left(\frac{T(\zeta)}{T_0}\right) - \frac{g_0 M_i}{k} \int_0^\zeta \frac{1}{T(\zeta')}\, d\zeta' \tag{2.14}$$

Next, an expression for the temperature profile $T(\zeta)$ needs to be defined. JB2008 and JBH2009 make use of an arctangent profile. Its description may be found in Emmert (2015). The more popular choice, which represents the baseline of MSISE-90/NRLMSISE-00, DTM-2012 and DTM-2013 and has the advantage of being analytically integrable, is the Bates temperature profile, as introduced in Bates (1959):

$$T(\zeta) = T_{ex} - (T_{ex} - T_0) \exp\left(-\frac{T_0'}{T_{ex} - T_0}\zeta\right) \tag{2.15}$$

$$\Leftrightarrow T(\zeta) = T_{ex} - (T_{ex} - T_0) \exp\left(-\sigma\zeta\right) \tag{2.16}$$

where $\sigma = \frac{T_0'}{T_{ex} - T_0}$ can be considered as an inverse scale height and is also known as "shape factor" (Emmert, 2015). The quantity T_0' represents the temperature gradient at the baseline altitude (commonly in units of [K/km]).

To obtain a description of the number density vs. altitude, it is required to solve the integral of the inverse temperature profile with respect to geopotential height. For a Bates temperature profile this can be done analytically, as shown in the following.

Inserting the Bates temperature profile into the integral in equation 2.14 results in:

$$\int_0^\zeta \frac{1}{T(\zeta')}\, d\zeta' = \int_0^\zeta \frac{1}{T_{ex} - (T_{ex} - T_0)\exp(-\sigma\zeta')}\, d\zeta' \tag{2.17}$$

After substituting $u = -\sigma\zeta \Rightarrow d\zeta' = -\frac{1}{\sigma}\, du'$, the integral becomes:

$$\int_0^\zeta \frac{1}{T(\zeta')}\, d\zeta' = -\frac{1}{\sigma} \int_0^{-\sigma\zeta} \frac{1}{T_{ex} - (T_{ex} - T_0)\exp(u')}\, du' \tag{2.18}$$

Next, substituting $\exp(u) = v \Rightarrow \frac{dv}{du} = v$ yields:

$$\int_0^{\zeta} \frac{1}{T(\zeta')}\, d\zeta' = -\frac{1}{\sigma} \int_1^{\exp(-\sigma\zeta)} \frac{1}{(T_{ex} - (T_{ex} - T_0)v')v'}\, dv' \tag{2.19}$$

Now the integrand needs to be replaced with its partial fraction expansion:

$$\begin{aligned}\int_0^{\zeta} \frac{1}{T(\zeta')}\, d\zeta' &= -\frac{1}{\sigma} \int_1^{\exp(-\sigma\zeta)} \frac{1}{T_{ex}v'} + \frac{T_0 - T_{ex}}{T_{ex}v'(T_{ex} - T_0) - T_{ex}^2}\, dv' \\ &= -\frac{1}{T_{ex}\sigma} \int_1^{\exp(-\sigma\zeta)} \frac{1}{v'}\, dv' - \frac{T_{ex} - T_0}{\sigma T_{ex}} \int_1^{\exp(-\sigma\zeta)} \frac{1}{T_{ex} - v'(T_{ex} - T_0)}\, dv'\end{aligned} \tag{2.20}$$

Finally, substituting $w = T_{ex} - v(T_{ex} - T_0) \Rightarrow \frac{dw}{dv} = -(T_{ex} - T_0)$ allows to obtain:

$$\begin{aligned}\int_0^{\zeta} \frac{1}{T(\zeta')}\, d\zeta' &= -\frac{1}{T_{ex}\sigma}(-\sigma\zeta - \ln(1)) + \frac{T_{ex} - T_0}{\sigma T_{ex}(T_{ex} - T_0)} \int_{T_0}^{T_{ex} - (T_{ex} - T_0)\exp(-\sigma\zeta)} \frac{1}{w'}\, dw' \\ &= \frac{\zeta}{T_{ex}} + \frac{1}{\sigma T_{ex}} \int_{T_0}^{T(\zeta)} \frac{1}{w'}\, dw' = \frac{\zeta}{T_{ex}} + \frac{1}{\sigma T_{ex}}(\ln(T(\zeta) - \ln(T_0))\end{aligned} \tag{2.21}$$

Therefore, the solution to the integral of the inverse Bates temperature profile with respect to geopotential height is:

$$\int_0^{\zeta} \frac{1}{T(\zeta')}\, d\zeta' = \frac{\zeta}{T_{ex}} - \frac{\ln\left(\frac{T_0}{T(\zeta)}\right)}{\sigma T_{ex}} \tag{2.22}$$

Introducing equation 2.22 to eq. 2.14 then yields the desired description of $n_i(\zeta)$, as for example published in Chamberlain and Hunten (1990) or Picone et al. (2013):

$$n_i = n_{i_0}\left(\frac{T_0}{T(\zeta)}\right)^{1+\alpha_i+\gamma_i} \exp\left(-\frac{g_0 M_i}{k T_{ex}}\zeta\right) \tag{2.23}$$

$$\Leftrightarrow n_i = n_{i_0}\left(\frac{T_0}{T(\zeta)}\right)^{1+\alpha_i+\gamma_i} \exp(-\gamma_i\sigma\zeta) \tag{2.24}$$

where

$$\gamma_i = \frac{g_0 M_i}{k T_{ex}\sigma} \tag{2.25}$$

Equation 2.24 is valid for all models assuming hydrostatic equilibrium and a Bates temperature profile. In all other cases (e.g. JB2008), its usage to derive relative density error estimates introduces undesired model uncertainty, however is still more accurate than the assumption of a constant thermospheric temperature, as applied in Schiemenz et al. (2019b).

2.2 ESTIMATION OF ATMOSPHERIC DENSITY INPUT COVARIANCE

Equation 2.24 is the fundamental equation used by many semi-empirical models to compute density estimates. Model dynamics, i.e. horizontal variations and forcings, result in changes of n_{i_0}, T_0 and T_{ex}, which then lead to the corresponding species number density changes and overall density variations. In the following it is used to derive an equation for the relative exospheric density error and its variance by performing an input sensitivity analysis of atmospheric density with respect to n_{i_0}, T_{ex} and T_0.

2.2.1 *Relative exospheric mass density error*

In a mixture of gases, mass density is described as the product of the mean mass, as defined in equation 2.7, and the overall number density of the species contained in the gas:

$$\rho = nM \tag{2.26}$$

For the remainder of this work the term density is used to refer to mass density, if not noted otherwise. The relative exospheric density error is defined in equation 2.27 according to the error convention listed in the beginning of this thesis. Recall that a hat-sign is used to indicate estimated quantities, i.e. the output obtained from a semi-empirical model.

$$\epsilon_\rho = \frac{\rho - \hat{\rho}}{\hat{\rho}} = \frac{\delta\rho}{\hat{\rho}} = \frac{\rho}{\hat{\rho}} - 1 \tag{2.27}$$

Similarly, the relative errors in the number density and the mean mass can be defined according to eqs. 2.28 and 2.29.

$$\epsilon_n = \frac{n}{\hat{n}} - 1 = \frac{\rho/M}{\hat{\rho}/\hat{M}} - 1 = \frac{\hat{M}}{M}(\epsilon_\rho + 1) - 1 = \frac{\hat{M}}{M}\epsilon_\rho + \frac{\hat{M}}{M} - 1 \tag{2.28}$$

$$\epsilon_M = \frac{M}{\hat{M}} - 1 \Leftrightarrow \frac{\hat{M}}{M} = \frac{1}{\epsilon_M + 1} \tag{2.29}$$

Inserting equation 2.29 into eq. 2.28 yields:

$$\epsilon_n = \frac{\epsilon_\rho}{\epsilon_M + 1} + \frac{1}{\epsilon_M + 1} - 1 \tag{2.30}$$

$$\Leftrightarrow \epsilon_\rho = \epsilon_n + \epsilon_M + \epsilon_n\epsilon_M \tag{2.31}$$

Equation 2.31 allows to compute the relative density error based on the underlying relative errors in the mean mass and number density. Since the masses of the individual species considered in a semi-empirical density model are known, only equation 2.24, the fundamental description of the number density per species and its sensitivity with respect to the three quantities n_{i_0}, T_0 and T_{ex} is needed to evaluate equation 2.31. The six required derivatives are therefore:

- number density and mean mass with respect to exospheric temperature
- number density and mean mass with respect to the temperature at the baseline altitude
- number density and mean mass with respect to the number density at the baseline altitude

RELATIVE EXOSPHERIC NUMBER DENSITY ERROR

The relative number density error is given to first order by:

$$\epsilon_n \cong \frac{\sum_{i=1}^{n_{species}} \frac{\partial n_i}{\partial n_{i_0}} \delta n_{i_0}}{\hat{n}} + \frac{\sum_{i=1}^{n_{species}} \frac{\partial n_i}{\partial T_{ex}} \delta T_{ex}}{\hat{n}} + \frac{\sum_{i=1}^{n_{species}} \frac{\partial n_i}{\partial T_0} \delta T_0}{\hat{n}} \tag{2.32}$$

Equation 2.32 expresses the relative number density error as the sum of the individual error contributors. Each of the terms needs to be formulated to derive an expression for ϵ_n.

Sensitivity of number density with respect to number density at the baseline altitude

The MSIS and DTM-class models evaluate the number density at the baseline altitude according to equation 2.33:

$$n_0 = \sum_{i=1}^{n_{species}} n_{i_{0_0}} \exp(G_i(L)) \tag{2.33}$$

where i denotes the i^{th} species (helium, hydrogen, atomic oxygen, atomic nitrogen, molecular oxygen and molecular nitrogen for the DTM-models and additionally argon, as well as ionized oxygen for NRLMSISE-00) and $n_{i_{0_0}}$ the global and temporal average number density of the species at the baseline altitude, prior to modification via the exponential of the spherical harmonics function. $n_{i_{0_0}}$ is a model-dependent coefficient. An error in the environmental parameters results in an error in n_0 according to:

$$\begin{aligned} \delta n_{i_0} &= n_{i_{0_0}} \left(\exp(G_i(\hat{L} + \delta L)) - \exp(G_i(\hat{L})) \right) \\ &= \underbrace{n_{i_{0_0}} \exp(G_i(\hat{L}))}_{:= n_{i_0}} (\exp(\delta G_i) - 1) \end{aligned} \tag{2.34}$$

From equation 2.24 it follows that:

$$\frac{\partial n_i}{\partial n_{i_0}} = \frac{n_i}{n_{i_0}} \tag{2.35}$$

The absolute error is then given by the introduction of eq. 2.35 and summation over the species:

$$\delta n_0 = \sum_{i=1}^{n_{species}} n_i (\exp(\delta G_i) - 1) \tag{2.36}$$

Inserting equation 2.36 with the model-estimated number density into the first summand of eq. 2.32 results in equation 2.37, which describes the sensitivity of the number density at the altitude of interest with respect to the number density at the baseline altitude, divided by the absolute number density estimate.

$$\epsilon_{n_0} := \frac{\sum_{i=1}^{n_{\text{species}}} \hat{n}_i (\exp(\delta G_i) - 1)}{\hat{n}} = \frac{\sum_{i=1}^{n_{\text{species}}} \hat{n}_i \exp(\delta G_i)}{\hat{n}} - 1 \tag{2.37}$$

Sensitivity of number density with respect to exospheric temperature
The sensitivity of the number density with respect to exospheric temperature requires the computation of the derivative $\partial n_i / \partial T_{ex}$:

$$\frac{\partial}{\partial T_{ex}} n_i = \frac{\partial}{\partial T_{ex}} n_{i_0} \left(\frac{T_0}{T(\zeta)} \right)^{1+\alpha_i+\frac{g_0 M_i}{k T_{ex} \sigma}} \exp\left(-\frac{g_0 M_i}{k T_{ex}} \zeta \right) \tag{2.38}$$

which can be written as $\frac{d}{dx} a f(x)^{g(x)} \exp(c(x))$. The corresponding rule for the differentiation is derived in appendix A.2 and reads:

$$\begin{aligned} &\frac{d}{dx} a f(x)^{g(x)} \exp(c(x)) \\ &\quad = a f(x)^{g(x)} \exp(c(x)) \left[\frac{d}{dx} c(x) + \ln(f(x)) \frac{d}{dx} g(x) + g(x) \frac{\frac{d}{dx} f(x)}{f(x)} \right] \end{aligned} \tag{2.39}$$

Using

$$\begin{aligned} x &= T_{ex}\,, \\ a &= n_{i_0}\,, \\ f(T_{ex}) &= \frac{T_0}{T(\zeta)} = \frac{T_0}{T_{ex} - (T_{ex} - T_0) \exp\left(-\frac{T_0' \zeta}{T_{ex} - T_0} \right)}\,, \\ g(T_{ex}) &= 1 + \alpha_i + \frac{M_i g_0 (T_{ex} - T_0)}{k T_{ex} T_0'} = 1 + \alpha_i + \gamma_i\,, \end{aligned}$$

and

$$c(T_{ex}) = -\frac{M_i g_0 \zeta}{k T_{ex}}\,,$$

the derivative is computed as:

$$\begin{aligned} \frac{\partial n_i}{\partial T_{ex}} = &\underbrace{n_{i_0} \left(\frac{T_0}{T(\zeta)} \right)^{1+\alpha_i+\gamma_i} \exp\left(-\frac{M_i g_0 \zeta}{k T_{ex}} \right)}_{n_i} \\ &\left[\frac{M_i g_0 \zeta}{k T_{ex}^2} + \frac{M_i g_0 T_0}{k T_0' T_{ex}^2} \ln\left(\frac{T_0}{T(\zeta)} \right) - (1 + \alpha_i + \gamma_i) \frac{\frac{T_0 [1 - \exp(-\sigma\zeta)(1+\sigma\zeta)]}{T(\zeta)^2}}{\frac{T_0}{T(\zeta)}} \right] \\ = n_i &\left[\frac{M_i g_0}{k T_{ex}^2} \left(\zeta + \frac{T_0}{T_0'} \ln\left(\frac{T_0}{T(\zeta)} \right) \right) - \frac{1 + \alpha_i + \gamma_i}{T(\zeta)} (1 - \exp(-\sigma\zeta)[1 + \sigma\zeta]) \right] \end{aligned} \tag{2.40}$$

From the definition of the Bates temperature profile (cf. equation 2.16), it follows that:

$$\exp(-\sigma\zeta) = \frac{T_{ex} - T(\zeta)}{T_{ex} - T_0} \tag{2.41}$$

Hence, equation 2.40 can be written as:

$$\frac{\partial}{\partial T_{ex}} n_i = n_i \left[\frac{M_i g_0}{k T_{ex}^2} \left(\zeta + \frac{T_0}{T_0'} \ln\left(\frac{T_0}{T(\zeta)} \right) \right) - \frac{1 + \alpha_i + \gamma_i}{T(\zeta)} \left(1 - \frac{T_{ex} - T(\zeta)}{T_{ex} - T_0} (1 + \sigma\zeta) \right) \right] \tag{2.42}$$

Defining the second summand of equation 2.32 as $\epsilon_{N_{T_{ex}}}$, the contribution of an absolute exospheric temperature error to the overall relative number density error is given by:

$$\begin{aligned} \epsilon_{N_{T_{ex}}} &= \frac{\sum_{i=1}^{n_{species}} \frac{\partial n_i}{\partial T_{ex}} \delta T_{ex}}{\hat{n}} \\ &= \delta T_{ex} \frac{g_0}{k T_{ex}^2} \left(\zeta + \frac{T_0}{T_0'} \ln\left(\frac{T_0}{T(\zeta)} \right) \right) \frac{\sum_{i=1}^{n_{species}} n_i M_i}{\hat{n}} \\ &\quad - \frac{\delta T_{ex} \left(1 - \frac{T_{ex} - T(\zeta)}{T_{ex} - T_0} (1 + \sigma\zeta) \right)}{T(\zeta)} \frac{\sum_{i=1}^{n_{species}} n_i (1 + \alpha_i + \gamma_i)}{\hat{n}} \end{aligned} \tag{2.43}$$

Making use of the definition of mean mass (eq. 2.7) and analogously defining the (number density weighted) mean thermal diffusion coefficient as

$$\alpha = \frac{\sum_{i=1}^{n_{species}} n_i \alpha_i}{n}, \tag{2.44}$$

and the (number density weighted) mean ratio of temperature and species scale heights as

$$\gamma = \frac{\sum_{i=1}^{n_{species}} n_i \gamma_i}{n}, \tag{2.45}$$

yields the desired description of $\epsilon_{N_{T_{ex}}}$ upon evaluation with the estimated quantities:

$$\epsilon_{N_{T_{ex}}} = \left[\frac{g_0 \hat{M}}{k \hat{T}_{ex}^2} \left(\zeta + \frac{\hat{T}_0}{\hat{T}_0'} \ln\left(\frac{\hat{T}_0}{\hat{T}(\zeta)} \right) \right) - \frac{1 + \hat{\alpha} + \hat{\gamma}}{\hat{T}(\zeta)} \left(1 - \frac{\hat{T}_{ex} - \hat{T}(\zeta)}{\hat{T}_{ex} - \hat{T}_0} (1 + \hat{\sigma}\zeta) \right) \right] \delta T_{ex} \tag{2.46}$$

Sensitivity of number density with respect to baseline temperature

Similarly as for the exospheric temperature, equation 2.39 can be used to derive the sensitivity with respect to the baseline temperature. Using the definitions:

$$x = T_0,$$

$$a = n_{i_0},$$

$$f(T_0) = \frac{T_0}{T(\zeta)} = \frac{T_0}{T_{ex} - (T_{ex} - T_0)\exp\left(-\frac{T_0'\zeta}{T_{ex}-T_0}\right)},$$

$$g(T_0) = 1 + \alpha_i + \frac{M_i g_0 (T_{ex} - T_0)}{k T_{ex} T_0'} = 1 + \alpha_i + \gamma_i,$$

and

$$c(T_0) = 0,$$

the derivative of the number density profile with respect to the baseline temperature is given by:

$$\frac{\partial}{\partial T_0} n_i = \underbrace{n_{i_0}\left(\frac{T_0}{T(\zeta)}\right)^{1+\alpha_i+\gamma_i} \exp\left(-\frac{M_i g_0 \zeta}{k T_{ex}}\right)}_{n_i} \left[-\frac{M_i g_0}{k T_0' T_{ex}} \ln\left(\frac{T_0}{T(\zeta)}\right) + (1 + \alpha_i + \gamma_i)\frac{\frac{T(\zeta) - T_0 \exp(-\sigma\zeta)[1+\sigma\zeta]}{T(\zeta)^2}}{\frac{T_0}{T(\zeta)}}\right] \tag{2.47}$$

which can be simplified to:

$$\frac{\partial n_i}{\partial T_0} = n_i \left[-\frac{M_i g_0}{k T_0' T_{ex}} \ln\left(\frac{T_0}{T(\zeta)}\right) + \frac{1 + \alpha_i + \gamma_i}{T(\zeta)} \frac{T(\zeta) - T_0 \frac{T_{ex} - T(\zeta)}{T_{ex} - T_0}[1 + \sigma\zeta]}{T_0}\right] \tag{2.48}$$

With the help of equation 2.48 it is now possible to compute the third term of equation 2.32:

$$\begin{aligned} \epsilon_{N_{T_0}} &= \frac{\sum_{i=1}^{n_{species}} \frac{\partial n_i}{\partial T_0} \delta T_0}{\hat{n}} \\ &= \delta T_0 \frac{g_0}{k T_{ex} T_0'} \ln\left(\frac{T(\zeta)}{T_0}\right) \frac{\sum_{i=1}^{n_{species}} M_i n_i}{\hat{n}} \\ &\quad + \frac{\delta T_0}{T(\zeta)} \frac{T(\zeta) - T_0 \frac{T_{ex} - T(\zeta)}{T_{ex} - T_0}(1 + \sigma\zeta)}{T_0} \frac{\sum_{i=1}^{n_{species}} n_i (1 + \alpha_i + \gamma_i)}{\hat{n}} \end{aligned} \tag{2.49}$$

The evaluation of eq. 2.49 with estimated quantities from an atmospheric model and equations 2.7, as well as 2.44 and 2.45 with the estimated number density, results in:

$$\epsilon_{N_{T_0}} = \left[\frac{g_0 \hat{M}}{k \hat{T}_{ex} \hat{T}_0'} \ln\left(\frac{\hat{T}(\zeta)}{\hat{T}_0}\right) - \frac{1 + \hat{\alpha} + \hat{\gamma}}{\hat{T}(\zeta)}\left(\frac{\hat{T}_{ex} - \hat{T}(\zeta)}{\hat{T}_{ex} - \hat{T}_0}(1 + \hat{\sigma}\zeta) - \frac{\hat{T}(\zeta)}{\hat{T}_0}\right)\right] \delta T_0 \tag{2.50}$$

Inserting equations 2.37, 2.46 and 2.50 into eq. 2.32 yields a first-order estimate of the relative error in the total number density, caused by

an absolute error in the function of the environmental parameters ($\delta G_i(L)$):

$$\begin{aligned}\epsilon_n \cong & \left[\frac{g_0\hat{M}}{k\hat{T}_{ex}^2}\left(\zeta+\frac{\hat{T}_0}{\hat{T}_0'}\ln\left(\frac{\hat{T}_0}{\hat{T}(\zeta)}\right)\right)-\frac{1+\hat{\alpha}+\hat{\gamma}}{\hat{T}(\zeta)}\left(1-\frac{\hat{T}_{ex}-\hat{T}(\zeta)}{\hat{T}_{ex}-\hat{T}_0}(1+\hat{\sigma}\zeta)\right)\right] \\ & \delta T_{ex}(\delta G_{T_{ex}}(L)) \\ & +\left[\frac{g_0\hat{M}}{k\hat{T}_{ex}\hat{T}_0'}\ln\left(\frac{\hat{T}(\zeta)}{\hat{T}_0}\right)-\frac{1+\hat{\alpha}+\hat{\gamma}}{\hat{T}(\zeta)}\left(\frac{\hat{T}_{ex}-\hat{T}(\zeta)}{\hat{T}_{ex}-\hat{T}_0}(1+\hat{\sigma}\zeta)-\frac{\hat{T}(\zeta)}{\hat{T}_0}\right)\right] \\ & \delta T_0(\delta G_{T_0}(L)) \\ & +\frac{\sum_{i=1}^{n_{species}}\hat{n}_i(\exp(\delta G_i))}{\hat{n}}-1\end{aligned} \tag{2.51}$$

Equation 2.51 only depends on the outputs of the atmospheric model and model-dependent coefficients (e.g. the thermal diffusion coefficients). It can be evaluated with a single model call, once $\delta G_i(L)$ is available and δT_{ex} and δT_0 are computed as shown in appendix A.3.

RELATIVE EXOSPHERIC MEAN MASS ERROR

To complete the estimation of the relative density error, also the sensitivities of the mean mass with respect to T_{ex}, T_0 and n_0 need to be derived. The relative error in the mean mass is given to first order by equation 2.52.

$$\epsilon_M \cong \frac{\sum_{i=1}^{n_{species}}\frac{\partial M}{\partial n_{i_0}}\delta n_{i_0}}{\hat{M}}+\frac{\frac{\partial M}{\partial T_{ex}}\delta T_{ex}}{\hat{M}}+\frac{\frac{\partial M}{\partial T_0}\delta T_0}{\hat{M}} \tag{2.52}$$

Sensitivity of mean mass with respect to number density at the baseline altitude

Using the definition of mean mass (eq. 2.7), the partial derivative with respect to the species number density at the baseline altitude can be computed:

$$\begin{aligned}\frac{\partial M}{\partial n_{i_0}} &= \frac{\partial}{\partial n_{i_0}}\frac{\sum_{j=1}^{n_{species}}M_j n_j}{n} \\ &= \frac{\frac{\partial}{\partial n_{i_0}}\left(\sum_{j=1}^{n_{species}}M_j n_j\right)n-\left(\sum_{j=1}^{n_{species}}M_j n_j\right)\frac{\partial}{\partial n_{i_0}}n}{n^2} \\ &= \frac{\sum_{j=1}^{n_{species}}M_j\frac{\partial}{\partial n_{i_0}}n_j}{n}-\frac{M}{n}\sum_{j=1}^{n_{species}}\frac{\partial}{\partial n_{i_0}}n_j\end{aligned} \tag{2.53}$$

The derivative with respect to the i^{th} species is only nonzero if $j = i$, otherwise it is zero. The sum can therefore be dropped and the index set equal to i:

$$\frac{\partial M}{\partial n_{i_0}} = \frac{M_i \frac{\partial}{\partial n_{i_0}} n_i}{n} - \frac{M}{n} \frac{\partial}{\partial n_{i_0}} n_i \tag{2.54}$$

Introducing equation 2.35 yields:

$$\frac{\partial M}{\partial n_{i_0}} = \frac{M_i \frac{n_i}{n_{i_0}}}{n} - \frac{M}{n} \frac{n_i}{n_{i_0}} \tag{2.55}$$

The contribution of an error in the number density at the baseline altitude therefore becomes:

$$\epsilon_{M_{n0}} = \frac{\sum_{i=1}^{n_{species}} \left[\frac{M_i \frac{n_i}{n_{i_0}}}{n} - \frac{M}{n} \frac{n_i}{n_{i_0}} \right] \delta n_{i_0}}{\hat{M}} = \frac{\sum_{i=1}^{n_{species}} \left[M_i \frac{n_i}{n_{i_0}} - M \frac{n_i}{n_{i_0}} \right] \delta n_{i_0}}{n\hat{M}} \tag{2.56}$$

Introducing the definition of the absolute error in the species number density from equation 2.34, results in:

$$\begin{aligned} \epsilon_{M_{n0}} &= \frac{\sum_{i=1}^{n_{species}} \left[M_i \frac{n_i}{n_{i_0}} - M \frac{n_i}{n_{i_0}} \right] n_{i_0} (\exp(\delta G_i) - 1)}{n\hat{M}} \\ &= \frac{\sum_{i=1}^{n_{species}} M_i n_i \exp(\delta G_i) - M_i n_i - M n_i \exp(\delta G_i) + M n_i}{n\hat{M}} \end{aligned} \tag{2.57}$$

Using estimated quantities and the definition of mean mass, equation 2.57 becomes:

$$\epsilon_{M_{n0}} := \frac{\sum_{i=1}^{n_{species}} M_i \hat{n}_i \exp(\delta G_i)}{\hat{n}\hat{M}} - \frac{\sum_{i=1}^{n_{species}} \hat{n}_i \exp(\delta G_i)}{\hat{n}} \tag{2.58}$$

Sensitivity of mean mass with respect to exospheric temperature

To obtain the contribution of an error in exospheric temperature to a relative error in mean mass, it is necessary to compute the derivative $\frac{\partial M}{\partial T_{ex}}$:

$$\begin{aligned} \frac{\partial}{\partial T_{ex}} M &= \frac{\left(\frac{\partial}{\partial T_{ex}} \sum_{i=1}^{n_{species}} M_i n_i \right) n - \left(\sum_{i=1}^{n_{species}} M_i n_i \right) \frac{\partial}{\partial T_{ex}} n}{n^2} \\ &= \frac{\sum_{i=1}^{n_{species}} M_i \frac{\partial}{\partial T_{ex}} n_i}{n} - \frac{M}{n} \sum_{i=1}^{n_{species}} \frac{\partial}{\partial T_{ex}} n_i \end{aligned} \tag{2.59}$$

The derivative of the number density with respect to exospheric temperature is already available (equation 2.42) and needs to be inserted. Hence:

$$\frac{\partial M}{\partial T_{ex}} = \frac{1}{n}\sum_{i=1}^{n_{species}} M_i n_i \left[\frac{M_i g_0}{k T_{ex}^2}\left(\zeta + \frac{T_0}{T_0'}\ln\left(\frac{T_0}{T(\zeta)}\right)\right) - \frac{1+\alpha_i+\gamma_i}{T(\zeta)}\left(1 - \frac{T_{ex}-T(\zeta)}{T_{ex}-T_0}(1+\sigma\zeta)\right)\right] - \frac{M}{n}\sum_{i=1}^{n_{species}} n_i \left[\frac{M_i g_0}{k T_{ex}^2}\left(\zeta + \frac{T_0}{T_0'}\ln\left(\frac{T_0}{T(\zeta)}\right)\right) - \frac{1+\alpha_i+\gamma_i}{T(\zeta)}\left(1 - \frac{T_{ex}-T(\zeta)}{T_{ex}-T_0}(1+\sigma\zeta)\right)\right] \quad (2.60)$$

which can be simplified to:

$$\frac{\partial M}{\partial T_{ex}} = \left(\zeta + \frac{T_0}{T_0'}\ln\left(\frac{T_0}{T(\zeta)}\right)\right)\frac{g_0}{k T_{ex}^2}\left(\frac{\sum_{i=1}^{n_{species}} M_i^2 n_i}{n} - M^2\right) + \frac{\left(1 - \frac{T_{ex}-T(\zeta)}{T_{ex}-T_0}(1+\sigma\zeta)\right)}{T(\zeta)}\left(M\alpha + M\gamma - \frac{\sum_{i=1}^{n_{species}} M_i n_i \alpha_i}{n} - \frac{\sum_{i=1}^{n_{species}} M_i n_i \gamma_i}{n}\right) \quad (2.61)$$

The contribution of the absolute exospheric temperature error to the relative mean mass error is given by $\epsilon_{M_{T_{ex}}} := \frac{(\partial M/\partial T_{ex})\delta T_{ex}}{\hat{M}}$, which upon evaluation with the estimated quantities, results in equation 2.62.

$$\epsilon_{M_{T_{ex}}} = \left[\left(\zeta + \frac{\hat{T}_0}{\hat{T}_0'}\ln\left(\frac{\hat{T}_0}{\hat{T}(\zeta)}\right)\right)\frac{g_0}{k\hat{T}_{ex}^2}\left(\frac{\sum_{i=1}^{n_{species}} M_i^2 \hat{n}_i}{\sum_{i=1}^{n_{species}} M_i \hat{n}_i} - \hat{M}\right) + \frac{\left(1 - \frac{\hat{T}_{ex}-\hat{T}(\zeta)}{\hat{T}_{ex}-\hat{T}_0}(1+\hat{\sigma}\zeta)\right)}{\hat{T}(\zeta)}\left(\hat{\alpha} + \hat{\gamma} - \frac{\sum_{i=1}^{n_{species}} M_i \hat{n}_i \alpha_i}{\hat{n}\hat{M}} - \frac{\sum_{i=1}^{n_{species}} M_i \hat{n}_i \gamma_i}{\hat{n}\hat{M}}\right)\right]\delta T_{ex} \quad (2.62)$$

Sensitivity of mean mass with respect to baseline temperature

Finally, the derivative of the mean mass with respect to the baseline temperature needs to be computed:

$$\frac{\partial}{\partial T_0}M = \frac{\sum_{i=1}^{n_{species}} M_i \frac{\partial}{\partial T_0} n_i}{n} - \frac{M}{n}\sum_{i=1}^{n_{species}} \frac{\partial}{\partial T_0} n_i$$

Introducing the derivative of the number density with respect to the baseline temperature from eq. 2.48, leads to:

$$\frac{\partial M}{\partial T_0} = \sum_{i=1}^{n_{species}} \frac{M_i n_i}{n}\left[-\frac{M_i g_0}{k T_0' T_{ex}}\ln\left(\frac{T_0}{T(\zeta)}\right) + \frac{1+\alpha_i+\gamma_i}{T(\zeta)}\frac{T(\zeta) - T_0\frac{T_{ex}-T(\zeta)}{T_{ex}-T_0}(1+\sigma\zeta)}{T_0}\right] - \frac{M}{n}\sum_{i=1}^{n_{species}} n_i\left[-\frac{M_i g_0}{k T_0' T_{ex}}\ln\left(\frac{T_0}{T(\zeta)}\right) + \frac{1+\alpha_i+\gamma_i}{T(\zeta)}\frac{T(\zeta) - T_0\frac{T_{ex}-T(\zeta)}{T_{ex}-T_0}(1+\sigma\zeta)}{T_0}\right] \quad (2.63)$$

Simplification yields:

$$\frac{\partial M}{\partial T_0} = \frac{g_0}{k T_0' T_{ex}}\ln\left(\frac{T(\zeta)}{T_0}\right)\left(\frac{\sum_{i=1}^{n_{species}} M_i^2 n_i}{n} - M^2\right) - \frac{\frac{T(\zeta)}{T_0} - \frac{T_{ex}-T(\zeta)}{T_{ex}-T_0}(1+\sigma\zeta)}{T(\zeta)}\left(M\alpha + M\gamma - \frac{\sum_{i=1}^{n_{species}} M_i n_i \alpha_i}{n} - \frac{\sum_{i=1}^{n_{species}} M_i n_i \gamma_i}{n}\right) \quad (2.64)$$

The contribution of an absolute error in the baseline temperature to a relative error in the mean mass is to first order given by $\epsilon_{M_{T_0}} := \frac{(\partial M/\partial T_0)\delta T_0}{\hat{M}}$, which upon evaluation with estimated quantities, yields:

$$\epsilon_{M_{T_0}} = \left[\frac{g_0}{k\hat{T}_0'\hat{T}_{ex}} \ln\left(\frac{\hat{T}(\zeta)}{\hat{T}_0}\right)\left(\frac{\sum_{i=1}^{n_{species}} M_i^2\hat{n}_i}{\sum_{i=1}^{n_{species}} M_i\hat{n}_i} - \hat{M}\right)\right.$$
$$\left. - \frac{\frac{\hat{T}(\zeta)}{\hat{T}_0} - \frac{\hat{T}_{ex}-\hat{T}(\zeta)}{\hat{T}_{ex}-\hat{T}_0}(1+\hat{\sigma}\zeta)}{\hat{T}(\zeta)}\left(\hat{\alpha}+\hat{\gamma} - \frac{\sum_{i=1}^{n_{species}} M_i\hat{n}_i\alpha_i}{\hat{n}\hat{M}} - \frac{\sum_{i=1}^{n_{species}} M_i\hat{n}_i\gamma_i}{\hat{n}\hat{M}}\right)\right]\delta T_0 \quad (2.65)$$

Inserting equations 2.58, 2.62 and 2.65 into eq. 2.52, allows to obtain a single expression for the relative error in the mean mass caused by an absolute error in the spherical harmonics function of the environmental parameters:

$$\epsilon_M \cong \left[\left(\zeta + \frac{\hat{T}_0}{\hat{T}_0'}\ln\left(\frac{\hat{T}_0}{\hat{T}(\zeta)}\right)\right)\frac{g_0}{k\hat{T}_{ex}^2}\left(\frac{\sum_{i=1}^{n_{species}} M_i^2\hat{n}_i}{\sum_{i=1}^{n_{species}} M_i\hat{n}_i} - \hat{M}\right) + \right.$$
$$\left. \frac{1 - \frac{\hat{T}_{ex}-\hat{T}(\zeta)}{\hat{T}_{ex}-\hat{T}_0}(1+\hat{\sigma}\zeta)}{\hat{T}(\zeta)}\left(\hat{\alpha}+\hat{\gamma} - \frac{\sum_{i=1}^{n_{species}} M_i\hat{n}_i\alpha_i}{\hat{n}\hat{M}} - \frac{\sum_{i=1}^{n_{species}} M_i\hat{n}_i\gamma_i}{\hat{n}\hat{M}}\right)\right]$$
$$\delta T_{ex}(\delta G_{T_{ex}}(L))$$
$$+ \left[\frac{g_0}{k\hat{T}_0'\hat{T}_{ex}} \ln\left(\frac{\hat{T}(\zeta)}{\hat{T}_0}\right)\left(\frac{\sum_{i}^{n_{species}} M_i^2\hat{n}_i}{\sum_{i}^{n_{species}} M_i\hat{n}_i} - \hat{M}\right) - \right.$$
$$\left. \frac{\frac{\hat{T}(\zeta)}{\hat{T}_0} - \frac{\hat{T}_{ex}-\hat{T}(\zeta)}{\hat{T}_{ex}-\hat{T}_0}(1+\hat{\sigma}\zeta)}{\hat{T}(\zeta)}\left(\hat{\alpha}+\hat{\gamma} - \frac{\sum_{i=1}^{n_{species}} M_i\hat{n}_i\alpha_i}{\hat{n}\hat{M}} - \frac{\sum_{i=1}^{n_{species}} M_i\hat{n}_i\gamma_i}{\hat{n}\hat{M}}\right)\right]$$
$$\delta T_0(\delta G_{T_0}(L))$$
$$+ \frac{\sum_{i=1}^{n_{species}} M_i\hat{n}_i\exp(\delta G_i)}{\hat{n}\hat{M}} - \frac{\sum_{i=1}^{n_{species}} \hat{n}_i\exp(\delta G_i)}{\hat{n}} \quad (2.66)$$

Equations 2.51 and 2.66 can now be inserted into equation 2.31 to obtain eq. 2.67, a single expression for the overall relative density error.

Note that the product of the relative errors in equation 2.67 is by far smaller than the individual contributions of the number density and the mean mass errors (typically $\sim|10^{-6}|$ vs. $\sim|10^{-3}|$). It should however not be neglected when computing the relative density error, as it is important in cases where ϵ_n and ϵ_M are of similar magnitude but opposite sign.

$$
\begin{aligned}
\epsilon_\rho \cong & \left[\frac{g_0\hat{M}}{k\hat{T}_{ex}^2}\left(\zeta+\frac{\hat{T}_0}{\hat{T}_0'}\ln\left(\frac{\hat{T}_0}{\hat{T}(\zeta)}\right)\right)-\frac{1+\hat{\alpha}+\hat{\gamma}}{\hat{T}(\zeta)}\left(1-\frac{\hat{T}_{ex}-\hat{T}(\zeta)}{\hat{T}_{ex}-\hat{T}_0}(1+\hat{\sigma}\zeta)\right)\right.\\
&+\frac{g_0}{k\hat{T}_{ex}^2}\left(\zeta+\frac{\hat{T}_0}{\hat{T}_0'}\ln\left(\frac{\hat{T}_0}{\hat{T}(\zeta)}\right)\right)\left(\frac{\sum_{i=1}^{n_{species}}M_i^2\hat{n}_i}{\sum_{i=1}^{n_{species}}M_i\hat{n}_i}-\hat{M}\right)\\
&\left.+\frac{1-\frac{\hat{T}_{ex}-\hat{T}(\zeta)}{\hat{T}_{ex}-\hat{T}_0}(1+\hat{\sigma}\zeta)}{\hat{T}(\zeta)}\left(\hat{\alpha}+\hat{\gamma}-\frac{\sum_{i=1}^{n_{species}}M_i\hat{n}_i\alpha_i}{\hat{n}\hat{M}}-\frac{\sum_{i=1}^{n_{species}}M_i\hat{n}_i\gamma_i}{\hat{n}\hat{M}}\right)\right]\delta T_{ex}\\
&+\left[\frac{g_0\hat{M}}{k\hat{T}_{ex}\hat{T}_0'}\ln\left(\frac{\hat{T}(\zeta)}{\hat{T}_0}\right)-\frac{1+\hat{\alpha}+\hat{\gamma}}{\hat{T}(\zeta)}\left(\frac{\hat{T}_{ex}-\hat{T}(\zeta)}{\hat{T}_{ex}-\hat{T}_0}(1+\hat{\sigma}\zeta)-\frac{\hat{T}(\zeta)}{\hat{T}_0}\right)\right.\\
&+\frac{g_0}{k\hat{T}_0'\hat{T}_{ex}}\ln\left(\frac{\hat{T}(\zeta)}{\hat{T}_0}\right)\left(\frac{\sum_{i=1}^{n_{species}}M_i^2\hat{n}_i}{\sum_{i=1}^{n_{species}}M_i\hat{n}_i}-\hat{M}\right)\\
&\left.-\frac{\frac{\hat{T}(\zeta)}{\hat{T}_0}-\frac{\hat{T}_{ex}-\hat{T}(\zeta)}{\hat{T}_{ex}-\hat{T}_0}(1+\hat{\sigma}\zeta)}{\hat{T}(\zeta)}\left(\hat{\alpha}+\hat{\gamma}-\frac{\sum_{i=1}^{n_{species}}M_i\hat{n}_i\alpha_i}{\hat{n}\hat{M}}-\frac{\sum_{i=1}^{n_{species}}M_i\hat{n}_i\gamma_i}{\hat{n}\hat{M}}\right)\right]\delta T_0\\
&+\frac{\sum_{i=1}^{n_{species}}M_i\hat{n}_i\exp(\delta G_i)}{\hat{n}\hat{M}}-1\\
&+\epsilon_n\epsilon_M
\end{aligned}
\tag{2.67}
$$

2.2.2 *Variance of relative exospheric mass density error*

This section is devoted to the approximation of the variance of equation 2.67. In the following, δG_i is assumed to follow a zero-mean Gaussian distribution. Under this assumption the uncertainty in the relative density error is fully described by its variance. $Var(\delta T_{ex})$ and $Var(\delta T_0)$ are linearly related to $Var(\delta G_i)$, as shown in appendix A.3 and are therefore also zero-mean Gaussian.

Since variances are always positive, the product of the relative errors in the number density and mean mass can be safely dropped. Masking the prefactors with scalars a and b, equation 2.67 can be written as:

$$
\epsilon_\rho \cong a\delta T_{ex}+b\delta T_0+\frac{\sum_{i=1}^{n_{species}}M_i\hat{n}_i\exp(\delta G_i)}{\hat{n}\hat{M}}-1 \tag{2.68}
$$

Using fundamental variance properties (cf. appendix A.1), the relative density error variance can be computed according to equation 2.69.

$$
\begin{aligned}
Var(\epsilon_\rho) = & \, a^2\,Var(\delta T_{ex})+b^2\,Var(\delta T_0)+\frac{Var\left(\sum_{i=1}^{n_{species}}M_i\hat{n}_i\exp(\delta G_i)\right)}{(\hat{n}\hat{M})^2}\\
&+2ab\,Cov(\delta T_{ex},\delta T_0)+2\frac{a}{\hat{n}\hat{M}}\,Cov\left(\delta T_{ex},\sum_{i=1}^{n_{species}}M_i\hat{n}_i\exp(\delta G_i)\right)\\
&+2\frac{b}{\hat{n}\hat{M}}\,Cov\left(\delta T_0,\sum_{i=1}^{n_{species}}M_i\hat{n}_i\exp(\delta G_i)\right)
\end{aligned}
\tag{2.69}
$$

$Var(\delta T_{ex})$ and $Var(\delta T_0)$ are given by equations A.19 and A.20 of appendix A.2 and derive from $\delta G_{T_{ex}}(L)$, as well as $\delta G_0(L)$. The remaining components that need to be computed are therefore:

- $Cov(\delta T_{ex}, \delta T_0)$
- $Var\left(\sum_{i=1}^{n_{species}} M_i \hat{n}_i \exp(\delta G_i)\right)$
- $Cov\left(\delta T_{ex}, \sum_{i=1}^{n_{species}} M_i \hat{n}_i \exp(\delta G_i)\right)$
- $Cov\left(\delta T_0, \sum_{i=1}^{n_{species}} M_i \hat{n}_i \exp(\delta G_i)\right)$

Covariance of exospheric temperature error and baseline temperature error
The definition of covariance (eq. A.8) implies that:

$$Cov(\delta T_{ex}, \delta T_0) = E[\delta T_{ex} \delta T_0] - E[\delta T_{ex}] E[\delta T_0] \tag{2.70}$$

As δT_{ex} and δT_0 are assumed to follow zero-mean Gaussian distributions, the covariance is equal to the expected value of the product of the absolute errors in the baseline and exospheric temperatures. Using equations A.17 and A.18 leads to:

$$\begin{aligned} Cov(\delta T_{ex}, \delta T_0) &= E[\delta T_{ex} \delta T_0] = T_{0_0} T_{ex_0} E[\delta G_{T_0} \delta G_{T_{ex}}] \\ &= T_{0_0} T_{ex_0} \, Cov(\delta G_{T_0}, \delta G_{T_{ex}}) \end{aligned} \tag{2.71}$$

Variance of the error in the particle density at the baseline altitude
The variance of the sum over all considered species is defined as:

$$\begin{aligned} Var\left(\sum_{i=1}^{n_{species}} M_i \hat{n}_i \exp(\delta G_i)\right) &= \sum_{i=1}^{n_{species}} M_i^2 \hat{n}_i^2 \, Var(\exp(\delta G_i)) \\ + 2 \sum_{i=1}^{n_{species}} \sum_{\substack{j=1, \\ j \neq i}}^{n_{species}} & M_i n_i M_j n_j \, Cov\big(\exp(\delta G_i), exp(\delta G_j)\big) \end{aligned} \tag{2.72}$$

Since δG_i is assumed to follow a zero-mean normal distribution, $\exp(\delta G_i)$ is lognormal with:

$$Var(\exp(\delta G_i)) = e^{Var(\delta G_i)}\left(e^{Var(\delta G_i)} - 1\right) \tag{2.73}$$

Applying the definition of covariance (eq. A.8) results in:

$$\begin{aligned} &Cov\big(\exp(\delta G_i), \exp(\delta G_j)\big) \\ &\quad = E\big[\exp(\delta G_i) \exp(\delta G_j)\big] - E\big[\exp(\delta G_i)\big] \, E\big[\exp(\delta G_j)\big] \\ &\quad = E\big[\exp(\delta G_i + \delta G_j)\big] - E\big[\exp(\delta G_i)\big] \, E\big[\exp(\delta G_j)\big] \end{aligned} \tag{2.74}$$

Each expectation may be computed from the first moment of the lognormal distribution. The addition of two correlated zero-mean Gaussians in the first term results in:

$$\begin{aligned} \delta G_i + \delta G_j &\sim \mathcal{N}\left(\mu_{\delta G_i} + \mu_{\delta G_j}, \sigma^2_{\delta G_i} + \sigma^2_{\delta G_j} + 2\, Cov\big(\delta G_i, \delta G_j\big)\right) \\ &\cong \mathcal{N}\big(0.0, Var(\delta G_i) + Var(\delta G_j) + 2\, Cov(\delta G_i, \delta G_j)\big) \end{aligned} \tag{2.75}$$

Using equation 2.75 to complete eq. 2.74 yields:

$$\begin{aligned}\mathrm{Cov}(\exp(\delta G_i), \exp(\delta G_j)) \cong\ & e^{\frac{1}{2}\left(\mathrm{Var}(\delta G_i)+\mathrm{Var}(\delta G_j)\right)+\mathrm{Cov}(\delta G_i, \delta G_j)} \\ & - e^{\frac{1}{2}\mathrm{Var}(\delta G_i)} e^{\frac{1}{2}\mathrm{Var}(\delta G_j)}\end{aligned} \tag{2.76}$$

Evaluating equation 2.72 using eqs. 2.73 and 2.76 then results in the desired expression for the variance of the error in the particle density:

$$\begin{aligned}\mathrm{Var}\left(\sum_{i=1}^{n_{\text{species}}} M_i \hat{n}_i \exp(\delta G_i)\right) =\ & \sum_{i=1}^{n_{\text{species}}} M_i^2 \hat{n}_i^2 e^{\mathrm{Var}(\delta G_i)}\left(e^{\mathrm{Var}(\delta G_i)} - 1\right) \\ & + 2 \sum_{i=1}^{n_{\text{species}}} \sum_{\substack{j=1, \\ j\neq i}}^{n_{\text{species}}} M_i n_i M_j n_j \left(e^{\frac{1}{2}\left(\mathrm{Var}(\delta G_i)+\mathrm{Var}(\delta G_j)\right)}\left(e^{\mathrm{Cov}(\delta G_i, \delta G_j)} - 1\right)\right)\end{aligned} \tag{2.77}$$

Covariance of exospheric temperature error and species number density errors at the baseline altitude

From the zero-mean assumption of δT_{ex} it follows that the covariance is equal to the expectation of the product:

$$\begin{aligned}\mathrm{Cov}\left(\delta T_{ex}, \sum_{i=1}^{n_{\text{species}}} M_i \hat{n}_i \exp(\delta G_i)\right) &= \mathrm{E}\left[\sum_{i=1}^{n_{\text{species}}} M_i \hat{n}_i \delta T_{ex} \exp(\delta G_i)\right] \\ &= \sum_{i=1}^{n_{\text{species}}} M_i \hat{n}_i T_{ex_0} \mathrm{E}\left[\delta G_T \exp(\delta G_i)\right]\end{aligned} \tag{2.78}$$

Equation 2.78 requires the expected value of the product of a normal and lognormal distribution. This type of distribution is sometimes denoted "Normal Lognormal Mixture" (NLNM), however it has only received little treatment in the literature. One of the very few papers on this type of distribution is Yang (2008). In his work, Yang states the first two moments of the NLNM created from a standard normal distribution with unit variance and a generic normal distribution without giving a derivation. Both are connected via an arbitrary correlation coefficient ρ. Equation 2.78 requires the first moment of the NLNM which is created from two correlated normal distributions with non-unit variance. The computation of the required first moment is given in appendix A.4 and extends the work of Yang for the case of non-standard normal Gaussians. Applying the final result, equation A.30, yields:

$$\mathrm{E}\left[\delta G_T \exp(\delta G_i)\right] = \sigma_{\delta G_T} \sigma_{\delta G_i} \rho e^{\frac{1}{2}\sigma_{\delta G_i}^2} \tag{2.79}$$

Introducing the definition of the correlation coefficient (eq. A.10), it follows that:

$$\begin{aligned}\mathrm{E}[\delta G_T \exp(\delta G_i)] &= \sigma_{\delta G_T} \sigma_{\delta G_i} \frac{\mathrm{Cov}(\delta G_T, \delta G_i)}{\sqrt{\mathrm{Var}(\delta G_T)}\sqrt{\mathrm{Var}(\delta G_i)}} e^{\frac{1}{2}\mathrm{Var}(\delta G_i)} \\ &= \mathrm{Cov}(\delta G_T, \delta G_i) e^{\frac{1}{2}\mathrm{Var}(\delta G_i)}\end{aligned} \tag{2.80}$$

Inserting eq. 2.80 into equation 2.78 allows the computation of the covariance as the sum of the individual covariances of the species:

$$\mathrm{Cov}\left(\delta T_{ex}, \sum_{i=1}^{n_{species}} M_i \hat{n}_i \exp(\delta G_i)\right) = \sum_{i=1}^{n_{species}} M_i \hat{n}_i T_{ex_0} \mathrm{Cov}(\delta G_T, \delta G_i)\, e^{\frac{1}{2}\mathrm{Var}(\delta G_i)} \tag{2.81}$$

Equation 2.81 is also valid for the last term to be computed, i.e. $\mathrm{Cov}(\delta T_0, \sum_{i=1}^{n_{species}} M_i \hat{n}_i \exp(\delta G_i))$, as δT_{ex} and δT_0 only differ by scalar coefficients. It therefore holds that:

$$\mathrm{Cov}\left(\delta T_0, \sum_{i=1}^{n_{species}} M_i \hat{n}_i \exp(\delta G_i)\right) = \sum_{i=1}^{n_{species}} M_i \hat{n}_i T_{0_0} \mathrm{Cov}(\delta G_0, \delta G_i)\, e^{\frac{1}{2}\mathrm{Var}(\delta G_i)} \tag{2.82}$$

This completes the equations to evaluate the variance of the relative density error. Inserting eqs. 2.71, 2.77 and 2.82 into eq. 2.69 leads to:

$$\begin{aligned}
\mathrm{Var}(\epsilon_\rho) = {} & a^2\,\mathrm{Var}(\delta T_{ex}) + b^2\,\mathrm{Var}(\delta T_0) \\
& + \frac{\sum_{i=1}^{n_{species}} M_i^2 \hat{n}_i^2 e^{\mathrm{Var}(\delta G_i)}\left(e^{\mathrm{Var}(\delta G_i)} - 1\right)}{(\hat{n}\hat{M})^2} \\
& + \frac{2}{(\hat{n}\hat{M})^2} \sum_{i=1}^{n_{species}} \sum_{\substack{j=1,\\ j\neq i}}^{n_{species}} M_i n_i M_j n_j \left(e^{\frac{1}{2}\left(\mathrm{Var}(\delta G_i)+\mathrm{Var}(\delta G_j)\right)}\left(e^{\mathrm{Cov}(\delta G_i, \delta G_j)} - 1\right)\right) \\
& + 2ab T_{0_0} T_{ex_0}\, \mathrm{Cov}(\delta G_{T_0}, \delta G_{T_{ex}}) \\
& + 2\frac{a}{\hat{n}\hat{M}} \sum_{i=1}^{n_{species}} M_i \hat{n}_i T_{ex_0}\, \mathrm{Cov}(\delta G_{T_{ex}}, \delta G_i) e^{\frac{1}{2}\mathrm{Var}(\delta G_i)} \\
& + 2\frac{b}{\hat{n}\hat{M}} \sum_{i=1}^{n_{species}} M_i \hat{n}_i T_{0_0}\, \mathrm{Cov}(\delta G_0, \delta G_i) e^{\frac{1}{2}\mathrm{Var}(\delta G_i)}
\end{aligned} \tag{2.83}$$

Equation 2.83 yields an approximate description of the relative density error variance. It only depends on outputs of a single atmospheric model call and model-dependent coefficients. The variances and covariances of the errors in the function of the environmental parameters are model and origin dependent. The computation for the cases of solar flux and magnetic index uncertainty is the subject of sections 2.3 and 2.4.

Reduced versions of equations 2.67 and 2.83 can be formulated if simplifying assumptions are introduced. In Schiemenz et al. (2019b) the dependencies on the baseline altitude quantities n_0 and T_0 have been dropped. Furthermore, the thermal diffusion coefficients have been set to zero and the assumption $T(\zeta) = T_{ex}$ has been introduced instead of considering the Bates profile. Under these assumptions the relative density error may be expressed by equation 2.84 and its variance by eq. 2.85.

$$\epsilon_\rho \cong \left(\frac{g_0 M(z)}{k T_{ex}^2} \frac{R_\oplus + z_0}{R_\oplus + z}(z - z_0) - \frac{1}{T_{ex}}\right) \delta T_{ex} \tag{2.84}$$

$$\mathrm{Var}(\epsilon_\rho) \cong T_{ex_0}^2 \left(\frac{g_0 M(z)}{k T_{ex}^2} \frac{R_\oplus + z_0}{R_\oplus + z}(z - z_0) - \frac{1}{T_{ex}}\right)^2 \mathrm{Var}(\delta G_{T_{ex}}(L)) \tag{2.85}$$

It has however been shown that the amount of simplifications degrade the error and variance estimates, such that a tuning-function was required to obtain appropriate estimates of $Var(\epsilon_\rho)$. Consequently, eqs. 2.67 and 2.83 should be favored over eqs. 2.84 and 2.85 for practical operations.

2.3 QUANTIFICATION OF PROPAGATED SOLAR FLUX UNCERTAINTY

As depicted in figure 2.1, errors in the solar flux proxies propagate to errors in the function of the spherical harmonics. The MSIS and DTM-class models considered in this thesis all make use of the F10.7 index to describe the solar forcing of the atmosphere. In section 1.3.3 it was outlined that two parameters for the solar flux input are commonly used: a daily flux input of the previous day and an 81-day flux-average. If not stated otherwise, the term "solar flux" is henceforth used to refer to the daily flux input argument (denoted E) and not the 81-day-average (denoted $\bar{E}$), as this is the parameter with the largest error in orbit propagation/estimation (the average flux parameter is much more stable).

The nature of solar flux input errors highly depends on the operational scenario. Simulations of past orbits can use fully processed solar flux data, whereas real-time applications or even predictions into the future (orbit forecasting) suffer from significantly larger uncertainties, as the uncertainty of the solar flux prediction increases with propagation time. Consequently different process noise models are required for these use-cases. Emmert et al. (2017) suggest a simple white noise process for orbit determination with published solar flux data and a Brownian motion process for orbit forecasting.

"The F10.7 values are deemed to be accurate to one solar flux unit or 1% of the flux value, whichever is the larger." (Tapping, 2013)

Published solar flux proxy values suffer from measurement uncertainty and temporal undersampling. Tapping (2013) mentions an accuracy in the order of 1-1.4% and notes the dependence on the flux monitor calibration process, as well as epistemic and aleatoric contributions. Further information about the F10.7 generation process, overall data accuracy and other flux issues such as temporal undersampling, may be found in Tapping (2013).

To predict the resulting relative atmospheric density error and variance using equations 2.67 and 2.83, an analytic mapping for the transition $\delta E \rightarrow \delta G_i$ is required. This mapping, its variance and the inter-species covariance are the subject of section 2.3.1.

2.3.1 *Absolute error, variance and covariance in $G_i(L)$ due to uncertain solar flux information*

As shown in Schiemenz et al. (2019b), an error in the solar flux input propagates to an error in the function of the spherical harmonics

according to equation 2.86, where a_i and b_i are model-dependent scalars that may depend on the 81-day solar flux component $\bar{E}$.

$$\delta G_i = a_i \delta E + b_i \delta E^2 \tag{2.86}$$

For the MSIS-class models equation 2.86 takes the form:

$$\begin{aligned}\delta G_i = &\underbrace{\left(c1_i + 2c2_i(\hat{E} - \bar{E})(1 + cc_i) + c1_i c3_i(\bar{E} - 150)\right)}_{a_i} \delta E \\ &+ \underbrace{c2_i(1 + cc_i)}_{b_i} \delta E^2\end{aligned} \tag{2.87}$$

with i denoting the species under consideration. $c1_i$, $c2_i$ and $c3_i$ are model and species-dependent coefficients. Some values are given in Schiemenz et al. (2019b). The remaining coefficients can be derived from the source codes of the models. cc_i is a scalar that accounts for cross-coupling terms. It describes the influence of the asymmetrical annual, diurnal, semidiurnal and terdiurnal variations in $G(L)$ which are coupled to the solar flux input. Any change in δE hence also causes a different response in these components and therefore a change in cc_i. Its exact form is very lengthy and tied to the model-specific implementation of $G(L)$. It can be derived from a sensitivity analysis of $G(L)$ for each model under consideration. General building blocks of cc_i are model-coefficients and Legendre polynomials in latitude/longitude. So far, the MSIS and DTM-class models each allow for unified descriptions that only differ in the model-coefficients.

For the DTM-class models the full form of equation 2.86 reads:

$$\begin{aligned}\delta G_i &= \left(\left(c1_i + 2c2_i(\hat{E} - \bar{E})\right)\delta E + c2_i \delta E^2\right)(1 + cc_i) \\ &= \underbrace{\left((1 + cc_i)(c1_i + 2c2_i(\hat{E} - \bar{E})\right)}_{a_i} \delta E + \underbrace{c2_i(1 + cc_i)}_{b_i} \delta E^2\end{aligned} \tag{2.88}$$

The DTM-models do not contain a dependency of δG_i on the term $(\bar{E} - 150)$. Also periodic cross-couplings are modeled differently. The exact equation for cc_i is given by the parameter f_p of the DTM-internal function `gldtm_XX`. Equations 2.87 and 2.88 assume similar shape. Therefore also the species variances and covariances only differ by the scalars a_i and b_i, which allows to derive them in a common fashion.

Variance of absolute error in $G_i(L)$

Applying the definition of variance (eq. A.3) and the linear combination property (eq. A.6) to equation 2.86, results in:

$$Var(\delta G_i) = a_i^2 \, Var(\delta E) + 2b_i^2 \, Var(\delta E)^2 + 2a_i b_i \, Cov(\delta E, \delta E^2) \tag{2.89}$$

Due to the Gaussian zero-mean assumption of δE, it holds that $Cov(\delta E, \delta E^2) = 0$, such that eq. 2.89 becomes:

$$Var(\delta G_i) = a_i^2 \, Var(\delta E) + 2b_i^2 \, Var(\delta E)^2 \tag{2.90}$$

Inter-species covariance
Next, a description of $Cov(\delta G_i, \delta G_j)$ is required. Since $E\left[\delta E^2\right] = Var(\delta E)$ the expected value of equation 2.86 is not formally zero-mean. Therefore, the covariance is computed as:

$$\begin{aligned} Cov(\delta G_i, \delta G_j) &= E[\delta G_i \delta G_j] - E[\delta G_i]E[\delta G_j] \\ &= E\left[(a_i \delta E + b_i \delta E^2)(a_j \delta E + b_j \delta E^2)\right] \\ &\quad - b_i Var(\delta E) b_j Var(\delta E) \\ &= a_i a_j E[\delta E^2] + (a_i b_j + a_j b_i) E[\delta E^3] + \\ &\quad b_i b_j E[\delta E^4] - b_i b_j Var(\delta E)^2 \end{aligned} \tag{2.91}$$

The third moment of a zero-mean normal distribution is zero ($E[\delta E^3] = 0$) and the fourth central moment equates to $3\,Var(\delta E)^2$ (cf. table A.1), which results in:

$$Cov(\delta G_i, \delta G_j) = a_i a_j Var(\delta E) + 2 b_i b_j Var(\delta E)^2 \tag{2.92}$$

Only equations 2.90 and 2.92 are required to evaluate eq. 2.83 for solar flux uncertainty. Since δG_i is not formally zero-mean for a zero-mean absolute solar flux error input, also the relative exospheric density error will not formally be zero-mean. However, since b_i is generally small, δG_i can be approximated as *effectively* zero-mean. Consequently also the relative density error is expected to be effectively zero-mean. The uncertainty quantification equations allow for the white noise and Brownian motion models suggested in Emmert et al. (2017) to be used.

2.3.2 *Simulation results*

To validate equations 2.67, 2.83 and 2.86-2.92, the estimated error is compared against a model propagation and the variance against a Monte-Carlo simulation.

ERROR PROPAGATION (EQUATIONS 2.86 AND 2.67)

If equation 2.67 agrees with the output of the semi-empirical model for a given $\delta G_i(\delta E)$ from equation 2.86, then the ratio $\frac{\epsilon_\rho}{\hat{\epsilon}_\rho}$ is equal to unity. Variations from the ideal value of 1.0 indicate that the error propagation equations are struggling to model the true error propagation path for a certain altitude/solar flux combination.

To show the general validity of equations 2.86 and 2.67, various models and solar flux/altitude combinations have been tested. From the MSIS-class models NRLMSISE-00 and from the DTM-class models DTM-2012 and DTM-2013 were selected. DTM-2013 operates using the F30 proxy instead of the F10.7 values. However, since the F30 data is scaled linearly to F10.7 flux units prior to being supplied to the model, there is no change in the error propagation equations (δE then corresponds to the scaled proxy). The altitude/flux-grid corresponds to the entire possible flux scale (67 sfu to 300 sfu). The altitude domain

was limited to the most populated LEO shell between 300 and 1200 km. The results are presented in figure 2.3 and demonstrate that the path shown in figure 2.1 has been modeled correctly.

VARIANCE PROPAGATION (EQUATIONS 2.83, 2.90 AND 2.92)

To validate the atmospheric density input uncertainty quantification, 300 Monte-Carlo iterations were computed for four selected altitude/flux combinations, covering low to high LEO altitudes and quiet to stormy solar conditions. To demonstrate compatibility with both process noise models suggested by Emmert, solar flux errors were simulated in two cases as Gaussian white noise and in the remaining cases as a Brownian motion process. The standard deviation of the white noise process (and therefore also the standard deviation at the first time-step of the Brownian motion model) was chosen as ~1.23 sfu, corresponding to a double-sided power spectral density of 130 000 $\mathrm{sfu}^2\mathrm{s}$ at daily sampling.

The results for NRLMSISE-00 are depicted in figure 2.4. In all cases the upper left plot contains the solar flux errors (daily sampling). The upper right plot shows the resulting errors and uncertainty in the baseline temperature and the lower left plot depicts the resulting exospheric temperature errors and their covariance. The corresponding relative density error is depicted in the lower right plot. The true variance of the Monte-Carlo iterations is colored red and the estimated variance blue. Monte-Carlo iterations are depicted in green. A thin black line illustrates the mean of the MC-iterations to validate the zero-mean assumption at the intermediate stages. All variances are correctly estimated using equations 2.83, 2.90 and 2.92, for both the white noise and Brownian motion error process models.

The DTM-2012 results are shown in figure 2.5. As the DTM-models formulate the baseline temperature independently of solar flux changes, the T_0 subplot has been dropped. All cases match the Monte-Carlo three sigma bounds. Case 2.5c is known to result in non-Gaussian relative density errors for DTM-2012 (Schiemenz et al., 2019b). A comparison with DTM-2013 (figure 2.6c) shows that the behavior is unique to DTM-2012 and therefore relates to the DTM-2012 model coefficients. As has been outlined in Schiemenz et al. (2019b), the behavior is caused by the parametrization of c_{2_i}, which causes the nonlinear part of equation 2.86 to dominate over the linear part, thereby corrupting Gaussianity for the chosen set of parameters. Despite the non-Gaussian nature of the errors and the slightly violated zero-mean assumption, the variance is still correctly estimated.

The uncertainty quantification for the error propagation from solar flux errors to relative density errors also yields correct estimates for DTM-2013 (cf. figure 2.6), justifying the Gaussian and zero-mean approximations. Comparing absolute error levels, DTM-2013 shows the strongest response in relative density to the input solar flux errors.

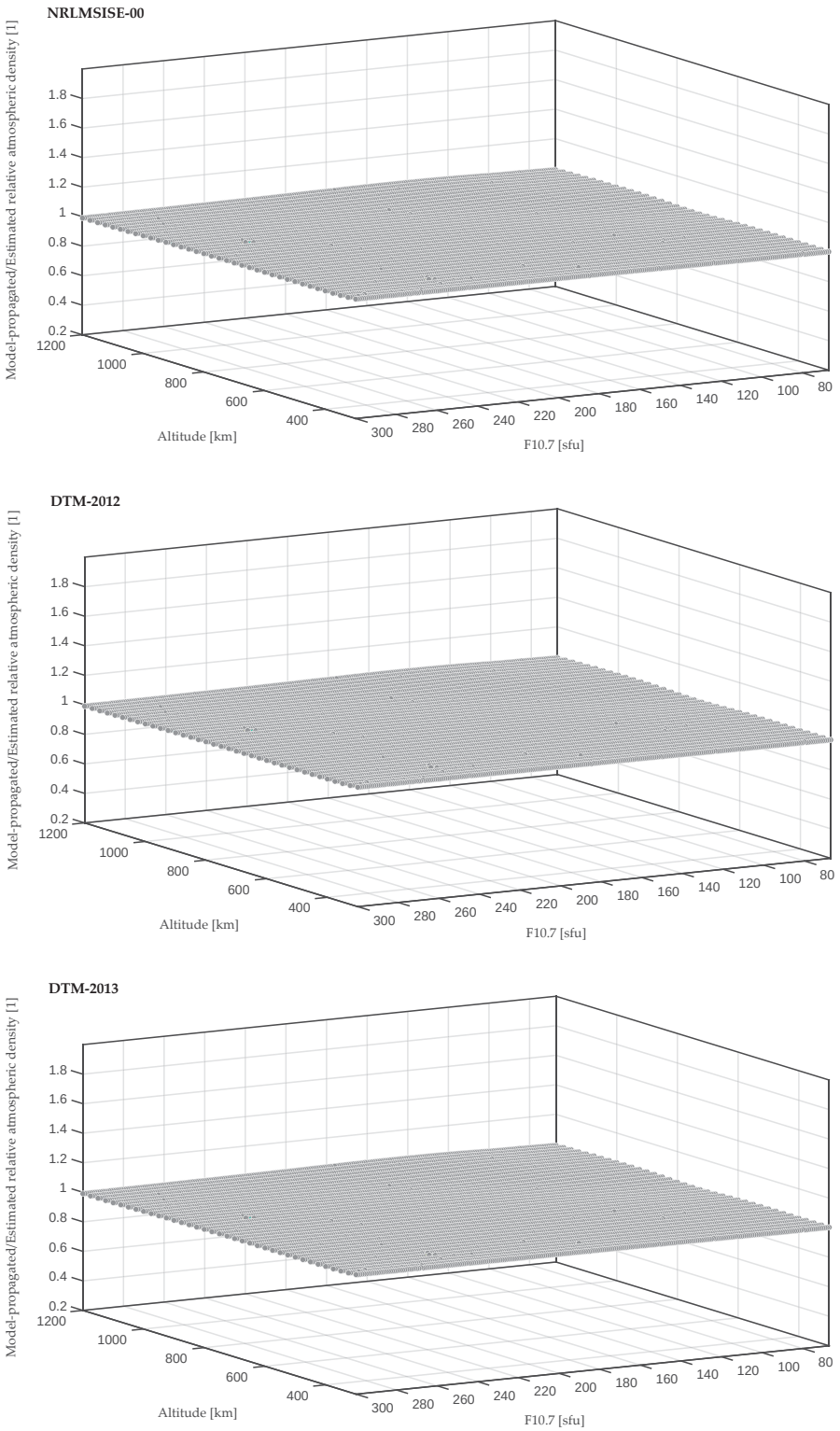
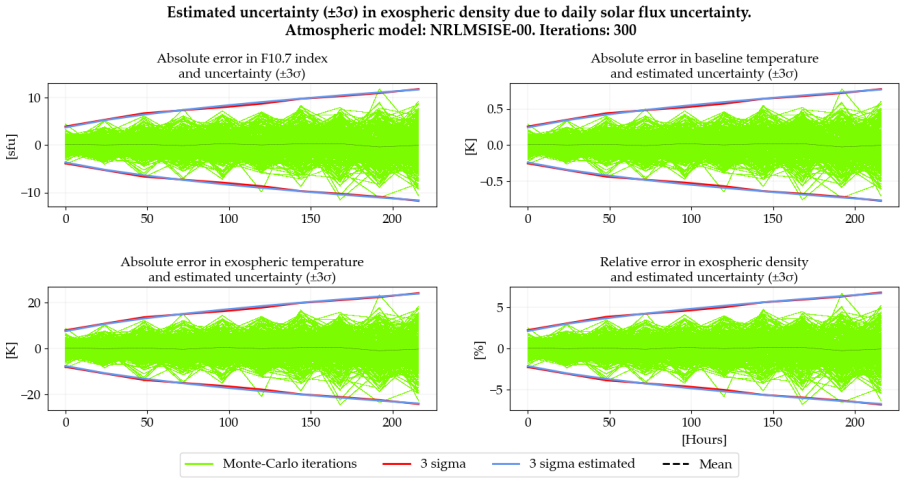
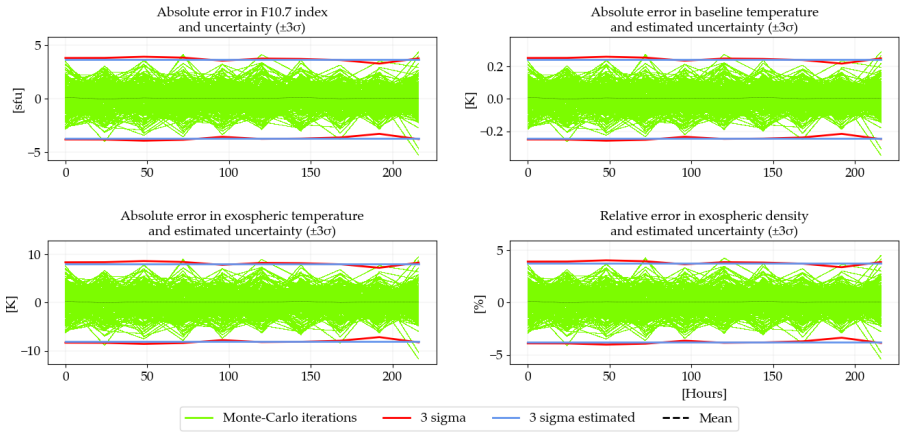


Figure 2.3: Plots of the ratio $\frac{\epsilon_\rho}{\hat{\epsilon}_\rho}$ for NRLMSISE-00, DTM-2012 and DTM-2013 with respect to altitude and daily F10.7 index. The estimated relative density error (equation 2.67 evaluated with eq. 2.87 for NRLMSISE-00 and eq. 2.88 for DTM-2012 and DTM-2013) matches the true value for all models, resulting in perfectly horizontal planes

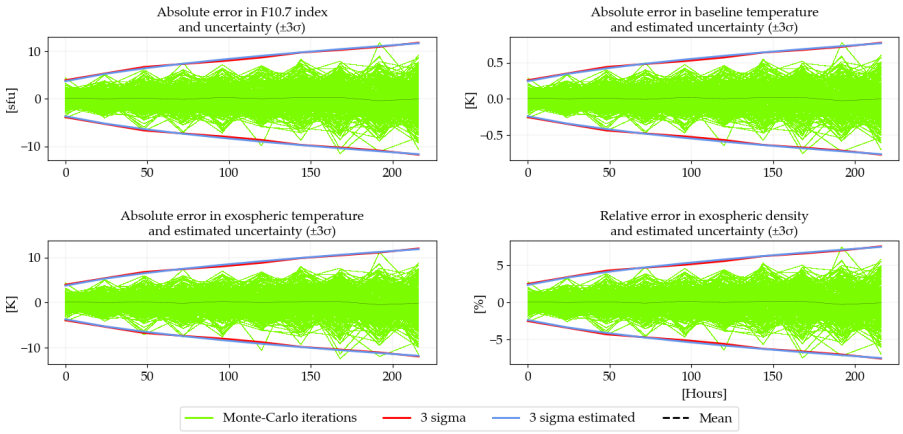
NRLMSISE-00



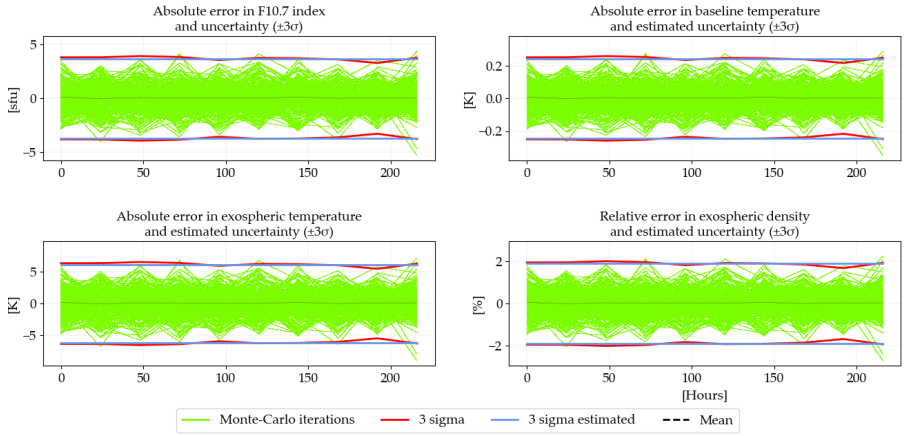
(a) $z = 300\,\mathrm{km}$, $\bar{\mathrm{E}} = 80\,\mathrm{sfu}$, $\hat{\mathrm{E}} = 80\,\mathrm{sfu}$, Brownian motion



(b) $z = 500\,\mathrm{km}$, $\bar{\mathrm{E}} = 130\,\mathrm{sfu}$, $\hat{\mathrm{E}} = 100\,\mathrm{sfu}$, white Gaussian noise



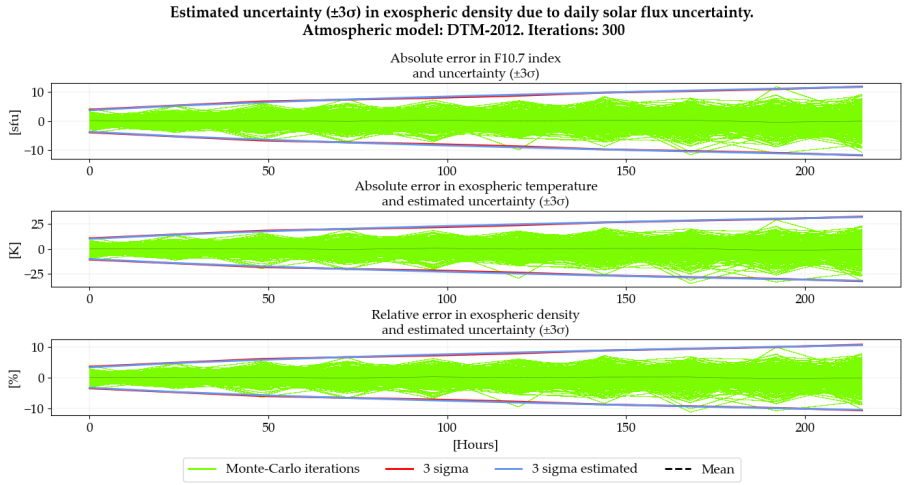
(c) $z = 700\,\mathrm{km}$, $\bar{\mathrm{E}} = 130\,\mathrm{sfu}$, $\hat{\mathrm{E}} = 200\,\mathrm{sfu}$, Brownian motion



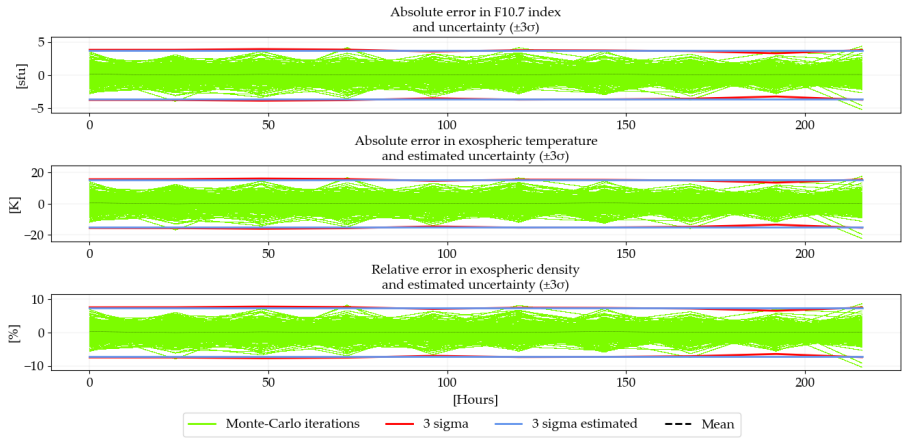
(d) $z = 1100\,\mathrm{km}$, $\bar{E} = 170\,\mathrm{sfu}$, $\hat{E} = 170\,\mathrm{sfu}$, white Gaussian noise

Figure 2.4: Uncertainty quantification of propagated solar flux errors using NRLMSISE-00

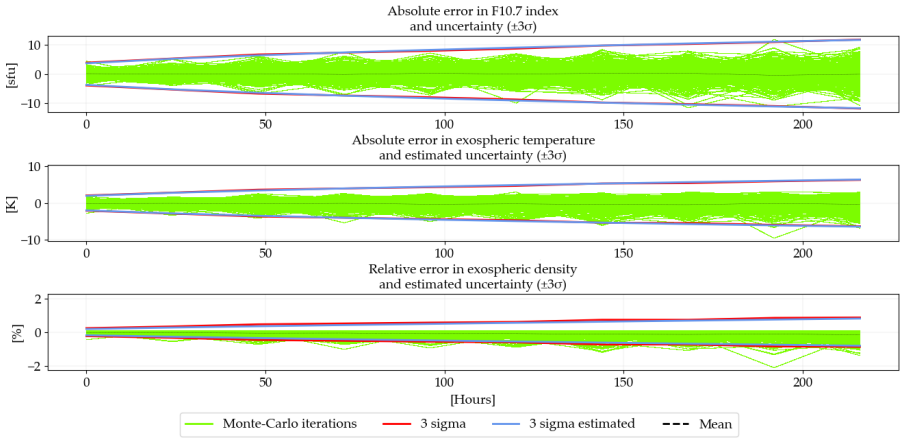
DTM-2012



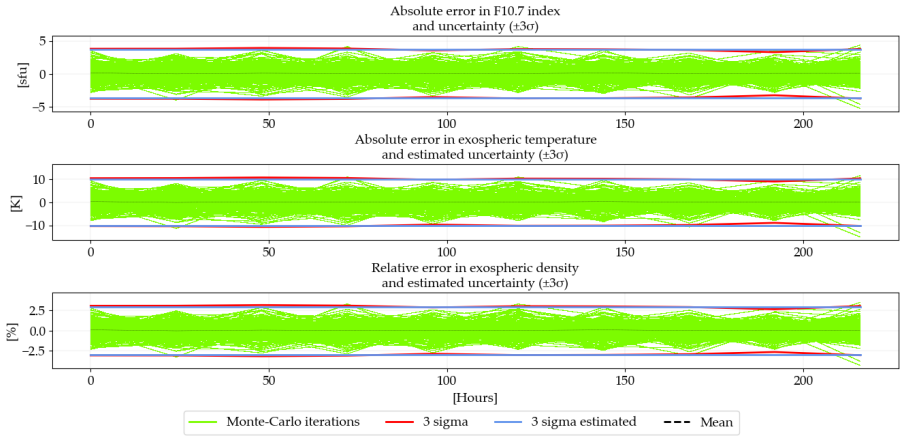
(a) $z = 300\,\mathrm{km}$, $\bar{E} = 80\,\mathrm{sfu}$, $\hat{E} = 80\,\mathrm{sfu}$, Brownian motion



(b) $z = 500\,\mathrm{km}$, $\bar{E} = 130\,\mathrm{sfu}$, $\hat{E} = 100\,\mathrm{sfu}$, white Gaussian noise



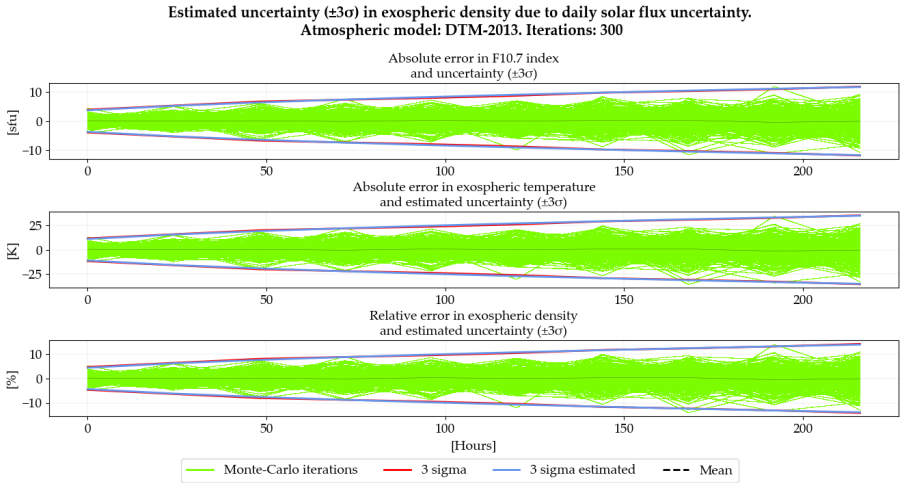
(c) $z = 700\,\mathrm{km}$, $\bar{E} = 130\,\mathrm{sfu}$, $\hat{E} = 200\,\mathrm{sfu}$, Brownian motion



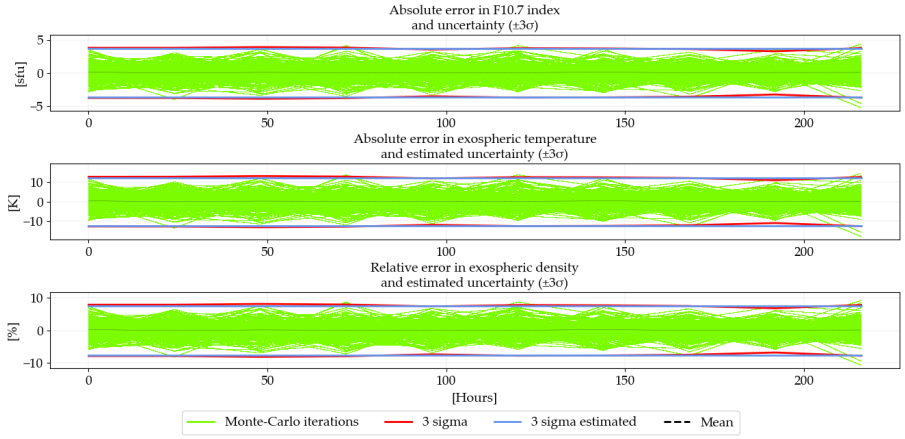
(d) $z = 1100\,\mathrm{km}$, $\bar{E} = 170\,\mathrm{sfu}$, $\hat{E} = 170\,\mathrm{sfu}$, white Gaussian noise

Figure 2.5: Uncertainty quantification of propagated solar flux errors using DTM-2012

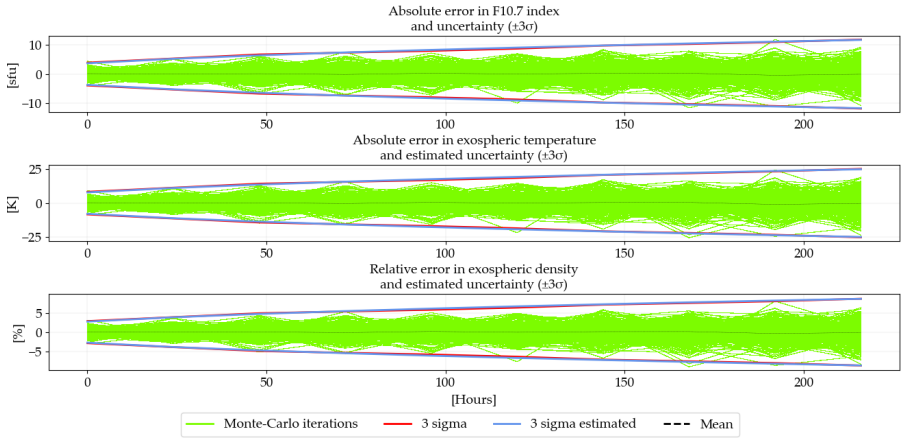
DTM-2013



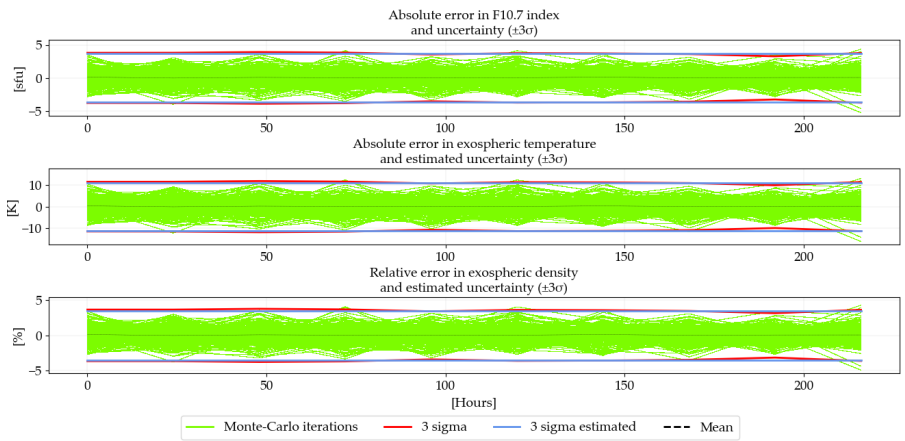
(a) $z = 300\,\mathrm{km}$, $\bar{E} = 80\,\mathrm{sfu}$, $\hat{E} = 80\,\mathrm{sfu}$, Brownian motion



(b) $z = 500\,\mathrm{km}$, $\bar{E} = 130\,\mathrm{sfu}$, $\hat{E} = 100\,\mathrm{sfu}$, white Gaussian noise



(c) $z = 700\,\mathrm{km}$, $\bar{E} = 130\,\mathrm{sfu}$, $\hat{E} = 200\,\mathrm{sfu}$, Brownian motion



(d) $z = 1100\,\mathrm{km}$, $\bar{E} = 170\,\mathrm{sfu}$, $\hat{E} = 170\,\mathrm{sfu}$, white Gaussian noise

Figure 2.6: Uncertainty quantification of propagated solar flux errors using DTM-2013

2.4 QUANTIFICATION OF PROPAGATED MAGNETIC INDEX UNCERTAINTY

Geomagnetic activity parameters are used in semi-empirical density models as indicators of the global state or specific parts of the magnetosphere, given the current forcing by the interplanetary magnetic field (IMF). Many different geomagnetic indices exist, which can be grouped into letter-prefixed classes. Some examples are:

- C-class indices: C_i, C_p, C_9
- K-class indices: kp/ap, km/am (kn/an, ks/as), aa
- Polar and auroral indices: PC, AE (AL, AO, AU)
- Storm indices: DST, SYM/ASY
- Newly proposed indices: IHV/IDV
- Pulsation indices: Wp, ULF, PC3

For the K-class indices a capitalized first letter typically implies a daily average value, whereas lowercase letters refer to shorter timescales (e.g. three hours). Not all authors follow this convention, however it is applied in this thesis. An excellent description of the various geomagnetic activity parameters and their derivation can be found in Mandea and Korte (2010, Section 8).

Geomagnetic indices can be broadly grouped into two classes: those that consider a planetary disturbance level and others that focus on disturbances in special regions (e.g. the Disturbance Storm Time (DST) index for the equatorial ring current or PC for the polar cap). Both types are suitable for orbit propagation and estimation, however long-term and continuously measured indices are preferred for their availability, relative accuracy and continuity. The most common geomagnetic proxies for orbit determination and propagation are the planetary K-class indices (e.g. kp and Ap), which have been introduced in Bartels et al. (1939), Bartels (1949), as well as Bartels and Veldkamp (1953), however also other geomagnetic proxies (e.g. DST in case of JB2008) are sometimes used.

MSISE-90 and NRLMSISE-00 are based on the daily Ap index, if this dependency is not disabled via the internal switch `SW(9)`. The DTM-class models preferably use the interpolated 3 h-delayed kp and the average Kp of the last 24 hours, however also the corresponding km and Km are supported.

Geomagnetic indices are regularly published online. Common data sources at the time of writing are:

- https://celestrak.com/SpaceData
- https://www.gfz-potsdam.de/kp-index
- https://www.swpc.noaa.gov/products-and-data
- http://wdc.kugi.kyoto-u.ac.jp/wdc/Sec3.html
- http://isgi.unistra.fr

The computation of the definitive indices traditionally takes place on a monthly scale with a delay (typically a few days) depending on when the final indices are made available by all observatories (Menvielle, 1998). Therefore, real-time or near real-time orbit estimation is performed with nowcasts (also called quicklooks) and forecasts of the geomagnetic indices.

Geomagnetic indices are subject to station uncertainty, data uncertainty and physical uncertainty (Xu, 2008)

All indices are subject to various types of uncertainty, such as "station uncertainty", "data uncertainty" and "physical uncertainty" (Xu, 2008). Data uncertainty relates to inadequate data processing. This does not only refer to the inherent uncertainty of the quicklook and forecast data, but also describes limitations in the derivation of an index. A common example of data uncertainty is the removal of the solar regular (S_R) and solar quiet (S_q) variations in the computation of the ap/kp indices. The effect of day-to-day variations in S_q on the resulting K-class indices has been analyzed in Xu (2008), who found that the common process of removing the average S_q of five international quite days for each month gives rise to variations in kp as large as $\delta kp = 3$.

Another example of data uncertainty is the publication of discrete K-class values, especially for the logarithmically scaled kp index. Atmospheric models commonly treat magnetic indices as continuous variables, which are sometimes additionally interpolated to a certain point in time. The kp/Kp and corresponding ap/Ap values on the other hand are typically published according to the discrete conversion table, which may be found in Mandea and Korte (2010). Also the quicklooks may vary depending on the data source. The NOAA SWPC index for example only uses 10 planetary K-values (0 to 9), whereas the GFZ indices are expressed in a scale of 1/3 and hence consist of 28 values.

The planetary K-class indices also suffer from station uncertainty, as all observatories, except for two which fall into the same longitude group (Canberra and Eyrewell), are located in the northern hemisphere (Mandea and Korte, 2010). This degrades the planetary validity of the index, as no complete picture of the southern sub-auroral stations is available in the derivation of the kp index.

Obviously also other indices are subject to various sources of uncertainty. The DST index for example suffers from physical uncertainty due to the co-existence of multiple current systems that contribute to the equatorial ring current without a unique separation. Hence, the DST index might increase without a corresponding decrease of the symmetric equatorial ring current due to its dependence on other environmental factors (Mandea and Korte, 2010). Furthermore, multiple ring current configurations may produce the same magnetic field on Earth (Xu, 2008).

Xu (2008) seems to be the only published study which both discusses magnetic index uncertainty and gives quantitative indications of the

resulting magnitude. Motivated by this work, magnetic index errors are subsequently assumed to follow a Gaussian distribution with a standard deviation of 0.5 ap units in geomagnetic amplitude[2]. Also slightly different choices would make sense, however further dedicated research is required to determine a higher-fidelity distribution of the geomagnetic index errors.

2.4.1 *Absolute error, variance and covariance in* $G_i(L)$ *due to uncertain magnetic index information*

To determine the effect of uncertain geomagnetic indices on orbit propagation/estimation, the quantities $Var(\delta G_i)$ and $Cov(\delta G_i, \delta G_j)$ need to be computed, given a standard deviation in geomagnetic amplitude. Due to the different geomagnetic proxies in the MSIS and DTM-class models, each class is treated separately in the following.

MSIS-class models: error/variance propagation from Ap *to* $G_i(L)$
In MSISE-90 and NRLMSISE-00 the relationship between a change in $G_i(L)$ and a change in the daily magnetic amplitude is given by:

$$\delta G_i = (c_{1_i} + c_{2_i} p_{40} + c_{3_i} p_{20} + c_{c_i})\, \delta A(\delta Ap) \tag{2.93}$$

where $A(Ap)$ is given by equation 2.94.

$$\begin{aligned} A(Ap) &= (Ap - 4) + (c_{5_i} - 1)\left[Ap - 4 + \frac{\exp(-c_{4_i}(Ap-4)) - 1}{c_{4_i}}\right] \\ &= Ap + (c_{5_i} - 1)Ap - 4c_{5_i} - \frac{c_{5_i} - 1}{c_{4_i}} + \frac{c_{5_i} - 1}{c_{4_i}} \exp(-c_{4_i} Ap + 4c_{4_i}) \\ &= c_{5_i} Ap + \exp(-c_{4_i} Ap) \underbrace{\frac{c_{5_i} - 1}{c_{4_i}} \exp(4c_{4_i})}_{:= d_i} - 4c_{5_i} - \frac{c_{5_i} - 1}{c_{4_i}} \\ &= c_{5_i} Ap + d_i \exp(-c_{4_i} Ap) - 4c_{5_i} - \frac{c_{5_i} - 1}{c_{4_i}} \end{aligned} \tag{2.94}$$

p_{20} and p_{40} represent Legendre-polynomials and c_{c_i} additional geomagnetic cross-correlation terms in latitude and longitude. c_{1_i}, c_{2_i}, c_{3_i}, c_{4_i} and c_{5_i} are model and species-dependent coefficients (which should not be confused with the solar flux coefficients introduced in section 2.3.1). The index i still denotes the species under consideration, which in a broader sense can be a chemical species (i.e. a gas constituent) or a "temperature species" (i.e. the exospheric temperature or the baseline temperature). The latter case is required to compute $\delta G_{T_{ex}}$ and δG_{T_0}, as well as the corresponding variances.

The parametrization of the coefficients for all species can be obtained from the model code. An exemplary set for cross-referencing is given in the following for exospheric temperature in NRLMSISE-00:

2 1 ap unit approximately corresponds to 2 nT

$c_{1_{T_{ex}}} = 4.93933E{-}03$, $c_{2_{T_{ex}}} = 2.50802E{-}03$, $c_{3_{T_{ex}}} = 5.72562E{-}03$, $c_{4_{T_{ex}}} = 8.47001E{-}02$ and $c_{5_{T_{ex}}} = 1.70147E{-}01$.

To complete equation 2.93, an expression for $\delta A(\delta Ap)$ is required. Following the error definition applied in this thesis ($Ap = \hat{A}p + \delta Ap$), equation 2.94 can be used to obtain:

$$\begin{aligned}\delta A(\delta Ap) &= c_{5_i} Ap - c_{5_i} \hat{A}p + d_i \exp(-c_{4_i} Ap) - d_i \exp(-c_{4_i} \hat{A}p) \\ &= c_{5_i} \delta Ap + \underbrace{d_i \exp(-c_{4_i} \hat{A}p)}_{:=f_i} \exp(-c_{4_i} \delta Ap) - d_i \exp(-c_{4_i} \hat{A}p) \\ &= c_{5_i} \delta Ap + f_i \exp(-c_{4_i} \delta Ap) - f_i \end{aligned} \tag{2.95}$$

Inserting eq. 2.95 into eq. 2.93, results in:

$$\begin{aligned}\delta G_i &= \underbrace{(c_{1_i} + c_{2_i} p_{40} + c_{3_i} p_{20} + c_{c_i})}_{c_i} (c_{5_i} \delta Ap + f_i \exp(-c_{4_i} \delta Ap) - f_i) \\ &= c_i c_{5_i} \delta Ap + c_i f_i \exp(-c_{4_i} \delta Ap) - c_i f_i \end{aligned} \tag{2.96}$$

Equation 2.96 relates an absolute error in Ap to an absolute error in the spherical harmonics function $G_i(L)$. It can now be used to derive the mean and variance of the distribution in $\delta G_i(L)$. Since the relationship between δAp and δG_i is not linear, the third and higher moments are nonzero. Nevertheless, the assumption of Gaussianity will be maintained, as $\hat{A}p$ is expected to be sufficiently close to Ap for the propagated errors to be near-Gaussian. For this moment-matching Gaussian to be a realistic description of the overall uncertainty, δAp needs to be confined to a small neighborhood around $\hat{A}p$, i.e. $\sigma_{\delta_{Ap}}$ is expected to be small in comparison to $\hat{A}p$.

Applying fundamental variance properties (cf. appendix A.1) to equation 2.96, it follows that:

$$\begin{aligned}Var(\delta G_i) = {} & c_i{}^2 c_{5_i}^2 \, Var(\delta Ap) + c_i{}^2 f_i{}^2 \, Var(\exp(-c_{4_i} \delta Ap)) \\ & + 2 c_i{}^2 c_{5_i} f_i \, Cov(\delta Ap, \exp(-c_{4_i} \delta Ap)) \end{aligned} \tag{2.97}$$

Since δAp is assumed to be zero-mean Gaussian, $\exp(-c_{4_i} \delta Ap)$ follows a lognormal distribution. The covariance therefore matches the first moment of the resulting NLNM with a correlation coefficient of -1.0. Using equation A.30 results in:

$$\begin{aligned}Cov(\delta Ap, \exp(-c_{4_i} \delta Ap)) &= E\left[\delta Ap \, e^{-c_{4_i} \delta Ap}\right] \\ &= -c_{4_i} Var(\delta Ap) \, e^{\frac{1}{2} c_{4_i}{}^2 Var(\delta Ap)} \end{aligned} \tag{2.98}$$

After application of the second moment of the lognormal distribution, $Var(\delta G_i)$ is found according to equation 2.99.

$$\begin{aligned}Var(\delta G_i) = {} & c_i{}^2 c_{5_i}{}^2 \, Var(\delta Ap) \\ & + c_i{}^2 f_i^2 e^{\left(c_{4_i}{}^2 Var(\delta Ap)\right)} \left(e^{c_{4_i}{}^2 Var(\delta Ap)} - 1\right) \\ & - 2 c_i{}^2 c_{5_i} c_{4_i} \, f_i Var(\delta Ap) \, e^{\frac{1}{2} c_{4_i}{}^2 Var(\delta Ap)} \end{aligned} \tag{2.99}$$

The first moment of equation 2.96 evaluates to:

$$\mathrm{E}[\delta G_i] = c_i f_i \left(e^{\frac{1}{2} {c_4}_i{}^2 \, \mathrm{Var}(\delta Ap)} - 1 \right) \tag{2.100}$$

which is nonzero. Since however ${c_4}_i$ is small (8.47001E−02 for the exospheric temperature in NRLMSISE-00) and taken to the square, the resulting mean is generally very close to zero, such that the assumption of δG_i being zero-mean Gaussian can be justified in the close neighborhood around $\hat{A}p$. The correctness of this claim can be verified with the aid of the simulation results presented in section 2.4.2.

With the help of equation 2.96, it is now also possible to compute an expression for the inter-species covariance $\mathrm{Cov}(\delta G_i, \delta G_j)$, as required by equation 2.83. Continuing with the assumption of δG_i being zero-mean, the covariance is approximated as the expected value of the product of the random variables:

$$\begin{aligned}
\mathrm{Cov}(\delta G_i, \delta G_j) &\cong \mathrm{E}\left[\delta G_i \delta G_j\right] \\
&= c_i {c_5}_i c_j {c_5}_j \underbrace{\mathrm{E}[\delta Ap^2]}_{\text{Var. of Guassian}} \\
&+ c_i {c_5}_i c_j f_j \underbrace{\mathrm{E}\left[\delta Ap \exp(-{c_4}_j \delta Ap)\right]}_{\text{Normal Lognormal Mixture}} \\
&+ c_i f_i c_j {c_5}_j \underbrace{\mathrm{E}\left[\delta Ap \exp(-{c_4}_i \delta Ap)\right]}_{\text{Normal Lognormal Mixture}} \\
&- c_i f_i c_j f_j \underbrace{\mathrm{E}\left[\exp(-{c_4}_i \delta Ap)\right]}_{\text{Lognormal}} \\
&- c_i f_i c_j f_j \underbrace{\mathrm{E}\left[\exp(-{c_4}_j \delta Ap)\right]}_{\text{Lognormal}} \\
&+ c_i f_i c_j f_j \underbrace{\mathrm{E}\left[\exp(-{c_4}_i \delta Ap) \exp(-{c_4}_j \delta Ap)\right]}_{\text{Lognormal}} \\
&+ c_i f_i c_j f_j
\end{aligned} \tag{2.101}$$

Inserting the corresponding moments of the lognormal distribution and the NLNM (eq. A.30), equation 2.101 can be evaluated to:

$$\begin{aligned}
\mathrm{Cov}(\delta G_i, \delta G_j) \cong \mathrm{Var}(\delta Ap) \Big[& c_i c_j {c_5}_i {c_5}_j - c_i c_j {c_5}_i f_j {c_4}_j e^{\frac{1}{2} {c_4}_j^2 \, \mathrm{Var}(\delta Ap)} \\
& - c_i c_j {c_5}_j f_i {c_4}_i e^{\frac{1}{2} {c_4}_i^2 \, \mathrm{Var}(\delta Ap)} \Big] \\
+ c_i c_j f_i f_j \Big[& e^{\frac{1}{2} ({c_4}_i {c_4}_j)^2 \mathrm{Var}(\delta Ap)} - e^{\frac{1}{2} {c_4}_i^2 \, \mathrm{Var}(\delta Ap)} \\
& - e^{\frac{1}{2} ({c_4}_j)^2 \, \mathrm{Var}(\delta Ap)} + 1 \Big]
\end{aligned} \tag{2.102}$$

Equations 2.99 and 2.102 finally allow to evaluate eq. 2.83 for uncertain daily magnetic amplitudes in case of the MSIS-class models.

DTM-class models: 3 h-delayed kp

The DTM-class models make use of two K-class indices. This section treats the faster index, i.e. the 3 h-delayed kp, as the longer index is more stable. Furthermore, the DTM-models require proxy interpolation, such that published kp (or km) data cannot be used directly, but the corresponding published ap data needs to be interpolated and then converted to kp. Consequently, the error propagation process similarly starts with a zero-mean Gaussian in the (3 h-delayed) magnetic amplitude.

As the relationship between ap and kp is not linear, the resulting error process in kp cannot be Gaussian, meaning that also higher moments have nonzero contribution. If the following conditions however are met, it is possible to estimate a near-Gaussian error process by only considering the resulting first and second moments (moment-matching Gaussian), as the propagation can be considered linear in the neighborhood of $\hat{a}p$:

- $\sigma_{\delta ap}$ must be small in comparison to $\mu_{\delta ap}$
- Due to the logarithmic mapping it must be ensured that $\hat{a}_p > 0$, as the logarithm is only defined for positive arguments

First of all, a continuous function is required to map ap to kp (or similarly Ap to Kp). The exact mapping of the linear magnetic amplitude to the quasi-logarithmic kp/Kp index is only available for discrete values of ap/Ap via conversion tables introduced by Bartels (1949) and Bartels and Veldkamp (1953). Wertz defines the following mapping, which is reported to be accurate to within ~10% (Wertz, 2012):

$$kp = 1.75 \ln ap - 1.6 \tag{2.103}$$

Equation 2.103 is not suitable for error propagation, as its conversion results are too inaccurate for small values of ap. Maximum conversion accuracy at small ap values is however especially important, as errors in kp are more pronounced for small geomagnetic amplitudes, since the knot size of the discrete ap to kp conversion table increases with increasing magnetic activity. Vallado postulates a cubic splining approach for its accuracy and consistency (Vallado, 2013). His algorithm is however also not suitable, as it results in undesired bumps at $ap = 2$ and $ap = 300$, which are

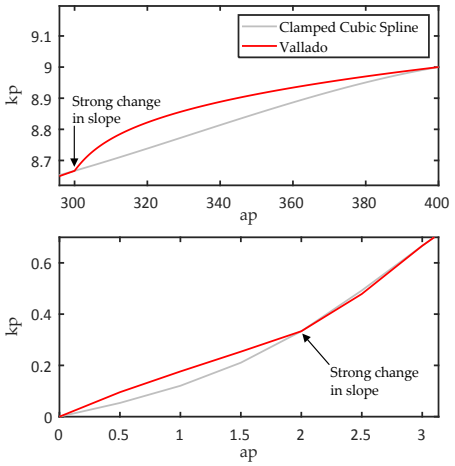


Figure 2.7: Differences of the ap to kp cubic splining algorithm given in Vallado (2013) and the postulated clamped cubic spline (appendix B.1). The Vallado-algorithm shows undesired strong changes in the slope at the outer two ap knots

the consequence of the undefined additional two degrees of freedom in the outer knots when performing the cubic splining (cf. figure 2.7). Vallado's cubic splining approach can be improved by transitioning to a clamped cubic spline with empirically chosen outer bounds to make sure that the transitions at $ap = 2$ and $ap = 300$ remain smooth. The respective coefficients of the third-order polynomials are given in appendix B.1.

Independently of the boundary conditions, a cubic splining approach generally results in a table of third-order coefficients that allow to obtain the matching cubic polynomial for the current value of ap:

$$kp = a_3 ap^3 + a_2 ap^2 + a_1 ap + a_0 \tag{2.104}$$

An absolute error in ap is defined as $\delta ap = ap - \hat{a}p$. Using equation 2.104, the corresponding error in kp is found via:

$$\delta kp = a_3 \delta ap^3 + (3a_3 \hat{a}p + a_2)\delta ap^2 + (3a_3 \hat{a}p^2 + 2a_2 \hat{a}p + a_1)\delta ap \tag{2.105}$$

The associated mean and variance in δkp then allow to formulate a Gaussian approximation of the resulting distribution.

Denoting $c_3 = a_3$, $c_2 = 3a_3 \hat{a}p + a_2$ and $c_1 = 3a_3 \hat{a}p^2 + 2a_2 \hat{a}p + a_1$, the variance in δkp is defined as:

$$\begin{aligned} Var(\delta kp) = {} & c_3^2 \, Var(\delta ap^3) + c_2^2 \, Var(\delta ap^2) + c_1^2 \, Var(\delta ap) \\ & + 2c_3 c_2 \, Cov(\delta ap^2, \delta ap^3) + 2c_3 c_1 \, Cov(\delta ap^3, \delta ap) \\ & + 2c_2 c_2 \, Cov(\delta ap^2, \delta ap) \end{aligned} \tag{2.106}$$

Since δap is assumed to follow a zero-mean Gaussian distribution, the variances and covariances can be evaluated using appendix A.1 and result in:

$$Var(\delta ap^3) = E[\delta ap^6] - E[\delta ap^3]^2 = 15 \, Var(\delta ap)^3 \tag{2.107}$$

$$Var(\delta ap^2) = E[\delta ap^4] - E\left[\delta ap^2\right]^2 = 2 \, Var(\delta ap)^2 \tag{2.108}$$

$$Cov(\delta ap^3, \delta ap) = E[\delta ap^4] - E[\delta ap^3]E[\delta ap] = 3 \, Var(\delta ap)^2 \tag{2.109}$$

$$Cov(\delta ap^3, \delta ap^2) = E[\delta ap^5] - E[\delta ap^3]E[\delta ap^2] = 0.0 \tag{2.110}$$

$$Cov(\delta ap^2, \delta ap) = E[\delta ap^3] - E[\delta ap^2]E[\delta ap] = 0.0 \tag{2.111}$$

Inserting equations 2.107 to 2.111 into eq. 2.106, the variance in δkp is evaluated as:

$$\begin{aligned} Var(\delta kp) = {} & 15 a_3^2 \, Var(\delta ap)^3 \\ & + \left(6a_3[3a_3 \hat{a}p^2 + 2a_2 \hat{a}p + a_1] + 2[3a_3 \hat{a}p + a_2]^2\right) \\ & Var(\delta ap)^2 + (3a_3 \hat{a}p^2 + 2a_2 \hat{a}p + a_1)^2 \, Var(\delta ap) \end{aligned} \tag{2.112}$$

Due to the linearity of expectation, the mean is computed from equation 2.105 as:

$$E[\delta kp] = (3a_3 \hat{a}p + a_2)E[\delta ap^2] = (3a_3 \hat{a}p + a_2)\sigma^2_{\delta ap} \tag{2.113}$$

which is nonzero. A little consideration with the coefficients of table B.1 however shows that for an assumed standard deviation of $\sigma_{\delta ap} = 0.5$ the mean in kp remains below $|0.05|$ and can therefore be neglected in almost all cases. Summarizing:

$$\begin{aligned}\mathcal{N}\left(0.0, \sigma^2_{\delta ap}\right) \overset{\approx}{\rightarrow} \mathcal{N}\big((3a_3\hat{a}p + a_2)\sigma^2_{\delta ap},\ & 15a_3^2\,Var(\delta ap)^3 \\ & + (6a_3[3a_3\hat{a}p^2 + 2a_2\hat{a}p + a_1] + \\ & \quad 2[3a_3\hat{a}p + a_2]^2)\,Var(\delta ap)^2 \\ & + (3a_3\hat{a}p^2 + 2a_2\hat{a}p + a_1)^2\,Var(\delta ap)\big) \quad (2.114)\end{aligned}$$

where the coefficients a_3, a_2 and a_1 are to be taken from the row of table B.1 that belongs to $\hat{a}p$.

Now that the approximate Gaussian in the kp error has been determined, the next step is to relate the error and its uncertainty to an error and its variance in the spherical harmonics of the environmental parameters, i.e. $\delta G_i(\delta kp)$. Both DTM-2012 and DTM-2013 contain the same description of $G(L)$, however use different parametrizations. 15 terms model the effects of the magnetic index inputs to the models. Only six of the terms however change if the 3 h-delayed kp index input changes. The dependency reads:

$$\begin{aligned}\delta G_i(\delta kp) = & (c_{1_i} + c_{2_i}p_{20} + c_{5_i}p_{40})\delta kp \\ & + (c_{3_i} + c_{4_i}p_{20})(kp^2 - \hat{k}p^2) + c_{6_i}(kp^4 - \hat{k}p^4) \quad (2.115)\end{aligned}$$

The c_{x_i}-coefficients are model and species-dependent, meaning that the appropriate values that match the species under study need to be loaded from the matching model coefficients file. p_{20} and p_{40} are Legendre polynomials and represent cross-couplings in geomagnetic activity/latitude and longitude. Resolving the differences in the powers results in:

$$\begin{aligned}\delta G_i(\delta kp) = & (c_{1_i} + c_{2_i}p_{20} + c_{5_i}p_{40} + 2\hat{k}p(c_{3_i} + c_{4_i}p_{20}) + 4\hat{k}p^3 c_{6_i})\,\delta kp \\ & + (c_{3_i} + c_{4_i}p_{20} + 6c_{6_i}\hat{k}p^2)\,\delta kp^2 + 4c_{6_i}\hat{k}p\delta kp^3 + c_{6_i}\delta kp^4 \quad (2.116)\end{aligned}$$

Based on equation 2.116, it is now possible to compute $Var(\delta G_i)$ and $Cov(\delta G_i, \delta G_j)$, which are needed by equation 2.83. To simplify notation, define $b_{1_i} = c_{1_i} + c_{2_i}p_{20} + c_{5_i}p_{40} + 2\hat{k}p(c_{3_i} + c_{4_i}p_{20}) + 4\hat{k}p^3 c_{6_i}$, $b_{2_i} = c_{3_i} + c_{4_i}p_{20} + 6c_{6_i}\hat{k}p^2$, $b_{3_i} = 4c_{6_i}\hat{k}p$ and $b_{4_i} = c_{6_i}$. Applying fundamental variance properties (cf. appendix A.1) to equation 2.116 leads to:

$$\begin{aligned}Var(\delta G_i) = & \, b_{4_i}^2 Var(\delta kp^4) + b_{3_i}^2 Var(\delta kp^3) \\ & + b_{2_i}^2 Var(\delta kp^2) + b_{1_i}^2 Var(\delta kp) \\ & + 2b_{4_i}b_{3_i}Cov(\delta kp^4, \delta kp^3) + 2b_{4_i}b_{2_i}Cov(\delta kp^4, \delta kp^2) \\ & + 2b_{4_i}b_{1_i}Cov(\delta kp^4, \delta kp) + 2b_{3_i}b_{2_i}Cov(\delta kp^3, \delta kp^2) \\ & + 2b_{3_i}b_{1_i}Cov(\delta kp^3, \delta kp) + 2b_{2_i}b_{1_i}Cov(\delta kp^3, \delta kp) \quad (2.117)\end{aligned}$$

For the variance terms it holds that:

$$\begin{aligned}\mathrm{Var}(\delta kp^4) &= \mathrm{E}[\delta kp^8] - \mathrm{E}[\delta kp^4]^2 = 105\,\mathrm{Var}(\delta kp)^4 - 9\,\mathrm{Var}(\delta kp)^2 \\ &= 96\,\mathrm{Var}(\delta kp)^4 \end{aligned} \tag{2.118}$$

$$\mathrm{Var}(\delta kp^3) = 15\,\mathrm{Var}(\delta kp)^3 \tag{2.119}$$

$$\mathrm{Var}(\delta kp^2) = 2\,\mathrm{Var}(\delta kp)^2 \tag{2.120}$$

Similarly, the definition of covariance leads to:

$$\mathrm{Cov}(\delta kp^4, \delta kp^3) = 0.0 \tag{2.121}$$

$$\mathrm{Cov}(\delta kp^4, \delta kp^2) = 15\,\mathrm{Var}(\delta kp)^3 - 3\,\mathrm{Var}(\delta kp)^3 = 12\,\mathrm{Var}(\delta kp)^3 \tag{2.122}$$

$$\mathrm{Cov}(\delta kp^4, \delta kp) = 0.0 \tag{2.123}$$

$$\mathrm{Cov}(\delta kp^3, \delta kp^2) = 0.0 \tag{2.124}$$

$$\mathrm{Cov}(\delta kp^3, \delta kp) = 3\,\mathrm{Var}(\delta kp)^2 \tag{2.125}$$

$$\mathrm{Cov}(\delta kp^2, \delta kp) = 0.0 \tag{2.126}$$

Inserting equations 2.118 to 2.126 into eq. 2.117 allows to obtain a compact description of the variance in $\delta G_i(\delta kp)$:

$$\begin{aligned}\mathrm{Var}(\delta G_i) = {} & 96 b_{4_i}^2 \mathrm{Var}(\delta kp)^4 + (15 b_{3_i}^2 + 24 b_{4_i} b_{2_i})\,\mathrm{Var}(\delta kp)^3 \\ & + (2 b_{2_i}^2 + 6 b_{3_i} b_{1_i})\,\mathrm{Var}(\delta kp)^2 + b_{1_i}^2 \mathrm{Var}(\delta kp)\end{aligned} \tag{2.127}$$

Similarly, the mean of equation 2.116 is found to result in equation 2.128 under the assumption of δkp being effectively zero-mean:

$$\mathrm{E}[\delta G_i] = b_{4_i}\mathrm{E}\left[\delta kp^4\right] + b_{2_i}\mathrm{E}\left[\delta kp^2\right] = 3 b_{4_i}\mathrm{Var}(\delta kp)^2 + b_{2_i}\mathrm{Var}(\delta kp) \tag{2.128}$$

Combining equations 2.127 and 2.128 now allows to formulate the approximate Gaussian in the spherical harmonics function error:

$$\begin{aligned}\mathcal{N}(0.0, \sigma_{\delta kp}^2) \xrightarrow{\approx} \mathcal{N}\big(& 3 b_{4_i}\mathrm{Var}(\delta kp)^2 + b_{2_i}\mathrm{Var}(\delta kp), \\ & 96 b_{4_i}^2 \mathrm{Var}(\delta kp)^4 + (15 b_{3_i}^2 + 24 b_{4_i} b_{2_i})\,\mathrm{Var}(\delta kp)^3 \\ & + (2 b_{2_i}^2 + 6 b_{3_i} b_{1_i})\,\mathrm{Var}(\delta kp)^2 \\ & + b_{1_i}^2 \mathrm{Var}(\delta kp)\big)\end{aligned} \tag{2.129}$$

The model-coefficients c_{x_i} for DTM-2012 and DTM-2013 in the case of exospheric temperature are given in table 2.1. The coefficients of the other species can be inferred form the respective model-coefficients file.

Coefficient	DTM-2012	DTM-2013
$c_{1_{T_{ex}}}$	0.128036E−01	0.124511E−01
$c_{2_{T_{ex}}}$	0.154477E−01	0.163993E−01
$c_{3_{T_{ex}}}$	0.792957E−03	0.184118E−02
$c_{4_{T_{ex}}}$	0.0	0.0
$c_{5_{T_{ex}}}$	0.0	0.0
$c_{6_{T_{ex}}}$	0.457116E−04	0.234802E−04

Table 2.1: Exospheric temperature coefficients for DTM-2012 and DTM-2013

The mean in δG_i is formally nonzero. To test its magnitude/relevance, a short numerical consideration of a prevailing species is required. For the case of exospheric temperature and a couple of representative scenarios the magnitude of the resulting $E[\delta T_{ex}]$ is given in table 2.2.

	$\sigma_{\delta ap} = 0.5$		$\sigma_{\delta ap} = 1.0$	
$\hat{a}p$	DTM-2012	DTM-2013	DTM-2012	DTM-2013
2.0	0.01762	0.03371	0.14341	0.31673
15.0	0.01073	0.01023	0.08867	0.08449
300.0	0.00006	0.00004	0.00057	0.00033

Table 2.2: Expected value of exospheric temperature error due to geomagnetic amplitude uncertainty (DTM-class models)

Table 2.2 indicates that the mean decreases with increasing $\hat{a}p$ and decreasing $\sigma_{\delta ap}$. Only for large $\sigma_{\delta ap}$ and small $\hat{a}p$ it exceeds 0.1K. Hence, the Gaussian approximation of δT_{ex} is also approximately zero-mean.

The last step is to determine the inter-species covariance. As reasoned above, it is assumed that the mean in δG_i can be neglected. Under this assumption the covariance is equal to the expected value of the product of the random variables, which results in:

$$\begin{aligned} Cov(\delta G_i, \delta G_j) \cong E\big[&(b_{4_i}\delta kp^4 + b_{3_i}\delta kp^3 + b_{2_i}\delta kp^2 + b_{1_i}\delta kp) \\ &(b_{4_j}\delta kp^4 + b_{3_j}\delta kp^3 + b_{2_j}\delta kp^2 + b_{1_j}\delta kp)\big] \end{aligned} \tag{2.130}$$

Equation 2.130 is a polynomial in δkp of order eight. Due to the zero-mean assumption the resulting expectations are given by the central moments of the normal distribution (cf. table A.1). These are nonzero in case of even powers of δkp and zero otherwise. Consequently the inter-species covariance for geomagnetic index uncertainties in case of the DTM-class models may be approximated by equation 2.131.

$$\begin{aligned} Cov(\delta G_i, \delta G_j) \cong\ & 105 b_{4_i} b_{4_j} Var(\delta kp)^4 \\ &+ 15(b_{4_i} b_{2_j} + b_{3_i} b_{3_j} + b_{2_i} b_{4_j})\, Var(\delta kp)^3 \\ &+ 3(b_{3_i} b_{1_j} + b_{2_i} b_{2_j} + b_{1_i} b_{3_j})\, Var(\delta kp)^2 \\ &+ b_{1_i} b_{1_j} Var(\delta kp) \end{aligned} \tag{2.131}$$

Equations 2.127 and 2.131, together with eq. 2.112, finally allow to evaluate eq. 2.83 in case of geomagnetic index uncertainty for the DTM-class models.

2.4.2 *Simulation results*

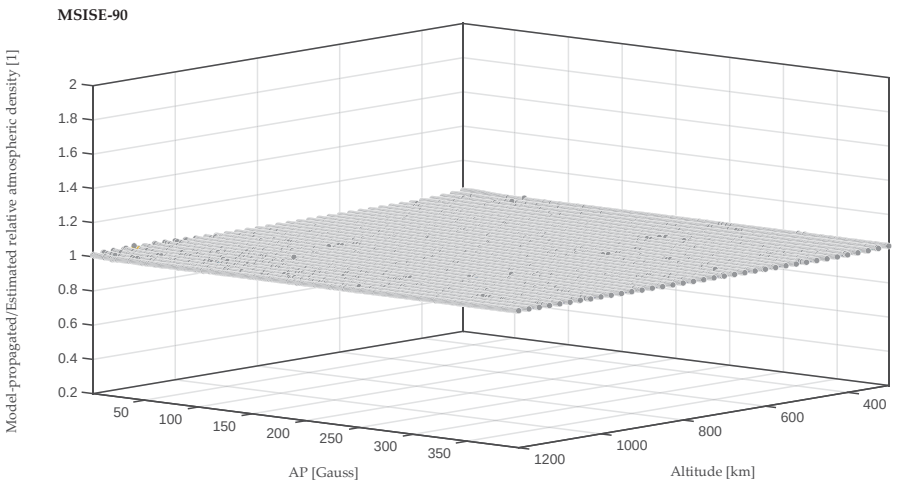
The error and uncertainty propagation equations 2.67 and 2.83 have already been validated, so they can be used to show that the complete

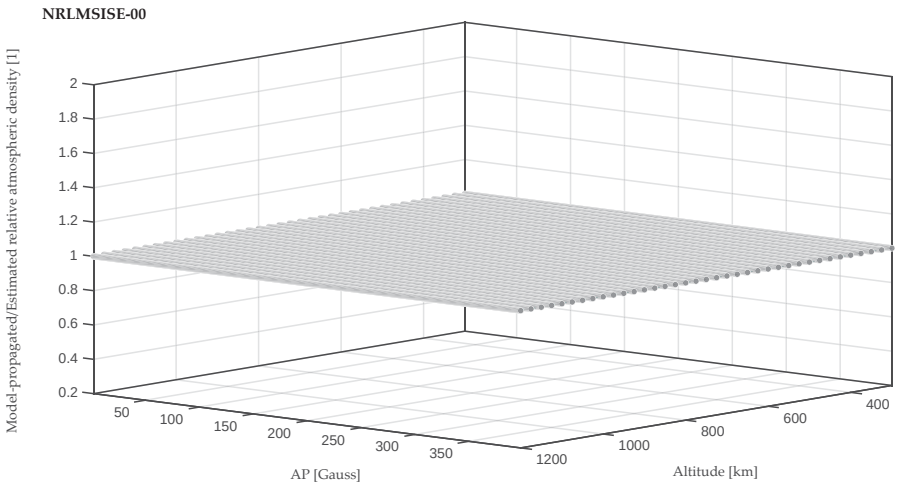
path from $\delta ap/\delta Ap$ to ϵ_ρ is correctly propagated for geomagnetic index uncertainties.

Similar to section 2.3, a geomagnetic activity/altitude grid covering the full ap scale and dominant LEO altitudes ($300 \leqslant z \leqslant 1200\,km$) has been established. The variance estimation is evaluated using $Var(\delta G_i)$ and $Cov(\delta G_i, \delta G_j)$ of the respective atmospheric model and validated with the help of a Monte-Carlo analysis.

ERROR PROPAGATION (EQUATIONS 2.67, 2.96, 2.105 AND 2.115)

Figure 2.8 depicts the ratios $\frac{\epsilon_\rho}{\hat{\epsilon}_\rho}$ over the magnetic amplitude/altitude-grid for MSISE-90, NRLMSISE-00, DTM-2012 and DTM-2013. A value of 1.0 indicates a perfect match of estimation and truth. As a value of 1.0 is obtained basically over the entire grid, no additional tuning is needed.





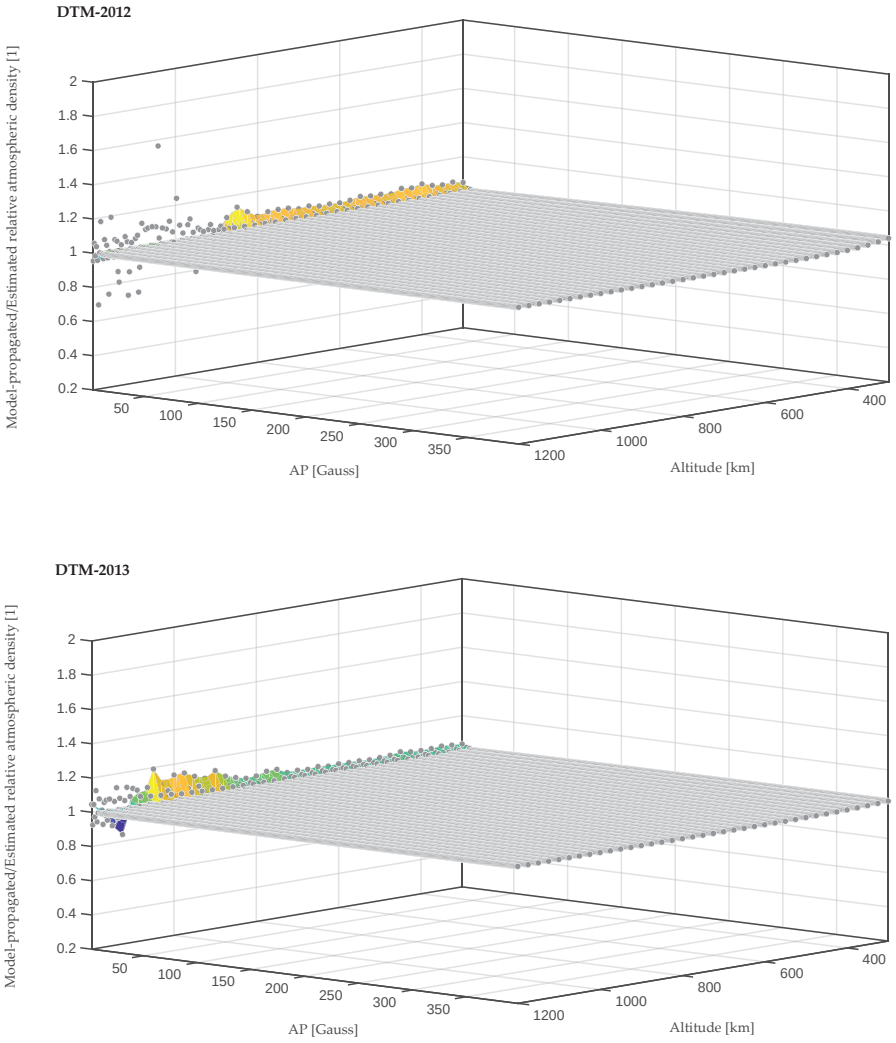


Figure 2.8: Uncertainty quantification of propagated magnetic index errors: MSISE-90, NRLMSISE-00, DTM-2012 and DTM-2013

For the MSIS-class models each grid-point resides at a value of 1.0, which indicates a perfect estimation of the relative density error via equations 2.96 and 2.67. The same also holds true for the DTM-models, except for $\hat{a}p = 2$. For this very low magnetic amplitude, the components of $\hat{e}_\rho$, i.e. $\hat{e}_n$ and $\hat{e}_M$, are of equal magnitude but opposite signs. In the computation of figure 2.8 the product of the relative number and relative mass density errors, as found in equation 2.67, was explicitly taken into account. Nevertheless, small errors in the estimation of $\hat{e}_n$ and $\hat{e}_M$ can cause relative errors in $\hat{e}_\rho$ of multiple 10%, despite the absolute error in the density being small. For practical applications the outliers are not of much concern, as the dominating orbital in-track errors depend on the double integral of the relative density error, which averages out fluctuations over the course of an orbit (Emmert et al., 2017).

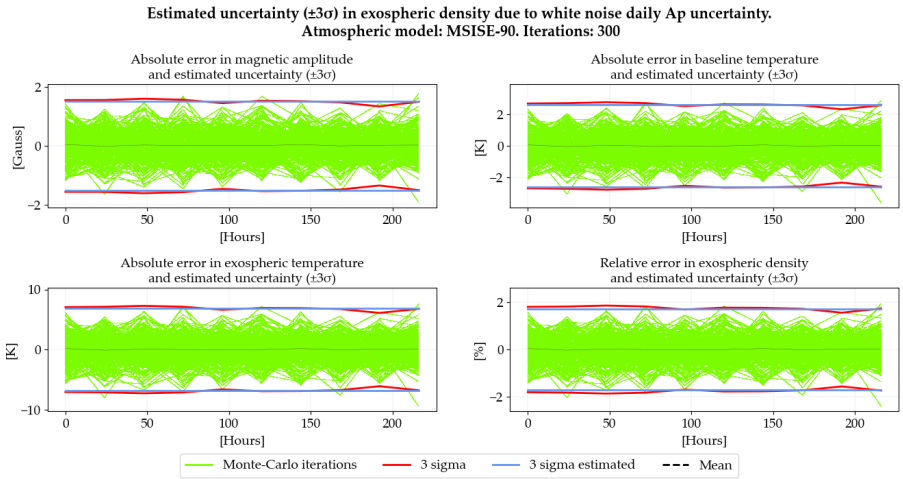
VARIANCE PROPAGATION (EQUATIONS 2.83, 2.99, 2.102, 2.112, 2.127 AND 2.131)

Each point of the grid in figure 2.8 corresponds to a unique altitude/magnetic amplitude combination for which the corresponding variance is estimated via equation 2.83. In the following, the complete variance propagation from geomagnetic amplitude to relative exospheric density is validated for each model using the representative scenarios:

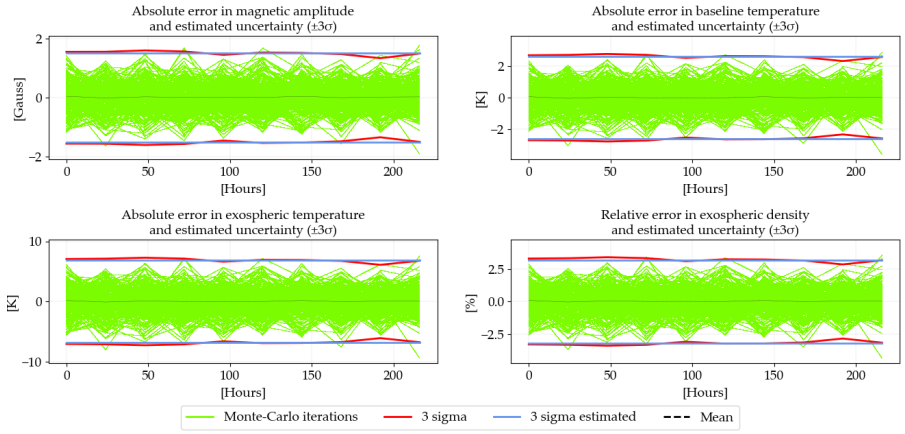
- Altitude: 300 km (low LEO), geomagnetic amplitude: 9 (low)
- Altitude: 500 km (medium LEO), geomag. ampl.: 300 (very high)
- Altitude: 700 km (high LEO), geomag. ampl.: 2 (very low)
- Altitude: 1100 km (very high LEO), geomag. ampl.: 50 (medium)

For each model and case 300 samples are computed. δAp and δap are treated as zero-mean Gaussian random variables. The resulting variance in the relative density error is subsequently compared to the actual spread of the Monte-Carlo iterations. The subplots depict the magnetic amplitude error (upper left), absolute exospheric temperature error (lower left) and the relative exospheric density error in the lower right. The upper right plot depicts the absolute baseline temperature error for the MSIS-class models. In case of DTM-2012 and DTM-2013 the baseline temperature is modeled constant. The upper right subplot thus depicts the variance propagation from the 3 h-delayed magnetic amplitude to the quasi-logarithmic kp index for these models. In each case the blue lines depict the estimated $\pm 3\sigma$ bounds, whereas the red lines give the Monte-Carlo $\pm 3\sigma$ intervals. The MC iterations are plotted in green color and a thin black line depicts the mean of the samples per time-step.

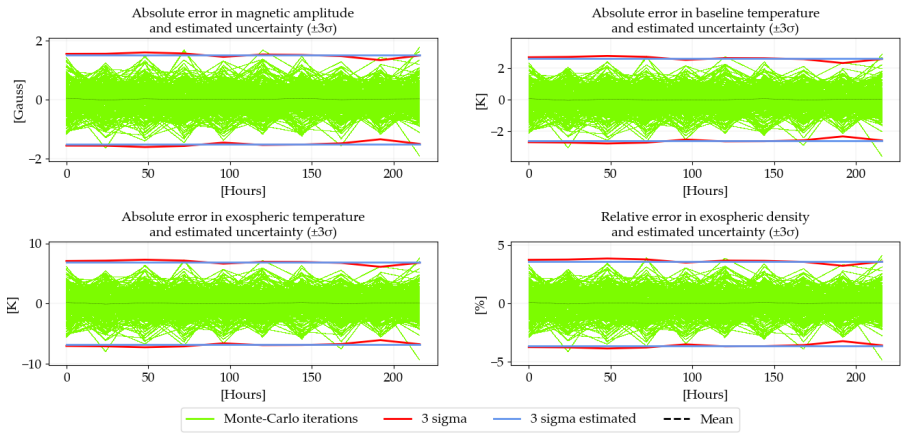
MSISE-90



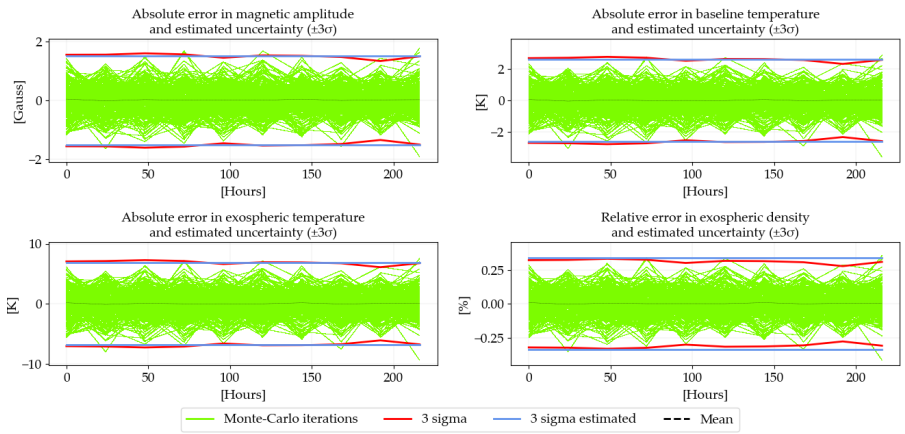
(a) $z = 300$ km, $\hat{\text{Ap}} = 9$



(b) $z = 500\,\mathrm{km}$, $\hat{\mathrm{A}}\mathrm{p} = 300$



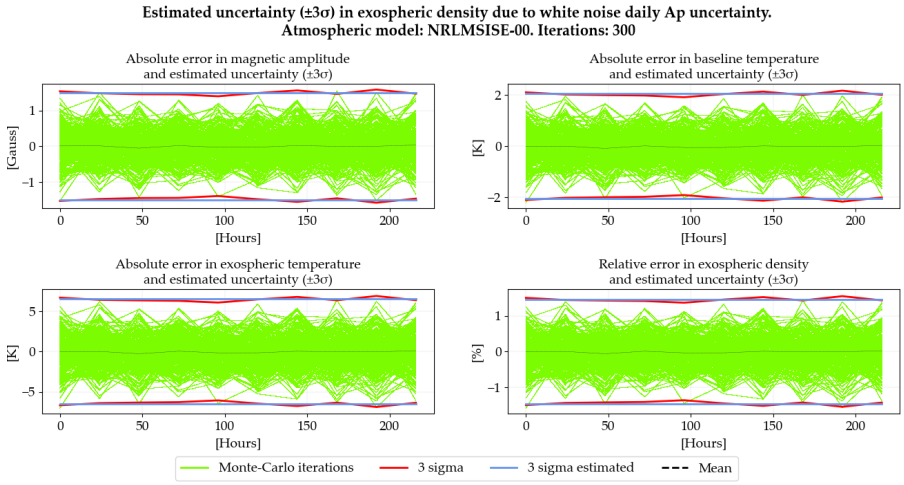
(c) $z = 700\,\mathrm{km}$, $\hat{\mathrm{A}}\mathrm{p} = 2$



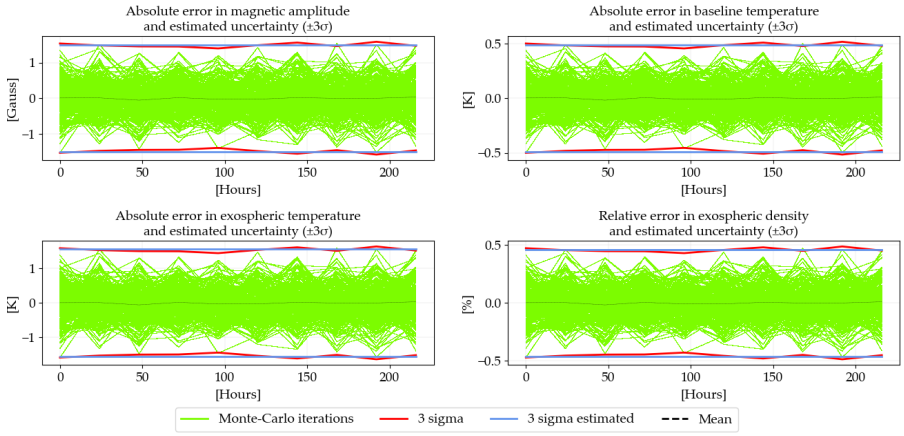
(d) $z = 1100\,\mathrm{km}$, $\hat{\mathrm{A}}\mathrm{p} = 50$

Figure 2.9: Uncertainty quantification of propagated magnetic index errors using MSISE-90

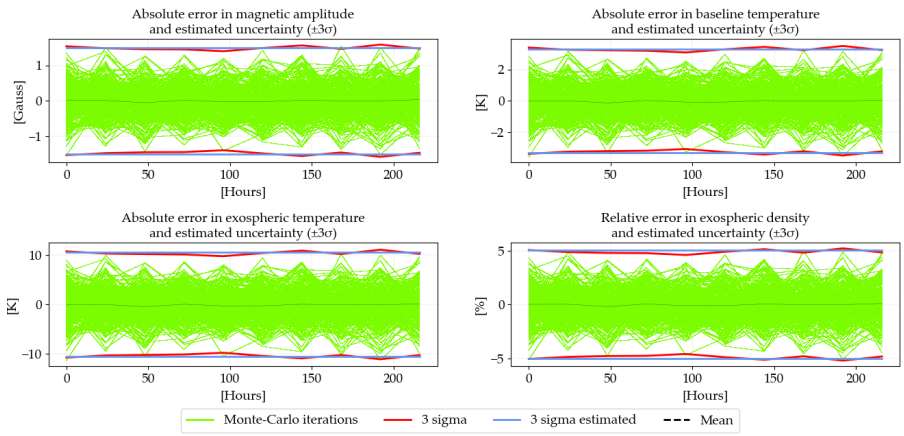
NRLMSISE-00



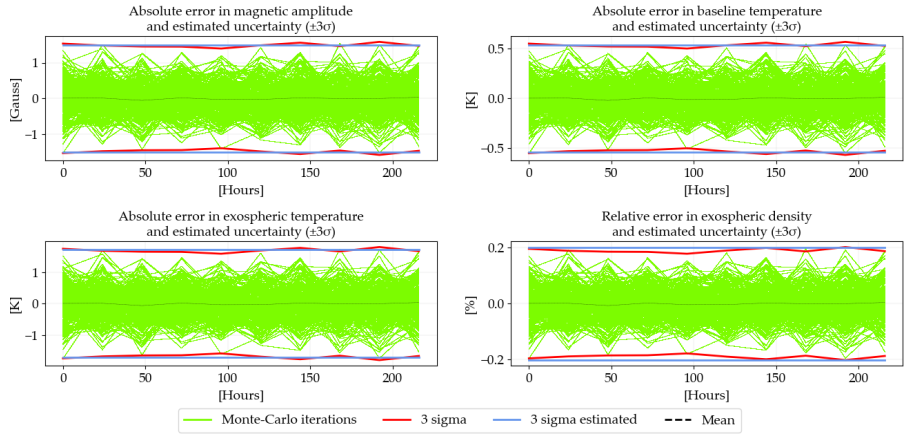
(a) $z = 300\,\mathrm{km}$, $\hat{\mathrm{A}}\mathrm{p} = 9$



(b) $z = 500\,\mathrm{km}$, $\hat{\mathrm{A}}\mathrm{p} = 300$



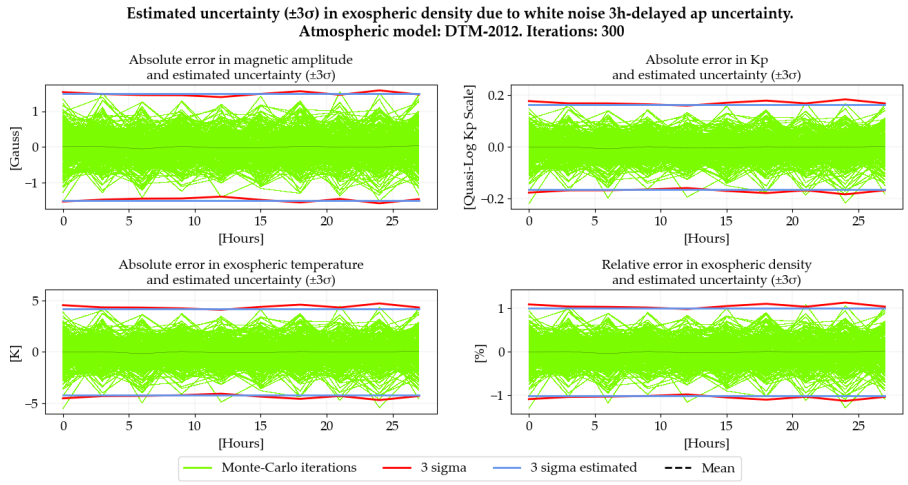
(c) $z = 700\,\mathrm{km}$, $\hat{\mathrm{A}}\mathrm{p} = 2$



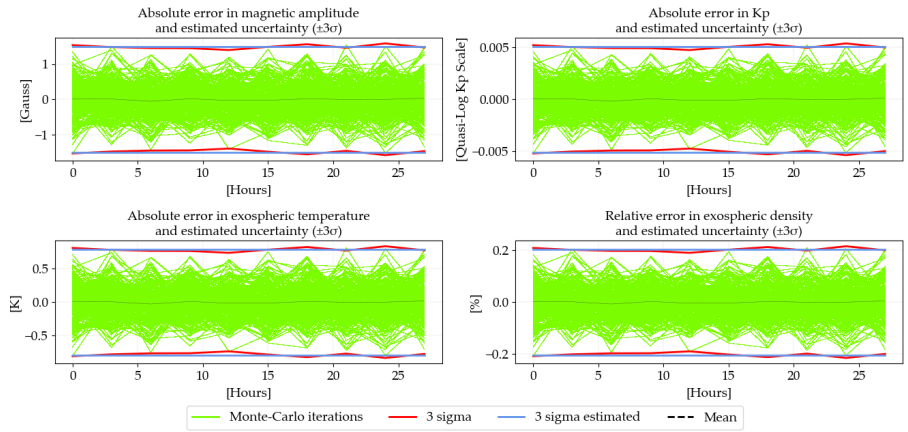
(d) $z = 1100\,\mathrm{km}$, $\hat{\mathrm{A}}\mathrm{p} = 50$

Figure 2.10: Uncertainty quantification of propagated magnetic index errors using NRLMSISE-00

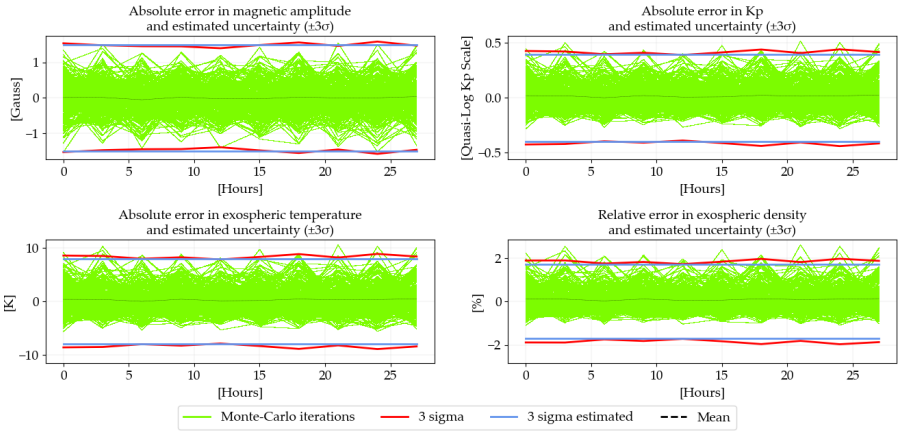
DTM-2012



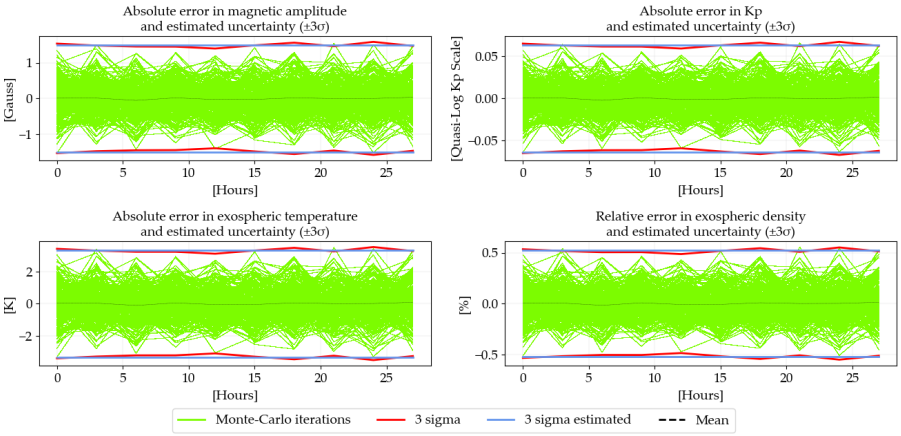
(a) $z = 300\,\mathrm{km}$, $\hat{\mathrm{a}}\mathrm{p} = 9$



(b) $z = 500\,\mathrm{km}$, $\hat{\mathrm{a}}\mathrm{p} = 300$



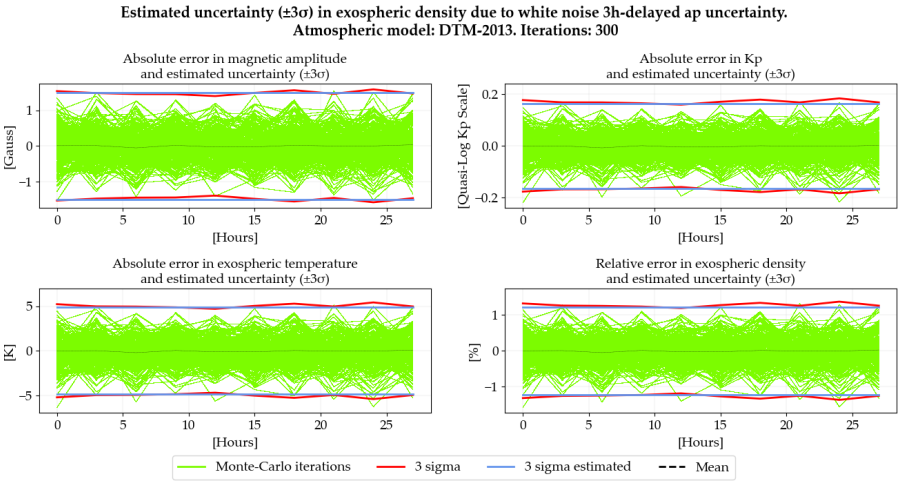
(c) $z = 700\,\mathrm{km}$, $\hat{a}p = 2$



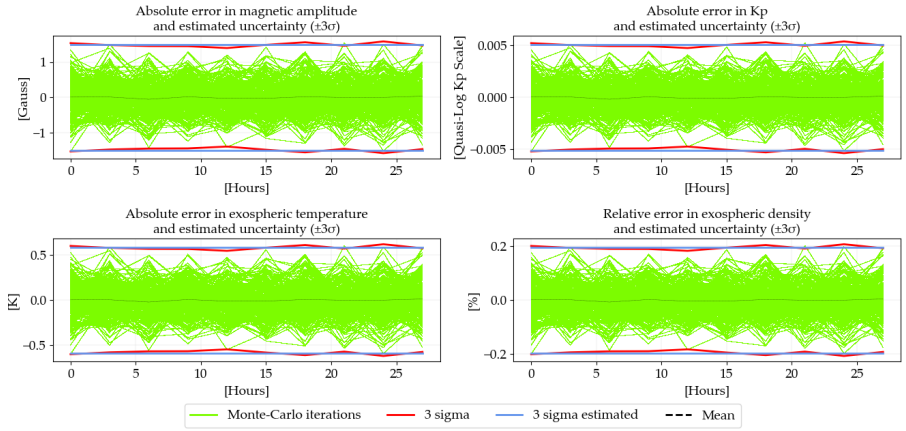
(d) $z = 1100\,\mathrm{km}$, $\hat{a}p = 50$

Figure 2.11: Uncertainty quantification of propagated magnetic index errors using DTM-2012

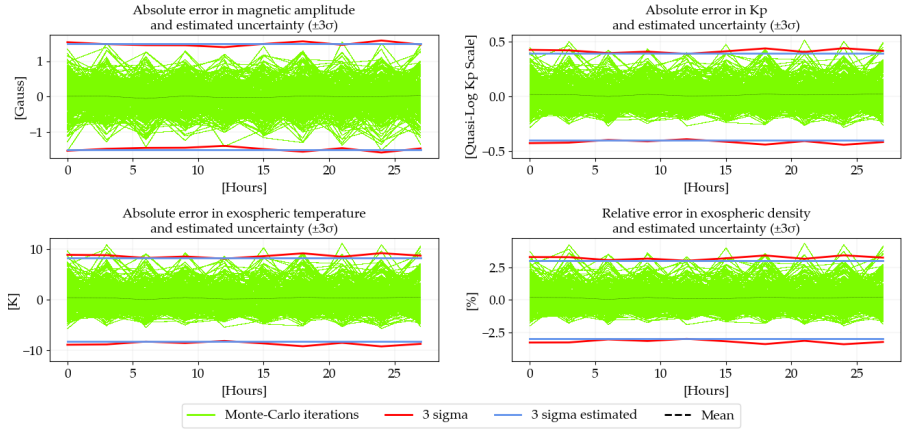
DTM-2013



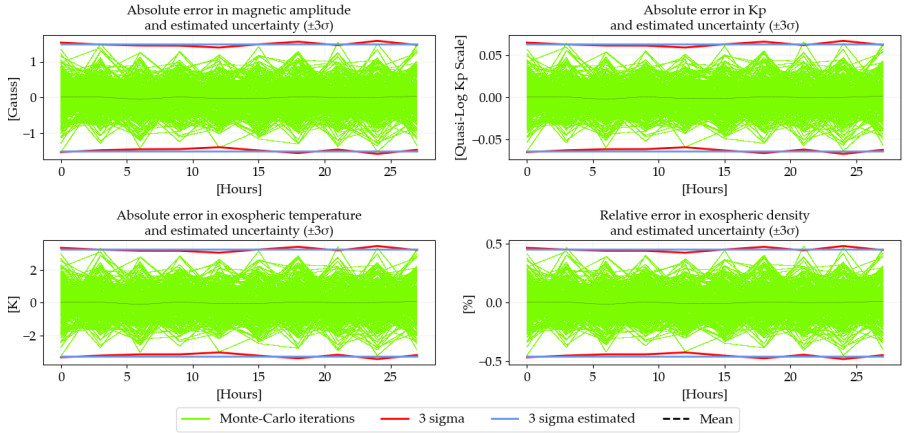
(a) $z = 300\,\mathrm{km}$, $\hat{a}p = 9$



(b) $z = 500\,\mathrm{km}$, $\hat{a}p = 300$



(c) $z = 700\,\mathrm{km}$, $\hat{a}p = 2$



(d) $z = 1100\,\mathrm{km}$, $\hat{a}p = 50$

Figure 2.12: Uncertainty quantification of propagated magnetic index errors using DTM-2013

For both MSISE-90 and NRLMSISE-00 the uncertainty is correctly estimated in all figures, which validates equations 2.99 and 2.102. Also the zero-mean assumptions hold in any case. The resulting relative

density error is largest for small magnetic amplitudes (up to 5% in figure 2.10c).

The DTM-class models show the same behavior, except for the case at 700 km altitude and a 3 h delayed âp of 2. As can be observed from figures 2.11c and 2.12c, the error propagation from magnetic amplitude to quasi-logarithmic kp introduces a slight positive bias and starts to degrade Gaussianity. These properties then propagate to the relative density error. The subsequent variance estimation via equation 2.83 is still accurate, even though it assumes $\delta G_i(L)$ to be zero-mean. The final relative density errors are again largest for small magnetic amplitudes and reach up to 4% in case of figure 2.12c.

Conclusion

With the help of the uncertainty estimation framework presented in sections 2.2-2.4 it is now possible to analytically account for the input uncertainty of the environmental parameters that need to be supplied to semi-empirical atmospheric models when performing orbit propagation/estimation. The need for a tuning-function, as it was necessary in Schiemenz et al. (2019b), has been removed.

For the propagation of geomagnetic index errors it has to be noted that the presented propagation from geomagnetic activity to relative density errors is in its current form most likely only directly applicable to the measurement uncertainty of the geomagnetic activity information. The more challenging case, i.e. geomagnetic forecast uncertainty, is expected to be too large to justify the assumption of a near-linear uncertainty propagation. For this case, nonlinear error/uncertainty propagations need to be considered to capture all moments of the resulting distribution. Modern techniques however exist that build upon Gaussian uncertainty quantification (e.g. Gaussian mixture models), such that the presented equations may be extended to geomagnetic forecast uncertainty.

2.5 PROPAGATION OF RELATIVE DENSITY ERROR TO SATELLITE ORBITS

Atmospheric density errors predominantly affect the mean motion and mean anomaly of a resident space object. To assess the impact of relative density errors on the state vectors of resident space objects, an error and uncertainty propagation from ϵ_ρ to the state $\mathbf{x}$ is required. The derivations are naturally expressed in orbital elements.

2.5.1 *Change of mean motion due to atmospheric drag*

To obtain the differential equation that expresses the secular change of the orbital mean motion due to atmospheric drag, consider the well-known description for the orbital period of elliptical orbits, where

μ is the standard gravitational parameter of Earth, a the semi-major axis of the orbit and T the orbital period:

$$T = 2\pi\sqrt{\frac{a^3}{\mu}} \Leftrightarrow \frac{2\pi}{T} = \sqrt{\frac{\mu}{a^3}} \tag{2.132}$$

The term $\frac{2\pi}{T}$ is the average angular velocity of the orbit and corresponds to the osculating mean motion, as it describes the constant angular speed that a satellite would possess on the circular orbit osculating the elliptical orbit. Denoting mean motion with n, it holds that:

$$n = \sqrt{\frac{\mu}{a^3}} \Leftrightarrow n^2 = \frac{\mu}{a^3} \Leftrightarrow n^2 a^3 = \mu \tag{2.133}$$

Under the influence of drag, both n and a vary with time. The variability is found by differentiating equation 2.133 with respect to time:

$$\begin{aligned} &\left(\frac{d}{dt}n^2\right)a^3 + n^2\left(\frac{d}{dt}a^3\right) = 0 \\ &\Leftrightarrow 2n\frac{dn}{dt}a^3 + n^2 3a^2\frac{da}{dt} = 0 \\ &\Leftrightarrow \frac{dn}{dt} = -\frac{3}{2}\frac{n}{a}\frac{da}{dt} \end{aligned} \tag{2.134}$$

Formula 2.134 represents a differential equation for the orbital mean motion. Once an expression for the semi-major axis change is found, it can be evaluated.

To derive an expression for $\dot{a}$ consider the orbital energy, which consists of both potential and kinetic energy and is also known as specific mechanical energy when remaining independent of mass:

$$E = \frac{mv^2}{2} - \frac{m\mu}{r} = \text{const.} \tag{2.135}$$

Additionally the angular momentum is conserved:

$$\mathbf{h} = \mathbf{r} \times \boldsymbol{v} \tag{2.136}$$

Denoting the perigee with a subscript p, equation 2.136 leads to:

$$h_p = r_p v_p \Leftrightarrow v_p^2 = \frac{h_p^2}{r_p^2} \tag{2.137}$$

Evaluating equation 2.135 at the perigee and inserting eq. 2.137 results in:

$$E = \frac{mv_p^2}{2} - \frac{m\mu}{r_p} = m\left(\frac{h^2}{2r_p^2} - \frac{\mu}{r_p}\right) \tag{2.138}$$

Since $r_p = a(1-e)$, with e denoting orbital eccentricity, this equates to:

$$E = m\left(\frac{h^2}{2a^2(1-e)^2} - \frac{\mu}{a(1-e)}\right) \tag{2.139}$$

To obtain an expression of the squared angular momentum, eq. 2.136 is next evaluated at the perigee, with the orbital velocity being formulated using the vis-viva equation:

$$\begin{aligned} h^2 = v_p^2 r_p^2 &= \left(\frac{2\mu}{r_p} - \frac{\mu}{a}\right) a^2(1-e)^2 \\ &= \left[\frac{2\mu}{a(1-e)} - \frac{\mu(1-e)}{a(1-e)}\right] a^2(1-e)^2 = \mu a(1-e^2) \end{aligned} \tag{2.140}$$

Introducing equation 2.140 into eq. 2.139 yields:

$$E = m\left(\frac{\mu(1-e)(1+e)}{2a(1-e)^2} - \frac{2\mu}{2a(1-e)}\right) = -\frac{m\mu}{2a} \tag{2.141}$$

The orbital energy is constant as long as no perturbations occur. Considering the semi-major axis to be time-varying and differentiating equation 2.141 with respect to time, yields the work done on the body per unit of time:

$$\frac{dE}{dt} = -\frac{m\mu}{2}\frac{d}{dt}\left[a(t)^{-1}\right] = \frac{m\mu}{2a^2}\frac{da}{dt} \tag{2.142}$$

Assuming next that the only force responsible for the work performed is the drag force, it is possible to state:

$$dE = \mathbf{F}_D \cdot d\mathbf{r} \Leftrightarrow \frac{dE}{dt} = \mathbf{F}_D \cdot \boldsymbol{v} = v\, \mathbf{F}_D \cdot \boldsymbol{e}_v \tag{2.143}$$

where $\boldsymbol{e}_v$ is a unit vector pointing in the direction of the speed relative to the atmosphere. Using Newton's second law to transition from force to acceleration yields:

$$\frac{dE}{dt} = \mathbf{F}_D \cdot \boldsymbol{v} = v\, m\, \mathbf{a}_D \cdot \boldsymbol{e}_v \tag{2.144}$$

Inserting equation 2.144 into eq. 2.142 then leads to the time-change of the semi-major axis:

$$\begin{aligned} v\, m\, \mathbf{a}_D \cdot \boldsymbol{e}_v &= \frac{m\mu}{2a^2}\frac{da}{dt} \\ \Leftrightarrow \frac{da}{dt} &= \frac{2a^2}{\mu} v\, \mathbf{a}_D \cdot \boldsymbol{e}_v \end{aligned} \tag{2.145}$$

The drag force always opposes the relative velocity vector. In reality there is also a small lift force perpendicular to the velocity vector, as well as a small turning moment accompanying drag. See King-Hele (1987, section 2.2) for a definition of the lift force, the associated turning moment and its effects on satellite orbits.

For the purpose of propagating uncertainty from relative density errors to orbital state vectors, the lift force has a negligible effect as it is much smaller than the drag-force opposing the velocity vector.

Hence, the dot product can be resolved by introducing a minus sign:

$$\frac{da}{dt} = -\frac{2a^2}{\mu} v \, a_D \tag{2.146}$$

Inserting the magnitude of the drag acceleration from equation 1.5 results in:

$$\frac{da}{dt} = -\frac{2a^2}{\mu} v \frac{1}{2} \rho c_D \frac{A}{m} v^2 = -\frac{a^2}{\mu} \rho B v^3 \tag{2.147}$$

where $B = c_D \frac{A}{m}$ is the ballistic coefficient and v denotes the velocity relative to the atmosphere. Inserting equation 2.147 into eq. 2.134 yields an expression for the impact of drag on the orbital mean motion:

$$\frac{dn}{dt} = \frac{3}{2} n \frac{a}{\mu} \rho B v^3 \tag{2.148}$$

Converting the dependency on the semi-major axis into one on mean motion via eq. 2.133 yields:

$$\frac{dn}{dt} = \frac{3}{2} n^{1/3} \mu^{-2/3} \rho B v^3 \tag{2.149}$$

To treat v as the absolute speed of a resident space object instead of the speed relative to the atmosphere, the relative speed may be expressed as the product of the actual orbital speed and a dimensionless wind-factor F, which models the motion of the atmosphere (e.g. a co-rotating atmosphere). This finally yields:

$$\frac{dn}{dt} = \frac{3}{2} n^{1/3} \mu^{-2/3} \rho B v^3 F \tag{2.150}$$

where v now denotes the orbital speed. For a definition of F in terms of the satellite and atmospheric velocity, see Emmert et al. (2017).

Equation 2.150 represents a separable ordinary differential equation (ODE) describing the secular change in the orbital mean motion due to atmospheric drag. As the change is positive, drag constantly increases the osculating mean motion of an elliptical orbit. The equation serves as the starting point for the work presented in Emmert et al. (2017), which is outlined in the next section.

2.5.2 *State vector error due to relative density error*

The problem of estimating mean motion and mean anomaly errors due to relative density errors has first been solved in Emmert et al. (2017). Based on equation 2.150, the study derives the resulting mean motion and mean anomaly errors and analyzes the corresponding variances for a white noise and Brownian motion error process.

The fundamental principles published in Emmert et al. (2017) are key to the subsequent developments achieved in this thesis. Due to the importance of the final equations and the relevance of the methodology used for deriving the associated white noise and Brownian motion variances, the original derivations given in Emmert et al. (2017) are outlined in this section with respect to the error propagation and section 2.5.3 regarding the variance propagation. Intermediate steps are introduced for ease of understanding.

The following derivations assume a constant ballistic coefficient and a perfectly known wind factor, such that atmospheric density is considered to be the only uncertain variable.

Mean motion error

Since equation 2.150 is separable, it can be solved by integrating from time t_0 to t:

$$\begin{aligned}
&\int_{t_0}^{t} n^{-1/3}\,dn = \frac{3}{2}\mu^{-2/3}B\int_{t_0}^{t}\rho v^3 F\,dt' \\
\Leftrightarrow &\left[\frac{1}{2/3}n^{2/3}\right]_{t_0}^{t} = \frac{3}{2}\mu^{-2/3}B\int_{t_0}^{t}\rho v^3 F\,dt' \\
\Leftrightarrow\ & n(t)^{2/3} = n(t_0)^{2/3} + \mu^{-2/3}B\int_{t_0}^{t}\rho v^3 F\,dt' \\
\Leftrightarrow\ & n(t)^{2/3} = n_0^{2/3}\left[1 + (n_0\mu)^{-2/3}B\int_{t_0}^{t}\rho v^3 F\,dt'\right] \\
\Leftrightarrow\ & n(t) = n_0\left[1 + (n_0\mu)^{-2/3}B\int_{t_0}^{t}\rho v^3 F\,dt'\right]^{3/2}
\end{aligned} \tag{2.151}$$

Emmert et al. next perform a Taylor linearization to drop terms of greater than first order. Note that equation 2.151 can be written as:

$$n(y) = n_0[1+y]^{3/2} \tag{2.152}$$

The zeroth and first derivatives of equation 2.152 with respect to y read:

$$n^{(0)}(y) = n_0[1+y]^{3/2} \tag{2.153}$$

$$n^{(1)}(y) = \frac{3}{2}n_0[1+y]^{1/2} \tag{2.154}$$

Since $n(t)$ only deviates slowly from n_0, it makes sense to evaluate the Taylor series at $y = 0$, which results in:

$$n(y) \cong n_0 + \frac{3}{2} n_0 y = n_0 \left[1 + \frac{3}{2} y \right] \tag{2.155}$$

Consequently the first order Taylor approximation becomes:

$$n(t) \cong n_0 \left[1 + \frac{3}{2} (n_0 \mu)^{-2/3} B \int_{t_0}^{t} \rho v^3 F \, dt' \right] \tag{2.156}$$

From the definition of absolute and relative errors, it follows that:

$$\begin{aligned} &\delta n_0 = n_0 - \hat{n}_0 = \epsilon_{n_0} \hat{n}_0 \\ \Leftrightarrow\ &n_0 = \hat{n}_0 (1 + \epsilon_{n_0}) \end{aligned} \tag{2.157}$$

Inserting equation 2.157 into eq. 2.156 yields:

$$n(t) \cong \hat{n}_0 (1 + \epsilon_{n_0}) \left[1 + \frac{3}{2} \hat{n}_0^{-2/3} (1 + \epsilon_{n_0})^{-2/3} \mu^{-2/3} B \int_{t_0}^{t} \rho v^3 F \, dt' \right] \tag{2.158}$$

The Taylor linearization of $(1 + \epsilon_{n_0})^{-2/3}$ at zero error is $1 - \frac{2}{3} \epsilon_{n_0}$, which results in:

$$n(t) \cong \hat{n}_0 (1 + \epsilon_{n_0}) \left[1 + \frac{3}{2} \hat{n}_0^{-2/3} \left(1 - \frac{2}{3} \epsilon_{n_0} \right) \mu^{-2/3} B \int_{t_0}^{t} \rho v^3 F \, dt' \right] \tag{2.159}$$

Applying equation 2.157 to express the atmospheric density ρ via its estimate and relative error, leads to:

$$n(t) \cong \hat{n}_0 (1 + \epsilon_{n_0}) \left[1 + \frac{3}{2} \hat{n}_0^{-2/3} \left(1 - \frac{2}{3} \epsilon_{n_0} \right) \mu^{-2/3} B \int_{t_0}^{t} \hat{\rho} (1 + \epsilon_\rho) v^3 F \, dt' \right] \tag{2.160}$$

Emmert et al. next define the drag estimate-weighted integral functional $D(x)$:

$$D(x) := \mu^{-2/3} B \int_{t_0}^{t} \hat{\rho} v^3 F \, x \, dt' \tag{2.161}$$

Applying equation 2.161 to eq. 2.160 simplifies the notation:

$$n(t) \cong \hat{n}_0 (1 + \epsilon_{n_0}) \left[1 + \frac{3}{2} \hat{n}_0^{-2/3} \left(1 - \frac{2}{3} \epsilon_{n_0} \right) (D(1) + D(\epsilon_\rho)) \right] \tag{2.162}$$

Next, the products are factored out and double-error terms are dropped:

$$
\begin{aligned}
n(t) &\cong \hat{n}_0 + \frac{3}{2}\hat{n}_0^{1/3}(1 - \frac{2}{3}\epsilon_{n_0})(D(1) + D(\epsilon_\rho)) \\
&\quad + \hat{n}_0\epsilon_{n_0} + \frac{3}{2}\hat{n}_0^{1/3}\epsilon_{n_0}\left(1 - \frac{2}{3}\epsilon_{n_0}\right)(D(1) + D(\epsilon_\rho)) \\
&\cong \hat{n}_0 + \frac{3}{2}\hat{n}_0^{1/3}(D(1) + D(\epsilon_\rho)) - \frac{3}{2}\hat{n}_0^{1/3}\frac{2}{3}\epsilon_{n_0}D(1) \\
&\quad + \hat{n}_0\epsilon_{n_0} + \frac{3}{2}\hat{n}_0^{1/3}\epsilon_{n_0}D(1) \\
&= \hat{n}_0\left[1 + \frac{3}{2}\hat{n}_0^{-2/3}D(1)\right] + \frac{3}{2}\hat{n}_0^{1/3}D(\epsilon_\rho) \\
&\quad - \hat{n}_0^{1/3}\epsilon_{n_0}D(1) + \hat{n}_0\epsilon_{n_0} + \frac{3}{2}\hat{n}_0^{1/3}\epsilon_{n_0}D(1)
\end{aligned}
\tag{2.163}
$$

Double-error product

Dbl.-err.

All terms that contain ϵ_{n_0} are error terms. Grouping accordingly results in:

$$
n(t) \cong \hat{n}_0\left[1 + \frac{3}{2}\hat{n}_0^{-2/3}D(1)\right] + \frac{1}{2}\hat{n}_0^{1/3}\epsilon_{n_0}D(1) + \hat{n}_0\epsilon_{n_0} + \frac{3}{2}\hat{n}_0^{1/3}D(\epsilon_\rho) \tag{2.164}
$$

From equation 2.157 it follows that $\delta_{n_0} = \hat{n}_0\epsilon_{n_0}$. Consequently it is possible to express the error term via the absolute error definition:

$$
n(t) \cong \underbrace{\hat{n}_0\left[1 + \frac{3}{2}\hat{n}_0^{-2/3}D(1)\right]}_{\hat{n}(t)} + \underbrace{\delta_{n_0}\left(1 + \frac{1}{2}\hat{n}_0^{-2/3}D(1)\right) + \frac{3}{2}\hat{n}_0^{1/3}D(\epsilon_\rho)}_{\delta_n(t)} \tag{2.165}
$$

The first term in formula 2.165 represents the mean motion estimate with respect to time. The second and third terms describe the evolution of the absolute mean motion error. $\delta_n(t)$ contains one term proportional to the initial error in the mean motion and another which is proportional to the integral of the relative density error. Uncertainty quantification focuses on the propagation of errors, hence the second term of equation 2.165 is considered further. Writing it as a separate formula, the absolute mean motion error reads:

$$
\delta_n(t) \cong \delta_{n_0}\left(1 + \frac{1}{2}\hat{n}_0^{-2/3}D(1)\right) + \frac{3}{2}\hat{n}_0^{1/3}D(\epsilon_\rho) \tag{2.166}
$$

Defining the quantity $\Delta\hat{n}$ as the estimate of the change of mean motion from its initial value:

$$
\Delta\hat{n}(t) = \hat{n}(t) - \hat{n}_0 \overbrace{=}^{\text{first part of eq. 2.165}} \frac{3}{2}\hat{n}_0^{1/3}D(1) \tag{2.167}
$$

and introducing equation 2.167 into eq. 2.166 yields the following alternative expression of the absolute mean motion error with respect to time:

$$\begin{aligned}
\delta_n(t) &\cong \delta_{n_0}\left(1 + \frac{1}{3}\frac{\hat{n}_0^{1/3}}{\hat{n}_0}\frac{3}{2}D(1)\right) + \frac{3}{2}\hat{n}_0^{1/3}D(\epsilon_\rho) \\
&\cong \delta_{n_0}\left(1 + \frac{1}{3}\frac{\Delta\hat{n}}{\hat{n}_0}\right) + \Delta\hat{n}\frac{D(\epsilon_\rho)}{D(1)} \\
&\cong \delta_{n_0}\left(1 + \frac{1}{3}\frac{\Delta\hat{n}}{\hat{n}_0}\right) + \Delta\hat{n}\frac{\int_{t_0}^{t}\hat{\rho}v^3F\,\epsilon_\rho\,dt'}{\int_{t_0}^{t}\hat{\rho}v^3F\,dt'} \\
&\cong \delta_{n_0}\left(1 + \frac{1}{3}\frac{\Delta\hat{n}}{\hat{n}_0}\right) + \Delta\hat{n}\,E_\rho
\end{aligned} \tag{2.168}$$

While equation 2.168 is already quite concise, it still does not separate the density error process from the orbital characteristics. This separation however is desired as it allows for the relative density errors to be quantified independently of the orbit propagation. Obviously the plan of taking the weighting factor $\hat{\rho}v^3F$ out of the numerator of equation 2.168 requires justified approximations. The weighting factor in E_ρ is largest at perigee, since at this location the orbital speed and the atmospheric density are greatest.

To perform the separation, Emmert et al. introduce the following assumptions:

- ϵ_ρ is a bias-free stochastic process, meaning that any bias in the density model estimate has been removed for the propagation duration of interest
- Variations in ϵ_ρ are uncorrelated with variations in the weighting factor

The approximation proceeds by evaluating the components of the numerator of E_ρ at the altitude/location of the dominant weighting factor, i.e. at perigee. Let $()_p$ denote the evaluation at perigee. Then $\hat{\rho}v^3F \to (\hat{\rho}v^3F)_p$. The relative density error remains along the orbital track, however for consistency it is also evaluated at the perigee altitude $\epsilon_\rho \to \epsilon_{\rho_p}$:

$$\int_{t_0}^{t}\hat{\rho}v^3F\epsilon_\rho\,dt' \to \int_{t_0}^{t}(\hat{\rho}v^3F)_p\,\epsilon_{\rho_p}\,dt'$$

These replacements overestimate the actual numerator, consequently a prefactor needs to be introduced that accounts for the actual perigee contribution along the track. This prefactor is taken as the time average of the weighting factor along the track with respect to its evaluation

at perigee. The resulting scalar scales down the overestimated integral approximately by the actual perigee contribution. Denoting the time-average as $\langle x \rangle = \frac{1}{t-t_0} \int_{t_0}^{t} x(t)\, dt'$, the numerator of E_ρ is approximated as follows:

$$\int_{t_0}^{t} \hat{\rho} v^3 F \epsilon_\rho \, dt' \cong \frac{\langle \hat{\rho} v^3 F \rangle}{\langle (\hat{\rho} v^3 F)_p \rangle} \int_{t_0}^{t} (\hat{\rho} v^3 F)_p \, \epsilon_{\rho_p} \, dt' \tag{2.169}$$

Next, the weighting factor at perigee in the integral is replaced with its time-average, which allows to reduce the prefactor:

$$\begin{aligned} \int_{t_0}^{t} \hat{\rho} v^3 F \epsilon_\rho \, dt' &\cong \frac{\langle \hat{\rho} v^3 F \rangle}{\langle (\hat{\rho} v^3 F)_p \rangle} \int_{t_0}^{t} \langle (\hat{\rho} v^3 F)_p \rangle \epsilon_{\rho_p} \, dt' \\ &= \langle \hat{\rho} v^3 F \rangle \int_{t_0}^{t} \epsilon_{\rho_p} \, dt' \end{aligned} \tag{2.170}$$

Inserting equation 2.170 into eq. 2.168 then results in the desired separation:

$$\begin{aligned} \delta_n(t) &\cong \delta_{n_0} \left(1 + \frac{1}{3} \frac{\Delta \hat{n}}{\hat{n}_0} \right) + \frac{\Delta \hat{n}}{t - t_0} \frac{\int_{t_0}^{t} \hat{\rho} v^3 F \, dt' \int_{t_0}^{t} \epsilon_{\rho_p} \, dt'}{\int_{t_0}^{t} \hat{\rho} v^3 F \, dt'} \\ \Leftrightarrow \delta_n(t) &\cong \delta_{n_0} \left(1 + \frac{1}{3} \frac{\Delta \hat{n}}{\hat{n}_0} \right) + \frac{\Delta \hat{n}}{\Delta t} \int_{t_0}^{t} \epsilon_{\rho_p}(t') \, dt' \end{aligned} \tag{2.171}$$

The absolute mean motion error is hence proportional to the average rate of change of the mean motion and the time-integral of the relative density error at perigee altitude. Emmert et al. note that "*for sufficiently elliptical orbits, only orbit-to-orbit variations in ϵ_{ρ_p} are relevant; that is, variations on shorter time scales should be filtered out*" (Emmert et al., 2017). This statement will become important later, as the input density uncertainty obtained from equation 2.83 yields varying estimates of $Var(\epsilon_{\rho_p})$ throughout the course of an orbit, such that some sort of filtering is recommended for the orbital uncertainty quantification process.

Mean anomaly error

Mean anomaly is defined as the time-integral of mean motion:

$$M(t) = M_0 + \int_{t_0}^{t} n(t') \, dt' \tag{2.172}$$

Introducing the absolute error definition, equation 2.172 may be written as:

$$\begin{aligned} M(t) &= M_0 + \int_{t_0}^{t} \hat{n}(t') + \delta n(t')\,dt' \\ &= \hat{M}_0 + \delta M_0 + \int_{t_0}^{t} \hat{n}(t')\,dt' + \int_{t_0}^{t} \delta n(t')\,dt' \\ &= \underbrace{\hat{M}_0 + \int_{t_0}^{t} \hat{n}(t')\,dt'}_{\hat{M}(t)} + \underbrace{\delta M_0 + \int_{t_0}^{t} \delta n(t')\,dt'}_{\delta M(t)} \end{aligned} \tag{2.173}$$

Equation 2.173 expresses the mean anomaly over time via the mean anomaly estimate (e.g. obtained from an orbit propagator) and the absolute mean anomaly error, which itself consists of an initial mean anomaly error and a drag-induced component. Inserting equation 2.171 into the second term of equation 2.173 yields:

$$\begin{aligned} \delta M(t) &\cong \delta M_0 + \int_{t_0}^{t} \left[\delta n_0 \left(1 + \frac{1}{3} \frac{\Delta\hat{n}(t')}{\hat{n}_0} \right) + \frac{\Delta\hat{n}(t')}{\Delta t'} \int_{t_0}^{t'} \epsilon_{\rho_p}(t'')\,dt'' \right] dt' \\ &= \delta M_0 + \int_{t_0}^{t} \delta n_0 \left(1 + \frac{1}{3} \frac{\Delta\hat{n}(t')}{\hat{n}_0} \right) dt' + \int_{t_0}^{t} \frac{\Delta\hat{n}(t')}{\Delta t'} \int_{t_0}^{t'} \epsilon_{\rho_p}(t'')\,dt''\,dt' \end{aligned} \tag{2.174}$$

Emmert et al. next assume the rate of change in the mean motion to be approximately constant over time, such that it can be taken out of the integral. Under this assumption equation 2.174 may be written as:

$$\delta M(t) \cong \delta M_0 + \delta n_0 \left(\Delta t + \frac{1}{3} \frac{1}{\hat{n}_0} \int_{t_0}^{t} \Delta\hat{n}\,dt' \right) + \frac{\Delta\hat{n}}{\Delta t} \int_{t_0}^{t} \int_{t_0}^{t'} \epsilon_{\rho_p}(t'')\,dt''\,dt' \tag{2.175}$$

The assumption of $\frac{\Delta\hat{n}}{\Delta t}$ being constant can also be used to simplify the integral of the mean motion difference:

Mean motion errors depend on the time-integral of relative density errors. Mean anomaly errors add another layer of integration, such that they are proportional to the twice time-integrated relative density error

$$\int_{t_0}^{t} \Delta\hat{n}\,dt' = \int_{t_0}^{t} \frac{\Delta\hat{n}}{\Delta t} \Delta t\,dt' \cong \frac{\Delta\hat{n}}{\Delta t} \int_{t_0}^{t} \Delta t\,dt' = \frac{\Delta\hat{n}}{\Delta t} \left[\frac{1}{2} \Delta t^2 \right] \tag{2.176}$$

Inserting formula 2.176 into eq. 2.175 then leads to the desired description of the mean anomaly error:

$$\delta M(t) \cong \delta M_0 + \delta n_0 \Delta t \left(1 + \frac{1}{6} \frac{\Delta\hat{n}}{\hat{n}_0} \right) + \frac{\Delta\hat{n}}{\Delta t} \int_{t_0}^{t} \int_{t_0}^{t'} \epsilon_{\rho_p}(t'')\,dt''\,dt' \tag{2.177}$$

Equation 2.177 expresses the absolute mean anomaly error via three terms, of which the first is the initial mean anomaly error, the second relates to initial errors in the mean motion which grow linearly with time and the third is due to atmospheric drag.

Emmert et al. note that the mean anomaly error, which linearly relates to the along-track error, is also accompanied by a radial error which needs to be derived in order to obtain error ellipses. This part is not covered by Emmert et al. (2017) but has received initial treatment in Schiemenz et al. (2019c). An optimized derivation based on orbital element to Cartesian frame conversions is presented in Schiemenz et al. (2020d) and section 2.7.

Equations 2.171 and 2.177 are the fundamental relationships that bridge the gap between relative density errors (as e.g. derived in sections 2.2-2.4) and the resulting impact on the orbital path. In their present form they are not yet directly applicable to orbit determination, however the missing steps are presented in section 2.7.

2.5.3 *State vector uncertainty due to relative density uncertainty*

Emmert et al. (2017, section 2.3) is concerned with uncertainty quantification of equations 2.171 and 2.177. The actual derivations are not given in the paper, but discussed in the accompanying electronic supplement. While leading to the correct results, the original approach does not differentiate between the ideal continuous density error process and its sampled discrete representation, which gives rise to confusion, as the same symbol σ^2 is used for both the continuous double-sided power spectral density (PSD) and the discrete observed variance. The derivation presented in the following corrects for this inaccuracy by clearly differentiating between the quantities.

As argued in the paper, density input uncertainties may be fruitfully treated via white noise and Brownian motion process models. Both derivations are presented in the following. In reality, both models are expected to represent boundary cases of the true experienced density errors.

Preliminaries for the upcoming derivations are the fundamental variance, covariance and expected value properties listed in appendix A.1, as well as equations 2.171 and 2.177. Assuming errors in the relative density error to be independent of initial errors in the orbital elements, it is possible to obtain the mean motion and mean anomaly variances via:

$$\begin{aligned} Var(\delta n(t)) &= Var\left(\delta n_0\left(1+\frac{1}{3}\frac{\Delta\hat{n}}{\hat{n}_0}\right)+\frac{\Delta\hat{n}}{\Delta t}\int_{t_0}^{t}\epsilon_{\rho_p}(t')\,dt'\right) \\ &= \left(1+\frac{1}{3}\frac{\Delta\hat{n}}{\hat{n}_0}\right)^2 Var(\delta n_0)+\left(\frac{\Delta\hat{n}}{\Delta t}\right)^2 Var\left(\int_{t_0}^{t}\epsilon_{\rho_p}(t')\,dt'\right) \end{aligned} \quad (2.178)$$

$$
\begin{aligned}
Var(\delta M(t)) = \; & Var\left(\delta M_0 + \delta n_0 \Delta t\left(1 + \frac{1}{6}\frac{\Delta \hat{n}}{\hat{n}_0}\right) + \frac{\Delta \hat{n}}{\Delta t}\int_{t_0}^{t}\int_{t_0}^{t'} \epsilon_{\rho_p}(t'')\,dt''\,dt'\right) \\
= \; & Var(\delta M_0) + \left(\Delta t\left(1 + \frac{1}{6}\frac{\Delta \hat{n}}{\hat{n}_0}\right)\right)^2 Var(\delta n_0) \\
& + 2\Delta t\left(1 + \frac{1}{6}\frac{\Delta \hat{n}}{\hat{n}_0}\right) Cov(\delta n_0, \delta M_0) \\
& + \left(\frac{\Delta \hat{n}}{\Delta t}\right)^2 Var\left(\int_{t_0}^{t}\int_{t_0}^{t'} \epsilon_{\rho_p}(t'')\,dt''\,dt'\right)
\end{aligned}
\tag{2.179}
$$

The covariance term in equation 2.179 has to be considered if the initial errors in the mean motion and mean anomaly are not independent. If they are, it can be dropped. For practical operations all non-density error related terms are usually not of particular relevance, as they are implicitly treated by covariance propagation techniques listed in figure 1.3. Of dominant interest are therefore the variance contributions of the time-integrated (eq. 2.178) and twice time-integrated (eq. 2.179) density error processes, i.e. $Var\left(\int_{t_0}^{t} \epsilon_{\rho_p}(t')\,dt'\right)$ and $Var\left(\int_{t_0}^{t}\int_{t_0}^{t'} \epsilon_{\rho_p}(t'')\,dt''\,dt'\right)$ for both a white noise and Brownian motion error process model. All four cases are treated in a common manner below, based on the following process model definition introduced in Emmert et al. (2017), which sets $t_0 = 0$:

$$
X_0(t) = \delta t \frac{dX_1(t)}{dt} \Rightarrow \text{dimensionless white noise process} \quad \left(\delta t \frac{dX_1(t)}{dt} = \frac{\delta t}{\delta t}\frac{d}{dt}\int_0^t X_0(s)\,ds = X_0(t)\right) \tag{2.180}
$$

$$
X_1(t) = W(t) = \frac{1}{\delta t}\int_0^t X_0(s)\,ds \Rightarrow \text{dimensionless Wiener process (Brownian motion)} \tag{2.181}
$$

$$
X_2(t) = \int_0^t X_1(s)\,ds \Rightarrow \text{time-dimensional integral of Wiener process} \tag{2.182}
$$

$$
X_3(t) = \int_0^t X_2(s)\,ds \Rightarrow \text{time squared-dimensional double integral of Wiener process} \tag{2.183}
$$

δt is a characteristic time scale and corresponds to the sampling time of the physically continuous processes. Above process model allows to treat both, the white noise and Brownian motion cases in a similar fashion if δt is set to unity, as explained in the following. For the white noise case $X_1(t)$ shall have the dimension of time and has to be modeled as $X_1(t) = \int_0^t X_0(s)\,ds$. $X_0(t)$ is required to be dimensionless. For the Brownian motion case $X_1(t)$ shall be dimensionless, i.e. $X_1(t) = \frac{1}{\delta t}\int_{t_0}^{t} X_0(s)\,ds$. Each integration increases the time-dimensionality by one and the original relative density error process is dimensionless. By setting $\delta t = 1$ both formulations become identical and the cases

may be treated in a similar manner. In a later step the assumption of $\delta t = 1$ is relaxed and the differences are elaborated. For Gaussian process models also the integrated processes are zero-mean, such that the expected value of any process is zero.

It should be noted that above process model defines continuous white noise as the derivative of a Brownian motion process. This is a common and accepted mathematical construct to describe white noise (as Brownian motion may itself be modeled as the time-integral of white noise), however it is not rigorous, since formally continuous Brownian motion is nowhere differentiable and requires an extended derivative interpretation. By defining the Wiener process as the integral of the idealized continuous white noise process, the model is nevertheless applicable and serves the purpose of determining the variances of the integrated density error processes. Also it should be noted that continuous-time white noise (CTWN) is an idealized theoretical construct with infinite variance which practically does not exist. Consequently X_0 is to be understood as band-limited white noise (BLWN).

For BLWN the covariance is given by (Jekeli, 2000, section 6.4):

$$\mathrm{E}\left[X_0(t)X_0(u)\right] = \mathrm{Cov}(X_0(t), X_0(u)) = P_{X_0}\delta(t-u) = P_{X_0}\delta(u-t) \quad (2.184)$$

where P_{X_0} is the constant double-sided PSD and δ is the Dirac delta-function. Using the definition of covariance, the process model for $X_1(t)$ and equation 2.184, the covariance of X_1 may be derived as follows:

$$\begin{aligned}
\mathrm{Cov}(X_1(t), X_1(u)) &= \mathrm{E}\left[\int_0^t X_0(s)\,ds \int_0^u X_0(s')\,ds'\right] \\
&= \int_0^t \int_0^u \mathrm{E}\left[X_0(s)X_0(s')\right] ds'\,ds \qquad (2.185) \\
&= \int_0^u \int_0^t \mathrm{E}\left[X_0(s)X_0(s')\right] ds\,ds' \qquad (2.186)
\end{aligned}$$

Equations 2.185 and 2.186 are equal, as can be seen by substituting $s' \to s$, $s \to s'$, $t \to u$ and $u \to t$. It is hence possible to continue with eq. 2.185 without loss of generality. Inserting equation 2.184 yields:

$$\mathrm{Cov}(X_1(t), X_1(u)) = \int_0^t \int_0^u P_{X_0}\delta(s'-s)\,ds'\,ds \quad (2.187)$$

Note that an integral over the Dirac-delta equates to unity if the domain of integration includes the location of the impulse. Otherwise the integral is zero.

It is easiest to think of the double-integral as a double sum with infinitesimal steps. At each time-step the Dirac impulse is at s, which is bound within $[0, t]$. If at each iteration of the outer sum the inner sum fulfills $s' \geqslant s$, then in the continuous case $u \geqslant t$ and $\mathrm{Cov}(X_1(t), X_1(u)) = \int_0^t P_{X_0}\, ds$. Therefore $\mathrm{Cov}(X_1(t), X_1(u)) = P_{X_0} t$, if $u \geqslant t$. However, if $u \leqslant t$, then there is no contribution for the domain $u \leqslant s \leqslant t$, as only the "sum iterations" where $s' \geqslant s$ contribute to the integral. Consequently the contribution is only from $0 \leqslant s \leqslant u$ and $\mathrm{Cov}(X_1(t), X_1(u)) = P_{X_0} u$, if $u \leqslant t$. Combining both cases yields:

$$\mathrm{Cov}(X_1(t), X_1(u)) = P_{X_0} \min(t, u) \tag{2.188}$$

From equation 2.188 the variance is found by setting $u = t$:

$$\mathrm{Var}(X_1(t)) = P_{X_0} t \tag{2.189}$$

Due to the zero-mean property of the Gaussians, the covariance of the Brownian motion processes equals the expected value of their product. Hence the next level of integration, $\mathrm{Cov}(X_2(t), X_2(u))$, is found as:

$$\begin{aligned}
\mathrm{Cov}(X_2(t), X_2(u)) &= \mathrm{E}\left[\int_0^t X_1(s)\, ds \int_0^u X_1(s')\, ds'\right] \\
&= \int_0^t \int_0^u \mathrm{E}\left[X_1(s) X_1(s')\right] ds'\, ds && (2.190) \\
&\overset{\text{eq. 2.188}}{=} \int_0^t \int_0^u P_{X_0} \min(s, s')\, ds'\, ds && (2.191)
\end{aligned}$$

Again both cases, $u \geqslant t$ and $u \leqslant t$, need to be analyzed separately to evaluate the minimum. As before it helps to think of the double integral as a double sum in ds' and ds.

Case 1: $u \geqslant t$
While the outer sum/integral is at a certain s, the inner sum (which ranges from $0 \leqslant s' \leqslant u$) is first in a domain where $s' \leqslant s$ and then in a section where $s' \geqslant s$. This latter case extends up to $s' = u$. To properly account for the minimum in these parts the inner integral has to be split at $s' = s$:

$$\begin{aligned}
\mathrm{Cov}(X_2(t), X_2(u)) &= \int_0^t \left[\int_0^s P_{X_0} s'\, ds' + \int_s^u P_{X_0} \overbrace{\min(s', s)}^{s}\, ds'\right] ds \\
&= P_{X_0} \int_0^t \left[\int_0^s s'\, ds' + \int_s^u s\, ds'\right] ds = P_{X_0} \left[\int_0^t \frac{1}{2}s^2 + s(u - s)\, ds\right] \\
&= P_{X_0} \left(\frac{1}{6}t^3 - \frac{1}{3}t^3 + \frac{1}{2}t^2 u\right) = P_{X_0} \left(\frac{1}{2}t^2 u - \frac{1}{6}t^3\right) && (2.192)
\end{aligned}$$

Case 2: $u \leqslant t$
Again, while the outer sum/integral is at a certain s, the inner sum ranges from $0 \leqslant s' \leqslant u$. However, since $u \leqslant t$, s' does not reach the values of s where $s \geqslant u$. Consequently $\min(s, s')$ always equals s' in this domain and also the outer integral needs to be split into the sections $[0, u]$ and $[u, t]$:

$$\operatorname{Cov}(X_2(t), X_2(u)) = \int_0^u \int_0^u P_{X_0} \min(s, s')\, ds'\, ds + \int_u^t \int_0^u P_{X_0} \min(s, s')\, ds'\, ds \quad (2.193)$$

Next, also the inner integral of the first term needs to be split to properly evaluate the minimum. The inner integral of the second term does not require splitting, as the lower boundary of the outer integral is equal to the upper boundary of the inner integral and therefore $0 \leqslant s' \leqslant u \leqslant s$:

$$\begin{aligned}
\operatorname{Cov}(X_2(t), X_2(u)) &= P_{X_0} \left[\int_0^u \left(\underbrace{\int_0^s s'\, ds'}_{0 \leqslant s' \leqslant s} + \underbrace{\int_s^u s\, ds'}_{s \leqslant s' \leqslant u} \right) ds + \int_u^t \underbrace{\int_0^u P_{X_0} s'\, ds'}_{0 \leqslant s' \leqslant u \leqslant s}\, ds \right] \\
&= P_{X_0} \left[\int_0^u \frac{1}{2}s^2 + s(u - s)\, ds + \int_u^t \frac{1}{2}u^2\, ds \right] \\
&= P_{X_0} \left(\frac{1}{6}u^3 - \frac{1}{3}u^3 + \frac{1}{2}u^3 + \frac{1}{2}u^2(t - u) \right) \\
&= P_{X_0} \left(\frac{1}{3}u^3 + \frac{1}{2}u^2(t - u) \right) = P_{X_0} \left(\frac{1}{2}u^2 t - \frac{1}{6}u^3 \right) \quad (2.194)
\end{aligned}$$

Combining both cases yields:

$$\operatorname{Cov}(X_2(t), X_2(u)) = P_{X_0} \left(\frac{1}{2} \min(t, u)^2 \max(t, u) - \frac{1}{6} \min(t, u)^3 \right) \quad (2.195)$$

And hence:

$$\operatorname{Var}(X_2(t)) = P_{X_0} \left(\frac{1}{2}t^3 - \frac{1}{6}t^3 \right) = \frac{1}{3} P_{X_0} t^3 \quad (2.196)$$

The Brownian motion case requires a third integration step:

$$\begin{aligned}
\operatorname{Cov}(X_3(t), X_3(u)) &= \int_0^t \int_0^u \mathrm{E}\left[X_2(s) X_2(s') \right] ds'\, ds \\
&= \int_0^t \int_0^u P_{X_0} \left(\frac{1}{2} \min(s', s)^2 \max(s', s) - \frac{1}{6} \min(s', s)^3 \right) ds'\, ds \quad (2.197)
\end{aligned}$$

Both time relationships need to be considered separately again.

Case 1: $u \geqslant t$

As the upper limit of the inner integral is greater than that of the outer, it suffices to split only the inner integral. The splitting has to be performed at $s' = s$. Hence:

$$\begin{aligned}
&\mathrm{Cov}(X_3(t), X_3(u)) \\
&\quad = P_{X_0} \int_0^t \Bigg[\underbrace{\int_0^s \frac{1}{2} s'^2 s - \frac{1}{6} s'^3 \, ds'}_{0 \leqslant s' \leqslant s} + \underbrace{\int_s^u \frac{1}{2} s^2 s' - \frac{1}{6} s^3 \, ds'}_{s \leqslant s' \leqslant u} \Bigg] ds \\
&\quad = P_{X_0} \int_0^t \frac{1}{6} s^3 s - \frac{1}{24} s^4 + \frac{1}{4} s^2 u^2 - \frac{1}{6} s^3 u - \frac{1}{4} s^4 + \frac{1}{6} s^3 s \, ds \\
&\quad = \frac{P_{X_0}}{24} \int_0^t s^4 - 4 s^3 u + 6 s^2 u^2 \, ds \\
&\quad = \frac{P_{X_0}}{24} \left[\frac{1}{5} t^5 - t^4 u + 2 t^3 u^2 \right]
\end{aligned} \tag{2.198}$$

Case 2: $u \leqslant t$

For $u \leqslant t$ the upper limit of the outer integral is greater than the upper limit of the inner integral. Hence it needs to be split at $s = u$ in order to correctly resolve all min / max domains:

$$\begin{aligned}
&\mathrm{Cov}(X_3(t), X_3(u)) \\
&\quad = P_{X_0} \Bigg[\underbrace{\int_0^u \int_0^s \frac{1}{2} s'^2 s - \frac{1}{6} s'^3 \, ds'}_{0 \leqslant s' \leqslant s} + \underbrace{\int_s^u \frac{1}{2} s^2 s' - \frac{1}{6} s^3 \, ds' \, ds}_{s \leqslant s' \leqslant u} + \underbrace{\int_u^t \int_0^u \frac{1}{2} s'^2 s - \frac{1}{6} s'^3 \, ds' \, ds}_{0 \leqslant s' \leqslant u \leqslant s} \Bigg] \\
&\quad = P_{X_0} \left[\int_0^u \frac{1}{6} s^3 s - \frac{1}{6}\frac{1}{4} s^4 + \frac{1}{4} s^2 u^2 - \frac{1}{6} s^3 u - \frac{1}{4} s^4 + \frac{1}{6} s^4 \, ds + \int_u^t \frac{1}{6} u^3 s - \frac{1}{6}\frac{1}{4} u^4 \, ds \right] \\
&\quad = \frac{P_{X_0}}{24} \left[\int_0^u 4 s^4 - s^4 - 6 s^4 + 4 s^4 + 6 s^2 u^2 - 4 s^3 u \, ds + \int_u^t 4 u^3 s - u^4 \, ds \right] \\
&\quad = \frac{P_{X_0}}{24} \left[\frac{6}{5} u^5 + 2 u^3 t^2 - u^4 t - 2 u^5 + u^5 \right] \\
&\quad = \frac{P_{X_0}}{24} \left[\frac{1}{5} u^5 - u^4 t + 2 u^3 t^2 \right]
\end{aligned} \tag{2.199}$$

Combining both cases leads to the generic result:

$$\begin{aligned}
\mathrm{Cov}(X_3(t), X_3(u)) = \frac{P_{X_0}}{24} \Big[&\frac{1}{5} \min(t,u)^5 - \min(t,u)^4 \max(t,u) \\
&+ 2 \min(t,u)^3 \max(t,u)^2 \Big]
\end{aligned} \tag{2.200}$$

Setting $t = u$ then yields:

$$\mathrm{Var}(X_3(t)) = \frac{P_{X_0}}{24} \left[\frac{1}{5} t^5 - t^5 + 2 t^5 \right] = \frac{P_{X_0}}{20} t^5 \tag{2.201}$$

Equation 2.201 describes the variance of the dimensionless Brownian motion process in the mean anomaly-case at time t (with $\delta t = 1$). Table 2.3 summarizes the findings so far. The results are not yet final, as it remains to relax the constraint $\delta t = 1$ and to account for the dimensionality of the processes.

Error	Mean motion	Mean anomaly
White noise	$Var\left(\int_0^t X_0(t')\,dt'\right) = P_{X_0} t$	$Var\left(\int_0^t \int_0^{t'} X_0(t'')\,dt''\,dt'\right) = \frac{1}{3} P_{X_0} t^3$
Brownian motion	$Var\left(\int_0^t X_1(t')\,dt'\right) = \frac{1}{3} P_{X_0} t^3$	$Var\left(\int_0^t \int_0^{t'} X_1(t'')\,dt''\,dt'\right) = \frac{P_{X_0}}{20} t^5$

Table 2.3: Summary of dimensionless variance computations (mean motion error and mean anomaly error)

In reality the continuous density error process can only be observed at discrete times. Hence it is only possible to estimate the properties of the continuous process using discrete observations, which are assumed to take place at a constant sampling time δt (e.g. daily for daily solar flux input errors or 3 h in case of magnetic index errors).

White noise

The covariance of idealized CTWN is given by $Cov(X_0(t), X_0(u)) = P_{X_0}\delta(u - t)$. The double-sided PSD has units of $\frac{\text{Signal}^2}{\text{Hz}}$ and is constant for all frequencies. Due to the observations taking place at δt, a band-limited version of the idealized CTWN is observed. For Gaussian BLWN the relationship between the variance and the sampling is (Jekeli, 2000, eq. 6.61):

$$\sigma^2 = \frac{\text{Double-sided PSD}}{\delta t} \tag{2.202}$$

The PSD is therefore replaced by $P_{X_0} = \sigma^2_{\rho_p}\delta t$ in equation 2.184:

$$Cov(X_0(t), X_0(u)) \cong \sigma^2_{\rho_p}\delta t\ \delta(t - u) = \sigma^2_{\rho_p}\delta t\ \delta(u - t) \tag{2.203}$$

Consequently equation 2.188 becomes:

$$\begin{aligned} Cov(X_1(t), X_1(u)) &\cong \int_0^t \int_0^u \sigma^2_{\rho_p}\delta t\ \delta(s' - s)\,ds'\,ds \\ &= \sigma^2_{\rho_p}\delta t \min(t, u) \end{aligned} \tag{2.204}$$

Comparing equations 2.204 and 2.188 it turns out that for the white noise case the PSD can be replaced by $Var(\epsilon_{\rho_p})\delta t$ in the subsequent integrations to X_2 and X_3. The white noise variances therefore become:

$$\mathrm{Var}\left(\int_0^t \epsilon_{\rho_p}(t')\,dt'\right) = \sigma^2_{\rho_p}\delta t\, t \tag{2.205}$$

$$\mathrm{Var}\left(\int_0^t\int_0^{t'} \epsilon_{\rho_p}(t'')\,dt''\,dt'\right) = \frac{1}{3}\sigma^2_{\rho_p}\delta t\, t^3 \tag{2.206}$$

Brownian motion

In the Brownian motion case $X_1(t)$ needs to be dimensionless, as it corresponds to the relative density error process. Hence the factor $\frac{1}{\delta t}$ needs to be accounted for when computing $\mathrm{Cov}(X_1(t), X_1(u))$. Inserting $\mathrm{Cov}(X_0(t), X_0(u)) \cong \sigma^2_{\rho_p}\delta t\,\delta(u-t)$ into eq. 2.181 results in:

$$\begin{aligned}\mathrm{Cov}(X_1(t), X_1(u)) &= \mathrm{E}\left[\frac{1}{\delta t}\left(\int_0^t X_0(s)\,ds\right)\frac{1}{\delta t}\left(\int_0^u X_0(s')\,ds'\right)\right]\\ &= \frac{1}{\delta t^2}\int_0^t\int_0^u \sigma^2_{\rho_p}\delta t\,\delta(s'-s)\,ds'\,ds\\ &= \frac{\sigma^2_{\rho_p}}{\delta t}\min(t-u)\end{aligned} \tag{2.207}$$

Comparing equations 2.207 and 2.188 then demonstrates that P_{X_0} may be replaced with $\sigma^2_{\rho_p}/\delta t$ in equations 2.189, 2.196 and 2.201 to account for the dimensionality of ϵ_{ρ_p} and its sampling. It should be noted that $\sigma^2_{\rho_p}$ is the constant variance of the fictitious white noise process that drives the Brownian motion. It resembles the variance of the Brownian motion density error process after the first characteristic time interval, which may be obtained as the instantaneous variance divided by the number of sampling steps: $\sigma^2_{\rho_p} = \sigma^2(n\delta t)/n$.

Extending table 2.3 with the proper dimensionality and sampling rate leads to the final results published in Emmert et al. (2017):

Error	Mean motion	Mean anomaly
White noise	$\mathrm{Var}\left(\int_0^t X_0(t')\,dt'\right) = \sigma^2_{\rho_p} t\delta t$	$\mathrm{Var}\left(\int_0^t\int_0^{t'} X_0(t'')\,dt''\,dt'\right) = \frac{1}{3}\sigma^2_{\rho_p} t^3\delta t$
Brownian motion	$\mathrm{Var}\left(\int_0^t X_1(t')\,dt'\right) = \frac{1}{3}\frac{\sigma^2_{\rho_p}}{\delta t} t^3$	$\mathrm{Var}\left(\int_0^t\int_0^{t'} X_1(t'')\,dt''\,dt'\right) = \frac{1}{20}\frac{\sigma^2_{\rho_p}}{\delta t} t^5$

Table 2.4: Summary of variance computations accounting for dimensionality and sampling (mean motion error and mean anomaly error)

Combining the results of table 2.4 with equations 2.178 and 2.179 yields the full expressions of the mean motion variance and mean anomaly variance in case of a white noise or Brownian motion input error process:

Error	Mean motion	Mean anomaly
White noise	$\mathrm{Var}(\delta n(t)) = \left(1+\frac{1}{3}\frac{\Delta\hat{n}}{\hat{n}_0}\right)^2 \mathrm{Var}(\delta n_0) + \left(\frac{\Delta\hat{n}}{\Delta t}\right)^2 \sigma^2_{\rho_p} t\delta t$ (2.208)	$\mathrm{Var}(\delta M(t)) = \left[\mathrm{Var}(\delta M_0) + \left(\Delta t\left(1+\frac{1}{6}\frac{\Delta\hat{n}}{\hat{n}_0}\right)\right)^2 \mathrm{Var}(\delta n_0) + 2\Delta t\left(1+\frac{1}{6}\frac{\Delta\hat{n}}{\hat{n}_0}\right)\mathrm{Cov}(\delta n_0, \delta M_0) + \frac{1}{3}\left(\frac{\Delta\hat{n}}{\Delta t}\right)^2 \sigma^2_{\rho_p} t^3\delta t\right]$ (2.209)
Brownian motion	$\mathrm{Var}(\delta n(t)) = \left(1+\frac{1}{3}\frac{\Delta\hat{n}}{\hat{n}_0}\right)^2 \mathrm{Var}(\delta n_0) + \frac{1}{3}\left(\frac{\Delta\hat{n}}{\Delta t}\right)^2 \frac{\sigma^2_{\dot{\rho}_p}}{\delta t} t^3$ (2.210)	$\mathrm{Var}(\delta M(t)) = \left[\mathrm{Var}(\delta M_0) + \left(\Delta t\left(1+\frac{1}{6}\frac{\Delta\hat{n}}{\hat{n}_0}\right)\right)^2 \mathrm{Var}(\delta n_0) + 2\Delta t\left(1+\frac{1}{6}\frac{\Delta\hat{n}}{\hat{n}_0}\right)\mathrm{Cov}(\delta n_0, \delta M_0) + \frac{1}{20}\left(\frac{\Delta\hat{n}}{\Delta t}\right)^2 \frac{\sigma^2_{\dot{\rho}_p}}{\delta t} t^5\right]$ (2.211)

Table 2.5: Variance of mean motion and mean anomaly error due to a relative density error process following a white noise or random walk process

The symbols reflect the following quantities:

- $\hat{n}$: estimated mean motion
- δn: absolute mean motion error
- $\hat{n}_0$: estimated mean motion at the epoch when the density error process is considered to start
- δM: absolute mean anomaly error
- $\hat{M}_0$: estimated mean anomaly at the start of the density error process
- t: time since the start of the density error process
- δt: interval at which the ideal continuous error process is sampled
- Δt: time since the start of the analysis ($\Delta t \leqslant t$)
- $\Delta\hat{n}$: estimated absolute change in mean motion since the start of the analysis

If only the portion of the uncertainty which is attributed to the relative density error is considered, the relationships simplify as follows:

Error	Mean motion	Mean anomaly
White noise	$\mathrm{Var}(\delta n(t)) = \left(\frac{\Delta\hat{n}}{\Delta t}\right)^2 \sigma^2_{\rho_p} t\,\delta t$ (2.212)	$\mathrm{Var}(\delta M(t)) = \frac{1}{3}\left(\frac{\Delta\hat{n}}{\Delta t}\right)^2 \sigma^2_{\rho_p} t^3\delta t$ (2.213)
Brownian motion	$\mathrm{Var}(\delta n(t)) = \frac{1}{3}\left(\frac{\Delta\hat{n}}{\Delta t}\right)^2 \frac{\sigma^2_{\dot{\rho}_p}}{\delta t} t^3$ (2.214)	$\mathrm{Var}(\delta M(t)) = \frac{1}{20}\left(\frac{\Delta\hat{n}}{\Delta t}\right)^2 \frac{\sigma^2_{\dot{\rho}_p}}{\delta t} t^5$ (2.215)

Table 2.6: Mean motion and mean anomaly variance growth due to a white noise or random walk relative density error process

The final variance results of tables 2.5 and 2.6 can be used together with the derivations of sections 2.2-2.4 to establish an analytic covariance matrix that describes Gaussian zero-mean atmospheric input uncertainty. The original error variance equations (2.178 and 2.179) are furthermore also applicable to other zero-mean Gaussian error processes, such as the Ornstein-Uhlenbeck error process used in the next section to model grid-scale density errors.

2.6 GRID-SCALE DENSITY MODEL UNCERTAINTY

While semi-empirical models are able to compute an estimate for any possible location around Earth, their internal resolution is limited to spatial scales of approximately 4000 km (Schiemenz et al., 2020b). Density variability with wave-lengths below this horizontal scale is therefore below the resolution of semi-empirical models and hence classified as subgrid-scale uncertainty. Bruinsma and Forbes (2008) showed that short-term fluctuations can yield one sigma standard deviations in neutral atmospheric density of >30%. As has been determined by Anderson et al. (2009) however, their integrated effect on the orbit turns out to be in the meter-level for near-circular low Earth orbits. Therefore subgrid-scale uncertainty is negligible in most practical applications.

Grid-scale density uncertainty on the other hand is typically the dominant source of thermospheric density uncertainty. Exceptions to this claim are situations in which a model-debiasing technique like HASDM (Storz et al., 2005) is available and the orbit forecasting requires an extrapolation of the space-weather parameters over multiple days (in this case input uncertainty is dominating).

Modern semi-empirical atmospheric models are constructed from least-squares fits of observational and drag-derived density data. The average one sigma accuracy of recent models is believed to be between 10% and 15%, given perfect knowledge of the model inputs (Vallado and Finkleman, 2014). As an example, the JB2008 model claims to have a mean one sigma accuracy of 10% at 400 km altitude and the recent releases of the DTM model series (DTM-2012 and DTM-2013) both claim to be overall more accurate than JB2008 (Bruinsma, 2013, 2015). Nevertheless the one sigma interval given by Vallado and Finkleman still represents a reasonable uncertainty perimeter for low Earth orbits, where depending on the model and altitude, the value may be closer to 10% or 15%.

Model-scale density variability has been studied by a variety of authors since the 1960s. In 1962 Kenneth Moe analyzed the effect of drag on artificial satellites by separating the effect of orbital drag into a sinusoidal and random component (Moe, 1962). Three years later H.E. Rauch worked on the estimation of satellite trajectories including random drag fluctuations which have been modeled as a first-order Gauss-Markov process, also known as Ornstein-Uhlenbeck Process (OUP) (Rauch, 1965). Rauch noted that the OUP can be fruitfully applied to the analysis of stochastic density variability, however he also pointed out that the stochastic model may be improved to increase accuracy. Also Wilkins and Alfriend stated that grid-scale density perturbations can be modeled by stationary, first-order Gauss-Markov stochastic processes (Wilkins and Alfriend, 2000). In their work they compared the results of a batch estimator with that of a Kalman Filter

which considered the time-correlated relative density error as seventh state vector element.

Recently Sagnières and Sharf made use of the Ornstein-Uhlenbeck process to study the effect of inter-model variability. In Sagnières and Sharf (2017) they present a modified version of the classical OUP which is expressed directly in atmospheric density (in contrast to Wilkins and Alfriend, who formulated the OUP in the relative density error). Sagnières and Sharf explicitly note that their modified Ornstein-Uhlenbeck process can be used to study both inter-model variability, as well as (grid-scale) model-inherent uncertainty (Sagnières and Sharf, 2017). Their modified OUP for atmospheric density is also used in this thesis to simulate truth samples of neutral atmospheric density in a Monte-Carlo analysis.

An important aspect of the Sagnières and Sharf modification is that it maintains the stochastic property of the classical OUP variance to converge towards a long-term variance ($s_t^2 = \frac{\sigma_t^2}{2\theta}$). When studying atmospheric model density uncertainty, the long-term variance (s_t^2) is to be set to the claimed model accuracy (~10%—15%). θ is a parameter of the OUP and model-independent. It can hence be kept constant. Consequently, increasing θ also increases the instantaneous volatility σ, whereas smaller θ result in longer deviations from the model estimates and therefore represent worst-case scenarios. The influence of this parameter is further discussed in section 2.6.1, when the orbital uncertainty approximations are derived and validated.

The stochastic differential equation of the Ornstein-Uhlenbeck process (eq. 2.216) needs to be solved concurrently with the orbit propagation to generate truth samples of density. A simple, yet effective approach using the Milstein algorithm is presented in Sagnières and Sharf (2017). Caution is required in the implementation of this technique when using solvers with variable step size, as this class of solvers adaptively controls the integration-step of the numerical integration. This results in iterations where the posterior step-size check requires a reevaluation of the previous iteration, as the step-size has been too large to maintain the required relative integration accuracy. Consequently time is reset and the step is recomputed with a smaller step-size. These situations must be detected and treated accordingly for the Milstein-algorithm to work with modern variable step-size integrators.

As demonstrated in Emmert et al. (2017) and section 2.5, relative density uncertainty needs to be integrated twice to estimate its effect on the satellite state vector. The first integration yields the impact on the orbital (mean) mean motion and the second integration on the orbital mean anomaly. Consequently any consideration of grid-scale density uncertainty that builds upon realistic covariance matrices to model the resulting uncertainty in the state vector, instead of appending elements to it, requires expressions for the variance and covariance of

the time-integrated and twice time-integrated OUP. These derivations are the subject of section 2.6.1.

2.6.1 *Variance and covariance of the time-integrated Ornstein-Uhlenbeck process*

The numerical simulations which are used to validate the derivations of this section utilize the modified Ornstein-Uhlenbeck process by Sagnières and Sharf. Their atmospheric density-adjustment of the classical time-dependent OUP stochastic differential equation (SDE) is given in equation 2.216.

$$d\rho_t = \theta(\mu_t - \rho_t)\,dt + \sigma_t\,dW_t + d\mu_t + \frac{\rho_t - \mu_t}{s_t}\,ds_t \tag{2.216}$$

The symbols have the following meaning:

- ρ_t is the time-dependent *true* density as computed by the OUP
- μ_t is the time-dependent density estimate as computed by the atmospheric model
- θ is the rate of mean reversion of the OUP
- σ_t is the time-dependent instantaneous standard deviation
- s_t is the long-term standard-deviation of the OUP and corresponds to the claimed model accuracy
- W_t is the Wiener process

The third and fourth term of the right-hand side represent the Sagnières and Sharf extensions to the classical Ornstein-Uhlenbeck process for an improved description of atmospheric density. These changes introduce a tendency to follow the time-varying mean of the density process (third term) and scale the strength of the density fluctuations about the mean to be in agreement with the long-term standard deviation of the classical OUP (fourth term). The effect of each modification is visualized in Sagnières and Sharf (2017, figure 3).

COVARIANCE OF TIME-INTEGRATED OUP

The following computations consider a standard (Wiener-driven) OUP in the relative density error in order to be compatible with the orbital error approximations of section 2.5. In Sagnières and Sharf (2017) the authors note that the initial value of the OUP is to be drawn from a Gaussian distribution with the current density estimate as the mean and the square root of the long-term OUP variance as the standard deviation. Transferring this concept to an OUP in the relative density error, the initial state is to be drawn from a Gaussian distribution with mean zero and the square root of the long-term OUP variance as standard deviation.

The following uncertainty propagation equations model the OUP in the relative density error, which allows to set $\mu = 0$. This corre-

sponds to the case of an unbiased atmospheric model, as has also been assumed in the derivations of equations 2.178 and 2.179. The randomization of the initial state impacts the orbital error approximations, however it leaves the expected value of the OUP at zero. As shown later, it is in fact possible to first consider a standard OUP without randomization of the initial state (the initial value is set to zero) and to correct for the initial randomization afterwards.

In the following the covariances and variances of the time-integrated and twice time-integrated Ornstein-Uhlenbeck processes are derived to complete equations 2.178 and 2.179 for the case of grid-scale model uncertainty. The methodology is similar to section 2.5.3.

Let $Y_t = \int_0^t X_s \, ds$, where X_s is the Ornstein-Uhlenbeck process. The autocovariance of Y_t is then computed as:

$$\mathrm{Cov}(Y_t, Y_u) = \mathrm{Cov}\left(\int_0^t X_s \, ds, \int_0^u X_{s'} \, ds' \right) \tag{2.217}$$

As reasoned earlier, $\mathrm{E}[X_t] = 0$. Hence the covariance equals the expectation of the product of the stochastic processes:

$$\mathrm{Cov}(Y_t, Y_u) = \mathrm{E}\left[\int_0^t \int_0^u X_s X_{s'} \, ds' \, ds \right] \tag{2.218}$$

Using Fubini's theorem, the expectation operator can be put into the integral, whereupon it becomes apparent that the autocovariance of Y_t equals the double-integrated autocovariance of X_t:

$$\begin{aligned} \mathrm{Cov}(Y_t, Y_u) &= \int_0^t \int_0^u \mathrm{E}\left[X_s X_{s'} \right] ds' \, ds \\ &= \int_0^t \int_0^u \mathrm{Cov}(X_s, X_{s'}) \, ds' \, ds \end{aligned} \tag{2.219}$$

The autocovariance of the standard OUP is known to be given by equation 2.220 (Åkesson and Lehoczky, 1998):

$$\mathrm{Cov}(X_t, X_u) = \frac{\sigma^2}{2\theta} \left(e^{2\theta \min(u,t) - \theta(u+t)} - e^{-\theta(u+t)} \right) \tag{2.220}$$

After inserting equation 2.220 into eq. 2.219, it becomes obvious that the domains of integration need to be split to correctly evaluate the min-expression. In general, two cases need to be considered: $u \geqslant t$ and $t \geqslant u$. Both match for $t = u$.

Case 1: $u \geqslant t$
If $u \geqslant t$ the inner integral needs to be split at $s' = s$. Then $s' < s$ in the first term and $s' > s$ in the second, which allows to evaluate the minimum when introducing equation 2.220:

$$\begin{aligned}\mathrm{Cov}(Y_t, Y_u) &= \frac{\sigma^2}{2\theta}\int_0^t \left[\int_0^s e^{-\theta(s-s')} - e^{-\theta(s+s')}\,ds'\right] + \left[\int_s^u e^{-\theta(s-s')} - e^{-\theta(s'+s)}\,ds'\right] ds \\ &= \frac{\sigma^2}{2\theta^2}\int_0^t 2 - 2e^{-\theta s} - e^{-\theta u}e^{\theta s} + e^{-\theta u}e^{-\theta s}\,ds \\ &= \frac{\sigma^2}{2\theta^3}\left(-e^{-\theta(t+u)} + 2e^{-\theta t} + 2e^{-\theta u} - e^{-\theta(u-t)} + 2t\theta - 2\right) \quad (2.221)\end{aligned}$$

Case 2: $t \geqslant u$
If $t \geqslant u$, also the outer integral needs to be split at $s = u$, since afterwards it holds that $s > s'$. Prior to this point the situation is identical to case 1. Hence:

$$\begin{aligned}\mathrm{Cov}(Y_t, Y_u) = \frac{\sigma^2}{2\theta}\Bigg[&\int_0^u \left(\int_0^s e^{-\theta(s-s')} - e^{-\theta(s+s')}\,ds' \right. \\ &\left. + \int_s^u e^{-\theta(s-s')} - e^{-\theta(s'+s)}\,ds'\right) ds \\ &+ \int_u^t \int_0^u e^{-\theta(s-s')} - e^{-\theta(s+s')}\,ds'\,ds\Bigg] \\ = \frac{\sigma^2}{2\theta^2}\Bigg[&\int_0^u 2 - 2e^{-\theta s} - e^{-\theta(u-s)} + e^{-\theta(s+u)}\,ds \\ &+ \int_u^t e^{-\theta(s-u)} + e^{-\theta(s+u)} - 2e^{-\theta s}\,ds\Bigg] \\ = \frac{\sigma^2}{2\theta^3}&\left(-e^{-\theta(t+u)} + 2e^{-\theta t} + 2e^{-\theta u} - e^{-\theta(t-u)} + 2u\theta - 2\right) \quad (2.222)\end{aligned}$$

Combining both cases then yields a universal description of the autocovariance of the time-integrated Ornstein-Uhlenbeck process:

$$\begin{aligned}\mathrm{Cov}(Y_t, Y_u) = \frac{\sigma^2}{2\theta^3}\Big(&-e^{-\theta(t+u)} + 2e^{-\theta t} + 2e^{-\theta u} \\ &- e^{-\theta(\max(t,u)-\min(t,u))} + 2\theta\min(t,u) - 2\Big) \quad (2.223)\end{aligned}$$

Setting $t = u$ in equation 2.223 results in the variance given by eq. 2.224.

$$\mathrm{Var}(Y_t) = \frac{\sigma^2}{2\theta^3}\left(-e^{-2\theta t} + 4e^{-\theta t} + 2\theta t - 3\right) \quad (2.224)$$

To estimate the mean anomaly or in-track orbital uncertainty the covariance of the twice time-integrated Ornstein-Uhlenbeck process is required. Let $Z_t = \int_0^t Y_s \, ds$. Then the autocovariance of the twice time-integrated Ornstein-Uhlenbeck process can be computed via equation 2.225, which makes use of Fubini's theorem and the fact that the integration does not change the expected value of zero.

$$\mathrm{Cov}(Z_t, Z_u) = \int_0^t \int_0^u \mathrm{E}\left[Y_s Y_{s'}\right] ds' \, ds = \int_0^t \int_0^u \mathrm{Cov}(Y_s, Y_{s'}) \, ds' \, ds \tag{2.225}$$

The cases $u \geqslant t$ and $t \geqslant u$ need to be treated separately again to evaluate the min and max-expressions when inserting equation 2.223.

Case 1: $u \geqslant t$
Splitting the inner integral at $s' = s$ results in:

$$\begin{aligned}
\mathrm{Cov}(Z_t, Z_u) = \frac{\sigma^2}{2\theta^3} \int_0^t \Bigg[& \int_0^s -e^{-\theta(s+s')} + 2e^{-\theta s} + 2e^{-\theta s'} - e^{-\theta(s-s')} + 2\theta s' - 2 \; ds' \\
& + \int_s^u -e^{-\theta(s+s')} + 2e^{-\theta s} + 2e^{-\theta s'} - e^{-\theta(s'-s)} + 2\theta s - 2 \, ds' \Bigg] ds \\
= \frac{\sigma^2}{2\theta^4} \Bigg[& \int_0^t e^{-\theta(u+s)} + e^{-\theta(u-s)} + 2\theta u e^{-\theta s} - 2e^{-\theta u} \\
& - \theta^2 s^2 + 2\theta^2 s u - 2\theta u \; ds \Bigg] \\
= \frac{\sigma^2}{2\theta^5} \Bigg[& - e^{-\theta(u+t)} + e^{-\theta(u-t)} - 2\theta u e^{-\theta t} - 2\theta t e^{-\theta u} \\
& - \frac{1}{3}\theta^3 t^3 + \theta^3 t^2 u - 2\theta^2 u t + 2\theta u \Bigg]
\end{aligned} \tag{2.226}$$

Case 2: $t \geqslant u$
A universal description of the covariance also requires an analysis of the second case. Similar to the first time-integration also the outer integral needs to be split at $s = u$ to properly evaluate $\mathrm{Cov}(Y_t, Y_u)$. Hence:

$$\begin{aligned}
\mathrm{Cov}(Z_t, Z_u) = \frac{\sigma^2}{2\theta^3} \Bigg[& \int_0^u \Bigg(\int_0^s -e^{-\theta(s+s')} + 2e^{-\theta s} + 2e^{-\theta s'} - e^{-\theta(s-s')} + 2\theta s' - 2 \, ds' \\
& + \int_s^u -e^{-\theta(s+s')} + 2e^{-\theta s} + 2e^{-\theta s'} - e^{-\theta(s'-s)} + 2\theta s - 2 \; ds' \Bigg) ds \\
& + \int_u^t \int_0^u -e^{-\theta(s+s')} + 2e^{-\theta s} + 2e^{-\theta s'} - e^{-\theta(s-s')} + 2\theta s' - 2 \, ds' \, ds \Bigg]
\end{aligned}$$

$$= \frac{\sigma^2}{2\theta^4}\left[\int_0^u e^{-\theta(s+u)} + e^{-\theta(u-s)} + 2u\theta e^{-\theta s} - 2e^{-\theta u} - \theta^2 s^2 + 2\theta^2 su - 2\theta u \, ds \right.$$
$$\left. + \int_u^t e^{-\theta(s+u)} - e^{-\theta(s-u)} + 2u\theta e^{-\theta s} - 2e^{-\theta u} + \theta^2 u^2 - 2\theta u + 2 \, ds\right]$$
$$= \frac{\sigma^2}{2\theta^5}\left[- e^{-\theta(u+t)} + e^{-\theta(t-u)} - 2u\theta e^{-\theta t} - 2t\theta e^{-\theta u} \right.$$
$$\left. - \frac{1}{3}\theta^3 u^3 + \theta^3 u^2 t - 2\theta^2 ut + 2\theta t\right] \tag{2.227}$$

Combining both cases (equations 2.226 and 2.227) allows to formulate a generic description of the autocovariance of the twice time-integrated OUP:

$$Cov(Z_t, Z_u) = \frac{\sigma^2}{2\theta^5}\left[- e^{-\theta(u+t)} + e^{-\theta(\max(t,u)-\min(t,u))} - 2u\theta e^{-\theta t} \right.$$
$$- 2t\theta e^{-\theta u} - \frac{1}{3}\theta^3 \min(t,u)^3 - 2\theta^2 ut$$
$$\left. + \theta^3 \min(t,u)^2 \max(t,u) + 2\theta \max(t,u)\right] \tag{2.228}$$

Setting $t = u$ in equation 2.228 finally yields a description of $Var(Z_t)$:

$$Var(Z_t) = \frac{\sigma^2}{2\theta^5}\left[\frac{2}{3}\theta^3 t^3 - 2\theta^2 t^2 + 2\theta t - 4\theta t e^{-\theta t} - e^{2\theta t} + 1\right] \tag{2.229}$$

Before making use of equations 2.224 and 2.229 to express the orbital uncertainties, the randomization of the initial state has yet to be accounted for, as model-derived densities are uncertain from the first estimate onwards.

EFFECT OF RANDOMIZING THE INITIAL OUP DENSITY SOLUTION

Drawing the initial OUP density solution from a Gaussian distribution with mean zero and a variance that corresponds to the long-term variance of the relative density error Ornstein-Uhlenbeck process, results in a superposition of the OUP with constant initial value zero and the function of the mean of the OUP with $\mu = 0$ (cf. equation 2.230 where A denotes the random process that models the randomization of the initial state).

$$A_t = A_0 e^{-\theta t} \tag{2.230}$$

Equation 2.230 has an expected value of 0, as $E[A_0] = 0$. Its variance is hence given by:

$$Var(A_t) = Var\left(A_0 e^{-\theta t}\right) = Var(A_0) e^{-2\theta t} = \frac{\sigma^2}{2\theta} e^{-2\theta t} \tag{2.231}$$

Next, a formulation of the autocovariance of equation 2.230 needs to be derived. Since it is zero-mean, it holds that the covariance is equal to the expected value of the product of the function with a time-delayed copy of itself:

$$\mathrm{Cov}(A_t, A_u) = \mathrm{E}[A_t A_u] = e^{-\theta t} e^{-\theta u} \mathrm{E}\left[A_{0_t} A_{0_u}\right] \tag{2.232}$$

Since A_0 is always drawn from the same distribution and independent of the lag, it holds that $\mathrm{E}[A_{0_t} A_{0_u}] = \mathrm{E}[A_0 A_0] = \frac{\sigma^2}{2\theta}$. Consequently the desired covariance is given by equation 2.233.

$$\mathrm{Cov}(A_t, A_u) = \frac{\sigma^2}{2\theta} e^{-\theta(t+u)} \tag{2.233}$$

Equation 2.233 describes an additional autocovariance term of the Ornstein-Uhlenbeck process, in case of a randomization of the initial state. It can either be added directly to the known expression of the classical OUP covariance or, since it does not contain any min/max-expressions, it is also possible to first integrate it with respect to time and then add the result to the respective integrated OUP stochastic process.

Adding equation 2.233 to the covariance expression of the Ornstein-Uhlenbeck process results in:

$$\begin{aligned}\mathrm{Cov}(X_t, X_u) &= \frac{\sigma^2}{2\theta}\left(e^{2\theta \min(u,t) - \theta(u+t)} - e^{-\theta(u+t)} + e^{-\theta(t+u)}\right) \\ &= \frac{\sigma^2}{2\theta} e^{2\theta \min(u,t) - \theta(u+t)}\end{aligned} \tag{2.234}$$

To demonstrate both paths, equation 2.233 is in the following first integrated with respect to time and the result is then added to equation 2.223. The resulting expression is finally integrated as a whole to compute the new equation of $\mathrm{Cov}(Z_t, Z_u)$.

As equation 2.233 contains no min/max-expressions, the domain of integration does not need to be split. Integration with respect to time yields:

$$\begin{aligned}\mathrm{Cov}(B_t, B_s) &= \frac{\sigma^2}{2\theta} \int_0^t \int_0^u e^{-\theta(s+s')} \, ds' \, ds = -\frac{\sigma^2}{2\theta^2} \int_0^t e^{-\theta(s+u)} - e^{-\theta s} \, ds \\ &= \frac{\sigma^2}{2\theta^2}\left(e^{-\theta(t+u)} - e^{-\theta u} - e^{-\theta t} + 1\right)\end{aligned} \tag{2.235}$$

Adding equation 2.235 to eq. 2.223 results in:

$$\begin{aligned}\mathrm{Cov}(Y_t, Y_u) = \frac{\sigma^2}{2\theta^3}\Big(&e^{-\theta t} + e^{-\theta u} - e^{-\theta(\max(t,u) - \min(t,u))} \\ &+ 2\theta \min(t,u) - 1\Big)\end{aligned} \tag{2.236}$$

Equation 2.236 represents the required covariance expression of the time-integrated OUP in the relative density error and allows to compute the mean motion variance after setting $t = u$, which results in an updated version equation 2.224:

$$\mathrm{Var}(Y_t) = \frac{\sigma^2}{2\theta^3}\left(2\theta t + 2e^{-\theta t} - 2\right) \tag{2.237}$$

An additional integration of equation 2.235 with respect to time results in:

$$\begin{aligned}
\mathrm{Cov}(C_t, C_u) &= \frac{\sigma^2}{2\theta^3}\int_0^t\int_0^u e^{-\theta(s+s')} - e^{-\theta s'} - e^{-\theta s} + 1\,\mathrm{d}s'\,\mathrm{d}s \\
&= \frac{\sigma^2}{2\theta^4}\int_0^t e^{-\theta s} - e^{-\theta(s+u)} + e^{-\theta u} - \theta u e^{-\theta s} + \theta u - 1\,\mathrm{d}s \\
&= \frac{\sigma^2}{2\theta^5}\Big(e^{-\theta(t+u)} - e^{-\theta t} - e^{-\theta u} + t\theta e^{-\theta u} \\
&\qquad + u\theta e^{-\theta t} - t\theta - u\theta + \theta^2 ut + 1\Big)
\end{aligned} \tag{2.238}$$

Equation 2.238 represents the expected difference of the following computation of $\mathrm{Cov}(Z_t, Z_u)$ including the randomization of the initial state and equation 2.228.

To obtain an updated expression of eq. 2.228, a second time-integration needs to be performed. In the following, eq. 2.236 is chosen as the baseline for $\mathrm{Cov}(Y_t, Y_u)$. Since it already models the effect of the initial state randomization, no posterior summation of an additional covariance component is required.

Case 1: $u \geqslant t$

$$\begin{aligned}
\mathrm{Cov}(Z_t, Z_u) &= \frac{\sigma^2}{2\theta^3}\int_0^t\Bigg[\int_0^s e^{-\theta s} + e^{-\theta s'} - e^{-\theta(s-s')} + 2\theta s' - 1\,\mathrm{d}s' \\
&\qquad \int_s^u e^{-\theta s} + e^{-\theta s'} - e^{-\theta(s'-s)} + 2\theta s - 1\,\mathrm{d}s'\Bigg]\,\mathrm{d}s \\
&= \frac{\sigma^2}{2\theta^4}\Bigg[\int_0^t u\theta e^{-\theta s} - e^{-\theta u} + e^{-\theta s} + e^{-\theta(u-s)} \\
&\qquad + 2\theta^2 su - \theta^2 s^2 - u\theta - 1\,\mathrm{d}s\Bigg] \\
&= \frac{\sigma^2}{2\theta^5}\Bigg[e^{-\theta(u-t)} - e^{-\theta t} - e^{-\theta u} - t\theta e^{-\theta u} - u\theta e^{-\theta t} \\
&\qquad -\frac{1}{3}\theta^3 t^3 + \theta^3 t^2 u - \theta^2 ut + u\theta - t\theta + 1\Bigg]
\end{aligned} \tag{2.239}$$

Comparing equation 2.239 with eq. 2.226, it becomes apparent that equation 2.239 = eq. 2.226 + eq. 2.238, which demonstrates that the

separate integration of equation 2.233 leads to the same result, as its incorporation into the covariance expressions of the standard OUP with mean zero ($X_0 = 0$).

Case 2: $t \geqslant u$
Following the argument that equation 2.238 can be added to the results of the time-integrated standard OUP, $\mathrm{Cov}(Z_t, Z_u)$ is found by summing equations 2.227 and 2.238, which yields:

$$\mathrm{Cov}(Z_t, Z_u) = \frac{\sigma^2}{2\theta^5}\left[e^{-\theta(t-u)} - e^{-\theta t} - e^{-\theta u} - u\theta e^{-\theta t} - t\theta e^{-\theta u} - \frac{1}{3}\theta^3 u^3 + \theta^3 u^2 t - \theta^2 u t + t\theta - u\theta + 1\right] \quad (2.240)$$

Similarly, the generic result which includes the min and max terms is found by adding equations 2.228 and 2.238 or simply by comparing equations 2.239 and 2.240:

$$\begin{aligned}\mathrm{Cov}(Z_t, Z_u) = \frac{\sigma^2}{2\theta^5}\Big[&e^{-\theta(\max(t,u)-\min(t,u))} - e^{\theta t} - e^{-\theta u} \\ &- t\theta e^{-\theta u} - u\theta e^{-\theta t} - \frac{1}{3}\theta^3 \min(t,u)^3 \\ &+ \theta^3 \min(t,u)^2 \max(t,u) - \theta^2 u t \\ &+ \theta \max(t,u) - \theta \min(t,u) + 1\Big]\end{aligned} \quad (2.241)$$

Setting $t = u$ allows to formulate an updated expression of the corresponding variance:

$$\mathrm{Var}(Z_t) = \frac{\sigma^2}{2\theta^5}\left[\frac{2}{3}\theta^3 t^3 - \theta^2 t^2 - 2e^{-\theta t} - 2\theta t e^{-\theta t} + 2\right] \quad (2.242)$$

Since $s^2_{\mathrm{Model}} = \frac{\sigma^2}{2\theta}$, it holds that:

$$\frac{\sigma^2}{2\theta^3} = \frac{s^2_{\mathrm{Model}}}{\theta^2} \quad (2.243)$$

and

$$\frac{\sigma^2}{2\theta^5} = \frac{s^2_{\mathrm{Model}}}{\theta^4} \quad (2.244)$$

Inserting equations 2.237 and 2.242 into eqs. 2.178 and 2.179, while accounting for eqs. 2.243 and 2.244, allows to formulate the final relationships governing the mean motion and mean anomaly variances, as given by equations 2.245 and 2.246.

Mean motion variance

$$\mathrm{Var}(\delta n(t)) = \left(1 + \frac{1}{3}\frac{\Delta\hat{n}}{\hat{n}_0}\right)^2 \mathrm{Var}(\delta n_0) + \left(\frac{\Delta\hat{n}}{\Delta t}\right)^2 \frac{s^2_{\mathrm{Model}}}{\theta^2}(2\theta t + 2e^{-\theta t} - 2) \quad (2.245)$$

Mean anomaly variance

$$\begin{aligned}\mathrm{Var}(\delta M(t)) = \mathrm{Var}(\delta M_0) + \left(\Delta t\left(1+\frac{1}{6}\frac{\Delta\hat{n}}{\hat{n}_0}\right)\right)^2 \mathrm{Var}(\delta n_0) + 2\Delta t\left(1+\frac{1}{6}\frac{\Delta\hat{n}}{\hat{n}}\right)\mathrm{Cov}(\delta M_0, \delta n_0) \\ + \left(\frac{\Delta\hat{n}}{\Delta t}\right)^2 \frac{s^2_{\mathrm{Model}}}{\theta^4}\left(\frac{2}{3}\theta^3 t^3 - \theta^2 t^2 - 2e^{-\theta t} - 2\theta t e^{-\theta t} + 2\right)\end{aligned} \tag{2.246}$$

Extracting only the portion of the variance that is attributed to the density error and not to initial errors in the state vector, equations 2.245 and 2.246 can be simplified to:

$$\mathrm{Var}(\delta n(t)) = \left(\frac{\Delta\hat{n}}{\Delta t}\right)^2 \frac{s^2_{\mathrm{Model}}}{\theta^2}\left(2\theta t + 2e^{-\theta t} - 2\right) \tag{2.247}$$

and

$$\mathrm{Var}(\delta M(t)) = \left[\left(\frac{\Delta\hat{n}}{\Delta t}\right)^2 \frac{s^2_{\mathrm{Model}}}{\theta^4}\left(\frac{2}{3}\theta^3 t^3 - \theta^2 t^2 - 2\theta t e^{-\theta t} - 2e^{-\theta t} + 2\right)\right] \tag{2.248}$$

Since errors in mean anomaly are to first order proportional to in-track errors via $\partial X_{||}/\partial M = (\partial X_{||}/\partial t)/(\partial M/\partial t) = v/n$, equation 2.248 can also be used to formulate an expression of the estimated in-track position error variance:

$$\mathrm{Var}(\delta X_{||}(t)) = \left(\frac{\hat{v}}{\hat{n}}\right)^2 \left[\left(\frac{\Delta\hat{n}}{\Delta t}\right)^2 \frac{s^2_{\mathrm{Model}}}{\theta^4}\left(\frac{2}{3}\theta^3 t^3 - \theta^2 t^2 - 2\theta t e^{-\theta t} - 2e^{-\theta t} + 2\right)\right] \tag{2.249}$$

Observing equations 2.247-2.249 closely, reveals that with growing time the mean motion variance growth will be dominated by the term $\frac{t}{\theta}$, whereas the mean anomaly/in-track error growth will become dominated by $\frac{t^3}{\theta}$. This consideration demonstrates the importance of the OUP θ-parameter, as it is inversely proportional to the expected orbital uncertainties. Larger values of θ will lead to smaller expected orbital uncertainties, whereas smaller values of θ result in larger orbital uncertainties.

An evaluation of this parameter can be found in section 2.6.2 alongside the numerical simulations used to validate above relationships.

2.6.2 *Simulation results*

In the following the results obtained with the newest of the MSIS and DTM-class models (NRLMSISE-00 and DTM-2013) are presented. The chosen satellite is TerraSAR-X, which has been launched in 2007 and orbits Earth sun-synchronously at a mean altitude of ~511 km (Pitz and Miller, 2010). The coefficient of drag has been fixed at 2.2 (flat plate model), which is also the value used in the on-board satellite propagator. The ballistic coefficient is modeled constant with a value of $0.006\,\mathrm{m}^2/\mathrm{kg}$ to avoid the inclusion of attitude information. The chosen epoch was set to February 18$^{\mathrm{th}}$, 2015, 09:00:00 UTC. The simulation was performed with the Special Perturbations Orbit determination

and Orbit analysis toolKit (SPOOK) of Airbus, which has been extended with the capability of grid-scale density-uncertainty driven orbit uncertainty propagation. More information about SPOOK can be found in (Rodriguez Fernandez et al., 2019b).

The simulation scenario is as follows: prior to the Monte-Carlo (MC) iterations an initial state vector is required for the subsequent propagation phases. This state vector was selected from the Position Velocity Time (PVT) telemetry of the active MOSAIC GNSS Receiver of TerraSAR-X and subsequently converted from EME2000 to GCRF. This initial orbit solution can be considered to be error-free for the purpose of this analysis, which allows to make use of equations 2.247 and 2.249 for the computation of the mean motion and in-track position uncertainties.

In each MC orbit propagation the density SDE (equation 2.216) is solved concurrently with the orbital state vector using a first-order Milstein algorithm that corresponds to the modified Ornstein-Uhlenbeck process. As suggested by Sagnières and Sharf (2017), the initial value of the density OUP is drawn from a Gaussian distribution with a standard deviation that corresponds to the assumed model uncertainty, centered at the model density estimate.

Knowledge of the model inputs is assumed to be perfect to avoid mixing grid-scale density uncertainty (described by equations 2.247 and 2.249) with input density uncertainty during the validation. Consequently also no unforeseen storms or measurement uncertainties are simulated and the same set of published space-weather data[3] is used in the truth computation and orbit/uncertainty estimations. The computation of a universal covariance matrix that accounts for both input and grid-scale uncertainty is described in section 2.7.

The validation is considered successful if the spread of the 300 MC propagations agrees with the three-sigma expected uncertainty as derived from equations 2.247 and 2.249.

As SPOOK uses a variable step-size Shampine-Gordon integrator, it is necessary to detect and treat step repetitions due to step size reductions. For this purpose also the quantities of the previous step are stored in each iteration, which allows to restore a prior state in case of a possible recomputation of a step due to a too-large step size.

In case of NRLMSISE-00 (figure 2.13), atmospheric drag is the only perturbation that has been enabled to show that the expected variances in the mean motion and in-track position agree with the Monte-Carlo iterations. For DTM-2013 also other perturbations than orbital drag (solar radiation pressure, solar and lunar gravity, as well as a 30×30 GGM02C non-spherical gravitational field) have been considered to demonstrate that the uncertainty predictions also remain valid in operationally relevant scenarios with all perturbations enabled. This desirable property is the result of the periodic mean motion varia-

3 Insofar as the models agree on the choice of solar and geomagnetic proxies

tions introduced by the additional perturbations to compensate in the subsequent integration step towards in-track position. In general the mean motion uncertainty of equations 2.245 and 2.247 hence describes mean mean motion uncertainty.

The color coding of figures 2.13 to 2.16 is as follows: MC iterations are plotted in green. The three sigma standard-deviation of the MC iterations is plotted in red. The blue lines represent the estimated three sigma uncertainties at each stage. The estimated orbital uncertainties are computed as three times the square root of equation 2.247 for the mean motion and three times the square root of equation 2.249 for the in-track position error. For NRLMSISE-00 a one sigma model uncertainty of $s_{\text{Model}} = 15\%$ was assumed, as it is a comparatively old model. For DTM-2013 the assumed model uncertainty was set to 10%.

NRLMSISE-00: Speed of mean reversion: $\theta = 10^{-4}$, Long-term standard deviation = assumed one sigma model uncertainty: $s_{\text{Model}} = 15\%$

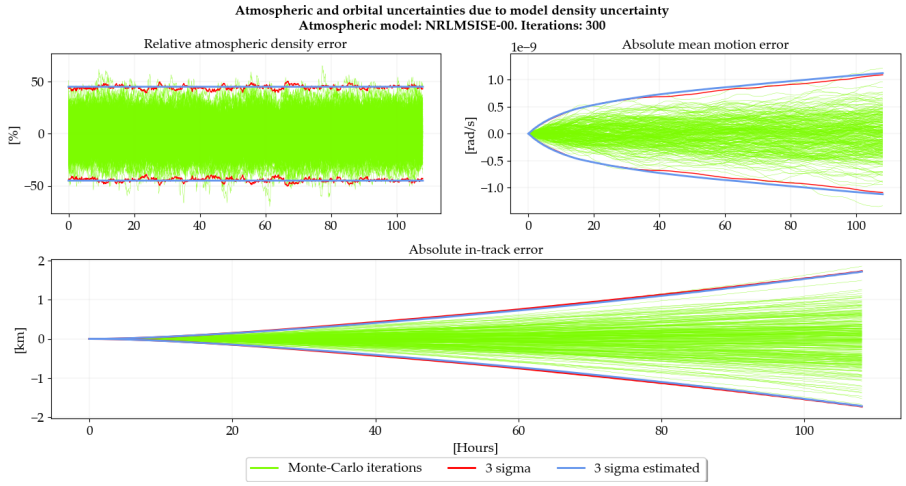


Figure 2.13: Propagation of initial state using NRLMSISE-00 as driver of the modified Ornstein-Uhlenbeck error process by Sagnières and Sharf. Atmospheric drag is the only perturbation which has been enabled

Figure 2.13 depicts the results obtained with NRLMSISE-00 as baseline density model. The estimated uncertainty in the mean motion closely follows the actual spread of the Monte-Carlo iterations. The same holds true after the additional integration to the in-track position. Therefore the estimated in-track position error uncertainty matches the actual Monte-Carlo uncertainty. With increasing values of θ, both the speed of mean reversion (the OUP attraction towards the mean) and the strength of the short-term density fluctuations increase. Consequently for $\theta = 10^{-4}$, relative density error samples quickly return to near-mean values, which results in comparatively small mean motion and in-track position errors. The maximum in-track error after 4.5 days of propagation is ~1.7 km.

In the following the simulation results for DTM-2013 are presented. Three cases with different values of θ have been computed. In addition to orbital drag also solar radiation pressure, a 30×30 GGM02C aspherical gravity field, thermospheric winds, as well as solar and lunar gravity perturbations have been enabled to demonstrate the compatibility of the derived equations with operational scenarios. The effect of particular values of θ is discussed afterwards.

Figure 2.14 displays the results obtained with θ set to 10^{-4} for comparison with figure 2.13. It is clearly visible that the additional perturbations introduce periodic variations in the mean motion errors. Consequently the mean motion uncertainty estimates of equation 2.247 match the actual mean motion uncertainty only for the first ~30 hours and represent the average mean motion uncertainty afterwards. The additional integration to the in-track error is able to compensate these periodic variations, resulting in consistent in-track uncertainty predictions. DTM-2013 agrees with the case of NRLMSISE-00 in that both models result in an in-track position error of ~1.6 km after 4.5 days of propagation.

DTM-2013: Speed of mean reversion: $\theta = 10^{-4}$, Long-term standard deviation = assumed one sigma model uncertainty: $s_{\text{Model}} = 10\%$

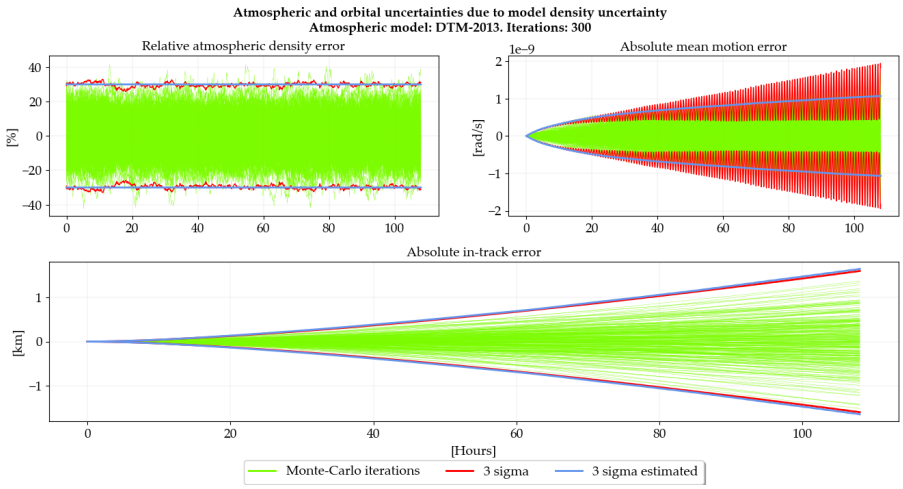


Figure 2.14: Propagation of initial state using DTM-2013 as driver of the modified Ornstein-Uhlenbeck error process by Sagnières and Sharf. Besides atmospheric drag also perturbations due to the Earth's aspherical gravitational field, solar radiation pressure and third body (solar and lunar) gravity have been activated

The simulation results of figure 2.15 confirm all expected effects for a smaller value of θ: first of all, the lower speed of mean reversion results in longer departures of the relative density errors from the mean. The intensity of the random fluctuations is decreased with respect to figure 2.14. As the mean motion and in-track uncertainties are inversely proportional to θ, the three sigma bounds show a notable growth when

compared to figure 2.14. Also for $\theta = 10^{-5}$ the uncertainty predictions follow the actual uncertainty of the Monte-Carlo iterations closely. The expected three sigma in-track position uncertainty is ~4.0 km after 4.5 days of propagation.

DTM-2013: Speed of mean reversion: $\theta = 10^{-5}$, Long-term standard deviation = assumed one sigma model uncertainty: $s_{\text{Model}} = 10\%$

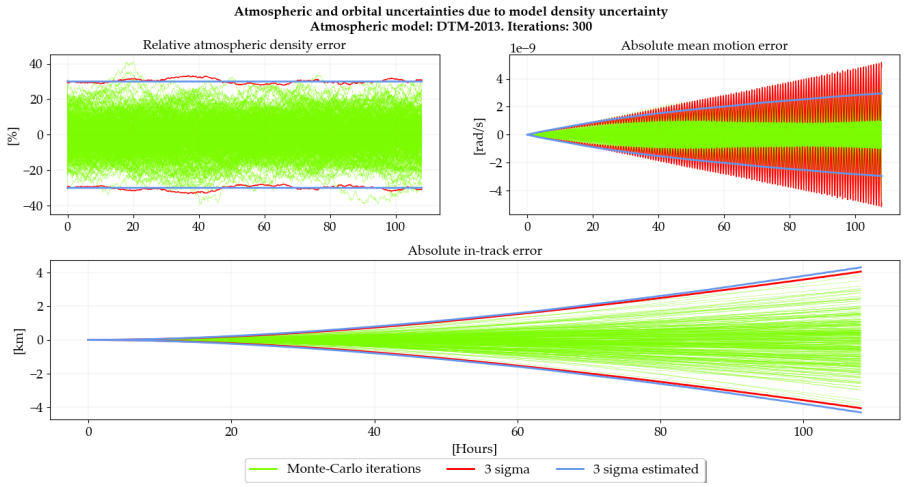


Figure 2.15: Orbital variations due to OUP density fluctuations. The simulation setup is identical to figure 2.14, however θ has been set to 10^{-5}

DTM-2013: Speed of mean reversion: $\theta = 10^{-6}$, Long-term standard deviation = assumed one sigma model uncertainty: $s_{\text{Model}} = 10\%$

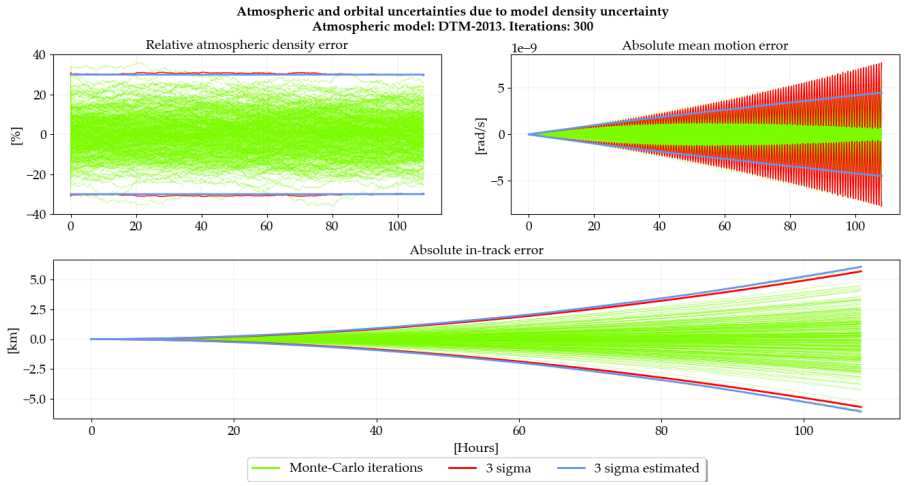


Figure 2.16: Orbital variations due to OUP density fluctuations. The simulation setup is identical to figures 2.14 and 2.15, with θ set to 10^{-6}

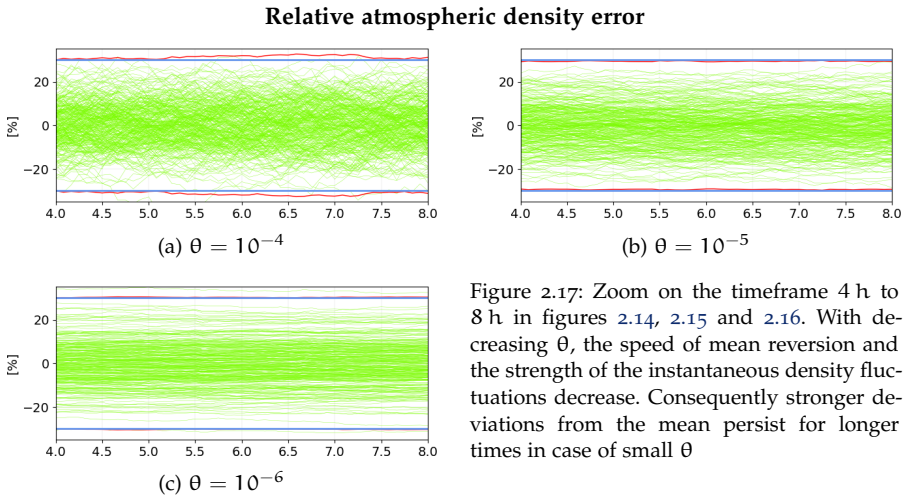
As θ is lowered further in figure 2.16, the discussed effects are even more pronounced. The parametrization of θ allows for density fluctuations to depart from the mean for multiple days, which can be well observed in the outer regions of the relative density error plot.

These long-time departures result in mean motion and in-track distributions with notably increased magnitude. The expected three sigma in-track position uncertainty is ~6.0 km after 4.5 days of propagation. As is the case for figures 2.14 and 2.15, the estimated mean motion uncertainty matches for the first ~30 h and then represents the average mean motion uncertainty.

The results depicted in figures 2.13 to 2.16 both validate the variance estimations (equations 2.245 to 2.249) and demonstrate that the (mean) mean motion and in-track position error uncertainty due to the modified OUP by Sagnières and Sharf can be approximated with the help of a standard OUP in the relative density error.

SELECTION OF θ

Figures 2.13 to 2.16 highlight the importance of the OUP θ-parameter, which represents the speed of mean reversion and therefore determines the duration of the periods in which the OUP results deviate from the model-estimates. The influence of θ on the density fluctuations is demonstrated in figure 2.17.



(a) $\theta = 10^{-4}$

(b) $\theta = 10^{-5}$

(c) $\theta = 10^{-6}$

Figure 2.17: Zoom on the timeframe 4 h to 8 h in figures 2.14, 2.15 and 2.16. With decreasing θ, the speed of mean reversion and the strength of the instantaneous density fluctuations decrease. Consequently stronger deviations from the mean persist for longer times in case of small θ

The expected uncertainty growth is inversely proportional to θ. While the derived uncertainty estimation equations (eqs. 2.245 to 2.249) work with any value of θ and are therefore universal, it must be the goal to set up the OUP in a way to represent natural grid-scale density deviations as closely as possible. Consequently a perimeter of realistic θ-values is required. These bounds can be inferred from the intensity of the density fluctuations and the magnitude of the resulting in-track errors.

Grid-scale density errors are the result of residuals in the least squares fitting process of semi-empirical density models. This type of density error is therefore expected to be comparatively smooth with correlation times in the order of orbital and daily scales. Judging from figure 2.17, these timeframes may be associated with the cases

$\theta = 10^{-5}$ and $\theta = 10^{-6}$. The case of $\theta = 10^{-4}$ depicts both unnaturally large density fluctuations and unrealistically small error growth.

The choices $\theta = 10^{-5}$ and $\theta = 10^{-6}$ are also supported by the corresponding in-track errors of figures 2.15 and 2.16, which indicate realistic maximum deviations of ~4 km to ~6 km after 4.5 days of propagation for the geometry of TerraSAR-X and the chosen epoch.

To further narrow down the perimeter of θ, an inter-model variability analysis was performed for the chosen epoch. Equations 2.247 and 2.249 are also expected to result in valid error bounds when computing the reference orbit with one density model and the propagation for comparison with another. For this case s_{Model} was set to the combined model uncertainty of $\sqrt{(10\%)^2 + (15\%)^2} = 18\%$. A combination of two models and the values $\theta = 10^{-5}$, as well as $\theta = 10^{-6}$ results in four scenarios. No error bound should be violated for a proper parametrization of θ.

Based on the results depicted in figure 2.18, a value of $\theta = 10^{-5}$ results in a violation of the error bounds in one of two cases. Setting $\theta = 10^{-6}$ does not lead to any violation. Therefore a realistic choice of θ is expected to be close to 10^{-6}. Sagnières and Sharf (2017) also performed an analysis of θ using Monte-Carlo propagated ensembles of Envisat, which orbits Earth ~250 km higher than TerraSAR-X and is about 6.5 times as heavy. Using a coefficient of variation-analysis they concluded that a value of $\theta = 10^{-6}$ is the most realistic parametrization, which is consistent with the inter-model variability analysis.

Allowing for some margin, realistic values for the speed of mean reversion are expected to lie in the range $\theta \in [5 \cdot 10^{-6}, 5 \cdot 10^{-7}]$. The following list summarizes the main findings about grid-scale density uncertainty:

- grid-scale density uncertainty can be fruitfully treated using a classical Ornstein-Uhlenbeck process in the relative density error or a modified OUP in atmospheric density
- the long-term mean motion variance due to OUP density uncertainty grows proportionally with time ($\sim t/\theta$)
- the long-term in-track position error variance due to OUP density uncertainty grows proportionally to the third power of time ($\sim t^3/\theta$)
- the OUP rate of mean reversion (θ) is inversely proportional to the long-term error growth and a key parameter of the uncertainty estimation. It influences both the instantaneous fluctuations, as well as the expected error growth. Realistic values of θ are expected to lie within the interval $\theta \in [5 \cdot 10^{-6}, 5 \cdot 10^{-7}]$, which is in agreement with the results of Sagnières and Sharf (2017)

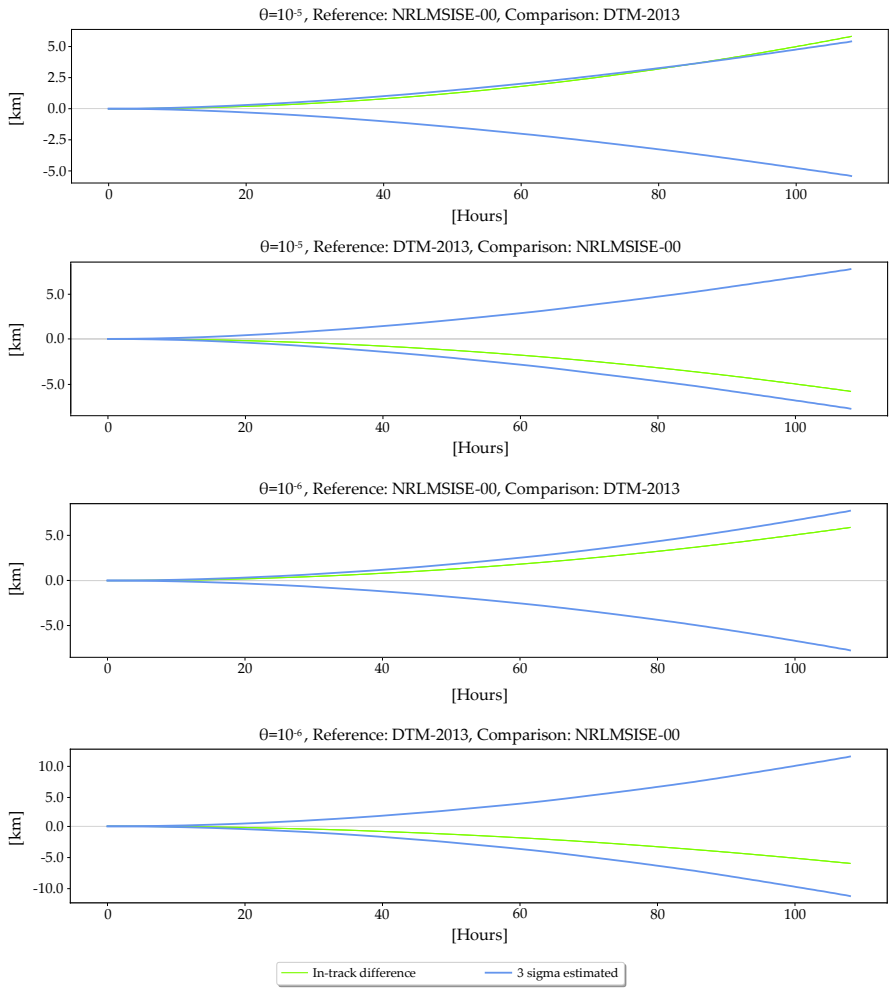


Figure 2.18: Inter-model variability. The first case violates the three sigma error bounds, which indicates that a value of $\theta = 10^{-5}$ is too large. Using $\theta = 10^{-6}$ does not result in any violation

2.7 COMBINED DENSITY UNCERTAINTY COVARIANCE

Sections 2.2 to 2.6 treated the quantification of atmospheric density uncertainty due to solar flux and magnetic index input uncertainty, as well as grid-scale density uncertainty. To finalize the density uncertainty framework, some further considerations are required. These include the derivations of the mean motion and mean anomaly covariance cross-correlations for the respective error processes (white noise, Brownian motion and Ornstein-Uhlenbeck). Furthermore, a robust method of covariance conversion to Cartesian coordinates is desirable, since many operational programs are operating in GCRF coordinates. Additionally, density uncertainty consideration for orbit determination requires a special treatment of the uncertainty propagation to properly model the time-evolution of the pdf. Finally, the different origins of density uncertainty need to be combined into a common covariance matrix. These topics shall be discussed in the current section.

2.7.1 *Mean motion and mean anomaly correlation coefficients*

As density uncertainty dominantly acts in the orbital plane, it can be quantified using a sparse 6×6 covariance matrix. Using orbital elements, only the mean motion and mean anomaly components are nonzero. To formulate the resulting covariance matrices for input uncertainty, the mean motion and mean anomaly correlation coefficient needs to be derived for a white noise and Brownian motion error process model. Grid-scale density uncertainty requires the corresponding correlation coefficient for an Ornstein-Uhlenbeck process. The derivation of these coefficients uses similar techniques as the corresponding variances and is shown in the following.

The correlation coefficient of two random variables is defined as

$$\rho_{XY} = \frac{E\left[(X-\bar{X})(Y-\bar{Y})\right]}{\sigma_X \sigma_Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \tag{2.250}$$

By definition it is a quantity that ranges between $-1 \leqslant \rho_{XY} \leqslant 1$. For input and grid-scale density uncertainty the random processes are the time-integrated and twice time-integrated relative density error processes, such that the term

$$Cov(X,Y) \to Cov\left(\int_0^t \int_0^{t'} \epsilon_\rho(t'')\,dt''\,dt', \int_0^t \epsilon_\rho(t')\,dt'\right) \tag{2.251}$$

needs to be found.

INPUT UNCERTAINTY: WHITE NOISE AND BROWNIAN MOTION

Considering the error process model given in equations 2.180-2.183, equation 2.251 corresponds to $Cov(X_2(t), X_1(u))$ (with the time-dimensionality still to be accounted for), if the relative density error process is treated as white noise. Similarly $Cov(X_3(t), X_2(u))$ (with the time-dimensionality still to be accounted for) needs to be computed if the relative density error process is considered as Brownian motion. Due to the zero-mean property the covariance may be obtained from the expected value of the random variable product:

$$\begin{aligned} Cov(X_2(t), X_1(u)) &= E\left[\int_0^t X_1(s)\,ds \int_0^u \frac{1}{\delta t} X_0(s')\,ds'\right] \\ &= \frac{1}{\delta t} E\left[\int_0^t \int_0^u X_1(s) X_0(s')\,ds'\,ds\right] \\ &= \frac{1}{\delta t} \int_0^t \int_0^u E\left[X_1(s) X_0(s')\right] ds'\,ds \end{aligned} \tag{2.252}$$

Since $E\left[X_1(s)X_0(s')\right] = Cov(X_1(s), X_0(s'))$, a description of the covariance one integration-level earlier is required to continue with equation 2.252. Inserting the definition of X_1 results in:

$$E\left[X_1(s)X_0(s')\right] = E\left[\frac{1}{\delta t}\int_0^t X_0(s)\,ds\, X_0(s')\right]$$

$$\Leftrightarrow E\left[X_1(s)X_0(s')\right] = \frac{1}{\delta t}\int_0^t E\left[X_0(s)X_0(s')\right] ds \tag{2.253}$$

From equation 2.184 it follows that:

$$E\left[X_1(s)X_0(s')\right] = \frac{1}{\delta t}\int_0^t P_{X_0}\delta(s-s')\,ds = \frac{P_{X_0}}{\delta t}\int_0^t \delta(s-s')\,ds \tag{2.254}$$

where δ is the Dirac-delta. If $s \geqslant s'$ then the integral is 1, else it is zero. Since $0 \leqslant s \leqslant t$ and $0 \leqslant s' \leqslant u$, $s \geqslant s'$ also implies $t \geqslant u$. Summarizing:

$$E\left[X_1(s)X_0(s')\right] = \begin{cases} \frac{P_{X_0}}{\delta t} & \text{if } s \geqslant s' \\ 0 & \text{otherwise} \end{cases} \tag{2.255}$$

Since equation 2.255 contains a case-differentiation, eq. 2.252 needs to be analyzed separately for both options.

Case 1: $s < s'$
If $s < s'$, then $E\left[X_1(s)X_0(s')\right] = 0$. Consequently also equation 2.252 becomes zero and there is no correlation.

Case 2: $s \geqslant s'$
The interesting case is $s \geqslant s'$, for which $E\left[X_1(s)X_0(s')\right] = \frac{P_{X_0}}{\delta t}$. Next, inserting equation 2.255 into eq. 2.252 and requiring $s \geqslant s'$ results in:

$$Cov(X_2(t), X_1(u)) = \frac{P_{X_0}}{(\delta t)^2}\int_0^t\int_0^u 1\,ds'\,ds, \text{ subject to } s \geqslant s' \tag{2.256}$$

While the outer integral is at a certain s, the inner integral ranges from $0 \leqslant s' \leqslant u$. As equation 2.256 is only defined for $s \geqslant s'$, the inner integral must be split at $s' = s$ to account for the domain of definition:

$$Cov(X_2(t), X_1(u)) = \frac{P_{X_0}}{(\delta t)^2}\int_0^t\left[\int_0^s 1\,ds' + \int_s^u 1\,ds'\right] ds \tag{2.257}$$

In equation 2.257 the second inner integral is zero, as $s' \geqslant s$. Consequently, requiring $s \geqslant s'$ is identical to setting the upper integration boundary of the inner integral to $s' = s$, which yields:

$$\begin{aligned} \mathrm{Cov}(X_2(t), X_1(u)) &= \frac{P_{X_0}}{(\delta t)^2} \int_0^t \int_0^s 1 \, ds' \, ds = \frac{P_{X_0}}{(\delta t)^2} \int_0^t s \, ds \\ &= \frac{1}{2} \frac{P_{X_0}}{(\delta t)^2} t^2 \end{aligned} \tag{2.258}$$

Combining both cases results in:

$$\mathrm{E}\left[X_2(t) X_1(u)\right] = \begin{cases} \frac{1}{2} \frac{P_{X_0}}{(\delta t)^2} t^2 & \text{if } s \geqslant s', \text{ i.e. } t \geqslant u \\ 0 & \text{otherwise} \end{cases} \tag{2.259}$$

In the final step of the variance/covariance definitions, the time variables are set equal. Consequently the first case holds, which yields $\mathrm{E}\left[X_2(t) X_1(u)\right] = \frac{1}{2} \frac{P_{X_0}}{(\delta t)^2}$.

So far, X_0 and X_1 were considered to be dimensionless. If ϵ_ρ follows a white noise process, then X_0 needs to be dimensionless and X_1 shall be time-dimensional. Hence, the model for X_1 changes from $X_1 = \frac{1}{\delta t} \int_0^t X_0(s) \, ds$ to $X_1 = \int_0^t X_0(s) \, ds$. Consequently equation 2.252 becomes:

$$\begin{aligned} \mathrm{Cov}(X_2(t), X_1(u)) &= \mathrm{E}\left[\int_0^t X_1(s) \, ds \int_0^u X_0(s') \, ds' \right] \\ &= \int_0^t \int_0^u \mathrm{E}\left[X_1(s) X_0(s')\right] ds' \, ds \end{aligned} \tag{2.260}$$

and equation 2.255 changes to:

$$\mathrm{E}\left[X_1(s) X_0(s')\right] = \begin{cases} P_{X_0} & \text{if } s \geqslant s' \\ 0 & \text{otherwise} \end{cases} \tag{2.261}$$

As equations 2.260 and 2.261 both lack the factor $\frac{1}{\delta t}$ in comparison to the derivation where X_1 was treated dimensionless, equation 2.259 becomes:

$$\mathrm{E}\left[X_2(t) X_1(u)\right] = \begin{cases} \frac{1}{2} P_{X_0} t^2 & \text{if } s \geqslant s', \text{ i.e. } t \geqslant u \\ 0 & \text{otherwise} \end{cases} \tag{2.262}$$

The double-sided PSD, P_{X_0}, may be computed as $\sigma^2_{\rho_p} \delta t$ within the Nyquist frequency for band-limited white noise. Therefore the desired expression for the white noise case is given by equation 2.263:

$$\mathrm{Cov}(X_2(t), X_1(t)) = \frac{1}{2} \sigma^2_{\rho_p} \delta t \, t^2 \tag{2.263}$$

For the Brownian motion case X_0 and X_1 are dimensionless. Hence $\mathrm{Cov}(X_3(t), X_2(u))$ needs to be computed using the error process model given by equations 2.180-2.183 without any posterior considerations of the time-dimensionality. Due to the zero-mean property, the covariance may be computed as the expected value of the product of the random variables, which yields:

$$\mathrm{Cov}(X_3(t), X_2(u)) = \mathrm{E}\left[\int_0^t X_2(s)\,\mathrm{d}s \int_0^u X_1(s')\,\mathrm{d}s'\right]$$

$$\Leftrightarrow \mathrm{Cov}(X_3(t), X_2(u)) = \int_0^t \int_0^u \mathrm{E}\left[X_2(s)X_1(s')\right]\mathrm{d}s'\,\mathrm{d}s \tag{2.264}$$

Inserting eq. 2.259 into eq. 2.264 results in:

$$\mathrm{Cov}(X_3(t), X_2(u)) = \begin{cases} \int_0^t \int_0^u \frac{1}{2}\frac{P_{X_0}}{(\delta t)^2} s^2\,\mathrm{d}s'\,\mathrm{d}s & \text{if } s \geqslant s' \\ 0 & \text{otherwise} \end{cases} \tag{2.265}$$

Again, the covariance is only nonzero if $s \geqslant s'$, which leads to:

$$\mathrm{Cov}(X_3(t), X_2(u)) = \frac{1}{2}\frac{P_{X_0}}{(\delta t)^2} \int_0^t \int_0^u s^2\,\mathrm{d}s'\,\mathrm{d}s \tag{2.266}$$

The requirement of $s \geqslant s'$ results in splitting the inner integral at $s' = s$. This in turn is equal to setting the upper boundary of the inner integral to $s' = s$, as the second integral, in which $s' > s$, is zero. Therefore the covariance is found via:

$$\mathrm{Cov}(X_3(t), X_2(u)) = \frac{1}{2}\frac{P_{X_0}}{(\delta t)^2} \int_0^t s^2 \int_0^s 1\,\mathrm{d}s'\,\mathrm{d}s = \frac{1}{2}\frac{P_{X_0}}{(\delta t)^2} \int_0^t s^3\,\mathrm{d}s$$

$$= \frac{1}{8}\frac{P_{X_0}}{(\delta t)^2} t^4 \quad \text{if } t \geqslant u \tag{2.267}$$

Using $P_{X_0} = \sigma^2_{\rho_p}\delta t$ and setting $u = t$, the desired Brownian motion covariance is given by equation 2.268:

$$\mathrm{Cov}(X_3(t), X_2(t)) = \frac{1}{8}\frac{\sigma^2_{\rho_p}}{\delta t} t^4 \tag{2.268}$$

Next, the covariance of the integrated relative density error processes must be used to derive an expression of the mean motion/mean anomaly cross-correlation. Considering only the part of the mean motion and mean anomaly errors which depends on the density error (last summand of eqs. 2.171 and 2.177 with $t_0 \to 0$), it becomes

apparent, that the mean motion and mean anomaly errors both depend linearly on the time-integrated and twice time-integrated relative density error processes. Since in orbital elements the prefactors are equal, they can be reduced such that the desired correlation coefficient is found via:

$$\begin{aligned}\rho_{\delta n M} &= \frac{\mathrm{Cov}(\delta n, \delta M)}{\sqrt{\mathrm{Var}(\delta n)}\sqrt{\mathrm{Var}(\delta M)}} \\ &= \frac{\left(\frac{\Delta \hat{n}}{\Delta t}\right)^2 \mathrm{E}\left[\int_0^t \epsilon_{\rho_p}(t')\,dt' \int_0^t \int_0^{t'} \epsilon_{\rho_p}(t'')\,dt''\,dt'\right]}{\left(\frac{\Delta \hat{n}}{\Delta t}\right)^2 \sqrt{\mathrm{Var}\left(\int_0^t \epsilon_{\rho_p}(t')\,dt'\right)}\sqrt{\mathrm{Var}\left(\int_0^t \int_0^{t'} \epsilon_{\rho_p}(t'')\,dt''\,dt'\right)}} \\ &= \frac{\mathrm{Cov}\left(\int_0^t \epsilon_{\rho_p}(t')\,dt', \int_0^t \int_0^{t'} \epsilon_{\rho_p}(t'')\,dt''\,dt'\right)}{\sqrt{\mathrm{Var}\left(\int_0^t \epsilon_{\rho_p}(t')\,dt'\right)}\sqrt{\mathrm{Var}\left(\int_0^t \int_0^{t'} \epsilon_{\rho_p}(t'')\,dt''\,dt'\right)}}\end{aligned} \tag{2.269}$$

Equation 2.269 demonstrates that the absolute mean motion error/mean anomaly error correlation coefficient is equal to the time-integrated and twice time-integrated relative density error correlation. Inserting the variance expressions from eqs. 2.212-2.215 and the covariances given by equations 2.263 and 2.268 yields for the white noise case:

$$\rho_{\delta n M} = \frac{\frac{1}{2}\sigma^2_{\rho_p}\delta t\, t^2}{\sqrt{\sigma^2_{\rho_p} t \delta t\, \frac{1}{3}\sigma^2_{\rho_p} t^3 \delta t}} = \frac{\sqrt{3}}{2} = \frac{\sqrt{12}}{4} \tag{2.270}$$

and for the Brownian motion case:

$$\rho_{\delta n M} = \frac{\frac{1}{8}\frac{\sigma^2_{\rho_p}}{\delta t} t^4}{\sqrt{\frac{1}{3}\frac{\sigma^2_{\rho_p}}{\delta t} t^3\, \frac{1}{20}\frac{\sigma^2_{\rho_p}}{\delta t} t^5}} = \frac{\sqrt{60}}{8} = \frac{\sqrt{15}}{4} \tag{2.271}$$

Equations 2.270 and 2.271 result in constant correlation coefficients for the white noise and Brownian motion error processes. If computations are done in orbital elements, the correlation is positive. In Schiemenz et al. (2019c) the covariances are directly expressed in Cartesian coordinates. In this case the prefactors cancel except for the sign, such that the correlation coefficients become negative. After the publication of Schiemenz et al. (2019c), however, it has been found that it is preferable to first formulate the entire covariance matrix in orbital elements and then to convert it to Cartesian coordinates using the approach shown in section 2.7.2.

Using equations 2.270 and 2.271 the mean motion error/mean anomaly error covariance is finally found by rearranging equation 2.250 for the covariance, i.e. via equation 2.273.

$$\mathrm{Cov}(\delta n(t), \delta M(t)) = \rho_{\delta n M}\sqrt{\mathrm{Var}(\delta n(t))}\sqrt{\mathrm{Var}(\delta M(t))} \tag{2.272}$$

$$= \begin{cases} \frac{1}{2}\left(\frac{\Delta \hat{n}}{\Delta t}\right)^2 \sigma^2_{\rho_p}\delta t\, t^2 & \text{White noise} \quad (2.273a) \\ \frac{1}{8}\left(\frac{\Delta \hat{n}}{\Delta t}\right)^2 \frac{\sigma^2_{\rho_p}}{\delta t} t^4 & \text{Brownian motion} \quad (2.273b) \end{cases}$$

GRID-SCALE MODEL UNCERTAINTY: ORNSTEIN-UHLENBECK

In case of average model uncertainty, the error process is best chosen as an Ornstein-Uhlenbeck process. Let $Y_t = \int_0^t X_s \, ds$, where X_s is the Ornstein-Uhlenbeck process including initial state randomization. Also let $Z_t = \int_0^t Y_s \, ds$ denote the twice time-integrated OUP with initial state randomization. Using this notation, it holds that $\mathrm{Cov}(\int_0^t \int_0^{t'} \epsilon_\rho(t'') \, dt'' \, dt', \int_0^t \epsilon_\rho(t') \, dt') = \mathrm{Cov}(Z_t, Y_t)$, which due to the zero-mean property of Z_t and Y_t is equal to the double integral of the unintegrated OUP and the single time-integrated OUP processes:

$$\mathrm{Cov}(Z_t, Y_u) = \int_0^t \int_0^u \mathrm{Cov}(Y_s, X_{s'}) \, ds' \, ds \tag{2.274}$$

Hence the description of the covariance one integration-level earlier is needed:

$$\mathrm{Cov}(Y_t, X_u) = \mathrm{E}[Y_t X_u] = \mathrm{E}\left[\int_0^t X_s \, ds \, X_u\right] \tag{2.275}$$

$$= \int_0^t \mathrm{E}[X_s X_u] \, ds = \int_0^t \mathrm{Cov}(X_s, X_u) \, ds \tag{2.276}$$

Equation 2.276 now allows to insert the expression for the classical OUP covariance (eq. 2.220), which results in:

$$\mathrm{Cov}(Y_t, X_u) = \frac{\sigma^2}{2\theta} \int_0^t e^{2\theta \min(s,u) - \theta(s+u)} - \underbrace{e^{-\theta(s+u)}}_{\substack{\text{Only in case of no} \\ \text{randomization of} \\ \text{the initial state}}} \, ds \tag{2.277}$$

To evaluate the minimum, the cases of $u \geqslant t$ and $t \geqslant u$ need to be analyzed separately.

Case 1: $u \geqslant t$

If $u \geqslant t$, there is no need to split the integral, as it always holds that $s \leqslant u$.

$$\begin{aligned} \mathrm{Cov}(Y_t, X_u) &= \frac{\sigma^2}{2\theta} \int_0^t e^{2\theta s - \theta(s+u)} \, ds = \frac{\sigma^2}{2\theta} \int_0^t e^{-\theta(u-s)} \, ds \\ &= \frac{\sigma^2}{2\theta} \left[\frac{1}{\theta}\left(e^{-\theta(u-t)} - e^{-\theta u}\right)\right] \\ &= \frac{\sigma^2}{2\theta^2}\left(e^{-\theta(u-t)} - e^{-\theta u}\right) \end{aligned} \tag{2.278}$$

Case 2: $t \geqslant u$
Since $t \geqslant u$ it is necessary to split the integral at $s = u$ to evaluate the minimum.

$$\begin{aligned}
\mathrm{Cov}(Y_t, X_u) &= \frac{\sigma^2}{2\theta}\left[\int_0^u e^{2\theta s-\theta(s+u)}\,\mathrm{d}s + \int_u^t e^{2\theta u-\theta(s+u)}\,\mathrm{d}s\right] \\
&= \frac{\sigma^2}{2\theta}\left[\int_0^u e^{-\theta(u-s)}\,\mathrm{d}s + \int_u^t e^{-\theta(s-u)}\,\mathrm{d}s\right] \\
&= \frac{\sigma^2}{2\theta^2}\left(-e^{-\theta(t-u)} - e^{-\theta u} + 2\right) \qquad (2.279)
\end{aligned}$$

Combining both cases yields a universal description of $\mathrm{Cov}(Y_t, X_u)$:

$$\mathrm{Cov}(Y_t, X_u) = \begin{cases} \frac{\sigma^2}{2\theta^2}\left(\quad e^{-\theta(u-t)} - e^{-\theta u}\right) & \text{if } t \leqslant u \\ \frac{\sigma^2}{2\theta^2}\left(-e^{-\theta(t-u)} - e^{-\theta u} + 2\right) & \text{if } u \leqslant t \end{cases} \qquad (2.280)$$

Equation 2.280 does not allow for a common description using min/max-expressions. Nevertheless it is valid regardless of the relationship between t and u, as long as the corresponding equation is used. This is also important in the next step, when equation 2.280 is inserted into equation 2.274 to evaluate the desired covariance. In principle two cases are possible again: $u \geqslant t$ and $t \geqslant u$. Since however this is the final integration and the last step will be to set $u = t$, it is theoretically sufficient to compute only one of the two cases, as they must yield the same result for $u = t$. Nevertheless, computing both cases is advised to use this property for validation.

Case 1: $u \geqslant t$
If $u \geqslant t$, only a single split of the inner integral is required at $s = u$:

$$\begin{aligned}
\mathrm{Cov}(Z_t, Y_u) &= \int_0^t \left[\int_0^s \underbrace{\mathrm{Cov}(Y_s, X_{s'})}_{s'<s}\,\mathrm{d}s' + \int_s^u \underbrace{\mathrm{Cov}(Y_s, X_{s'})}_{s'>s}\,\mathrm{d}s'\right]\mathrm{d}s \\
&= \frac{\sigma^2}{2\theta^2}\left[\int_0^t \left(\int_0^s -e^{-\theta(s-s')} - e^{-\theta s'} + 2\,\mathrm{d}s' \right.\right. \\
&\qquad \left.\left. + \int_s^u e^{-\theta(s'-s)} - e^{-\theta s'}\,\mathrm{d}s'\right)\mathrm{d}s\right] \\
&= \frac{\sigma^2}{2\theta^2}\left[\int_0^t \frac{1}{\theta}\left(-1 + e^{-\theta s}\right) + \frac{1}{\theta}\left(e^{-\theta s} - 1\right) + 2s \right. \\
&\qquad \left. - \frac{1}{\theta}\left(e^{-\theta(u-s)} - 1\right) + \frac{1}{\theta}\left(e^{-\theta u} - e^{-\theta s}\right)\mathrm{d}s\right]
\end{aligned}$$

$$= \frac{\sigma^2}{2\theta^3} \int_0^t e^{-\theta s} + e^{-\theta u} - e^{-\theta(u-s)} + 2s\theta - 1 \, ds$$

$$= \frac{\sigma^2}{2\theta^4} \left(t^2\theta^2 + e^{-\theta u}(t\theta + 1) - e^{-\theta t} - e^{-\theta(u-t)} - t\theta + 1 \right) \quad (2.281)$$

Setting $u = t$ in equation 2.281 now allows to obtain the desired description of the time-integrated and twice time-integrated relative density error OUP:

$$\mathrm{Cov}(Z_t, Y_t) = \frac{\sigma^2}{2\theta^4} \left(t^2\theta^2 + t\theta e^{-\theta t} - t\theta \right) \quad (2.282)$$

For the sake of completeness, $\mathrm{Cov}(Z_t, Y_u)$ is calculated in the following also for the second case, where $t \geqslant u$. When setting $t = u$ also this solution has to yield the same expression as equation 2.282.

Case 2: $t \geqslant u$
If $t \geqslant u$, both the outer and inner integrals need to be split:

$$\begin{aligned}
\mathrm{Cov}(Z_t, Y_u) = & \int_0^u \left[\int_0^s \underbrace{\mathrm{Cov}(Y_s, X_{s'})}_{s'<s} \, ds' + \int_s^u \underbrace{\mathrm{Cov}(Y_s, X_{s'})}_{s'>s} \, ds' \right] ds \\
& + \int_u^t \int_0^u \underbrace{\mathrm{Cov}(Y_s, X_{s'})}_{s'<s} \, ds' \, ds \\
= & \frac{\sigma^2}{2\theta^2} \left[\int_0^u \left(\int_0^s -e^{\theta(s-s')} - e^{-\theta s'} + 2 \, ds' + \int_s^u e^{-\theta(s'-s)} - e^{-\theta s'} \, ds' \right) ds \right. \\
& \left. + \int_u^t \int_0^u -e^{\theta(s-s')} - e^{-\theta s'} + 2 \, ds' \, ds \right] \\
= & \frac{\sigma^2}{2\theta^3} \left[\int_0^u e^{-\theta s} + e^{-\theta u} - e^{-\theta(u-s)} + 2s\theta - 1 \, ds \right. \\
& \left. + \int_u^t e^{-\theta s} + e^{-\theta u} - e^{-\theta(s-u)} + 2u\theta - 1 \, ds \right] \\
= & \frac{\sigma^2}{2\theta^4} \left(-u^2\theta^2 + e^{-\theta u}(t\theta + 1) - e^{-\theta t} + e^{-\theta(t-u)} - t\theta + 2u\theta^2 t - 1 \right) \quad (2.283)
\end{aligned}$$

Setting $u = t$ yields:

$$\mathrm{Cov}(Z_t, Y_t) = \frac{\sigma^2}{2\theta^4} \left(t^2\theta^2 + t\theta e^{-\theta t} - t\theta \right) \quad (2.284)$$

which equals the case of $u \geqslant t$ (eq. 2.282). Hence, the generic form of the covariance is given by equation 2.285.

$$\mathrm{Cov}(Z_t, Y_u) = \begin{cases} \frac{\sigma^2}{2\theta^4} \left(t^2\theta^2 + e^{-\theta u}(t\theta + 1) - e^{-\theta t} - e^{-\theta(u-t)} - t\theta + 1 \right) & \text{if } u \geqslant t \\ \frac{\sigma^2}{2\theta^4} \left(-u^2\theta^2 + e^{-\theta u}(t\theta + 1) - e^{-\theta t} + e^{-\theta(t-u)} - t\theta + 2u\theta^2 t - 1 \right) & \text{if } t \geqslant u \end{cases} \quad (2.285)$$

The corresponding correlation coefficient is found using equation 2.269 together with eqs. 2.282, 2.237 and 2.242, which yields:

$$\begin{aligned}\rho_{\delta n M} &= \frac{\frac{\sigma^2}{2\theta^4}\left(t^2\theta^2 + t\theta e^{-\theta t} - t\theta\right)}{\sqrt{\frac{\sigma^2}{2\theta^3}(2\theta t + 2e^{-\theta t} - 2)}\sqrt{\frac{\sigma^2}{2\theta^5}\left(\frac{2}{3}\theta^3 t^3 - \theta^2 t^2 - 2e^{-\theta t} - 2\theta t e^{-\theta t} + 2\right)}} \\ &= \frac{t^2\theta^2 - t\theta\left(1 - e^{-\theta t}\right)}{\sqrt{\left(\frac{2}{3}\theta^3 t^3 - \theta^2 t^2 - 2e^{-\theta t} - 2\theta t e^{-\theta t} + 2\right)(2\theta t + 2e^{-\theta t} - 2)}}\end{aligned} \tag{2.286}$$

The cross-correlation coefficient given by eq. 2.286 cannot be reduced further. It is therefore time-dependent (in contrast to the constant values of $\rho_{\delta n M} = \sqrt{12}/4$ for a white noise error process and $\rho_{\delta n M} = \sqrt{15}/4$ for a Brownian motion error process). For large t, the upper limit can be approximated using only the dominant terms, which results in:

$$\rho^*_{\delta n M} = \frac{t^2\theta^2}{\sqrt{\frac{4}{3}\theta^4 t^4}} = \frac{\sqrt{12}}{4} \tag{2.287}$$

The result of equation 2.287 equals the value obtained for a white noise error process. For $t \to 0$, the correlation coefficient converges against 1.0, as can be quickly verified using a computer algebra system like Mathematica®:

```
In[1]:= Limit[(θ^2 t^2 - t θ (1 - e^(-θ t))) / Sqrt[(2/3 θ^3 t^3 - θ^2 t^2 - 2 e^(-θ t) - 2 θ t e^(-θ t) + 2) (2 θ t + 2 e^(-θ t) - 2)], t -> 0, Direction -> "FromAbove", Assumptions -> θ >= 0]
Out[1]= 1
```

Figure 2.19: Limit of correlation coefficient for positive values of θ as t approaches zero from above

The evaluation of equation 2.286 on a computer leads to discontinuities due to floating point arithmetics for small t. Hence, the result of the expression should be bound to 1.0 for short propagation times when programmatically evaluated in the context of density uncertainty. A plot of equation 2.286 for different t and θ is given in figure 2.20. The discontinuities for small t can be clearly identified as spikes on the left boundary of the plot. The larger θ, the faster the value of $\rho_{\delta n M}$ approaches the white noise limit. For $\theta = 10^{-6}$ the cross-correlation is effectively 1.0 for all simulation periods within a couple of days. In case of $\theta = 10^{-4}$ however, the cross-correlation reaches the white noise limit after ~1.2 days.

Inserting equation 2.286 together with eqs. 2.237 and 2.242 into equation 2.272 yields the final mean motion error/mean anomaly error covariance for grid-scale OUP model uncertainty:

$$Cov(\delta n(t), \delta M(t)) = \frac{\sigma^2}{2\theta^4}\left(\frac{\Delta\hat{n}}{\Delta t}\right)^2\left(t^2\theta^2 + t\theta e^{-\theta t} - t\theta\right) \tag{2.288}$$

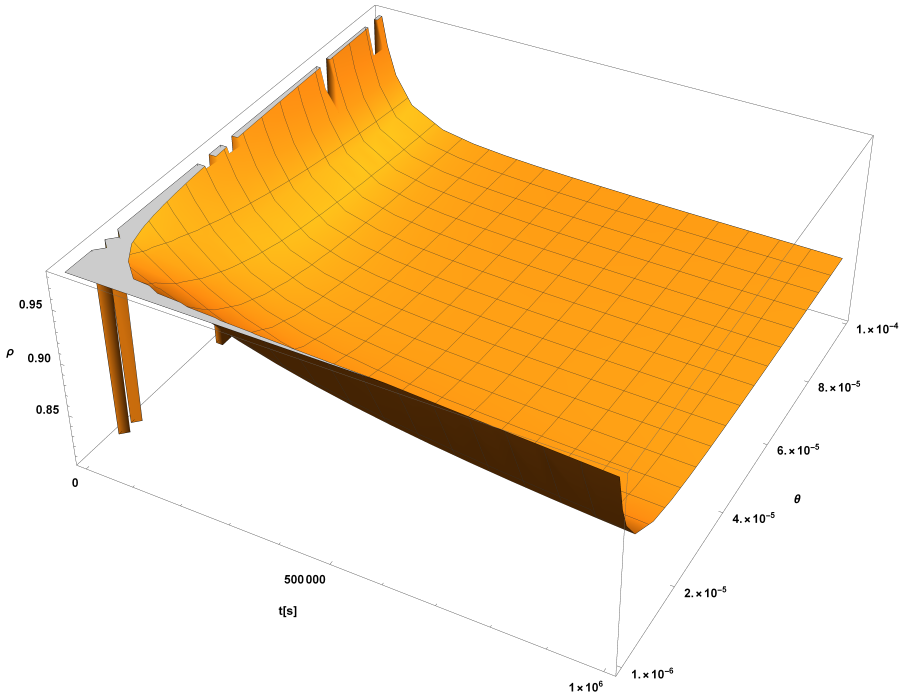


Figure 2.20: Time-dependent correlation coefficient of the time-integrated and twice time-integrated Ornstein-Uhlenbeck error processes with randomization of the initial state

2.7.2 *Conversion of density uncertainty to Cartesian coordinates*

Taking mean motion as the first and mean anomaly as the sixth orbital element, density uncertainty can be described via the sparse 6×6 covariance matrix depicted in equation 2.289.

$$\mathbf{Q}_{\mathrm{OE}} = \begin{pmatrix} \mathrm{Var}(\delta n) & 0 & 0 & 0 & 0 & \mathrm{Cov}(\delta n, \delta M) \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \mathrm{Cov}(\delta n, \delta M) & 0 & 0 & 0 & 0 & \mathrm{Var}(\delta M) \end{pmatrix} \tag{2.289}$$

Since only the mean motion and mean anomaly uncertainty is nonzero, the uncertainty in mean anomaly equals the uncertainty in mean longitude, such that the matrix computed via eq. 2.289 is valid for both classical orbital elements (COE) and equinoctial orbital elements (EOE).

Orbit determination is often conducted in inertial Cartesian coordinates using the Geocentric Celestial Reference Frame (GCRF). Consequently a method for converting the orbital element uncertainties to Cartesian coordinates is required. In Schiemenz et al. (2019c) simplified expressions are derived which approximately map the covariance matrix in classical orbital elements to GCRF coordinates. These

transformations however are not recommended anymore, as it is more precise to perform the conversion from COE/EOE to GCRF using unscented transformation (UT) or via the analytic Jacobians of the frame conversions. Both methods are briefly explained in the following.

UNSCENTED TRANSFORMATION

Unscented Transformation is the first step of the Unscented Kalman Filter (UKF). UT is a sampling based covariance transformation/propagation scheme. The details are presented in Julier and Uhlmann (2004). In principle, UT consists of three steps. First, special sampling points (so-called "sigma points") are selected. The second step is to transform each of these sigma points using the full nonlinear transformation function $f(\mathbf{x})$. The spread of the transformed sigma points and their weights are then used to construct an estimate of the transformed mean and covariance matrix. The principle is illustrated in figure 2.21. A major advantage of the UT is that it is free of derivatives. Consequently also very complex nonlinear transformations can be computed with the same algorithm as simple test cases. In contrast to a Monte-Carlo propagation, only few sampling points are needed. Different strategies have been developed, however the most common is the symmetric sigma point set that consists of $2n+1$ sigma points, where n denotes the state vector dimension. For a classical orbit determination $n = 6$, such that 13 sigma points are required. This directly leads to the downside of the method. If the nonlinear transformation is not analytic but represents a time-consuming numerical orbit propagation, the calculation of 13 propagations is computationally much more expensive than covariance transformation using Jacobi matrices. For analytic coordinate frame conversions however the performance difference is negligible.

To use UT for covariance conversion, the orbital elements covariance is first sampled to obtain the set of sigma points. Each point is then transformed using the state vector at time t and the full nonlinear transformation from COE/EOE to GCRF. Finally, an estimate of the transformed covariance is computed from the transformed sigma points. The process is mathematically summarized as:

$$\mathbf{Q}_{\mathrm{GCRF}} = \mathrm{UT}\left(\mathbf{Q}_{\mathrm{OE}}\right) \tag{2.290}$$

The UT transformation solution is not unique. Despite different options for the overall number of sigma points and their weighting, also the spread of the sigma points is not uniquely defined, but subject to a degree of freedom and can therefore be selected by the user. The choice of this spread-parameter α determines the quality of the transformation. In many cases the selection of α is not critical, however some cases exist for which additional testing is required to obtain a robust parametrization. This is also true for the orbital elements

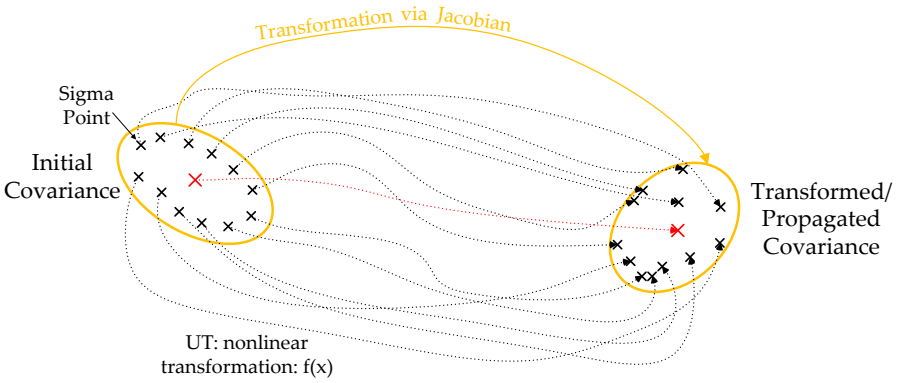


Figure 2.21: Jacobian and sigma-point based covariance transformation

to GCRF covariance conversion, as the mean motion covariance is typically multiple orders of magnitude smaller than the mean anomaly uncertainty and adopts values in the order of 10^{-20}–$10^{-18}\,rad^2/s^2$ after a few orbital revolutions, leading to UT covariance conversion results which critically depend on the spread of the sigma points and the numerical stability of the transformations.

JACOBI MATRICES

The transformation of equation 2.289 via Jacobians is more robust than the UT approach[4], however 36 analytic partial derivatives are required for the transformation. Over the past two decades David Vallado has authored two papers devoted to a summary of Jacobian coordinate frame conversions (Vallado, 2004; Vallado and Alfano, 2015). Both papers contain a couple of typographical errors, the second revision of Vallado and Alfano (2015) however is nevertheless a recommended reference to study the conversions and obtain further literature.

Kinematic state vector and covariance transformation routines between COE, EOE and GCRF were needed in many places throughout the work conducted as part of this thesis. Some examples are the conversion of equation 2.289 to GCRF, the developments presented in chapter 5, the extension of the Airbus-tool SPOOK to orbit estimation in EOE coordinates or the study of the coordinate frame relevance for the validity of the Gaussian pdf assumption described in section 1.3. Consequently the transformation routines between Cartesian and orbital element coordinates outlined in Vallado and Alfano (2015) have been implemented in computer code. Typographical errors have been corrected and transformations missing in Vallado and Alfano (2015) have been derived. The Fortran source code of the transformations may be found in appendix B.2.

4 Note that this statement is only appropriate for the transformation of equation 2.289. In general there are many examples where UT-based transformations show superior performance over the usage of Jacobians

It should be stressed that a Gaussian covariance in orbital elements only yields an approximate Gaussian in Cartesian GCRF coordinates if the covariance is sufficiently small. With increasing magnitude of the eigenvalues, a Gaussian probability density function in orbital elements results in non-Gaussian Cartesian pdfs, such that more advanced conversion techniques (e.g. Gaussian mixture models) are required. This case is further elaborated in chapter 5.

Given a complete 6×6 Jacobian $\mathbf{J}$, the orbital elements covariance can be transformed via:

$$\mathbf{Q}_{GCRF} = \mathbf{J}\,\mathbf{Q}_{OE}\,\mathbf{J}^T \tag{2.291}$$

where the superscript ^{T} denotes the matrix transpose. After conversion to Cartesian coordinates, the covariance matrix is no longer sparse.

2.7.3 *Covariance propagation and inclusion of additional perturbations*

The linear state error covariance propagation from time t_j to t_k, including process noise, is given by equation 2.292:

$$\hat{\mathbf{P}}_k = \mathbf{\Phi}(t_k, t_j)\hat{\mathbf{P}}_j\mathbf{\Phi}(t_k, t_j)^T + \mathbf{dQ}_k \tag{2.292}$$

where $\hat{\mathbf{P}}$ denotes the estimated state vector covariance matrix, $\mathbf{\Phi}(t_k, t_j)$ denotes the State error Transition Matrix (STM) that models the linear covariance propagation from time t_j to time t_k and $\mathbf{dQ}_k$ describes the process noise increment of the propagation step (e.g. due to force-model uncertainty). Equation 2.292 is independent of the selected coordinate frame. Formally the covariances refer to the kinematic state *error*. However, since OD methods internally use the state error as state vector and not the overall kinematic state, it is common to omit the error designation and use the terms "state vector covariance" and "state transition matrix". Special caution is required for the STM, as there are different formulations for the satellite state vector and its error. Both are treated in Vallado (2013), together with background information on the calculation. For OD however, the STM is almost always referring to the state error.

So far descriptions of $\mathbf{Q}_{OE}$ and $\mathbf{Q}_{GCRF}$ have been derived. Equations 2.289, 2.290 and 2.291 are special in nature, as they represent analytic descriptions of the state vector covariance with respect to time and therefore do not require STMs to facilitate the time propagation. This allows for fast and efficient computations of the density process noise covariance, however it complicates the sequential propagation scheme of Kalman filters. To perform incremental covariance propagation, a description of $\mathbf{dQ}_k$ is required that only makes use of $\mathbf{Q}_k$, $\mathbf{Q}_j$ and $\mathbf{\Phi}(t_k, t_j)$. Starting from time t_0, the propagation to time t_k is computed using:

$$\hat{\mathbf{P}}_k = \mathbf{\Phi}(t_k, t_0)\hat{\mathbf{P}}_0(t_k, t_0)^T + \mathbf{Q}(t_k) \tag{2.293}$$

Similarly the propagation to time t_{k+1} is expressed as:

$$\hat{\mathbf{P}}_{k+1} = \mathbf{\Phi}(t_{k+1}, t_0)\hat{\mathbf{P}}_0\mathbf{\Phi}(t_{k+1}, t_0)^T + \mathbf{Q}(t_{k+1}) \tag{2.294}$$

Equation 2.293 allows to state:

$$\mathbf{\Phi}(t_k, t_0)\hat{\mathbf{P}}_0\mathbf{\Phi}(t_k, t_0)^T = \hat{\mathbf{P}}_k - \mathbf{Q}(t_k) \tag{2.295}$$

The linearity of the STM yields:

$$\mathbf{\Phi}(t_{k+1}, t_0) = \mathbf{\Phi}(t_{k+1}, t_k)\mathbf{\Phi}(t_k, t_0) \tag{2.296}$$

Multiplying equation 2.295 with $\mathbf{\Phi}(t_{k+1}, t_k)$ from the left and with $\mathbf{\Phi}(t_{k+1}, t_k)^T$ from the right, results in:

$$\begin{aligned} &\mathbf{\Phi}(t_{k+1}, t_k)\mathbf{\Phi}(t_k, t_0)\hat{\mathbf{P}}_0\mathbf{\Phi}(t_k, t_0)^T\mathbf{\Phi}(t_{k+1}, t_k)^T \\ &= \mathbf{\Phi}(t_{k+1}, t_k)\left[\hat{\mathbf{P}}_k - \mathbf{Q}(t_k)\right]\mathbf{\Phi}(t_{k+1}, t_k)^T \end{aligned} \tag{2.297}$$

which can also be written as:

$$\mathbf{\Phi}(t_{k+1}, t_0)\hat{\mathbf{P}}_0\mathbf{\Phi}(t_{k+1}, t_0)^T = \mathbf{\Phi}(t_{k+1}, t_k)\left[\hat{\mathbf{P}}_k - \mathbf{Q}(t_k)\right]\mathbf{\Phi}(t_{k+1}, t_k)^T \tag{2.298}$$

by making use of eq. 2.296. Equation 2.294 on the other hand also allows to obtain an expression for $\mathbf{\Phi}(t_{k+1}, t_0)\hat{\mathbf{P}}_0\mathbf{\Phi}(t_{k+1}, t_0)^T$:

$$\mathbf{\Phi}(t_{k+1}, t_0)\hat{\mathbf{P}}_0\mathbf{\Phi}(t_{k+1}, t_0)^T = \hat{\mathbf{P}}_{k+1} - \mathbf{Q}(t_{k+1}) \tag{2.299}$$

Setting equation 2.299 equal to eq. 2.298 yields:

$$\begin{aligned} \hat{\mathbf{P}}_{k+1} - \mathbf{Q}(t_{k+1}) &= \mathbf{\Phi}(t_{k+1}, t_k)\left[\hat{\mathbf{P}}_k - \mathbf{Q}(t_k)\right]\mathbf{\Phi}(t_{k+1}, t_k)^T \\ \Leftrightarrow \hat{\mathbf{P}}_{k+1} &= \mathbf{\Phi}(t_{k+1}, t_k)\left[\hat{\mathbf{P}}_k - \mathbf{Q}(t_k)\right]\mathbf{\Phi}(t_{k+1}, t_k)^T \\ &\quad + \mathbf{Q}(t_{k+1}) \end{aligned} \tag{2.300}$$

$$\begin{aligned} \Leftrightarrow \hat{\mathbf{P}}_{k+1} &= \mathbf{\Phi}(t_{k+1}, t_k)\hat{\mathbf{P}}_k\mathbf{\Phi}(t_{k+1}, t_k)^T \\ &\quad - \mathbf{\Phi}(t_{k+1}, t_k)\mathbf{Q}(t_k)\mathbf{\Phi}(t_{k+1}, t_k)^T \\ &\quad + \mathbf{Q}(t_{k+1}) \end{aligned} \tag{2.301}$$

which can also be written as:

$$\hat{\mathbf{P}}_{k+1} = \mathbf{\Phi}(t_{k+1}, t_k)\hat{\mathbf{P}}_k\mathbf{\Phi}(t_{k+1}, t_k)^T + \mathbf{dQ}_{k+1} \tag{2.302}$$

where $\mathbf{dQ}_{k+1}$ is given by equation 2.303:

$$\mathbf{dQ}_{k+1} = \mathbf{Q}(t_{k+1}) - \mathbf{\Phi}(t_{k+1}, t_k)\mathbf{Q}(t_k)\mathbf{\Phi}(t_{k+1}, t_k)^T \tag{2.303}$$

Equations 2.302 and 2.303 allow to facilitate incremental covariance propagation using the analytical density uncertainty framework. Alternatively also eq. 2.300 can be used. In contrast to equation 2.294 both descriptions have the advantage that the covariance can be propagated in small time steps, instead of enforcing $t = t_0$ as starting time, which would be a lot more time-consuming.

Taking a closer look at equations 2.300 and 2.303, it turns out that $\mathbf{dQ}_{k+1}$ essentially removes any prior influence of the process noise on the state vector and then adds the new contribution for the time t_{k+1}. Since $\mathbf{\Phi}(t_{k+1}, t_k)$ has to be computed in any case when using linear covariance propagation, the only additional computation required at time t_{k+1} is $\mathbf{Q}(t_{k+1})$, which is evaluated analytically. Some caution is required when thinking of $\mathbf{dQ}_{k+1}$ as the process noise from time t_k to time t_{k+1}, since $\mathbf{dQ}_{k+1}$ represents a delta-covariance matrix which, unlike regular covariance matrices, is not required to be positive semidefinite (PSD).

INCLUSION OF ADDITIONAL PERTURBATIONS

The density uncertainty framework has been derived under the assumption of drag being the only perturbation acting on the satellite state. Operational applications however generally model also further kinds of disturbances, e.g. solar radiation pressure or a non-spherical gravity field. The formulation of density uncertainty however is also compatible with this operationally relevant case, given the proper perturbation settings when evaluating equation 2.302.

When performing the covariance propagation using arbitrary perturbations, it is important to compute $\mathbf{\Phi}(t_{k+1}, t_k)$ using the entire set of activated perturbations and not only drag. $\mathbf{Q}(t_{k+1})$ on the other hand needs to be computed using drag as the only perturbation, such that $\hat{n}(t)$ in the prefactor $\frac{\hat{n}(t)-\hat{n}_0}{\Delta t}$, which is found in all nonzero components of equation 2.289, requires a separate drag-only propagation.

UNSCENTED TRANSFORMATION

STMs are only used in case of linear covariance propagation via Jacobians. For sigma-point based propagation schemes, equation 2.303 cannot be used, however the general concept of adding $\mathbf{Q}(t_{k+1})$ to the version of $\hat{\mathbf{P}}_k$ which is free of any process-noise contributions can be easily generalized.

For the unscented Kalman Filter this operation principle is actually beneficial. Once the cloud of sigma-points is generated, it is to be propagated until the desired propagation end. This is efficiently implemented by saving the sigma-point cloud after every intermediate propagation step and then using this saved cloud to continue the propagation in the next step ("cloud-saving"), instead of recomputing the cloud from the reconstructed propagated covariance matrix every propagation step. Using cloud-saving with posterior addition of $\mathbf{Q}_{k+1}$ there is no need to subtract any prior additions of $\mathbf{Q}_k$, since the saved cloud never experiences any contribution of process noise. Consequently, using UT it is possible to propagate the covariance matrix according to equation 2.304:

$$\hat{\mathbf{P}}_{k+1} = \mathrm{UT}(f(\boldsymbol{\chi}_k), \boldsymbol{w}_k) + \mathbf{Q}_{k+1} \tag{2.304}$$

where $\boldsymbol{\chi}_k$ denotes the set of sigma points at the start of the propagation phase and $\boldsymbol{w}_k$ the corresponding weights vector.

Further details about covariance propagation in the framework of orbit determination are outlined in part III, when the force model uncertainty framework is incorporated into classical and modern orbit determination algorithms.

2.7.4 *Unified input and grid-scale density uncertainty*

Equation 2.289 allows to formulate a density uncertainty covariance matrix for each type of origin: solar flux input uncertainty, magnetic index input uncertainty and grid-scale model uncertainty. In the following the respective covariance matrices are denoted as $\mathbf{Q}_f$, $\mathbf{Q}_m$ and $\mathbf{Q}_g$ with the subscript indicating the origin. Considering the different origins of density uncertainty to be independent, the overall density uncertainty covariance can be obtained via summation:

$$\mathbf{Q} = \mathbf{Q}_f + \mathbf{Q}_m + \mathbf{Q}_g \tag{2.305}$$

In orbital elements (classical or equinoctial) $\mathbf{Q}$ has the structure shown in equation 2.306:

$$\mathbf{Q}(t) = \begin{pmatrix} \sum_{x\in f,m,g} \mathrm{Var}(\delta n_x(t)) & 0 & 0 & 0 & 0 & \sum_{x\in f,m,g} \mathrm{Cov}(\delta n_x(t), \delta M_x(t)) \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \sum_{x\in f,m,g} \mathrm{Cov}(\delta n_x(t), \delta M_x(t)) & 0 & 0 & 0 & 0 & \sum_{x\in f,m,g} \mathrm{Var}(\delta M_x(t)) \end{pmatrix} \tag{2.306}$$

A final point that has been noted on page 107 applies to the computation of $\mathbf{Q}_f$ and $\mathbf{Q}_m$, i.e. input density uncertainty. Both covariances rely on equation 2.83 to obtain an expression of $\sigma^2_{\rho_p}$, as required by equations 2.212-2.215. The white noise process model expects $\sigma^2_{\rho_p}$ to be constant. Similarly, $\sigma^2_{\rho_p}$ is also expected to be constant for the Brownian motion error process, as it represents the variance of the density uncertainty of the underlying white noise process (i.e. the observed density uncertainty after the first characteristic sampling step). Equation 2.83 on the other hand can be used to obtain varying estimates of the relative density error variance at the perigee altitude throughout the course of the propagation duration. As noted on page 107, variations in $\sigma^2_{\rho_p}$ should be filtered on time scales shorter than an orbital revolution to obtain a constant value used for the next propagation phase during orbit determination.

A myriad of options exist to perform this filtering. Some examples are averaging over the propagation duration, averaging per orbital revolution, taking the minimum/maximum/median value of $\sigma^2_{\rho_p}$, and so forth. Also deciding for different strategies depending on the origin

(solar flux/magnetic index) or error process model (white noise/Brownian motion) is possible, giving rise to empirical optimization. The work conducted in this thesis makes use of an empirical strategy to perform the $\sigma^2_{\rho_p}$ filtering, however averaging over the propagation duration also turned out to yield accurate results in most cases.

2.8 SUMMARY OF DENSITY UNCERTAINTY QUANTIFICATION EQUATIONS

Table 2.7 lists the most important equations that have been derived throughout this chapter. Orbit determination simulation results are presented in part III of this thesis, when the force model uncertainty framework is integrated into classical and modern OD algorithms.

<table>
<tr><td rowspan="5">Relative density uncertainty</td><td colspan="2">Input uncertainty</td><td>Grid-scale model uncertainty</td></tr>
<tr><td>Solar flux</td><td>$Var(\delta G_i)$: eq. 2.90
$Cov(\delta G_i, \delta G_j)$: eq. 2.92</td><td rowspan="4">$s_{Model} = 10\% - 15\%$</td></tr>
<tr><td rowspan="2">Geom. index</td><td>Class: $Var(\delta G_i)$: eq. 2.99
MSIS $Cov(\delta G_i, \delta G_j)$: eq. 2.102</td></tr>
<tr><td>Class: $Var(\delta kp)$: eq. 2.112
DTM $Var(\delta G_i)$: eq. 2.127
$Cov(\delta G_i, \delta G_j)$: eq. 2.131</td></tr>
<tr><td colspan="2">$Var(\epsilon_\rho)$: eq. 2.83, needs to be evaluated at the perigee altitude along the track to obtain $Var(\epsilon_{\rho_p})$</td></tr>
<tr><td rowspan="4">Orbital uncertainty</td><td colspan="3">Error process</td></tr>
<tr><td>White noise</td><td>Brownian motion</td><td>Ornstein-Uhlenbeck</td></tr>
<tr><td>$Var(\delta n(t))$: eq. 2.212
$Var(\delta M(t))$: eq. 2.213</td><td>$Var(\delta n(t))$: eq. 2.214
$Var(\delta M(t))$: eq. 2.215</td><td>$Var(\delta n(t))$: eq. 2.247
$Var(\delta M(t))$: eq. 2.248</td></tr>
<tr><td>$Cov(\delta n(t), \delta M(t))$: eq. 2.273a</td><td>$Cov(\delta n(t), \delta M(t))$: eq. 2.273b</td><td>$Cov(\delta n(t), \delta M(t))$: eq. 2.288</td></tr>
<tr><td rowspan="3">Unified covariance</td><td colspan="2">Covariance combination</td><td>Eq. 2.306</td></tr>
<tr><td colspan="2">Covariance propagation</td><td>Linear: eq. 2.300 or eqs. 2.302 and 2.303
UT: eq. 2.304</td></tr>
<tr><td colspan="2">Covariance frame conversion</td><td>Linear: eq. 2.291, appendix B.2
UT: eq. 2.290</td></tr>
</table>

Table 2.7: Summary of essential density uncertainty equations

3

GRAVITY UNCERTAINTY

Corresponding publications: Schiemenz et al. (2020c)

Besides density uncertainty also Earth's non-spherical gravity field may act as a notable source of uncertainty in low Earth orbits. As outlined in section 1.3.3, gravity fields are nowadays highly precise. Accurate field modeling on the other hand requires large amounts of computation time, such that it is customary to perform orbit determination with truncated potential fields, especially in case of real-time applications. For LEO satellites the IERS recommends 90×90 gravity fields. Practical applications however often treat the additional accuracy for improved runtime by setting a lower truncation degree/order, e.g. 36×36 or 30×30. Gravity uncertainty is therefore predominantly caused by potential field truncation (errors of omission) and to a much lesser extent by the uncertainty of the field coefficients (errors of commission).

Errors of omission are of epistemic nature, since it is possible to acquire better knowledge of the potential field by increasing the truncation point, whereas errors of commission are aleatoric. In the following, epistemic errors of omission are treated by stochastic, i.e. aleatoric, means. This decision is rooted in the existence of fundamental prior work, including the availability of the complete orbit gravity acceleration error autocovariance matrix. In particular, gravity errors of omission and commission will be treated by a unified process noise matrix. Using process noise theory allows for an efficient algorithm capable of running in real-time and faster than real-time applications. Alternatively, uncertain gravitational coefficients could be added to the estimation state space or their effects could be accounted for as consider parameters. These solutions however require a large amount of additional computations and increased computer memory due to the extensive number of gravitational coefficients involved. The choice to use a stochastic process noise representation of the gravitation acceleration errors is therefore an engineering decision that can be justified through the validation of the resulting model.

The chapter is structured as follows: after a brief overview of prior work on gravity uncertainty in section 3.1, section 3.2 is devoted to a presentation of the theory that leads to the only state of the art algorithm capable of running in real-time (Wright et al., 2008a,b,c). Subsequently, section 3.3 is devoted to the development of an improved gravity uncertainty covariance estimation algorithm that builds upon the derivations presented in section 3.2, but is more robust and yields superior performance. The computed orbital gravity error covariances seamlessly integrate with orbit determination algorithms capable of

incorporating process noise. Numerical validation results, including techniques for a proper validation of errors of omission, are presented in section 3.5.

3.1 PRIOR RESEARCH ON ORBITAL GRAVITY UNCERTAINTY

The theory of statistical gravity error covariance dates back to 1959, when William Kaula published his "Statistical and harmonic analysis of gravity", in which he studied gravity errors of omission on the Earth's surface (Kaula, 1959). In his work, Professor Kaula published expressions for the gravity acceleration errors in the Radial/In-track/Cross-Track (RIC) frame[1] due to errors of omission, i.e. potential function truncation. The corresponding theory on statistical geodesy has been worked out later in detail by authors such as Heiskanen and Moritz (1967) or Moritz (1980).

In 1967 Gersten et al. transferred Kaula's theory to orbital altitudes and derived approximations of the resulting orbital element covariances via full sphere averaging (Gersten et al., 1967). Some years later, in 1978, James Wright attempted to numerically integrate the double integral which describes the propagation of the acceleration error covariance function to the orbital state vector (Wright et al., 2008a). However, long runtimes and matrix asymmetries showed that direct numerical integration of the acceleration error covariance is not suitable for operational applications, such as orbit determination.

To solve these problems, Wright recognized the need to separate the inner from the outer integration to efficiently compute the double integral. In 1981 he published the theory behind this separation together with a first sequential orbit determination algorithm capable of considering autocorrelated gravity modeling errors (Wright, 1981). Wright also defined the degree variance due to errors of commission, i.e. the potential field coefficient uncertainty, in his work of 1981. Over a decade later, Wright finally published the details of a key step in his 1981-derivation in Wright (1994).

More than two decades after the presentation of the original sequential estimation algorithm, Wright et al. published an improved algorithm architecture in a three-paper series (Wright et al., 2008a,b,c). Reference Wright et al. (2008a) explains the main theory of the improved algorithm and outlines the steps towards obtaining the orbit gravity error covariance matrix. Wright et al. (2008b) is devoted to the validation of the filter covariance matrix and Wright et al. (2008c) focuses on the computation of the inner integral (**I1**) approximation using polynomials or splining functions.

1 The term "in-track" does not apply correctly for nonzero eccentricity and always refers to transverse. For consistency with Wright et al. (2008a,b,c) the terminology is used nonetheless. The frame is also known as Radial/Transverse/Normal (RTN)

This three-paper algorithm (henceforth called "WK algorithm" for Wright-Kaula algorithm) has proven to be beneficial for long-term filter stability (Vallado, 2013, p.777), however Wright et al. (2008b) also showed that the double integrated gravity acceleration error covariance matrices are only realistic for a limited time (a couple of orbits) and that there is room for further improvements.

Primarily, the WK algorithm suffers from an underestimation of the in-track error component of the position covariance. Also the radial and normal position components only agree in some cases with the observed (numerically propagated) covariance. In Wright et al. (2008b) only the plots of the position error covariances for errors of commission are given, however these deficiencies were found to gain momentum when also errors of omission are applicable, i.e. if the gravity field potential function is truncated for the purpose of orbit propagation (Schiemenz et al., 2020c).

Before improvements to the WK algorithm may be developed, its theory first needs to be studied to identify the root causes of these shortcomings.

3.2 THEORETICAL BACKGROUND: THE WK ALGORITHM

The orbit error covariance matrix $\mathbf{P}_{k+1,k}$ is defined by equation 3.1,

$$\mathbf{P}_{k+1,k} = E\left[\delta\mathbf{x}(t_{k+1}|t_k)\delta\mathbf{x}(t_{k+1}|t_k)^T\right] = E\left[\delta\mathbf{x}_{k+1,k}\delta\mathbf{x}_{k+1,k}^T\right] \quad (3.1)$$

where $\delta\mathbf{x}$ is the state vector error that represents the difference between the true and the estimated state vectors. The WK algorithm considers a Cartesian representation (position and velocity vectors) of the satellite state. Therefore $\delta\mathbf{x}$ denotes the vector of the position and velocity errors. Equation 3.2 describes the linear model for propagating $\delta\mathbf{x}$ with time (Wright et al., 2008a):

$$\delta\mathbf{x}(t_{k+1}|t_k) = \mathbf{\Phi}(t_{k+1},t_k)\delta\mathbf{x}(t_k|t_k) + \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1},\tau)\mathbf{G}(\tau)\delta\mathbf{g}(\tau)\,d\tau \quad (3.2)$$

where

- $\mathbf{\Phi}(t_{k+1},t_k)$ is the 6×6 State error Transition Matrix (STM) which linearly propagates $\delta\mathbf{x}$ from time t_k to time t_{k+1}
- $\delta\mathbf{g}(\tau)$ is the 3×1 vector describing the gravity acceleration error at time τ
- $\mathbf{G}(\tau)$ is the 6×3 matrix which translates the gravity acceleration error from its frame of definition to the inertial frame used for computing $\mathbf{P}_{k+1,k}$. The upper 3×3 part of $\mathbf{G}(t)$ is known to be zero from theory (Wright et al., 2008a). The lower 3×3 part is the transformation matrix, which performs the Radial/Transverse/Normal (RTN) to inertial transformation[2]

2 See Vallado (2013, Eq. 3.20) or Wright et al. (2008c) for the required steps to obtain this frame transformation matrix

Inserting equation 3.2 into eq. 3.1 gives:

$$\mathbf{P}_{k+1,k} = \mathbf{\Phi}(t_{k+1}, t_k)\mathbf{P}_{k,k}\mathbf{\Phi}(t_{k+1}, t_k)^T + \mathbf{P}^Q_{k+1,k} \tag{3.3}$$

where

$$\mathbf{P}^Q_{k+1,k} = \mathbf{I}^C_{k+1,k} + \mathbf{I}^L_{k+1,k} + \mathbf{I}^R_{k+1,k} \tag{3.4}$$

is formed from three covariance components:

$$\begin{aligned}\mathbf{I}^C_{k+1,k} &= \left(\mathbf{I}^C_{k+1,k}\right)^T \\ &= \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1}, \tau)\ \mathbf{G}(\tau)\mathrm{E}\left[\delta\mathbf{g}(\tau)\delta\mathbf{g}(t)^T\right]\mathbf{G}(t)^T\mathbf{\Phi}(t_{k+1}, t)^T\,\mathrm{d}\tau\,\mathrm{d}t\end{aligned} \tag{3.5}$$

$$\mathbf{I}^L_{k+1,k} = \mathbf{\Phi}(t_{k+1}, t_k) \int_{t_k}^{t_{k+1}} \mathrm{E}\left[\delta\mathbf{x}(t_k|t_k)\delta\mathbf{g}(t)^T\right]\mathbf{G}(t)^T\mathbf{\Phi}(t_{k+1}, t)^T\,\mathrm{d}t \tag{3.6}$$

$$\mathbf{I}^R_{k+1,k} = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1}, \tau)\ \mathbf{G}(\tau)\mathrm{E}\left[\delta\mathbf{g}(\tau)\delta\mathbf{x}(t_k|t_k)^T\right]\mathrm{d}\tau\ \mathbf{\Phi}(t_{k+1}, t_k)^T \tag{3.7}$$

The evolution of the orbit error $\delta\mathbf{x}$ is formally non-Markov, since $\mathbf{I}^L_{k+1,k}$ depends on a time-integral containing $\mathrm{E}\left[\delta\mathbf{x}(t_k|t_k)\delta\mathbf{g}(t)^T\right]$ (with $t_k \leqslant t \leqslant t_{k+1}$), where $\delta\mathbf{x}(t_k|t_k)$ itself depends on prior gravity acceleration errors (cf. eq. 3.11) and therefore introduces unlimited history. Kalman filters however are constructed using Markov chains as underlying dynamics model, meaning that the probability at each time step is assumed to only depend on the probability of the previous step. This allows swift calculations and leads to computational tractability, as propagations are minimized. Hence, making orbit gravity errors compatible with sequential orbit estimation requires an approximately Markovian formulation of $\mathbf{P}^Q_{k+1,k}$. If to this end an approximation of $\mathbf{I}^L_{k+1,k}+\mathbf{I}^R_{k+1,k}$ was available which does not require past gravity acceleration errors, the orbital gravity error process would be approximately Markovian.

To ease understanding of the subsequent assumptions and simplifications, it makes sense to graphically depict the integration domains, as has been done in figure 3.1. The first term of equation 3.3 propagates the uncertainty at time t_k to time t_{k+1} and hence encapsulates the entire uncertainty between time t_0 and time t_k. $\mathbf{I}^C_{k+1,k}$ is the part of the propagated covariance that accounts for the contribution of gravity errors that occur between t_k and t_{k+1}. The Markov-boundary is depicted in figure

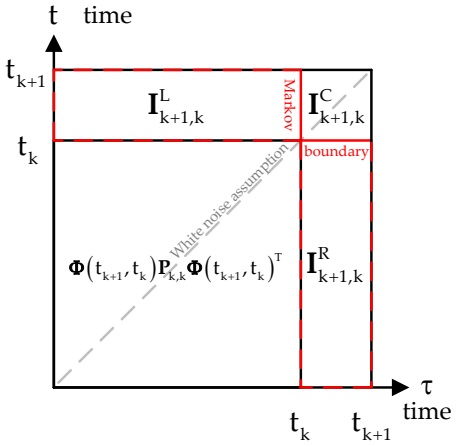


Figure 3.1: Orbit gravity error covariance: initial integration domain

3.1 by continuous red lines. As $\mathbf{I}^{R}_{k+1,k}$ and $\mathbf{I}^{L}_{k+1,k}$ depend on prior gravity error information, they reach beyond this boundary, which makes orbital gravity errors formally non-Markov and is indicated graphically by the hatched integration domains.

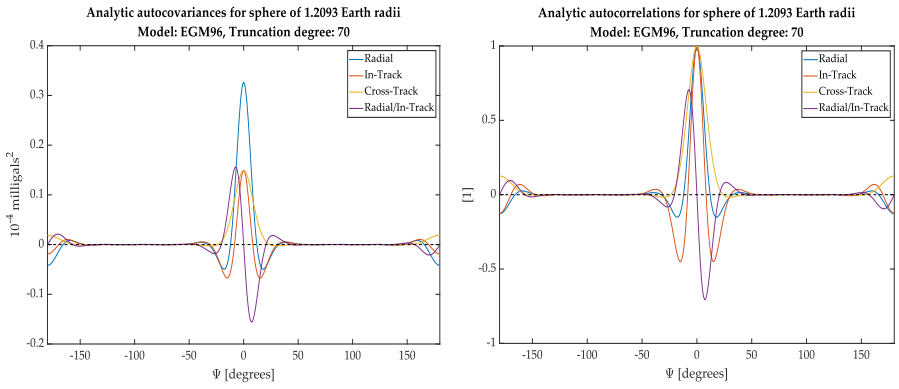
The well-known white noise approximation would be to assume $\mathbf{I}^{R}_{k+1,k} = \left(\mathbf{I}^{L}_{k+1,k}\right)^{T} = \mathbf{0}$ and set $E\left[\delta\mathbf{g}(\tau)\delta\mathbf{g}(t)\right] = \mathbf{Q}\,\delta(\tau - t)$ where δ denotes the Dirac delta function. The integration over $\delta(\tau - t)$ would then reduce the two-dimensional time integral in equation 3.5 into a one-dimensional integral, which can graphically be depicted as the diagonal gray dashed line at $t = \tau$ in figure 3.1. As the autocorrelation function of gravity errors however is not close to white noise (see e.g. figure 3.2b), the white noise hypothesis needs to be rejected and autocorrelated gravity errors need to be considered.

The next step in the derivation was undocumented between 1981 and 1994, since Wright (1981), after introducing two assumptions (eqs. 3.24 and 3.25) that allow to make the process Markovian, directly states that "with the aid of the assumptions of integral symmetries it is straightforward" to arrive at equation 3.30. The fact that 13 years later Wright devoted a 20 page conference paper, reference Wright (1994), to the justification of this "step" however indicates that it is probably not too straightforward.

The key idea of Wright (1994) to transform the orbital gravity error process into a Markov process is to first convert $\mathbf{I}^{L}_{k+1,k}$ and $\mathbf{I}^{R}_{k+1,k}$ into descriptions that depend on the gravity acceleration error autocovariance instead of $E\left[\delta\mathbf{x}(t_k)\delta\mathbf{g}(t)^{T}\right]$ and then switching the inner integration from the time domain into the Earth central angle, such that the integration over prior gravity acceleration errors then corresponds to an integration over prior Earth central angles.

The derivation proceeds according to Wright (1994). The state vector error at time t_k is formulated according to the absolute error definition:

$$\delta\mathbf{x}(t_k|t_k) = \delta\mathbf{x}_{k|k} = \mathbf{x}_{k|k} - \hat{\mathbf{x}}_{k|k} \tag{3.8}$$



(a) Autocovariances (b) Autocorrelations

Figure 3.2: Typical shapes of gravity acceleration error autocovariances and corresponding autocorrelations

The estimated state at time t_k, $\hat{\mathbf{x}}_{k|k}$, is the a priori state $\hat{\mathbf{x}}_{k|0}$ corrected by a measurement update (if available):

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|0} + \mathbf{K}_k(\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|0})) \tag{3.9}$$

where $\mathbf{K}_k$ represents the Kalman gain, $\mathbf{z}_k$ the measurement vector and $h(\hat{\mathbf{x}}_{k|0})$ denotes the estimated observation based on the nonlinear measurement function h and the estimated initial state vector. Inserting equation 3.9 into eq. 3.8 and defining the a priori error at time t_k as $\delta\mathbf{x}_{k|0} = \mathbf{x}_{k|k} - \hat{\mathbf{x}}_{k|0}$, as well as $\delta\mathbf{z}_{k|0} = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|0})$, yields:

$$\delta\mathbf{x}_{k|k} = \delta\mathbf{x}_{k|0} - \mathbf{K}_k\delta\mathbf{z}_{k|0} \tag{3.10}$$

Inserting equation 3.2 for $\delta\mathbf{x}_{k|0}$ with t_0 as initial time and t_k as target time into eq. 3.10 results in:

$$\delta\mathbf{x}_{k|k} = \mathbf{\Phi}_{k,0}\delta\mathbf{x}_{0|0} + \int_{t_0}^{t_k} \mathbf{\Phi}_{k,\omega}\mathbf{G}(\omega)\delta\mathbf{g}(\omega)\,d\omega - \mathbf{K}_k\delta\mathbf{z}_{k|0} \tag{3.11}$$

Equation 3.11 allows to relate the a priori error at time t_k to the prediction of an initial error, the contribution of gravity acceleration errors between t_0 and t_k and a measurement update at t_k. The expected value $E\left[\delta\mathbf{x}(t_k)\delta\mathbf{g}(t)^T\right]$ can hence be written as:

$$E\left[\delta\mathbf{x}(t_k)\delta\mathbf{g}(t)^T\right] = \mathbf{\Phi}_{k,0}E\left[\delta\mathbf{x}_{0|0}\delta\mathbf{g}(t)^T\right] + \int_{t_0}^{t_k} \mathbf{\Phi}_{k,\omega}\mathbf{G}(\omega)E\left[\delta\mathbf{g}(\omega)\mathbf{g}(t)^T\right]d\omega \tag{3.12}$$

since $E\left[\delta\mathbf{z}_{k|0}\delta\mathbf{g}(t)^T\right] = 0$ (Wright, 1994). Wright reasons next that the second term of equation 3.12 dominates the first. This claim is rooted in the property of the gravity acceleration autocorrelation function to level off rapidly around any value of t (cf. figure 3.2b). Due to the time gap between t_0 and t ($t_0 < t_k$ and $t_k \leqslant t \leqslant t_{k+1}$), the expected value $E\left[\delta\mathbf{x}_{0|0}\delta\mathbf{g}(t)^T\right]$ in the first term of equation 3.12 is hence assumed to be dominated by the second term. Consequently, $E\left[\delta\mathbf{x}(t_k)\delta\mathbf{g}(t)^T\right] \cong \int_{t_0}^{t_k} \mathbf{\Phi}_{k,\omega}\ \mathbf{G}(\omega)E\left[\delta\mathbf{g}(\omega)\mathbf{g}(t)^T\right]d\omega$ which, upon insertion into eq. 3.6, leads to:

$$\mathbf{I}^L_{k+1,k} \cong \int_{t_k}^{t_{k+1}}\int_{t_0}^{t_k} \mathbf{\Phi}(t_{k+1},\omega)\ \mathbf{G}(\omega)E\left[\delta\mathbf{g}(\omega)\delta\mathbf{g}(t)^T\right]\mathbf{G}(t)^T\mathbf{\Phi}(t_{k+1},t)^T\,d\omega\,dt \tag{3.13}$$

Similarly it follows for $\mathbf{I}^R_{k+1,k}$:

$$\mathbf{I}^R_{k+1,k} \cong \int_{t_k}^{t_{k+1}}\int_{t_0}^{t_k} \mathbf{\Phi}(t_{k+1},\tau)\mathbf{G}(\tau)E\left[\delta\mathbf{g}(\tau)\delta\mathbf{g}(\omega)^T\right]\mathbf{G}(\omega)^T\mathbf{\Phi}(t_{k+1},\omega)^T\,d\tau\,d\omega \tag{3.14}$$

Eqs. 3.5, 3.13 and 3.14 now contain the same integrand and only differ in the integration boundaries, which motivates further approximations of equation 3.4 that are based on the modification of integral boundaries. Equations 3.13 and 3.14 are not yet Markovian, as they still depend on information prior t_k.

As a first step, the lower integral bounds of $\mathbf{I}^{R}_{k+1,k}$ and $\mathbf{I}^{L}_{k+1,k}$ are changed from t_0 to a constant offset T with respect to the integration variable. The constant offsets cut off the total integration domain parallel to the Kalman line in a distance of $t-T$ and $\tau-T$. This also influences the shape of $\mathbf{\Phi}_{k+1,k}\mathbf{P}_{k,k}\mathbf{\Phi}^{T}_{k+1,k}$, as depicted in figure 3.3. Mathematically the inner integral limits are changed as follows:

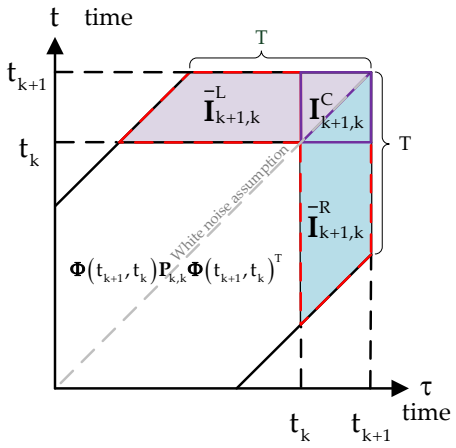


Figure 3.3: Orbit gravity error covariance: intermediate integration domain

$$\bar{\mathbf{I}}^{L}_{k+1,k} \cong \int_{t_k}^{t_{k+1}} \int_{t-T}^{t_k} \mathbf{K}(\tau,t)\,d\tau\,dt \tag{3.15}$$

and:

$$\bar{\mathbf{I}}^{R}_{k+1,k} \cong \int_{t_k}^{t_{k+1}} \int_{\tau-T}^{t_k} \mathbf{K}(t,\tau)\,dt\,d\tau \tag{3.16}$$

where $\mathbf{K}$ denotes the common integrand of eqs. 3.5, 3.13 and 3.14.

Splitting $\mathbf{I}^{C}_{k+1,k}$ at $t=\tau$ and assigning the resulting triangles to the domains $\bar{\mathbf{I}}^{L}_{k+1,k}$ and $\bar{\mathbf{I}}^{R}_{k+1,k}$ allows to formulate the purple ($\widetilde{\mathbf{I}}^{L}_{k+1,k}$) and blue ($\widetilde{\mathbf{I}}^{R}_{k+1,k}$) shaded integration domains:

$$\widetilde{\mathbf{I}}^{L}_{k+1,k} \cong \int_{t_k}^{t_{k+1}} \int_{t-T}^{t} \mathbf{K}(\tau,t)\,d\tau\,dt \tag{3.17}$$

$$\widetilde{\mathbf{I}}^{R}_{k+1,k} \cong \int_{t_k}^{t_{k+1}} \int_{\tau-T}^{\tau} \mathbf{K}(t,\tau)\,dt\,d\tau \tag{3.18}$$

Next, introducing a change of variables that replaces the absolute integration times with the time difference

$$\gamma = t-\tau \Leftrightarrow t = \tau+\gamma \Rightarrow d\gamma = dt \tag{3.19}$$

allows to transform the integration bounds with respect to the time difference γ:

$$\widetilde{\mathbf{I}}^{R}_{k+1,k} \cong \int_{t_k}^{t_{k+1}} \int_{-T}^{0} \mathbf{K}(\tau+\gamma,\tau)\,d\gamma\,d\tau \tag{3.20}$$

For orbital motion, time differences are directly linked to changes in the orbital mean anomaly. However, for gravity acceleration errors it is more typical to express the angular difference as a change in the Earth central angle Ψ than in mean anomaly. Regardless of the naming, angular changes are connected to time differences via the orbital mean motion n:

$$\Psi = n\gamma \tag{3.21}$$

Transferring the integration domain of the inner integral from time to Earth central angle results in:

$$\widetilde{\mathbf{I}}^{R}_{k+1,k} \cong \int_{t_k}^{t_{k+1}} \left[\frac{1}{n} \int_{-T\cdot n}^{0} \mathbf{K}\left(\tau + \frac{\Psi}{n}, \tau\right) d\Psi \right] d\tau \tag{3.22}$$

Considering only the direct time dependence, equation 3.22 is now Markovian, since it only depends on time information between t_k and t_{k+1}. Prior time information is converted to prior positional information via the Earth central angle and the orbital mean motion. As prior position information is linked to the current time t_k via orbit propagation, this conceptually solves the problem of satisfying the Markov property.

As can be inferred from figure 3.2a, $E\left[\delta\mathbf{g}(\tau)\delta\mathbf{g}(t)^T\right]$ consists only of even and odd functions in the Earth central angle. Wright (1994) stresses that $\widetilde{\mathbf{I}}^{L}_{k+1,k} \neq \left(\widetilde{\mathbf{I}}^{R}_{k+1,k}\right)^T$ due to the presence of odd functions in the integrand $\mathbf{K}$, which are integrated to zero in $\mathbf{I}^{C}_{k+1,k}$, however nonzero for each of the triangles building $\mathbf{I}^{C}_{k+1,k}$. Consequently, it must be made sure that the computation of the inner integral guarantees that the odd functions integrate to zero. This is achieved by transitioning to the final integration domain depicted in figure 3.4 (Wright, 1994). The computation of the final integral according to figure 3.4 is realized by expanding the upper bound in equation 3.20 to $T \cdot n$:

$$\mathbf{P}^{Q}_{k+1,k} \cong \int_{t_k}^{t_{k+1}} \left[\frac{1}{n} \int_{-T\cdot n}^{T\cdot n} \mathbf{K}\left(\tau + \frac{\Psi}{n}, \tau\right) d\Psi \right] d\tau \tag{3.23}$$

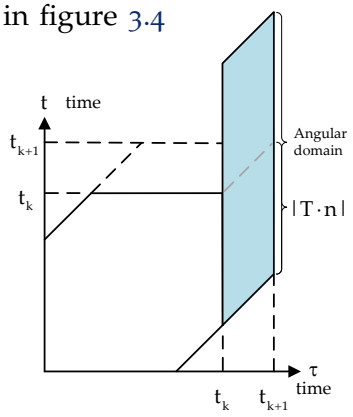


Figure 3.4: Orbit gravity error covariance: final integration domain

Despite guaranteeing proper handling of odd functions, equation 3.23 is only an approximation of the intermediate integration domain depicted in figure 3.3. The corresponding assumptions for their equivalence are (Wright, 1981):

$$\bar{\mathbf{I}}^{R}_{k+1,k} \cong \left(\bar{\mathbf{I}}^{L}_{k+1,k}\right)^T \tag{3.24}$$

and

$$\bar{\mathbf{I}}^{L}_{k+1,k} \cong \bar{\mathbf{I}}^{L}_{k+2,k+1} \tag{3.25}$$

A point that has not been discussed so far is the choice of the time constant T. An upper limit for orbital motion is half the orbital period ($T < P/2$) to keep the inner integral limits of equation 3.23 within one orbital revolution. To avoid transients when the integration is close to completing a full revolution, Wright (1981) suggests to choose $2/9 \leqslant T/P \leqslant 4/9$, which corresponds to 80-160 degrees difference in Earth central angle. A discussion of this limit is part of section 3.3.

The derivation presented so far builds upon Wright (1981, 1994) and demonstrates the need to a) modify integral boundaries and b) transition from time to space to make the solution conceptually Markovian. Finally, the choice of T introduces some additional variability. A significantly faster derivation is possible, if formal rigidity is slightly relaxed. This path is explained in the following and applied in Schiemenz et al. (2020c) to swiftly obtain equation 3.30. The following three properties are key to the subsequent reasoning:

- The dependence on prior gravity acceleration errors in equation 3.4 is limited to at most half of the orbital period
- The constituents of equation 3.4 (eqs. 3.5, 3.13 and 3.14) share the same integrand
- The autocorrelation integrals tend to form constant plateaus after integrating over differences in the Earth central angle that correspond to a fraction of the orbital revolution (typically less than a quarter of an orbit[3])

These characteristics motivate the general idea to first approximate $\mathbf{I}^{L}_{k+1,k} + \mathbf{I}^{R}_{k+1,k}$ as being close to zero and then to compensate for the significant part around $t = \tau$ by adding sufficient margin to the inner integration boundaries of $\mathbf{I}^{C}_{k+1,k}$.

Dropping $\mathbf{I}^{L}_{k+1,k} + \mathbf{I}^{R}_{k+1,k}$ results in:

$$\begin{aligned}\mathbf{P}^{Q}_{k+1,k} &\cong \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1},\tau)\mathbf{G}(\tau)\mathrm{E}\left[\delta\mathbf{g}(\tau)\delta\mathbf{g}(t)^{T}\right]\mathbf{G}(t)^{T}\mathbf{\Phi}(t_{k+1},t)^{T}\,\mathrm{d}\tau\,\mathrm{d}t \\ &\cong \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1},\tau)\,\mathbf{G}(\tau)\left[\int_{t_k}^{t_{k+1}} \mathrm{E}\left[\delta\mathbf{g}(\tau)\delta\mathbf{g}(t)^{T}\right]\mathbf{G}(t)^{T}\mathbf{\Phi}(t_{k+1},t)^{T}\,\mathrm{d}t\right]\mathrm{d}\tau \end{aligned} \quad (3.26)$$

Using equation 3.19 to transition the inner integral from absolute times to the time difference γ, results in:

$$\begin{aligned}\mathbf{P}^{Q}_{k+1,k} \cong \int_{t_k}^{t_{k+1}} &\mathbf{\Phi}(t_{k+1},\tau)\,\mathbf{G}(\tau) \\ &\left[\int_{t_k-\tau}^{t_{k+1}-\tau} \mathrm{E}\left[\delta\mathbf{g}(\tau)\delta\mathbf{g}(\tau+\gamma)^{T}\right]\mathbf{G}(\tau+\gamma)^{T}\mathbf{\Phi}(t_{k+1},\tau+\gamma)^{T}\,\mathrm{d}\gamma\right]\mathrm{d}\tau \end{aligned} \quad (3.27)$$

3 Wright (1981) suggests stability after two ninth of an orbital revolution, i.e. 80 degrees

To ease notation, define $\mathbf{H}(t_{k+1}, t) = \mathbf{\Phi}(t_{k+1}, t)\mathbf{G}(t)$ and $\mathbf{R}(\tau, \tau+\gamma) = E[\delta\mathbf{g}(\tau)\delta\mathbf{g}(\tau+\gamma)^T]$. Then:

$$\mathbf{P}^{Q}_{k+1,k} \cong \int_{t_k}^{t_{k+1}} \mathbf{H}(t_{k+1}, \tau) \left[\int_{t_k-\tau}^{t_{k+1}-\tau} \mathbf{R}(\tau, \tau+\gamma)\mathbf{H}(t_{k+1}, \tau+\gamma)^T \, d\gamma \right] d\tau \quad (3.28)$$

It is a happy fact that for near-circular orbits $\mathbf{H}(t_{k+1}, t)$ is only slowly varying (Wright, 1981). Thus, $\mathbf{H}(t_{k+1}, \tau+\gamma)^T$ is next assumed to be constant with respect to γ, which leads to:

$$\mathbf{P}^{Q}_{k+1,k} \cong \int_{t_k}^{t_{k+1}} \mathbf{H}(t_{k+1}, \tau) \left[\int_{t_k-\tau}^{t_{k+1}-\tau} \mathbf{R}(\tau, \tau+\gamma) \, d\gamma \right] \mathbf{H}(t_{k+1}, \tau)^T \, d\tau \quad (3.29)$$

Spherically averaged gravity acceleration error autocovariance matrices are only sensitive to changes in the Earth central angle. Consequently $\mathbf{R}$ does not depend on absolute times or absolute positions and is therefore stationary. Hence $\mathbf{R}(\tau, \tau+\gamma) = \mathbf{R}(0, \gamma) = \mathbf{R}(\gamma)$. In equation 3.29 the inner integral bounds depend on τ, which presently ranges from t_k to t_{k+1}. To complete the decoupling of the inner from the outer integral, this link has to be broken. For low Earth orbits Gravity acceleration error autocorrelation integrals tend to converge to a near constant value after a fraction of the orbital period. When reformulating the inner integral bounds it hence makes sense to choose a timeframe that corresponds to a difference in the Earth central angle which is sufficiently large to capture this plateau. As increasing the inner integral domain beyond the value of Ψ for which it stabilizes does not change the result of the integral, the bounds can be safely expanded until the value of T suggested in the Wright (1981, 1994) derivation. Hence gravity errors may be approximated via:

$$\mathbf{P}^{Q}_{k+1,k} = \int_{t_k}^{t_{k+1}} \mathbf{H}(t_{k+1}, t) \left[\int_{-T}^{T} \mathbf{R}(\gamma) \, d\gamma \right] \mathbf{H}(t_{k+1}, t)^T \, dt \quad (3.30)$$

Graphically depicted, the reasoning that was applied to arrive at equation 3.30 starting from eq. 3.29 essentially broadens the integration domain of $\mathbf{I}^{C}_{k+1,k}$ in figure 3.1 to arrive at figure 3.4. The reasoning behind this integral bounds expansion is not as rigorous as Wright's derivation, however it makes use of practical experience to arrive at the same result. A similar integral bounds modification was also applied in Gersten et al. (1967) and motivated by the fact that the integral limits should be chosen based on the maximum possible angular separation between orbital positions, which corresponds to half of the orbital period.

Apart from the separation of $\mathbf{H}$ and the inner integral (which is also the subsequent step in the original derivation after obtaining eq. 3.23), equations 3.30 and 3.23 represent two sides of the same coin. The

only visual difference is that equation 3.23 explicitly formulates the inner integral in the angular domain and then uses $1/n$ as prefactor to convert back to time, whereas equation 3.30 contains the equivalent expression of the inner integral in the time domain.

The WK algorithm assumes $\mathbf{R}(\gamma)$ to be diagonal, i.e. free of cross-correlations (this assumption is relaxed in section 3.3). In this case the gravity acceleration error autocovariance can be decomposed into a product of the instantaneous autocovariance $\mathbf{R}(0)$ and the 3×3 gravity acceleration error autocorrelation function $\boldsymbol{\rho}(\gamma)$:

$$\mathbf{R}(\gamma) = \mathbf{R}(0)\boldsymbol{\rho}(\gamma) \tag{3.31}$$

Inserting equation 3.31 into eq. 3.30 leads to:

$$\mathbf{P}^{Q}_{k+1,k} = \int_{t_k}^{t_{k+1}} \mathbf{H}(t_{k+1}, t)\, \mathbf{R}(0, t)\, \mathbf{I1}\, \mathbf{H}(t_{k+1}, t)^T \, dt \tag{3.32}$$

where $\mathbf{I1} = \int_{-T}^{T} \boldsymbol{\rho}(\gamma)\, d\gamma$. To avoid the computationally expensive numerical evaluation of the outer integral in equation 3.29, Wright et al. (2008a) use a mean value theorem to approximate the integrand as an iterated sum ("Riemann partitioning"). The time granularity for the Riemann partitioning is chosen as $\Delta \leqslant P/36$ with P denoting the orbital period (Wright, 1981), which for LEO objects leads to the recommendation of $\Delta = 2$ min found in Wright et al. (2008a). In its final form, the WK algorithm therefore uses equation 3.33 to approximate the orbit gravity error covariance matrix $\mathbf{I2} = \mathbf{P}^{Q}_{k+1,k}$:

$$\mathbf{I2} = \Delta \sum_{j=0}^{n-1} \mathbf{K}(t_{k+1}, \tau_j + \Delta/2) \tag{3.33}$$

where

$$\begin{aligned} \mathbf{K}(t_{k+1}, \tau_j + \Delta/2) = {} & \boldsymbol{\Phi}(t_{k+1}, \tau_j + \Delta/2)\, \mathbf{G}(\tau_j + \Delta/2) \\ & \mathbf{R}(0, \tau_j + \Delta/2)\, \mathbf{I1}(\tau_j + \Delta/2) \\ & \mathbf{G}(\tau_j + \Delta/2)^T\, \boldsymbol{\Phi}(t_{k+1}, \tau_j + \Delta/2)^T \end{aligned} \tag{3.34}$$

together with $\tau_0 = t_k$ and $\tau_{j+1} = \tau_j + \Delta$.

The inner integral **I1** is separated from the outer integral **I2** in equations 3.32 and 3.33. This allows to precompute **I1** with respect to altitude for a certain gravity model, which results in a matrix of time constants[4]. Since the matrices are sensitive to the orbital altitude, the position part of the state vector, $\mathbf{r}(t)$, is required for evaluation. The computation of the inner and outer integrals is outlined in the following.

4 The inner integral computations are performed in the angular domain, which conceptually satisfies the Markov property. The results are then converted from the angular domain to time using the orbital mean motion as indicated by equation 3.23

INNER INTEGRAL COMPUTATION

Both $\mathbf{R}(\Psi = 0, \mathbf{r}(t))$ and $\mathbf{I1}(\mathbf{r}(t))$ derive from the Kaula-Gersten-Pechenick theory, which defines the RIC/RTN gravity acceleration error variances by averaging $E\left[\delta\mathbf{g}(\mathbf{r}(\tau))\delta\mathbf{g}(\mathbf{r}(t))^T\right]$ over all possible products separated by the arc distance Ψ (Gersten et al., 1967; Wright et al., 2008a). This makes $\mathbf{R}(\Psi = 0, \mathbf{r}(t))$ only sensitive to angular/time differences but not to absolute position/time information.

Acceleration error variances

The RIC acceleration error variances for orbital altitudes have been given in Gersten et al. (1967):

$$\sigma^2_{RR}(\Psi) = \sum_{n=2}^{N_{max}} \left[\frac{n+1}{n-1}\right]^2 \left(\frac{a_E}{r}\right)^{2n+4} P_{n0}(\cos\Psi)\ \sigma^2_n \tag{3.35}$$

$$\sigma^2_{II}(\Psi) = \frac{1}{2}\sum_{n=2}^{N_{max}} \left[\frac{n(n+1)}{(n-1)^2}\right] \left(\frac{a_E}{r}\right)^{2n+4} \left[P_{n0}(\cos\Psi) - \frac{P_{n2}(\cos\Psi)}{n(n+1)}\right]\sigma^2_n \tag{3.36}$$

$$\sigma^2_{CC}(\Psi) = \frac{1}{2}\sum_{n=2}^{N_{max}} \left[\frac{n(n+1)}{(n-1)^2}\right] \left(\frac{a_E}{r}\right)^{2n+4} \left[P_{(n-1)0}(\cos\Psi) + \frac{P_{(n-1)2}(\cos\Psi)}{n(n+1)}\right]\sigma^2_n \tag{3.37}$$

where

- P_{n0} is the associated Legendre function[5] of degree n and order 0. Note that $P_{nm} = 0$ if $m > n$ by definition, which occurs for $n = 2$ in σ^2_{CC}
- Ψ is the Earth central angle
- a_E is the Earth's mean equatorial radius
- N_{max} is the maximum considered model degree (full-field degree)
- σ^2_n is the degree variance valid for the current value of the running index n

Degree variances

Degree variances define the contribution of a certain potential field degree to the acceleration error covariances. Degree variances are either due to errors of omission (potential field truncation) or errors of commission (uncertainty of the considered coefficients). The uncertainty of truncated degrees is generally not considered (no errors of commission on truncated degrees). For a certain $n = m$, σ^2_n is therefore defined as (Wright et al., 2008a):

$$\sigma^2_n = \begin{cases} \sigma^2_T, & \text{if } n > N_T \\ \sigma^2_C, & \text{otherwise} \end{cases} \tag{3.38}$$

5 Geodesy typically uses the definition of the associated Legendre functions without the Condon-Shortley phase $(-1)^m$. Since however only even orders are included in the autocovariances, it makes no difference if the Condon-Shortley phase is included or not

where N_T is the truncation degree ($N_T \leqslant N_{max}$) and equals the degree/order used for the orbit propagation. The variances σ_T^2 and σ_C^2 are defined as:

$$\sigma_T^2(n) = \left[\frac{\mu(n-1)}{a_E^2}\right]^2 \sum_{m=0}^{n} (\bar{C}_{nm}^2 + \bar{S}_{nm}^2) \tag{3.39}$$

$$\sigma_C^2(n) = \left[\frac{\mu(n-1)}{a_E^2}\right]^2 \sum_{m=0}^{n} (E[\delta\bar{C}_{nm}^2] + E[\delta\bar{S}_{nm}^2]) \tag{3.40}$$

$\bar{C}_{nm}^2$ and $\bar{S}_{nm}^2$ denote fully normalized coefficients, as typically found in the gravity potential field coefficient files. The constant μ represents the product of the universal gravitational constant and the mass of Earth.

Equations 3.35, 3.36 and 3.37 define the diagonal components of the gravity acceleration error covariance matrix and require equations 3.38, 3.39 and 3.40 for evaluation. The derivation of equation 3.35 is shown in Gersten et al. (1967). Work on the derivation of the off-diagonal components started in the 1980's, first by Kay Pechenick and later by Eric Delaye. As noted by Pechenick (1988), most of her research was performed in 1982, however not published before 1988. Delaye's work on the analysis of the effect of gravity model errors on the state error covariance was published in 1986 (Delaye, 1986). Delaye and Pechenick used different approaches for the derivations, however their results agree up to the sign of the radial-/in-track covariance. Pechenick pointed out that both solutions are correct and that the sign depends on the definition of the in-track direction when deriving σ_{RI}^2 (Pechenick, 1988). The proper selection of the sign is discussed in section 3.3.

The symmetric off-diagonal components, where the minus-sign corresponds to the Pechenick solution and the plus-sign to the Delaye solution, read as follows:

$$\sigma_{RI}^2(\Psi) = \pm\frac{1}{2}\sum_{n=2}^{N_{max}} \left[\frac{n(n+1)^2}{(n-1)^2}\right] \left(\frac{a_E}{r}\right)^{2n+4} \left[P_{(n-1)0}(\cos\Psi) + \frac{P_{(n-1)2}(\cos\Psi)}{n(n+1)}\right] \sin(\Psi)\ \sigma_n^2 \tag{3.41}$$

$$\sigma_{IR}^2 = \sigma_{RI}^2 \tag{3.42}$$

Delaye and Pechenick also showed that any other off-diagonal component has zero contribution (Delaye, 1986; Pechenick, 1988). This is by no means an obvious result and the respective derivations are lengthy. Hence:

$$\sigma_{CR}^2 = \sigma_{RC}^2 = 0 \tag{3.43}$$

$$\sigma_{CI}^2 = \sigma_{IC}^2 = 0 \tag{3.44}$$

Summarizing the results, the 3×3 gravity acceleration error covariance matrix is computed according to equation 3.45:

$$\mathbf{R}(\Psi) = \begin{bmatrix} \sigma^2_{RR}(\Psi) & \sigma^2_{RI}(\Psi) & 0 \\ \sigma^2_{RI}(\Psi) & \sigma^2_{II}(\Psi) & 0 \\ 0 & 0 & \sigma^2_{CC}(\Psi) \end{bmatrix} \tag{3.45}$$

Wright (1981) already noticed that it is beneficial to decompose the autocovariance matrix into an instantaneous component ($\mathbf{R}(\Psi = 0)$) and an autocorrelation matrix ($\boldsymbol{\rho}(\Psi)$). The generic decomposition equations are given in Wright et al. (2008a):

$$\boldsymbol{\rho}(\Psi) = \mathbf{R}(0)^{-1/2}\mathbf{R}(\Psi)\mathbf{R}(0)^{-1/2} \tag{3.46}$$

and

$$\mathbf{R}(\Psi) = \mathbf{R}(0)^{1/2}\boldsymbol{\rho}(\Psi)\mathbf{R}(0)^{1/2} \tag{3.47}$$

As reasoned in Wright et al. (2008a) and shown in section 3.3, $\mathbf{R}(0)$ is highly dependent on the current Earth central distance r, whereas the changes in $\boldsymbol{\rho}(\Psi)$ with respect to r are comparatively small. For $\mathbf{R}(\Psi = 0)$ the off-diagonal components in equation 3.45 vanish, which yields:

$$\begin{aligned} \mathbf{R}(0) = \operatorname{diag}\Bigg(& \sum_{n=2}^{N_{max}} \left[\frac{n+1}{n-1}\right]^2 \left(\frac{a_E}{r}\right)^{2n+4} \sigma_n^2\,, \\ & \frac{1}{2}\sum_{n=2}^{N_{max}} \left[\frac{n(n+1)}{(n-1)^2}\right] \left(\frac{a_E}{r}\right)^{2n+4} \sigma_n^2\,, \\ & \frac{1}{2}\sum_{n=2}^{N_{max}} \left[\frac{n(n+1)}{(n-1)^2}\right] \left(\frac{a_E}{r}\right)^{2n+4} \sigma_n^2 \Bigg) \end{aligned} \tag{3.48}$$

To compute $\mathbf{I1}$, it is necessary to evaluate:

$$\mathbf{I1} = \frac{1}{n} \underbrace{\int_{-T\cdot n}^{T\cdot n} \rho(\Psi)\, d\Psi}_{\substack{\text{Evaluation in} \\ \text{angular domain} \\ \text{(radians)}}} \Leftrightarrow \frac{1}{n}\frac{2\pi}{360\,\text{deg}} \underbrace{\int_{-T\cdot n}^{T\cdot n} \rho(\Psi)\, d\Psi}_{\substack{\text{Evaluation in} \\ \text{angular domain} \\ \text{(degrees)}}} \Leftrightarrow \underbrace{\int_{-T}^{T} \rho(\gamma)\, d\gamma}_{\substack{\text{Evaluation in} \\ \text{time domain}}} \tag{3.49}$$

One of the central parts of the WK algorithm is the approximation of the diagonal components of $\mathbf{I1}$ by time constants. These values are found by detecting plateaus in the autocorrelation integrals and a subsequent transformation from angle to time. In the WK algorithm, $\mathbf{I1}$ is only computed for positive values of Ψ. The result is then multiplied by two due to the symmetry of the diagonal components of $\mathbf{I1}$ with respect to the ordinate. This computation is performed for a range of r values, e.g. $r \in (1.0, 2.0)a_E$. The resulting curves are then approximated by techniques of curve fitting (e.g. polynomial interpolation, Fourier series interpolation or cubic splining). The curve coefficients are finally stored together with the degree variances to compute $\mathbf{I1}$ and $\mathbf{R}(0)$ as part of the OD algorithm.

OUTER INTEGRAL I2 COMPUTATION

The covariance matrix $\mathbf{I2} = \mathbf{Q}_F(t_{k+1}, k)$ needs to be computed as part of the OD covariance propagation phase. Given times t_k and t_{k+1}, it is therefore required to evaluate equation 3.33 to obtain the corresponding orbit gravity error covariance matrix. This involves the computation of $\mathbf{\Phi}(t_{k+1}, \mathbf{r}(t))$, $\mathbf{G}(\mathbf{r}(t))$, $\mathbf{R}(0, \mathbf{r}(t))$ (based on the precomputed degree variances) and $\mathbf{I1}(\mathbf{r}(t))$ (based on the curve-fit coefficients) for each discontinuous circular arc of duration Δ between t_k and t_{k+1}.

3.3 IMPROVED ORBIT GRAVITY ERROR COVARIANCE

In the past, the WK algorithm has proven its positive effects on the long-term filter stability, together with its real-time capability, which is why it has been integrated into the software product ODTK of Analytical Graphics Inc. (AGI). Nevertheless, the primary shortcomings of the WK algorithm remain its underestimation of the in-track position covariance, along with a lack of robustness in the radial and normal direction components for increased propagation durations.

These shortcomings shall now be fixed by revisiting the theory of the **I1** computation. Essentially, two steps are required to solve these main issues and improve covariance realism.

3.3.1 *Underestimation of in-track position error covariance and radial velocity error covariance*

In the WK algorithm **I1** is a diagonal matrix of the following shape:

$$\mathbf{I1}_{WK} = \mathrm{diag}(T_{RR}, 0.0^*, T_{CC}) \tag{3.50}$$

where T_{RR} and T_{CC} denote the correlation times from the evaluation of the curve fits of the respective autocorrelation integrals. The value of 0.0 for the in-track direction arises from the fact that the in-track acceleration error components annihilate themselves after integration, as explained in Wright et al. (2008a).

Using this form of **I1** leads to an underestimation of the in-track position error covariance, which is coupled to the radial velocity error covariance. This is depicted in figure 3.5 that shows the case of an artificial LEO reference satellite at an altitude of 511 km and a propagation duration of 4.2 days (instead of only a couple of orbits as shown in Wright et al., 2008b). GGM02C is used as geopotential model and truncated at degree/order: $N_T = 30$. Similar to Wright et al. (2008b), 1000 numerical propagations are performed to validate the covariance bounds. In each Monte-Carlo iteration the orientation orbit parameters are changed to validate the errors of omission. This validation strategy is explained in section 3.5.

* The WK algorithm recommends a value of 10^{-10} deg scaled to time for numerical stability. 0.0 is the value to be known correct from theory (Wright et al., 2008a)

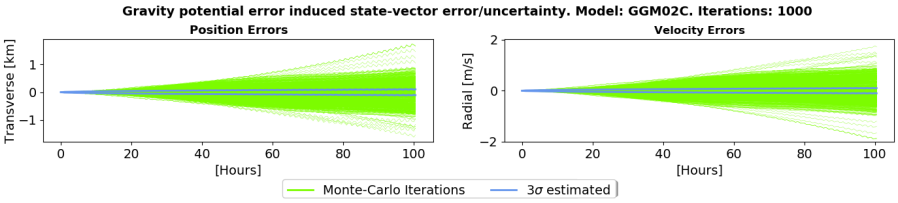


Figure 3.5: Underestimation of the WK algorithm: the actual errors (green) are not contained within the predicted 3σ bounds

While the diagonal components given in equation 3.50 can successfully be used to approximate the respective components of the inner integral, the WK algorithm completely neglects the RI acceleration error cross-correlation terms (equations 3.41 and 3.42).

In fact, Wright et al. (2008a) quote the results derived by Pechenick and the corresponding generic decomposition of $\mathbf{R}(\Psi)$ into $\mathbf{R}(0)$ and $\boldsymbol{\rho}(\Psi)$ (equations 3.46 and 3.47), however it is explicitly stated that this decomposition "enables the definition of a *diagonal* 3×3 matrix autocorrelation function", whereas equation 3.46 does not result in a diagonal matrix if $\sigma^2_{RI} = \sigma^2_{IR}$ is taken into account.

Numerical simulations revealed the root-cause of these underestimations to be the neglect of the RI cross-correlations in $\mathbf{I1}$ (Schiemenz et al., 2020c). These cross-correlations are required for an accurate estimation of the in-track position error and radial velocity error covariances, despite the fact that the in-track acceleration errors cancel during integration. The neglected RI cross-correlations need to be treated in a similar, but slightly different manner as the diagonal terms in order to solve the covariance underestimations, which is shown in the following.

σ^2_{RR}, σ^2_{II} and σ^2_{CC} are symmetric with respect to the ordinate, however σ^2_{RI} is pointwise symmetric with respect to the origin due to the dependency of equation 3.41 on the sine-function. This is depicted in figure 3.6, in which each curve represents σ^2_{RI} for an altitude between $1.0\,a_E$ and $2.0\,a_E$, spaced in equal distances of $20\,\mathrm{km}$.

The symmetry of σ^2_{RR}, σ^2_{II} and σ^2_{CC} with respect to the ordinate make the cumulative integration of the corresponding autocorrelation functions independent of the integration direction (Ψ from $-\pi$ to π or vice versa). For σ^2_{RI} however this is not the case due to its different symmetry properties. This is also the reason why the (inaccurate) WK integration strategy (Ψ from 0 to π) would not hold for ρ_{RI}, as obtained from equation 3.46. Instead, it is required to choose the integration bounds such that the RI plateaus are revealed. For ρ_{RI} the bounds are shifted by π from $[-\pi,\pi]$ to $[0,2\pi]$. The sign of the integral needs to be chosen according to the definition of σ^2_{RI} in equation 3.41 ($-$ for the Pechenick form and $+$ for the Delaye form). The minus sign in the Pechenick-form essentially changes the integration bounds to $[2\pi,0]$, which reflects the opposite definition of the in-track direction with respect to Delaye.

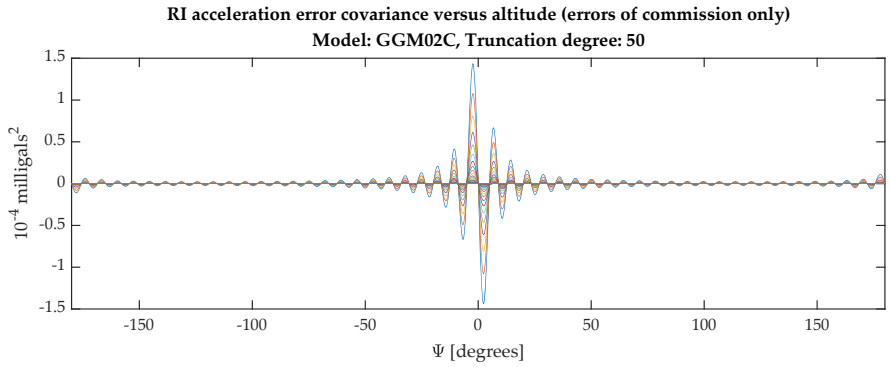


Figure 3.6: σ^2_{RI} versus altitude for $N_T = 50$ and GGM02C as obtained using the Pechenick form of equation 3.41

Consequently, the RI cross-correlation integral is obtained according to equation 3.51,

$$I_{RI} = \pm \int_0^{2\pi} \rho_{RI}(\Psi)\, d\Psi \tag{3.51}$$

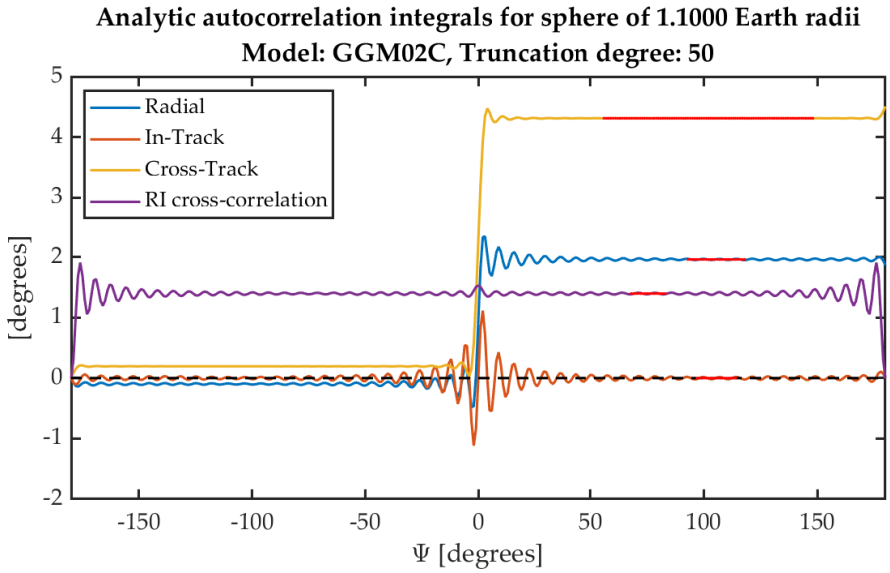
which leads to graphs similar to figure 3.7a for each altitude and truncation degree (in figure 3.7a the result of the integration for I_{RI} has been shifted by $-\pi$ to facilitate the comparison). The plot containing the altitude range $1.0a_E$ to $2.0a_E$ is given in figure 3.7b.

A key finding that has been uncovered after the publication of Schiemenz et al. (2020c) is the physical relationship of the sign choice with respect to the propagation direction when computing I_{RI}. Forward propagation requires positive cross correlation time constants, whereas backward propagation requires negative time constants and hence a change of the sign. The precomputation of the inner integral curve coefficients can hence be performed using either the Pechenick or Delaye formulation with subsequent integration according to equation 3.51, which leads to positive correlation times. The real-time part of the orbit gravity error covariance estimation algorithm then needs to check the propagation direction and reverse the sign of T_{RI} if necessary.

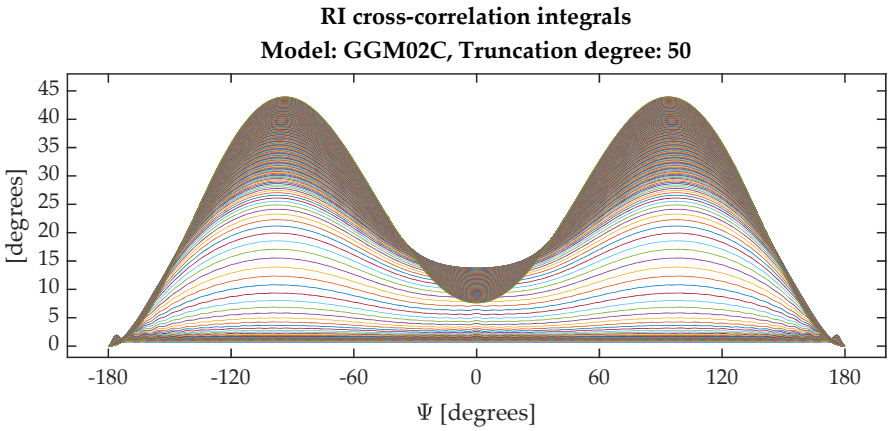
With the inclusion of the radial/in-track cross-correlation, the multiplicative decomposition of the instantaneous autocovariance and the inner integral matrix $\mathbf{I1}$ takes the form given in equation 3.52.

$$\begin{aligned}\mathbf{R}(0,r)^{1/2}\mathbf{I1}(r)\mathbf{R}(0,r)^{1/2} = {} & \operatorname{diag}\left(R(0,r)_{RR}^{1/2}, R(0,r)_{II/CC}^{1/2}, R(0,r)_{II/CC}^{1/2}\right) \\ & \cdot \begin{bmatrix} T_{RR}(r) & T_{RI}(r) & 0 \\ T_{RI}(r) & 0 & 0 \\ 0 & 0 & T_{CC}(r) \end{bmatrix} \cdot \\ & \operatorname{diag}\left(R(0,r)_{RR}^{1/2}, R(0,r)_{II/CC}^{1/2}, R(0,r)_{II/CC}^{1/2}\right)\end{aligned} \tag{3.52}$$

Equation 3.52 is to be used instead of eq. 3.31, with time implicitly defined by $\mathbf{r}(t)$ in equation 3.32, if the RI cross-correlation is taken into



(a) Plot of equation 3.51 for $r = 1.1a_E$, $N_T = 50$ and GGM02C (purple) along with the results for I_{RR}, I_{II}, I_{CC}. Detected plateaus are highlighted in red



(b) Plot of equation 3.51 for GGM02C and $N_T = 50$. Each curve corresponds to the cumulative integration for a certain altitude within the range $1.0a_E$ and $2.0a_E$

Figure 3.7: RI cross-correlation integral evaluation and approximation using time constants, similar to I_{RR} and I_{CC} in the WK algorithm

account, as eq. 3.31 only holds for $\mathbf{R}(0,t)$ and $\mathbf{I1}(t)$ being diagonal. Also note that $\mathbf{R}(0,r)_{CC}$ and $\mathbf{R}(0,r)_{II}$ are equal from theory, as shown in equation 3.48. The updated definition of equation 3.34 therefore reads:

$$\begin{aligned}\mathbf{K}(t_{k+1},\tau_j+\Delta/2) = {} & \mathbf{\Phi}(t_{k+1},\tau_j+\Delta/2)\ \mathbf{G}(\tau_j+\Delta/2) \\ & \mathbf{R}(0,\tau_j+\Delta/2)^{1/2}\ \mathbf{I1}(\tau_j+\Delta/2)\ \mathbf{R}(0,\tau_j+\Delta/2)^{1/2} \\ & \mathbf{G}(\tau_j+\Delta/2)^T\ \mathbf{\Phi}(t_{k+1},\tau_j+\Delta/2)^T \end{aligned} \tag{3.53}$$

with $I1(1,2) = I1(2,1) \neq 0$ due to the RI acceleration error covariances. The explicit consideration of the RI acceleration error cross-correlation is the single measure required to fix the underestimation of the WK

algorithm in the in-track position error and the radial velocity error components, as shown in section 3.5.

An example of the inner integral curve fits via cubic splining is given in figure 3.8. Due to the splining technique, the original data-points are exactly met, which results in zero root mean square error (RMSE) and is why the evaluated splines cannot be distinguished from the original data (this is not the case for polynomial curve fits). The lower subplot of figure 3.8 demonstrates a splining of the $\mathbf{R}(0)$ diagonal elements, which may be used to increase the algorithm performance, as explained in section 3.4.

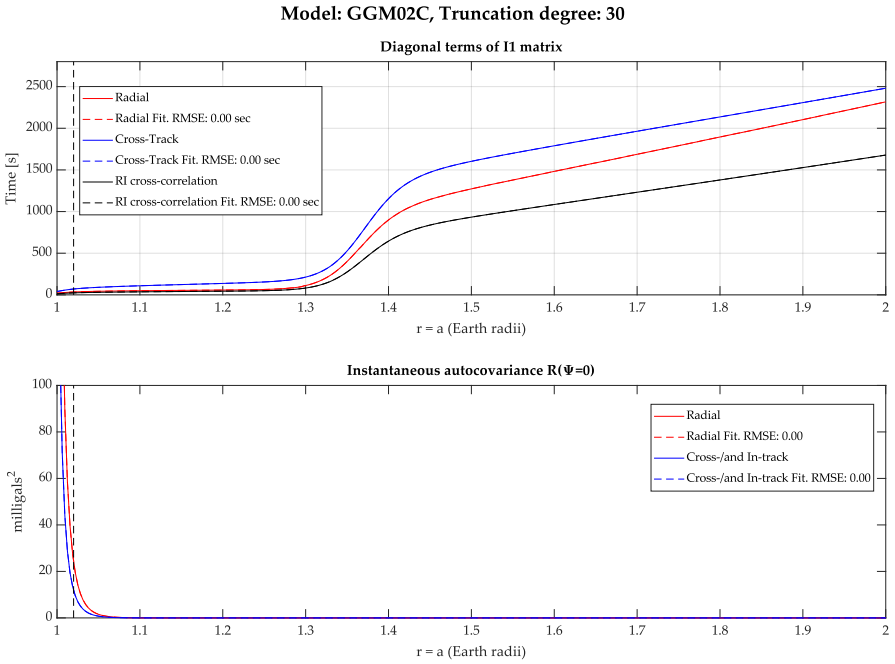


Figure 3.8: Inner integral **I1** curves (upper subplot) and $\mathbf{R}(0)$ diagonal elements (lower subplot) together with the cubic splining fits for GGM02C and $N_T = 30$

IMPACT OF IN-TRACK UNCERTAINTY UNDERESTIMATION ON ORBIT DETERMINATION

The underestimation of the in-track position error and radial velocity error variances is rooted in the omission of the radial/in-track cross-correlation terms, as can be validated by comparing the results of section 3.5.1 with figure 3.5. The practical relevance of this neglect for orbit determination purposes however deserves a separate discussion.

Figure 3.9 depicts a zoom on the first 10 hours of figure 3.5. When performing orbit determination, the measurement update times depend on the availability of sensors for taking observations. Given a filter-based OD process and sufficient resources to keep the propagation times within a couple of orbits, the neglect of ρ_{RI} may be acceptable for practical operations. However, in case of limited sensor resources or if the orbit determination is performed using batch estima-

tors, the propagation times are longer than a couple of orbits. In these cases ρ_{RI} *should not* be neglected. The reason for this soft wording is twofold. First, in orbit determination it is often possible to compensate for small energy errors via small adjustments of the semi-major axis - a phenomenon sometimes referred to as "favorable aliasing" (Woodburn, 2019). The second point is the dichotomy identified in Wright et al. (2008b), when a case of apparent in-track *overestimation* is discussed. This dichotomy (theoretical underestimation vs. practical overestimation) traces back to the full-sphere averaging in the definition of the stochastic gravity error process. Recall that due to this averaging property all orbits shown in figure 3.5 result in the same covariance estimates. Depending on the orbit under consideration however, the actual gravity error process may be dominated more by the local geographic properties, which in turn can lead to a practical overestimation of uncertainty, whereas from a theoretical perspective the transverse error is clearly underestimated. The domination of local effects is increasing with orbital altitude, as the theoretical foundation of the gravity error covariance is based on the extrapolation of Kaula's work to near-Earth altitudes. A further example of this dichotomy is given in section 3.5 as part of the numerical validation of the improved algorithm.

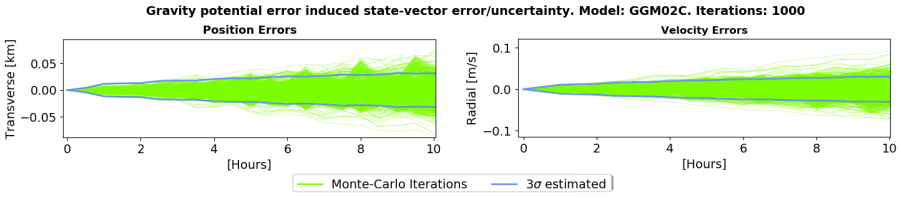


Figure 3.9: Zoom on the first 10 hours of figure 3.5. As shown, the neglect of ρ_{RI} may be acceptable for a couple of orbits propagation duration

3.3.2 *Robustness of radial and normal position error covariance components*

To ensure computational tractability, the WK algorithm approximates the inner autocorrelation integrals by representative time constants. Figure 3.10 depicts the autocorrelation functions of the radial direction (similar results are obtained for the normal direction) versus central angle Ψ. All curves for the altitudes between $r = 1.0\ a_E$ and $r = 2.0\ a_E$ in steps of 20 km are combined into the graphs. The autocorrelation functions are symmetric about the ordinate, which is the justification of the WK algorithm for only evaluating the positive range of Ψ (Wright et al., 2008a). However, when performing the integration over the Earth central angle, it makes a difference if the integration is started at $\Psi = -\pi$ or at $\Psi = 0$, as depicted in figure 3.11 and reasoned in the following.

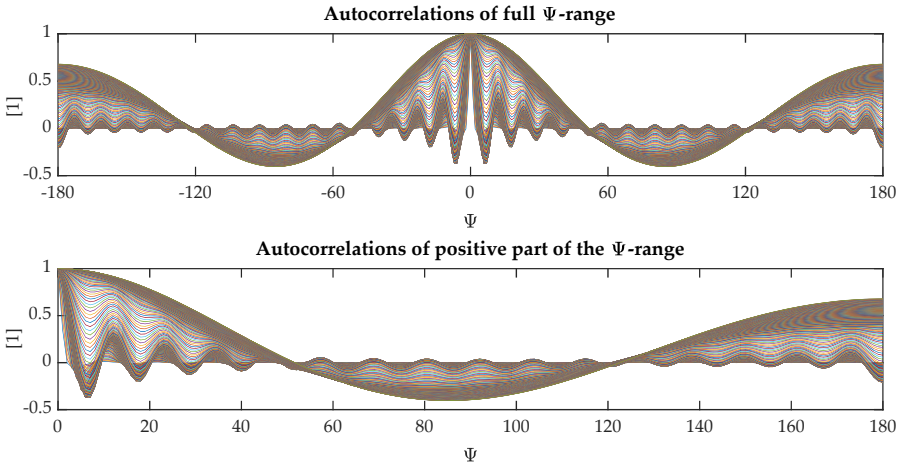
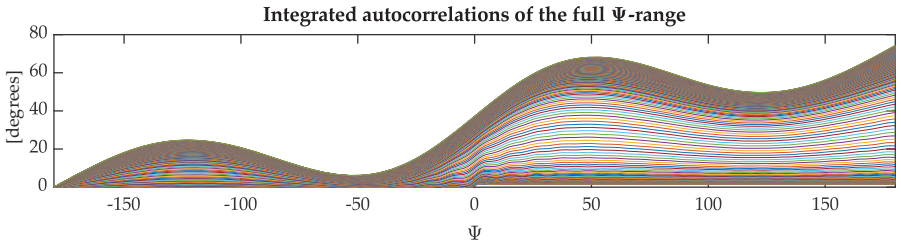
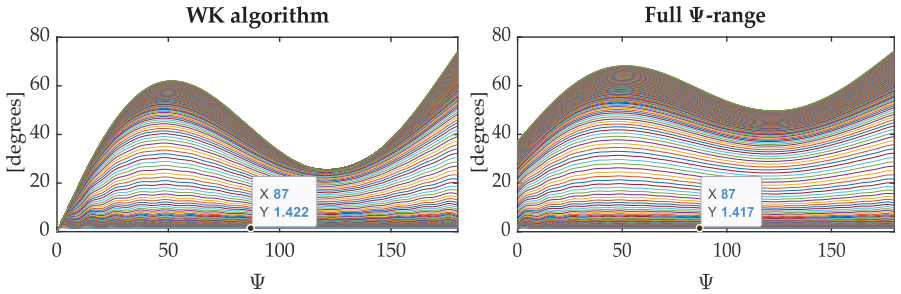


Figure 3.10: Radial autocorrelations for $N_T = 30$ and GGM02C. Each curve depicts $\rho_{RR}(\Psi)$ for an altitude within 1.0 to 2.0 a_E. The autocorrelations are symmetric with respect to the ordinate



(a) Cumulative integration of $\rho_{RR}(\Psi)$ starting from $-\pi$, which is the actual integral to be considered for **I1**



(b) Cumulative integration of $\rho_{RR}(\Psi)$ starting from $\Psi = 0$ and taking the result times two (WK algorithm)

(c) Cut of figure 3.11a on the positive part of the abscissa

Figure 3.11: Difference in radial autocorrelation integrals depending on integration strategy

Multiple interesting properties can be inferred from figure 3.11:

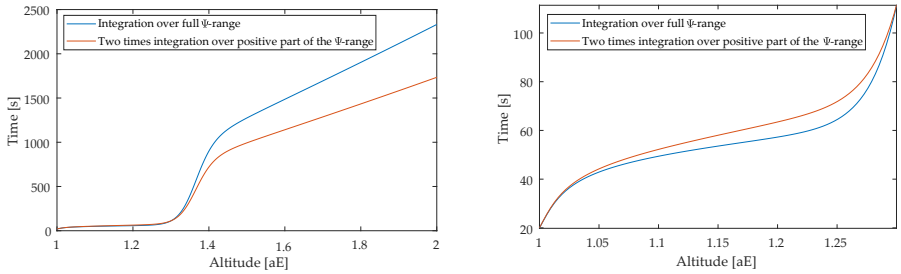
- For altitudes below a certain boundary r_b ($\approx 1.3a_E$ for $N_T = 30$ and GGM02C) the autocorrelation integrals form true plateaus once the domain of integration is sufficiently large. The values of these plateaus are taken as the constants to approximate the autocorrelation integrals in the WK algorithm

- Above r_b the plateaus vanish and the autocorrelation integrals become sinusoidal
- If true plateaus are reached, their plateau values are only approximately equal between both approaches ($\int_{-\pi}^{\pi} \rho(\Psi)\, d\Psi$ and $2\int_{0}^{\pi} \rho(\Psi)\, d\Psi$). A reference plateau value is highlighted in figure 3.11 for comparison
- As there are no true plateaus above r_b, fictive plateaus have to be created, which are less reliable than true plateaus ("badly conditioned"). In these situations the plateau forming algorithm has a strong influence on the resulting scalar constants

Since the WK algorithm starts the integration from $\Psi = 0$, its integrated autocorrelations rise from zero, whereas this is not the case if the integration is started from $\Psi = -\pi$. The relationship $\int_{-\pi}^{\pi} \rho(\Psi)\, d\Psi = 2\int_{0}^{\pi} \rho(\Psi)\, d\Psi$ only holds if the integration is performed over the full range of Ψ. Therefore only the endpoints ($\Psi = \pi$) match exactly in the lower graphs of figure 3.11. All other points differ from each other, with increasing magnitude the further Ψ is away from π. Since transients are not to be taken into account in the plateau detection, the plateau values in each direction are obtained by integrating over a sufficiently large part of the orbit, but never the full orbit. As the plateau values are subsequently converted to time using the inverse orbital mean motion, even small differences can scale to notably different $\mathbf{I1}$ curve fits for low altitudes. The partial lack of robustness of the WK algorithm in the normal and radial position error covariance components can therefore be explained by its simplified integration strategy and the selection of the averaging domain (40-160 degrees in case of the original implementation in ODTK).

For altitudes above r_b the plateaus vanish and the autocorrelation integrals become sinusoidal. In the absence of true plateaus other strategies are required to obtain representative time constants. One option is to average over the entire integration domain. So far this has been the strategy of ODTK, however this approach requires switching the integration bounds back to $2\int_{0}^{\pi} \rho(\Psi)\, d\Psi$, which may lead to sharp transitions around r_b, when combined with $\int_{-\pi}^{\pi} \rho(\Psi)\, d\Psi$ for altitudes below r_b.

Another approach that does not require a change of the integration bounds is to average over the domain of Ψ which is free of transient behavior. This domain depends on the truncation degree N_T, but is constant with altitude (the sinusoids are stacked on top of each other in figure 3.11). Transient behavior can be identified in many ways. A local standard deviation filter identified transients above $\Psi = 122\,deg$ and below $\Psi = 52\,deg$ for altitudes above r_b in figure 3.11. Each fictive plateau may hence be created as the mean autocorrelation integral between the sinusoid peaks (max and min). Local standard deviation filtering is also able to detect the true plateau boundaries. As such the method can be used for both $r \leqslant r_b$ and $r > r_b$. It allows to obtain



(a) Autocorrelation integrals of $\rho_{RR}(\Psi)$ scaled to time for $N_T = 30$ and GGM02C

(b) Zoom on figure 3.12a for the altitude region $r < 1.3a_E$

Figure 3.12: Difference in autocorrelation integrals between the WK algorithm (**I1** integration strategy: $2\int_0^{\pi}\rho(\Psi)\,d\Psi$, red) and the theoretical integral to be considered for the **I1** computation ($\int_{-\pi}^{\pi}\rho(\Psi)\,d\Psi$, blue)

figure 3.12 after posterior conversion from central angle to correlation time and curve-fitting, as outlined in eq. 3.49 and detailed in Wright et al. (2008a,c).

Figure 3.12a reveals the differences in the autocorrelation integrals versus altitude for $r > r_b$ between the WK algorithm (red) and the integration starting from $\Psi = -\pi$ (blue). However, a zoom on the domain $r \leqslant r_b$ shows that the slight angular differences in the plateaus also lead to considerable changes of the autocorrelation time curve-fits in the most populated LEO altitudes (cf. figure 3.12b).

It is this difference in the true and fictive plateaus, which explains the partial lack of robustness in the radial and normal position directions of the WK algorithm. That the improved version (domain of integration: $\Psi \in [-\pi, \pi]$) is indeed able to yield accurate state vector error covariances is demonstrated in section 3.5.

3.3.3 *Inclusion of additional perturbations*

The WK algorithm uses a two-body STM in equation 3.34 to propagate the orbital error covariance over time. The rationale behind this decision is to "enable a fast-running filter, sufficiently accurate to capture the essence of the potential function covariance matrix" (Wright et al., 2008a).

Using a two-body STM, $\mathbf{\Phi}_{2B}$, is appropriate as long as no additional perturbations are modeled or only short propagation times are considered. For practical operations however, all possible perturbation models are typically enabled. Only the gravity field is recommended to be truncated for the purpose of execution speed (Petit and Luzum, 2010). Consequently, these perturbations also need to be reflected in the state error transition matrices, as otherwise the accumulation of the orbit error covariance diverges rather quickly. This is shown in figure 3.13, in which the covariance breaks down between 13 h and 17 h of simulation time, since $\mathbf{\Phi}_{2B}$ is not accurate enough for the summing of

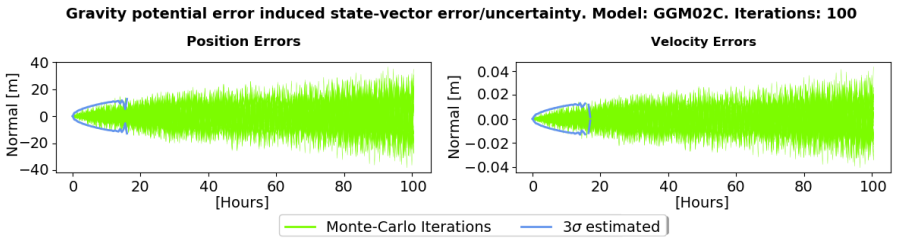


Figure 3.13: Breakdown of normal position and velocity orbit error covariances using a two-body STM as suggested by the WK algorithm

the covariances as also perturbations other than an aspherical gravity field are modeled. Modern special perturbations (SP) orbit analysis applications, such as AGI's ODTK or Airbus' SPOOK compute the STMs numerically. It is therefore straightforward to include the full set of perturbations in the STM computations.

Depending on the implementation of the propagation steps in between measurement updates, it may be necessary to perform an additional propagation (or interpolation) in order to obtain precise state vector data at all covariance evaluation times $(\tau_j + \Delta/2)$ to evaluate the lower part of the frame conversion matrix $\mathbf{G}(\tau_j + \Delta/2)$. If an additional propagation is required (for example because there is no sufficiently small propagation grid in the absence of measurement updates), it is important to stress that $\mathbf{G}(\tau_j + \Delta/2)$ must always be computed with the maximum precision possible. This avoids errors in the frame conversions to accumulate additionally to possible errors in the STMs due to floating-point arithmetic or measures of reducing the runtime.

3.4 RUNTIME CONSIDERATIONS AND IMPLEMENTATION

Equation 3.33 requires the evaluation of $\mathbf{K}(t_{k+1}, \tau_j + \Delta/2)$ for each time granularity step between t_k and t_{k+1} (p steps). This involves one (special perturbations) orbit propagation to compute $\mathbf{G}(\tau_j + \Delta/2)$ with p output times and p (numerical) computations of $\mathbf{\Phi}(t_{k+1}, \tau_j + \Delta/2)$.

Also, as noted above, $\mathbf{G}(\tau_j + \Delta/2)$ needs to be computed with the full accuracy available and $\mathbf{\Phi}(t_{k+1}, \tau_j + \Delta/2)$ with all considered perturbations. These parts of $\mathbf{K}(t_{k+1}, \tau_j + \Delta/2)$ are the main runtime drivers when computing $\mathbf{Q}_F(t_k, t_{k+1})$. If a non-sequential batch algorithm capable of incorporating process noise is used for orbit determination (e.g. the extended weighted least squares algorithm presented in Schiemenz et al., 2019c), this load is even squared, since $\mathbf{Q}_F(t_0, t_{k+1})$ is required for each individual measurement time (t_0 denotes the reference epoch of a batch estimator).

The additional computational burden due to the gravity process noise computation depends on the orbit determination algorithm and its implementation. In some cases, for example for filters using a small propagation-grid in the absence of measurements, full accuracy

integrated quantities may be available and can be harnessed to obtain the required matrices for the evaluation of $\mathbf{Q}_F$ via interpolation. If the availability of quantities ready for interpolation is not assumed (for example because the time grid of propagations in the absence of measurements is too wide for interpolation, or if batch estimators are considered), the computation of $\mathbf{Q}_F(t_k, t_{k+1})$ can quickly become the most costly component of the covariance propagation. The following discussion is concerned with this latter scenario.

In the light of the accuracy requirements formulated during the discussion of the inclusion of additional perturbations, it becomes clear that the main options to reduce the computational runtime are $\mathbf{\Phi}$ and the integration accuracy of the numerical integrator. Δ should not be increased, as otherwise the constraint $\Delta \leqslant P/36$ gets violated, which leads to erroneous orbit gravity error covariance results.

The following means were found to reduce the time needed to compute $\mathbf{Q}_F$ for a typical low Earth orbit at an average altitude of 511 km over 4.2 days of propagation with covariance outputs computed every 15 minutes by approximately 50%; while still maintaining qualitatively equal covariance data compared to the case of evaluating $\mathbf{Q}_F$ with the full accuracy available (gravity field up to the truncation degree/order and relative/absolute integrator accuracy of 10^{-12}):

- The gravity field degree is reduced to $n = 4$ in the computation of $\mathbf{\Phi}(t_{k+1}, \tau_j + \Delta/2)$
- The gravity field order is reduced to $m = 0$ in the computation of $\mathbf{\Phi}(t_{k+1}, \tau_j + \Delta/2)$

  } J_4 gravity
- The absolute and relative numerical integration accuracy of the numerical integrator used to compute $\mathbf{\Phi}(t_{k+1}, \tau_j + \Delta/2)$ can be significantly decreased based on the orbital altitude. For 511 km altitude, an accuracy of 10^{-12} allowed for ~5 days of covariance propagation without breakdown using a Shampine-Gordon integrator. The higher the orbital altitude, the further the integration accuracy can be lowered

In addition to above runtime improvements, two further tricks need to be heeded when implementing equation 3.33 in software for the case of multiple output times (i.e. for batch estimators):

- When computing $\mathbf{\Phi}(t_{k_i}, \tau_j + \Delta/2)$ for each measurement time i, the integrator-internal state-variables shall be saved in order to continue the integration from the current time $\tau_j + \Delta/2$ to the next measurement time $t_{k_{i+1}}$
- Since $\mathbf{\Phi}(t_{k_{i+1}}, \tau_j + \Delta/2) = \mathbf{\Phi}(t_{k_{i+1}}, \tau_{j-1} + \Delta/2)\mathbf{\Phi}(\tau_{j-1} + \Delta/2, \tau_j + \Delta/2)$, it suffices to compute only the STMs for the first output/measurement time and the delta STMs between the output times. This strategy is exemplified in figure 3.14, which illustrates that instead of numerically computing the paths in red, it is equivalent to compute the blue paths once (with integrator state backups) and then to propagate them via the delta STMs (black)

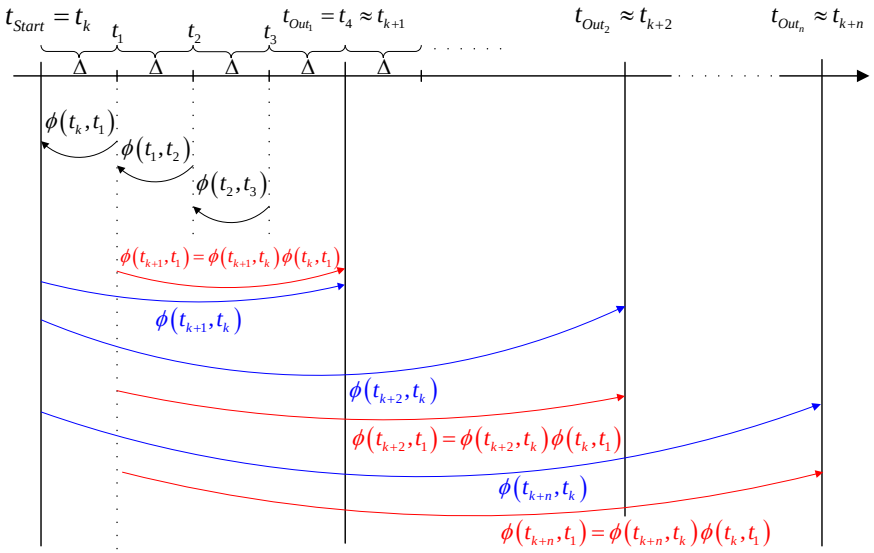


Figure 3.14: Efficient STM computation strategy. The red paths are computed using the blue and black ones, which requires a minimum amount of runtime

NUMERICAL ROBUSTNESS

As noted in Wright et al. (2008a), the numerical computations sometimes lead to non positive semidefinite (PSD) covariance estimates (e.g. due to a negative element on the main diagonal of the orbit gravity error variances or via correlation coefficients with a magnitude > 1.0, or due to matrix asymmetries).

The occurrence of indefinite orbit error covariance estimates could be replicated independently of the value selected for I_{II}, for which the WK algorithm suggests to put 10^{-10} for the purpose of numerical stability (Wright et al., 2008a). Consequently, I_{II} has been reverted to its theoretical value of 0.0 and a more powerful technique to correct indefinite covariance estimates has been sought. A suitable algorithm for this posterior checking has been published in Higham (1988). Higham's method is able to estimate a nearest (in the sense of Frobenius norm minimization) PSD covariance matrix to an arbitrary real matrix. Therefore it can be used to correct indefinite covariance estimates. The covariance correction is especially successful in Cartesian (e.g. GCRF) coordinates due to the inherent magnitude differences between the position and velocity sub-covariances.

ALGORITHM SUMMARY

The complete real-time part of the $\mathbf{Q}_F$ computation algorithm is summarized in algorithm 3.1. The improved **I1** integral components are assumed to be available from offline computations. The listed pseudocode can be used for filters as well as batch estimators and supports the efficient computation of $\mathbf{Q}_F$ at multiple output times (required for batch estimation). The availability of integrated trajectories and STMs to be used for interpolation is not assumed, which is why these computations are listed as separate steps.

The reduced form of the algorithm, which is suitable only for filters performing the propagation phase in sufficiently small time increments for the state and STM computations to be interpolated, is given as algorithm 3.2 for the sake of completeness.

Algorithm 3.1: Generic computation of orbit gravity error covariance (filters and batch estimators)

```
1:  function computeQFMatrixGeneric(x(t_Start), t_Start, t_End)
2:      n_Out ← length(t_End)
3:      t_eval, t_dir ← computeEvalTimesAndTimeDir(t_Start, t_End(n_Out))
4:      n_eval ← length(t_eval)
5:      x_eval ← propagate(x(t_Start), t_eval)
6:      idx ← 1
7:      Q_F ← 0.0
8:      for i ← 1, n_eval do
9:          if ((t_dir = 1 and t_eval(i) > t_End(idx)) or
                (t_dir = −1 and t_eval(i) < t_End(idx))) then
10:             idx ← idx + 1
11:         end if
12:         G ← buildG(x_eval(t_i))
13:         r ← |x_eval_pos(t_i)|/aE
14:         if i = 1 then
15:             I1 ← evaluateI1(r)
16:             if t_dir = −1 then
17:                 I1(2, 1) ← −I1(2, 1)
18:                 I1(1, 2) ← −I1(1, 2)
19:             end if
20:         end if
21:         R0 ← computeR0(r)
22:         F  ← G ∗ R(0)^{1/2}
23:         C  ← F ∗ I1 ∗ F^T
24:         if i = 1 then
25:             for j ← idx, n_Out do
26:                 Φ_j ← computeSTM(x(t_i), t_i, t_End(j))
27:                 storeIntegratorState()
28:                 Q_F(j) ← Q_F(j) + Δ · (Φ_j ∗ C ∗ Φ_j^T)
29:             end for
30:         else
31:             δΦ ← computeSTM(x(t_i), t_i, t_{i−1})
32:             for j ← idx, n_Out do
33:                 Φ_j ← Φ_j ∗ δΦ
34:                 Q_F(j) ← Q_F(j) + Δ · (Φ_j ∗ C ∗ Φ_j^T)
35:             end for
36:         end if
37:     end for
38:     for j ← 1, n_Out do
39:         Q_F(j) ←nearPSD(Q_F(j))
40:     end for
41:     return Q_F
42: end function
```

Comments:

- *1: t_{End} is an array of output times*
- *3: $t_{eval} = \tau_j + \Delta/2$*
- *5: Use all perturbations and full integration accuracy*
- *7: Initialize the output covariance matrix array*
- *13: Compute r as the magnitude of the position part*
- *15: Evaluate the precomputed (according to section 3.3) I1 curve-fits*
- *17: Swap I_{RI} sign for backward propagation*
- *21: Compute R0 via the stored degree variances or $\mathbf{R}(0)$ curve-fits*
- *26: Use prepared/stored integrator state and reduced gravity field/integrator accuracy*
- *28: Compute Riemann-partitioned integral*
- *33: Propagate each STM via multiplication with δΦ. Use reduced gravity field degree/order and integrator accuracy*
- *39: nearPSD implements the Higham algorithm and checks if fixing is necessary*
- *41: Return the array of covariance matrices (one for each element in t_{End})*

Algorithm 3.2: Computation of filter orbit gravity error covariance using interpolation

Comments:
2: Δ needs to fulfill $\Delta \leqslant P/36$
3: Select sign depending on propagation direction
7: Negate I_{RI} for backward propagation

1: **function** COMPUTEQFMATRIXFILTER($x(t_{Start})$, t_{Start}, $x(t_{End})$, t_{End})
2: $\Delta \leftarrow t_{End} - t_{Start}$
3: $t_{eval} \leftarrow t_{Start} \pm \Delta/2$
4: $x_{eval}, \Phi \leftarrow \text{interpolate}(x(t_{Start}), t_{Start}, x(t_{End}), t_{End}, t_{eval})$
5: $G \leftarrow \text{buildG}(x_{eval})$
6: $r \leftarrow |x_{eval_{pos}}|/aE$
7: $I1 \leftarrow \text{evaluateI1}(r)$
8: $R0 \leftarrow \text{computeRo}(r)$
9: $F \leftarrow G * R(0)^{1/2}$
10: $C \leftarrow F * I1 * F^T$
11: $Q_F \leftarrow \Delta(\Phi * C * \Phi^T)$
12: $Q_F \leftarrow \text{nearPSD}(Q_F)$
13: **return** Q_F
14: **end function**

3.5 SIMULATION RESULTS

The orbit gravity error covariance validation is performed at two stages: propagation only and orbit determination. Using orbit and covariance propagation only allows to verify that the errors between the reference orbit (truth) and the propagated Monte-Carlo (MC) orbits remain within the 3σ error bounds for each of the state vector components.

The type of gravity errors considered also determines the validation strategy: errors of commission can be validated by randomizing the model potential field coefficients in accordance with its covariance matrix. This procedure is further explained in Wright et al. (2008b). Consequently a reference orbit (with the nominal potential field coefficients) is computed first and then the nominal coefficients are perturbed in a Gaussian fashion in each of the Monte-Carlo iterations.

Error bounds on the errors of omission are more difficult to validate. When truncating the potential field, there is no randomness in the resulting path (the errors are epistemic in nature). The (simulated) space environment leads to a single and distinct orbit, which differs from the full-field reference orbit. Therefore one would not expect the errors to fill the error bounds as in the case of errors of commission. In fact, the zero-mean property of the stochastic gravity error process is achieved only due to a) the full-sphere averaging of gravity acceleration errors and b) the fact that there are no zero-degree harmonics in the disturbing potential.

To validate the orbit gravity error of omission covariance estimates, it is however possible to explicitly make use of the averaging property in the definition of the stochastic gravity error process. Recall that "gravity covariance is defined as the average of all possible products of [...]

gravity components separated by the arc distance Ψ'' (Gersten et al., 1967). The same sequence of orbit error covariance matrices can thus be expected when changing orbit parameters that only influence the orientation of the orbit but not its shape. This allows the error bounds for gravity errors of omission to be validated by first computing the expected uncertainty for the nominal orbit and then performing two numerical orbit propagations for each Monte-Carlo iteration. The first of these propagations is the reference propagation and uses the full potential field, whereas the second uses the truncated potential field.

In each MC iteration one of the orbit plane orientation parameters is varied according to a uniform distribution[6] in the parameter ranges given in table 3.1, which cover the typical LEO parameters. If errors of commission are to be simulated additionally, the potential field coefficients are randomized for the second propagation.

Orbit plane orientation parameter	**Range**
Argument of Perigee	$0 - 2\pi$
Right ascension of ascending node	$0 - 2\pi$
True anomaly	$0 - 2\pi$
Inclination	$65\,deg - 105\,deg$

Table 3.1: Bounds on orbit parameter changes to validate errors of omission

In the following cases the full potential field is considered to be of degree/order equal to 85 ($N_{max} = 85$ in the $\mathbf{I1}$ computations), which is of typical magnitude for LEO applications. The presented framework also holds for higher altitudes, however it has already been mentioned in section 3.3 that the $\mathbf{I1}$ computation becomes increasingly ill conditioned above the LEO regime. Furthermore, the influence of potential field errors and their associated uncertainty decreases rapidly with altitude, which is why the validation is limited to the LEO regime.

The simulations have been performed with the Airbus tool SPOOK (Rodriguez Fernandez et al., 2019b), which has been extended with the improved orbit gravity error covariance capabilities. The reference orbit (prior to any changes when validating errors of omission) is similar to that of the Airbus satellite TerraSAR-X (Pitz and Miller, 2010), which orbits Earth at a mean altitude of 511 km with an eccentricity of ≈ 0.0001476. Errors of omission are simulated using a 30×30 potential field. The truncation level was selected as a reference for operational choices dominated by runtime considerations.

6 The uniform distribution shall ensure that the orbit plane orientation parameters are drawn equally likely from their domain of definition. A Gaussian distribution would artificially bias the observed state vector errors with respect to the mean of the orbit plane orientation parameter range, which is undesired. After all, the aleatoric treatment of epistemic omission uncertainty is able to quantify the resulting state vector error distribution, but does not allow to *characterize* it (UC only applies for pure errors of commission; see section 1.3.1 for the terminology)

3.5.1 *Propagation only*

The results for errors of omission and commission are depicted in figure 3.15 for EGM-96 and figure 3.16 for GGM02C. Despite the aspherical gravity field, also atmospheric drag including thermospheric winds (using DTM-2013 (Bruinsma, 2015) and HWM-93 (Hedin et al., 1996)), as well as solar radiation pressure and solar/lunar third body gravity fields were enabled to demonstrate that additional perturbations can be handled as explained in section 3.3.3.

The results are similar in both cases: during the propagation time of 4.2 days the radial and in-track orbit errors are consistent with the 3σ boundaries as computed by algorithm 3.1. The normal direction uncertainty is consistently estimated up to ~40 h of simulation time, which is when some samples begin to exceed the error bounds. The in-track position error is dominating the radial and normal position errors due to the radial/in-track cross-coupling. The obtained results justify the stochastic model approach.

Errors of commission only (no potential field truncation) are presented in figure 3.17 for GGM02C. The performance is similar to figures 3.15 and 3.16. The orbit errors remain in the predicted bounds for the entire propagation time in the radial and in-track directions. The normal directions again contain samples which start to exceed the error bounds after ~40 h of propagation time. In contrast to the cases of omission and commission, the error magnitudes have decreased by a factor of ~300 (5 meter transverse position uncertainty vs. 1.5 km and 0.1 m radial/cross-track vs. 30 m in figure 3.16).

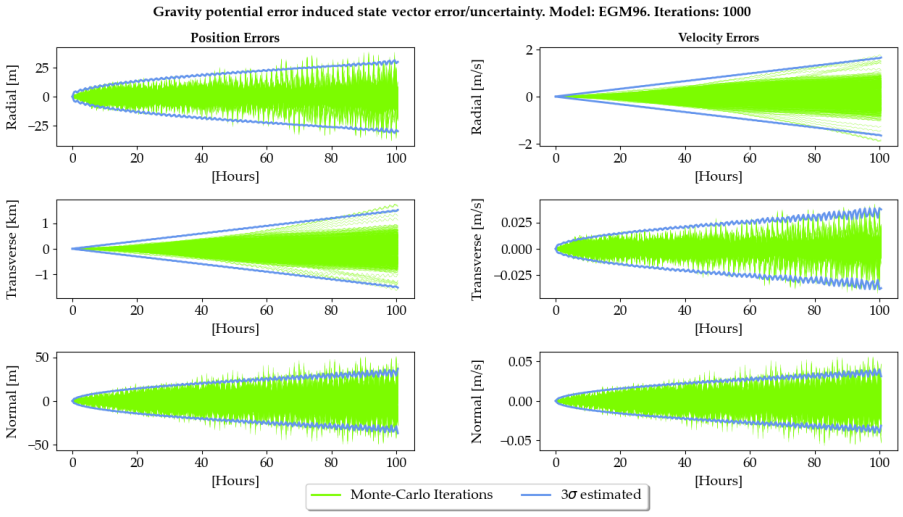


Figure 3.15: Errors of omission and commission for EGM-96 obtained after 1000 Monte-Carlo iterations, corresponding to 2001 numerical propagations

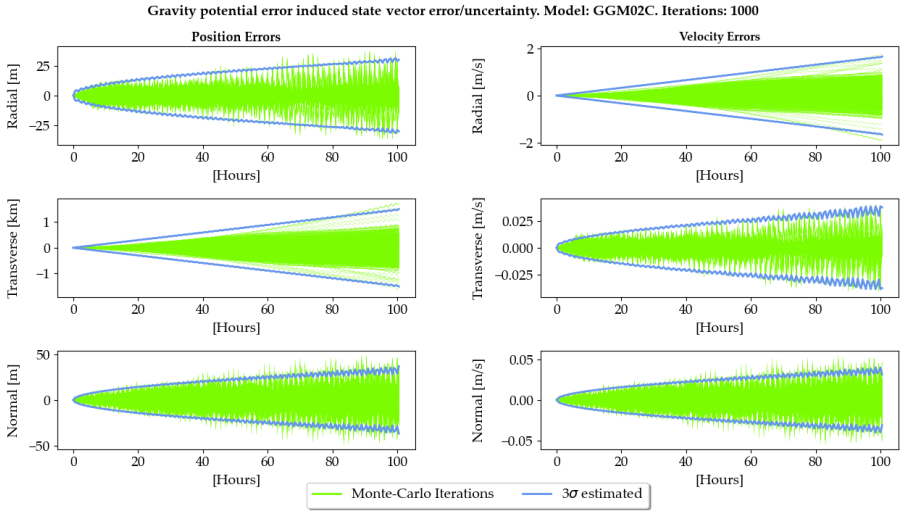


Figure 3.16: Errors of omission and commission for GGM02C obtained after 1000 Monte-Carlo iterations, corresponding to 2001 numerical propagations

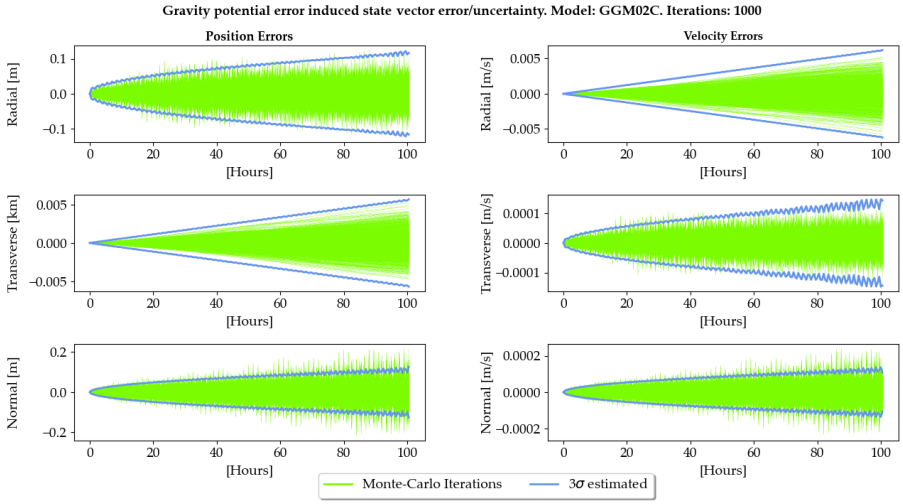


Figure 3.17: Errors of commission for GGM02C obtained after 1000 Monte-Carlo iterations, corresponding to 2001 numerical propagations

3.5.2 *Orbit determination*

The covariance propagation results of section 3.5.1 qualify the orbit gravity error covariance matrices for their application in orbit determination algorithms. Naturally the usage of process noise falls into the domain of the sequential filters. However, also batch estimators can be enabled to consider physics-based process noise, as for example demonstrated in section 4.2 and Schiemenz et al. (2019c). Since the gravity process noise matrix is computed incrementally, it can be added directly to the propagated covariance of sequential estimators without further considerations.

The technique of orbit orientation parameter variation cannot be applied in case of orbit determination applications, as different orbits

result in different target observation times. Consequently, to validate the physical process noise matrices as part of an OD algorithm, the orbit state vector errors are expected to remain within the 3σ error bounds without filling them, given that errors of omission are applicable. If only errors of commission are simulated, it is expected that the error bounds are filled and match the 3σ bounds of the Monte-Carlo iterations. The following simulation results were obtained using SPOOK with the same orbit configuration as in section 3.5.1 and a single simulated optical observer located in central Germany. Only the position parts of the state vector errors are shown for brevity.

Figures 3.18, 3.19 and 3.21 depict the results obtained using an Extended Kalman Filter (EKF)[7] with orbit gravity error process noise. For comparison, the results of a classical EKF without process noise are shown in figure 3.20.

While the oscillatory motion in the radial and normal directions in figure 3.18 is perfectly contained within the expected uncertainty bounds, the in-track direction in figure 3.18 exemplifies the dichotomy discussed in section 3.3: the in-track uncertainty estimates appear pessimistic due to the definition of the stochastic gravity error process. A change of the simulated orbit orientation parameters from $(t, a, e, i, \Omega, \omega, \nu) = (2011/10/01\text{—}20{:}00\text{h}, 6887.0\,\text{km}, 0.000\,147\,6, 97.1\,\text{deg}, 141.578\,\text{deg}, 56.55\,\text{deg}, 80.2\,\text{deg})$ to $(t, a, e, i, \Omega, \omega, \nu) = (2015/02/18\text{—}09{:}00\text{h}, 6887.0\,\text{km}, 0.000\,147\,6, 76.389\,\text{deg}, 84.771\,\text{deg}, 108.631\,\text{deg}, 162.348\,\text{deg})$, however, demonstrates that for this parameter combination the errors propagate exactly along the bounds. Without the explicit consideration of the RI cross-correlation this case would be a perfect example of the in-track underestimation of the WK algorithm, as shown in figure 3.19.

In both cases featuring orbital gravity error consideration (figures 3.18 and 3.19) the measurement updates of the simulated optical observer are able to significantly reduce the orbital uncertainty. The performance of a classical EKF without process noise consideration is depicted in figure 3.20. The position orbit errors are not contained within the error bounds, leading to suboptimal and inconsistent estimates. Since process noise is neglected, there is no increase of the estimated root variances between the measurement times. The estimated uncertainties also remain inconsistent after measurement updates of the optical observer. These examples demonstrate the need to account for physics-based process noise in low Earth orbits.

Errors of commission only are depicted in figure 3.21. Since the full potential field is used in the orbit determination, the orbit errors and their actual/estimated uncertainties are small compared to the case of errors of omission. In all axes the three sigma root variances of the

7 Strictly speaking, the filter is not a classical EKF, as gravity noise is not white. The terminology is used nonetheless, since this property does not alter the remaining filter equations (Wright, 1981)

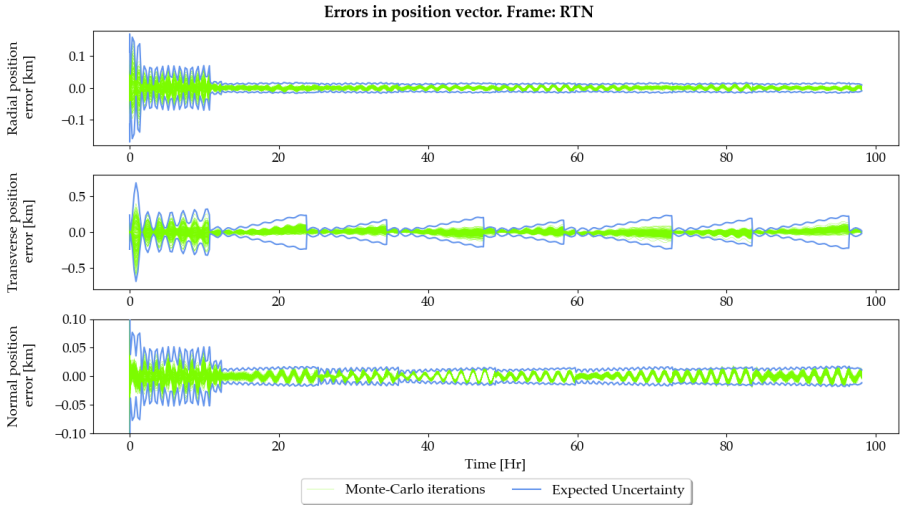


Figure 3.18: OD including errors of omission and commission obtained using an EKF. Measurement updates can be identified via the dips in the estimated root variances. Model: GGM02C

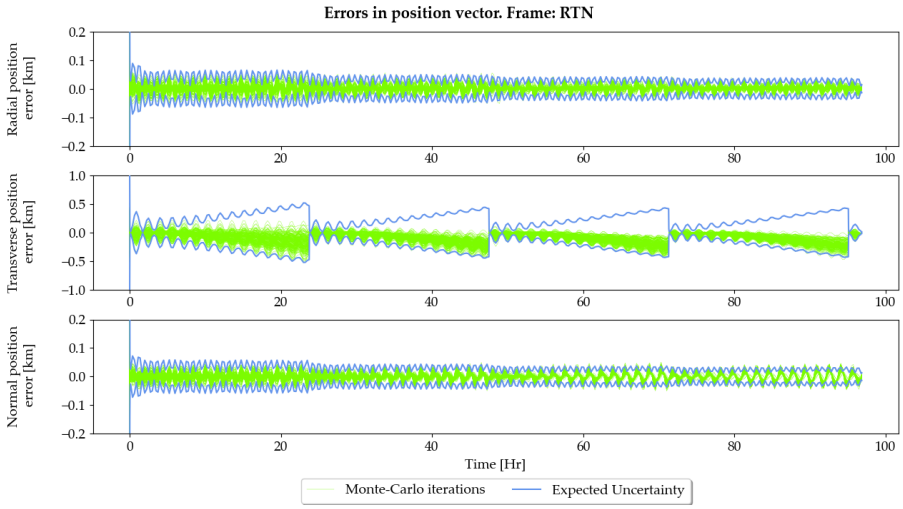


Figure 3.19: Same scenario as in figure 3.18, except for modified orbit orientation parameters to demonstrate the need to consider the RI cross-correlation

MC iterations (red) match the estimated three sigma root variances (blue) once the filter is settled (cf. zoom on the timeframe 80 h to 98 h).

Gravity acceleration errors are a perfect example for the treatment of epistemic uncertainties by aleatoric means. The simulation results both demonstrate the need for physics-based process noise matrices and validate the extension of the original WK algorithm. The improved formulation, as given in listing 3.1, not only provides more realistic in-track uncertainty estimates, but also increases the stability of the radial and normal uncertainty components. For the first time, also the physical connection of the RI cross-correlation sign with respect to the direction of the propagation phases was recognized and implemented. The inclusion of arbitrary perturbations into the STM furthermore permits significantly longer propagation times in

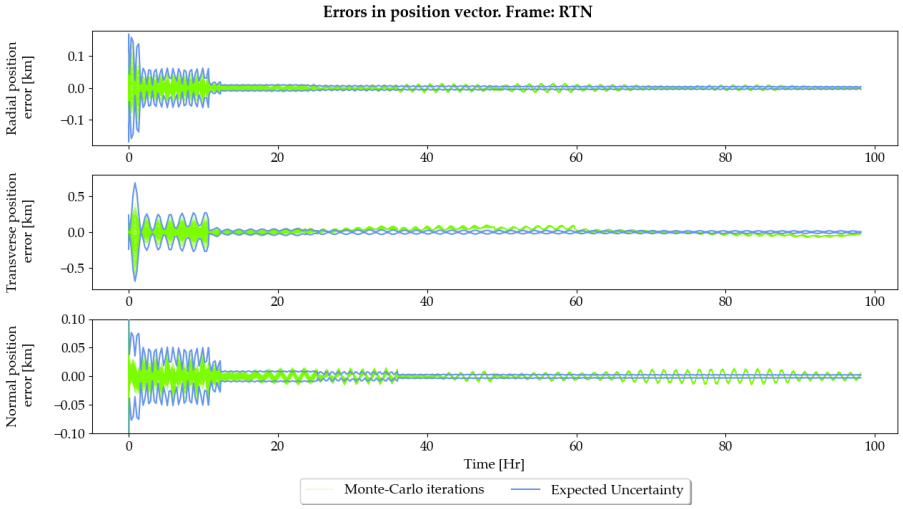
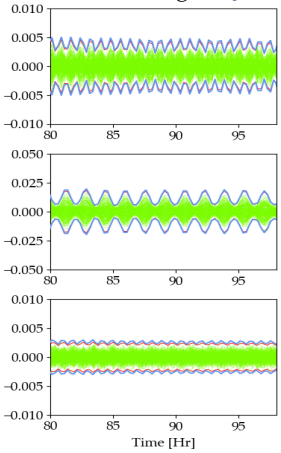


Figure 3.20: Performance of a classical EKF. As process noise is not considered, the filter is not able to produce consistent estimates

Zoom on timeframe 80 h to 98 h in figure 3.21:



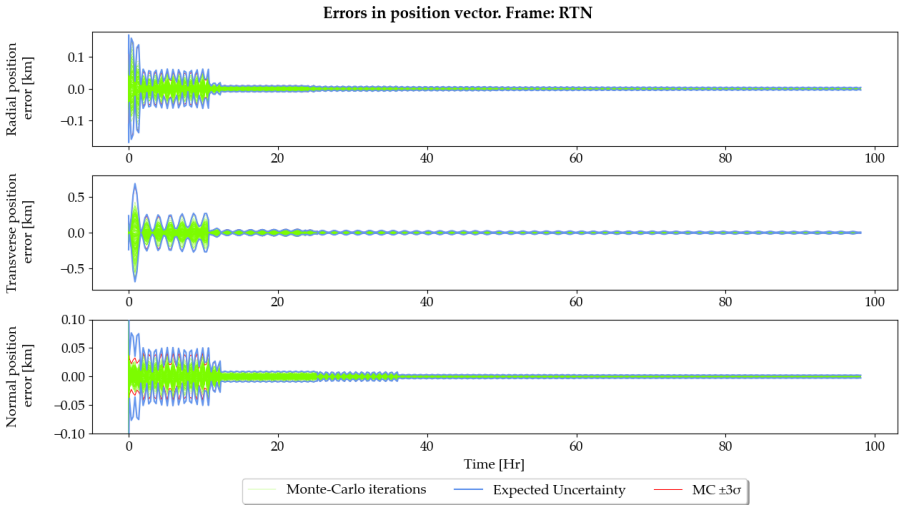


Figure 3.21: Orbit determination results for gravity errors of commission. Model: GGM02C

the order of days compared to a couple of orbits for the original WK description. Numerical stability during the evaluation of the outer integral was achieved by using Higham's nearest covariance matrix algorithm. To ensure the highest practical success of this step, it is important to express the state vector uncertainty in Cartesian coordinates before a correction is made. This is due to the different units of the position and velocity subcovariances favoring the existence of a Frobenius norm closest covariance which is practically representative of the original covariance. For orbital elements (COE or EOE) however only the size variable differs significantly in magnitude from the other orbital elements, which tends to result in corrections less representative of the algorithm input.

Part III

ORBIT DETERMINATION WITH PHYSICS-BASED PROCESS NOISE

4

INTEGRATION OF FORCE MODEL UNCERTAINTIES INTO CLASSIC ORBIT ESTIMATORS

Corresponding publications: Schiemenz et al. (2019c, 2020d)

This chapter is dedicated to the integration of the covariance quantification framework developed in chapters 2 and 3 into operationally proven methods for orbit determination.

Several objectives are pursued. First, the simulation architecture is described to facilitate the understanding of the data flow, as well as the required processing stages. Subsequently, the classic OD batch estimators are extended for the incorporation of process noise. This is an important step, because it closes the gap between the force model uncertainty framework and this widely used class of orbit determination methods, which traditionally were designed without the possibility to account for uncertainties other than of measurement origin.

Further, operational considerations that apply to the application of the uncertainty framework with the respective OD algorithms will be discussed. Simulation results complete the respective sections.

Finally, the extensibility of the force model uncertainty quantification framework is discussed with respect to the GP orbital theory. In some ways, this section broadens the scope of the GP theory. This may be controversial, as the analytic GP theory was published without a complementary uncertainty modeling theory. However, workarounds are presented that allow to obtain an approximate quantification of force model uncertainties also with the GP propagation model.

4.1 SIMULATION ARCHITECTURE

All simulations carried out in the scope of this thesis were performed with the Airbus SST-tool SPOOK, which has been enhanced by the presented features. Further information about SPOOK from its early design phase is provided by Rodriguez Fernandez et al. (2019b). Today, SPOOK has evolved to a powerful tool suite with work in progress on all building blocks of the SST loop identified in figure 1.3, including catalog maintenance and sensor management[1].

A high-level abstraction of the simulation architecture including the sequential filter OD-layer is depicted in figure 4.1. All computations are performed in Fortran, using OpenMP® for parallelization where possible.

1 Airbus operates an optical telescope in Extremadura, Spain, which is tasked using SPOOK. See Utzmann et al. (2019) for further information on the Airbus Robotic Telescope (ART)

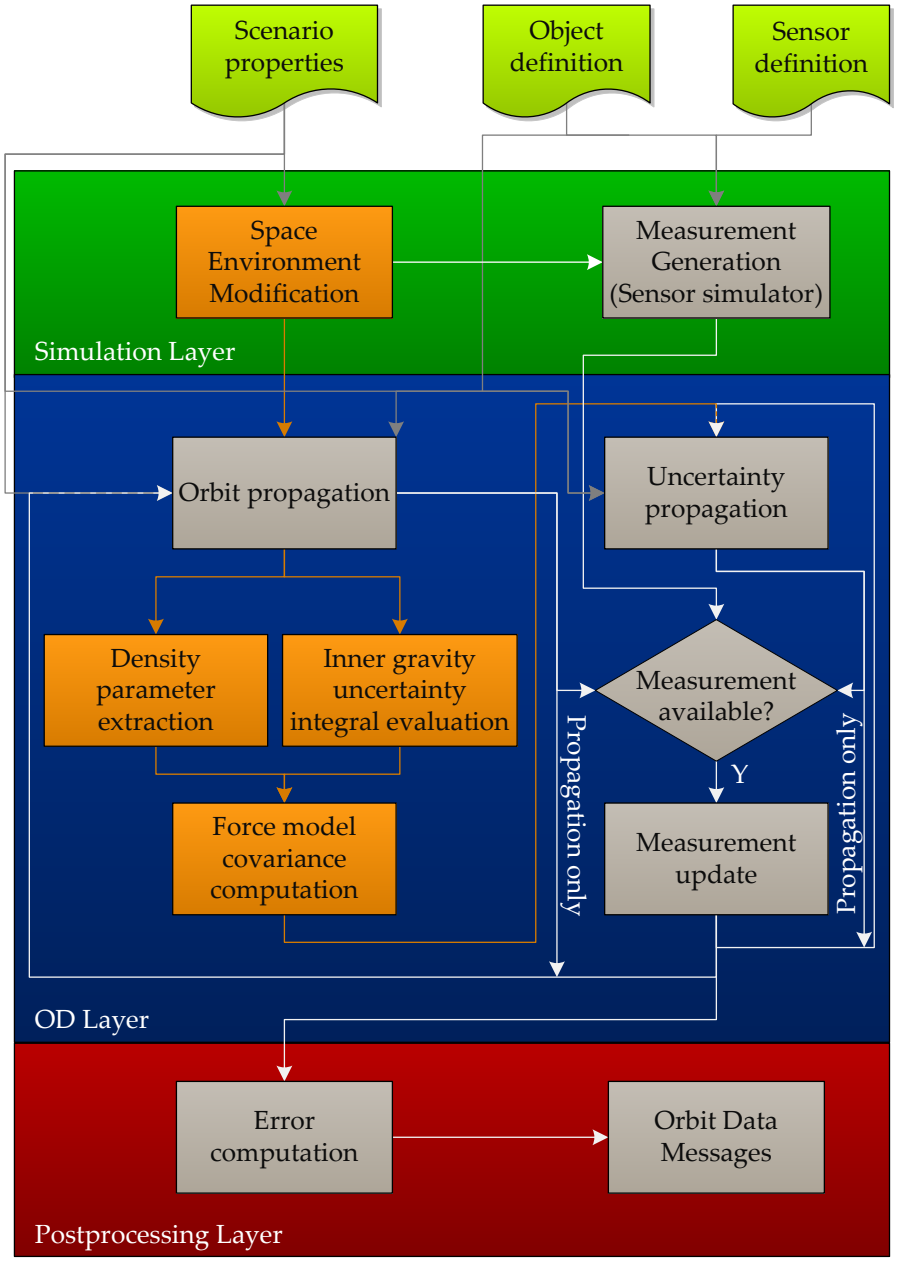


Figure 4.1: High-level architecture of the SPOOK processing core-subset used for the numerical Monte-Carlo simulations

Modules added for the evaluation of force model uncertainties are highlighted in orange. The blocks evaluate the unified density covariance according to table 2.7 and orbital gravity uncertainty according to algorithm 3.1. To properly validate the force model uncertainties, stochastic perturbations are applied in each Monte-Carlo iteration for density errors and gravity errors of commission. Gravity errors of omission are taken into account by using different potential field granularities for the OD layer and the postprocessing layer when computing the orbit errors. The batch estimation OD implementation

presented in section 4.2 makes use of the same force model uncertainty quantification routines as the sequential filters, however the data flow is slightly different.

Prior to starting an orbit determination, the object population (n RSOs), as well as m sensors (ground/space based, optical/radar, ...) need to be defined. Also scenario properties like the force model selection, the uncertainty model/propagation method, the type of numerical integrator and its accuracy, the stochastic properties of the space environment variability, the coordinate frame to be used for measurement updates, the OD method, the number of MC iterations, etc. need to be specified. Default values are applied for options with no explicit user choice.

Using this information, it is possible to generate measurements for the observer(s) when the object(s) pass their field of view (FOV), given the space environment modification (solar flux deltas/magnetic index deltas/OUP atmospheric density realization/perturbed potential field coefficients) that is applicable within each Monte-Carlo iteration. Additional processing is required to limit the visibility timeframes to the closest common subset for each sensor, as the changes in the environmental parameters throughout the Monte-Carlo iterations influence the FOV entry and exit times. These measurements can then be used to perform measurement updates that reduce the accumulated uncertainty during the propagation phases. If no measurements are available, only orbit and uncertainty propagation is performed in configurable time-steps.

4.2 BATCH ESTIMATORS

As noted in section 1.2, OD algorithms can be grouped into batch estimators and sequential filters. Even today, batch estimation is still the most widely used technique for orbit determination, which is primarily due to its robustness. Furthermore, these methods do not require initial covariance estimates, which on the other hand are particularly important for sequential filters to maximize their capture domain (i.e. the maximum range of errors for which the filters are still able to converge on a solution during their initialization phase) and minimize their convergence time.

It is therefore not surprising that the WLS technique is still the standard OD method at CSpOC, where it is used thousands of times a day. Traditionally, however, batch estimators for orbit determination have been developed neglecting process noise (see e.g. section 4.3 in Schutz et al., 2004) and are therefore only able to include the measurement noise in their solution. The astrodynamic force models are implicitly assumed to be perfect. Without an extension of the batch procedures it is thus not possible to make use of the developments

derived in chapters 2 and 3 for this class of orbit estimators. This extension is derived in sections 4.2.1 and 4.2.2.

4.2.1 *Weighted Least Squares*

Classical WLS estimators compute the a priori covariance according to equation 4.1,

$$\mathbf{P}_k = (\mathbf{A}_k^T \mathbf{R} \mathbf{A}_k)^{-1} \tag{4.1}$$

where $\mathbf{A}_k$ is the sensitivity of the observations at time t_k with respect to the state at epoch t_0 and $\mathbf{R}$ is a weighting matrix. It is well-known that the result obtained in this way is also the minimum variance estimate, if $\mathbf{R} = \mathbf{W}^{-1}$, i.e. the weighting matrix is built from the inverse matrix of the measurement variances (Schutz et al., 2004). Equation 4.1 contains no variable relating to model uncertainty and hence needs to be updated. The presented derivation of this extended WLS-algorithm closely follows the classical derivation of the minimum variance estimator, which for example is given in Schutz et al. (2004, section 4.4).

Given a linear (either by the nature of the problem or due to linearization) system of state and observation equations, the task is to find the minimum variance estimate of the state and the corresponding minimum variance.

Each state vector of dimension $n \times 1$ may be propagated in time via equation 4.2,

$$\mathbf{x}_i = \mathbf{\Phi}(t_i, t_k)\, \mathbf{x}_k + \boldsymbol{\nu}_i \tag{4.2}$$

and each $p \times 1$ observation vector relates to the state vector according to equation 4.3,

$$\mathbf{y}_i = \mathbf{H}_i \mathbf{x}_i + \boldsymbol{\epsilon}_i \tag{4.3}$$

where

- $\mathbf{\Phi}(t_i, t_k)$ is the STM from time t_k to t_i
- $\boldsymbol{\nu}_i$ is the propagation error associated with the transition from t_k to t_i
- $\mathbf{H}_i$ is the observation matrix at time t_i and
- $\boldsymbol{\epsilon}_i$ is the measurement error at time t_i

Letting the time index i range from 1 to l, it is possible to define:

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_l \end{bmatrix},\ \mathbf{H} = \operatorname{diag}(\mathbf{H}_1, \ldots, \mathbf{H}_l),\ \boldsymbol{\epsilon} = \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \vdots \\ \boldsymbol{\epsilon}_l \end{bmatrix},\ \boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_1 \\ \vdots \\ \boldsymbol{\nu}_l \end{bmatrix},\ \mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}(t_1, t_k) \\ \vdots \\ \mathbf{\Phi}(t_l, t_k) \end{bmatrix} \tag{4.4}$$

Then

$$\mathbf{x} = \mathbf{\Phi} \mathbf{x}_k + \boldsymbol{\nu} \tag{4.5}$$

and

$$\mathbf{y} = \mathbf{H} \mathbf{x} + \boldsymbol{\epsilon} \tag{4.6}$$

where $\mathbf{x}_k$ is $n \times 1$, $\mathbf{x}$ is $l \cdot n \times 1$, $\boldsymbol{\nu}$ is $l \cdot n \times 1$, $\mathbf{\Phi}$ is $l \cdot n \times n$, $\mathbf{H}$ is $m \times n \cdot l$ and $\mathbf{y}$ is $m \times 1$, with $m = l \cdot n$ being equal to the total number of observations. Next, define

$$\mathbf{A} = \mathbf{H} \mathbf{\Phi} \tag{4.7}$$

which is $m \times n$. Since $\mathbf{H}$ relates the i^{th} measurement set to the i^{th} state and $\mathbf{\Phi}$ relates the state at time t_i to the state at time t_k, $\mathbf{A}$ relates the state at time t_k to the measurement at time t_i. Inserting equation 4.5 into eq. 4.6 and applying 4.7, yields:

$$\mathbf{y} = \mathbf{A} \mathbf{x}_k + \mathbf{H} \boldsymbol{\nu} + \boldsymbol{\epsilon} \tag{4.8}$$

The measurement error covariance is given by

$$E[\boldsymbol{\epsilon} \boldsymbol{\epsilon}^T] = \mathbf{R} \tag{4.9}$$

and the propagation error covariance by

$$E[\boldsymbol{\nu} \boldsymbol{\nu}^T] = \mathbf{Q} \tag{4.10}$$

Assuming the propagation error to be zero-mean, the variance associated with $\mathbf{d} = \mathbf{H} \boldsymbol{\nu}$ is:

$$Var(\mathbf{H} \boldsymbol{\nu}) = E[\mathbf{d} \mathbf{d}^T] = E[\mathbf{H} \boldsymbol{\nu} \boldsymbol{\nu}^T \mathbf{H}^T] = \mathbf{H} \mathbf{Q} \mathbf{H}^T \tag{4.11}$$

The task is to find the linear, unbiased, minimum variance estimate $\hat{\mathbf{x}}_k$ of the state $\mathbf{x}_k$. The requirement of a linear estimate can be translated into the estimate being a linear combination of the observations, i.e.

$$\hat{\mathbf{x}}_k = \mathbf{M} \mathbf{y} = \mathbf{M} (\mathbf{A} \mathbf{x}_k + \mathbf{H} \boldsymbol{\nu} + \boldsymbol{\epsilon}) \tag{4.12}$$

$\mathbf{M}$ needs to be found and is $n \times m$, since $\hat{\mathbf{x}}_k$ is $n \times 1$ and $\mathbf{y}$ is $m \times 1$. The requirement of the estimate being unbiased is expressed as:

$$E[\hat{\mathbf{x}}_k] = \mathbf{x}_k \tag{4.13}$$

Inserting equation 4.12 into eq. 4.13 yields:

$$\begin{aligned} \mathbf{x}_k &= E[\mathbf{M} \mathbf{y}] = E\big[\mathbf{M} (\mathbf{A} \mathbf{x}_k + \mathbf{H} \boldsymbol{\nu} + \boldsymbol{\epsilon})\big] \\ &= E[\mathbf{M} \mathbf{A} \mathbf{x}_k] + E[\mathbf{M} \mathbf{H} \boldsymbol{\nu}] + E[\mathbf{M} \boldsymbol{\epsilon}] \end{aligned} \tag{4.14}$$

Since $\boldsymbol{\nu}$ and $\boldsymbol{\epsilon}$ are zero-mean, the last two terms equate to zero, which results in:

$$\mathbf{MA} = \mathbf{I} \tag{4.15}$$

Equation 4.15 is the mathematical result of the estimate being linear and unbiased. Since $\mathbf{A}$ is $m \times n$, the classical inverse is not defined and equation 4.15 is to be treated as a constraint to be fulfilled by the solution of $\mathbf{M}$, which needs to make sure to yield the minimum variance estimate.

The definition of the estimation error covariance matrix reads:

$$\mathbf{P}_k = E\left[\left((\hat{\mathbf{x}}_k - \mathbf{x}_k) - E[\hat{\mathbf{x}}_k - \mathbf{x}_k]\right)\left((\hat{\mathbf{x}}_k - \mathbf{x}_k) - E[\hat{\mathbf{x}}_k - \mathbf{x}_k]\right)^T\right] \tag{4.16}$$

Since $E[\hat{\mathbf{x}}_k - \mathbf{x}_k] = E[\hat{\mathbf{x}}_k] - E[\mathbf{x}_k] = \mathbf{x}_k - \mathbf{x}_k = \mathbf{0}$, the covariance expression becomes:

$$\mathbf{P}_k = E\left[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T\right] \tag{4.17}$$

Inserting equation 4.12 into eq. 4.17 yields:

$$\mathbf{P}_k = E\left[(\mathbf{My} - \mathbf{x}_k)(\mathbf{My} - \mathbf{x}_k)^T\right] \tag{4.18}$$

Next, inserting equation 4.8 into eq. 4.18 leads to:

$$\mathbf{P}_k = E\left[(\mathbf{MAx}_k + \mathbf{MH}\boldsymbol{\nu} + \mathbf{M}\boldsymbol{\epsilon} - \mathbf{x}_k)(\mathbf{MAx}_k + \mathbf{MH}\boldsymbol{\nu} + \mathbf{M}\boldsymbol{\epsilon} - \mathbf{x}_k)^T\right] \tag{4.19}$$

In order to simplify equation 4.19, the products and transposes need to be resolved, which results in:

$$\begin{aligned}
\mathbf{P}_k &= E\left[(\mathbf{MAx}_k + \mathbf{MH}\boldsymbol{\nu} + \mathbf{M}\boldsymbol{\epsilon} - \mathbf{x}_k)(\mathbf{x}_k^T\mathbf{A}^T\mathbf{M}^T + \boldsymbol{\nu}^T\mathbf{H}^T\mathbf{M}^T + \boldsymbol{\epsilon}^T\mathbf{M}^T - \mathbf{x}_k^T)\right] \\
&= E\left[\mathbf{MAx}_k\mathbf{x}_k^T\mathbf{A}^T\mathbf{M}^T + \mathbf{MAx}_k\boldsymbol{\nu}^T\mathbf{H}^T\mathbf{M}^T + \mathbf{MAx}_k\boldsymbol{\epsilon}^T\mathbf{M}^T\right. \\
&\quad - \mathbf{MAx}_k\mathbf{x}_k^T + \mathbf{MH}\boldsymbol{\nu}\mathbf{x}_k^T\mathbf{A}^T\mathbf{M}^T + \mathbf{MH}\boldsymbol{\nu}\boldsymbol{\nu}^T\mathbf{H}^T\mathbf{M}^T + \mathbf{MH}\boldsymbol{\nu}\boldsymbol{\epsilon}^T\mathbf{M}^T \\
&\quad - \mathbf{MH}\boldsymbol{\nu}\mathbf{x}_k^T + \mathbf{M}\boldsymbol{\epsilon}\mathbf{x}_k^T\mathbf{A}^T\mathbf{M}^T + \mathbf{M}\boldsymbol{\epsilon}\boldsymbol{\nu}^T\mathbf{H}^T\mathbf{M}^T + \mathbf{M}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T\mathbf{M}^T \\
&\quad \left. - \mathbf{M}\boldsymbol{\epsilon}\mathbf{x}_k^T - \mathbf{x}_k\mathbf{x}_k^T\mathbf{A}^T\mathbf{M}^T - \mathbf{x}_k\boldsymbol{\nu}^T\mathbf{H}^T\mathbf{M}^T - \mathbf{x}_k\boldsymbol{\epsilon}^T\mathbf{M}^T + \mathbf{x}_k\mathbf{x}_k^T\right]
\end{aligned} \tag{4.20}$$

Applying the linearity of expectation and recognizing that $E[\boldsymbol{\nu}] = E[\boldsymbol{\epsilon}] = \mathbf{0}$, results in the simplification:

$$\begin{aligned}
\mathbf{P}_k = {} & \mathbf{MA}E[\mathbf{x}_k\mathbf{x}_k^T]\mathbf{A}^T\mathbf{M}^T - \mathbf{MA}E[\mathbf{x}_k\mathbf{x}_k^T] + \mathbf{MH}E[\boldsymbol{\nu}\boldsymbol{\nu}^T]\mathbf{H}^T\mathbf{M}^T + \mathbf{MH}E[\boldsymbol{\nu}\boldsymbol{\epsilon}^T]\mathbf{M}^T \\
& + \mathbf{M}E[\boldsymbol{\epsilon}\boldsymbol{\nu}^T]\mathbf{H}^T\mathbf{M}^T + \mathbf{M}E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T]\mathbf{M}^T - E[\mathbf{x}_k\mathbf{x}_k^T]\mathbf{A}^T\mathbf{M}^T + E[\mathbf{x}_k\mathbf{x}_k^T]
\end{aligned} \tag{4.21}$$

Assuming further that there is no correlation between the observation error and the propagation error, equation 4.21 reduces to:

$$\begin{aligned}
\mathbf{P}_k = {} & \mathbf{MAx}_k\mathbf{x}_k^T\mathbf{A}^T\mathbf{M}^T - \mathbf{MAx}_k\mathbf{x}_k^T + \mathbf{MH}E[\boldsymbol{\nu}\boldsymbol{\nu}^T]\mathbf{H}^T\mathbf{M}^T \\
& + \mathbf{M}E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T]\mathbf{M}^T - \mathbf{x}_k\mathbf{x}_k^T\mathbf{A}^T\mathbf{M}^T + \mathbf{x}_k\mathbf{x}_k^T
\end{aligned} \tag{4.22}$$

The third and fourth terms can be resolved by making use of equations 4.9 and 4.10. Applying further equation 4.15 yields:

$$\begin{aligned}\mathbf{P}_k &= \mathbf{x}_k\mathbf{x}_k^T - \mathbf{x}_k\mathbf{x}_k^T + \mathbf{M}\mathbf{H}\mathbf{Q}\mathbf{H}^T\mathbf{M}^T + \mathbf{M}\mathbf{R}\mathbf{M}^T - \mathbf{x}_k\mathbf{x}_k^T + \mathbf{x}_k\mathbf{x}_k^T \\ &= \mathbf{M}\mathbf{H}\mathbf{Q}\mathbf{H}^T\mathbf{M}^T + \mathbf{M}\mathbf{R}\mathbf{M}^T \\ &= \mathbf{M}(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R})\mathbf{M}^T \end{aligned} \tag{4.23}$$

The central term of equation 4.23 describes the sum of the observation error covariance $\mathbf{R}$ and the observation-matrix transformed propagation error covariance $\mathbf{H}\mathbf{Q}\mathbf{H}^T$. The matrix reduces to the classical covariance description ($\mathbf{P}_k = \mathbf{M}\mathbf{R}\mathbf{M}^T$) if no propagation error covariance is included.

The next step is to determine $\mathbf{M}$, such that equation 4.15 is fulfilled and equation 4.23 is minimized.

Computation of $\mathbf{M}$

The derivation of $\mathbf{M}$ can be treated as the optimization problem of finding a minimum value of $\mathbf{P}_k$ under the imposed condition of $\mathbf{M}\mathbf{A} = \mathbf{I}$. A suitable solution technique to this class of problems is the Lagrange-multiplier method. This technique defines a Lagrange-function, which incorporates the original function, the constraint, and a Lagrange-multiplier $\boldsymbol{\lambda}$. Finding the minimum of this function then results in the desired solution of $\mathbf{M}$. The Lagrange-function for $\mathbf{P}_k$ also has to make sure, that $\mathbf{P}_k$ remains symmetric, as it is a covariance matrix. Hence, any modification of the original function needs to be symmetric as well.

For any square matrix $\mathbf{C}$ it holds that $\mathbf{C} + \mathbf{C}^T$ is symmetric. A suitable Lagrange-function is thus given by equation 4.23 + $\mathbf{C}(\boldsymbol{\lambda}) + \mathbf{C}(\boldsymbol{\lambda})^T$, where $\mathbf{C}(\boldsymbol{\lambda}) = (\mathbf{I} - \mathbf{M}\mathbf{A})\boldsymbol{\lambda}$:

$$\mathbf{P}_k = \mathbf{M}(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R})\mathbf{M}^T + (\mathbf{I} - \mathbf{M}\mathbf{A})\boldsymbol{\lambda} + \boldsymbol{\lambda}^T(\mathbf{I} - \mathbf{M}\mathbf{A})^T \tag{4.24}$$

$\mathbf{M}$ is to minimize equation 4.24 and fulfill eq. 4.15. An intuitive approach for this would be to set the derivative of eq. 4.24 with respect to $\mathbf{M}$ equal to zero. Since $\mathbf{P}_k$ and $\mathbf{M}$ are matrices however, this would yield a tensor. A more elegant way is to require that a change in $\mathbf{M}$ results in zero change in $\mathbf{P}_k$. For this purpose, small matrix increments ($\delta\mathbf{M}$ and $\delta\mathbf{P}_k$) may be added to equation 4.24 whenever $\mathbf{M}$ and $\mathbf{P}_k$ are encountered. Using $\tilde{\mathbf{R}} = \mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R}$ then results in:

$$\begin{aligned}\mathbf{P}_k + \delta\mathbf{P}_k &= (\mathbf{M} + \delta\mathbf{M})\tilde{\mathbf{R}}(\mathbf{M} + \delta\mathbf{M})^T + \boldsymbol{\lambda} - (\mathbf{M} + \delta\mathbf{M})\mathbf{A}\boldsymbol{\lambda} \\ &\quad + \boldsymbol{\lambda}^T - \boldsymbol{\lambda}^T\mathbf{A}^T(\mathbf{M} + \delta\mathbf{M})^T \\ &= \mathbf{M}\tilde{\mathbf{R}}\mathbf{M}^T + \mathbf{M}\tilde{\mathbf{R}}\delta\mathbf{M}^T + \delta\mathbf{M}\tilde{\mathbf{R}}\mathbf{M}^T + \delta\mathbf{M}\tilde{\mathbf{R}}\delta\mathbf{M}^T + \boldsymbol{\lambda} \\ &\quad - \mathbf{M}\mathbf{A}\boldsymbol{\lambda} - \delta\mathbf{M}\mathbf{A}\boldsymbol{\lambda} + \boldsymbol{\lambda}^T - \boldsymbol{\lambda}^T\mathbf{A}^T\mathbf{M}^T - \boldsymbol{\lambda}^T\mathbf{A}^T\delta\mathbf{M}^T \end{aligned} \tag{4.25}$$

$\delta\mathbf{P}_k$ is then found by subtraction of equations 4.25 and 4.24:

$$\begin{aligned}\delta\mathbf{P}_k &= \mathbf{M}\tilde{\mathbf{R}}\delta\mathbf{M}^T + \delta\mathbf{M}\tilde{\mathbf{R}}\mathbf{M}^T + \delta\mathbf{M}\tilde{\mathbf{R}}\delta\mathbf{M}^T - \delta\mathbf{M}\mathbf{A}\boldsymbol{\lambda} - \boldsymbol{\lambda}^T\mathbf{A}^T\delta\mathbf{M}^T \\ &= (\mathbf{M}\tilde{\mathbf{R}} - \boldsymbol{\lambda}^T\mathbf{A}^T)\delta\mathbf{M}^T + \delta\mathbf{M}(\tilde{\mathbf{R}}\mathbf{M}^T - \mathbf{A}\boldsymbol{\lambda}) + \delta\mathbf{M}\tilde{\mathbf{R}}\delta\mathbf{M}^T \end{aligned} \tag{4.26}$$

Neglecting terms quadratic in the change to maintain a first-order relationship, results in:

$$\delta \mathbf{P}_k = (\mathbf{M}\tilde{\mathbf{R}} - \boldsymbol{\lambda}^T\mathbf{A}^T)\delta\mathbf{M}^T + \delta\mathbf{M}(\tilde{\mathbf{R}}\mathbf{M}^T - \mathbf{A}\boldsymbol{\lambda}) \stackrel{!}{=} \mathbf{0} \tag{4.27}$$

One way to fulfill equation 4.27 is to require

$$\mathbf{M}\tilde{\mathbf{R}} - \boldsymbol{\lambda}^T\mathbf{A}^T \stackrel{!}{=} \mathbf{0} \tag{4.28}$$

It should be noted that a zero change of $\mathbf{P}_k$ for a nonzero change of $\mathbf{M}$ could also apply to a maximum of $\mathbf{P}_k$. The proof that in fact a minimum value is found, is given in appendix A.5.

Equations 4.15 and 4.28 now yield two constraints for $\mathbf{M}$. Solving eq. 4.28 for $\mathbf{M}$ results in:

$$\mathbf{M} = \boldsymbol{\lambda}^T\mathbf{A}^T(\tilde{\mathbf{R}}^{-1})^T \tag{4.29}$$

Since $\tilde{\mathbf{R}}$ is regular, its inverse and the transpose commute. Furthermore, due to $\tilde{\mathbf{R}}^{-1}$ being symmetric, the transpose vanishes, which allows to write:

$$\mathbf{M} = \boldsymbol{\lambda}^T\mathbf{A}^T\tilde{\mathbf{R}}^{-1} \tag{4.30}$$

Inserting equation 4.30 into eq. 4.15 leads to:

$$\boldsymbol{\lambda}^T\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{A} = \mathbf{I} \tag{4.31}$$

Assuming that $\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{A}$ is invertible, an expression for $\boldsymbol{\lambda}^T$ can be found:

$$\boldsymbol{\lambda}^T = (\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{A})^{-1} \tag{4.32}$$

Equation 4.32 can now be inserted into eq. 4.30, which yields the desired expression for $\mathbf{M}$ that satisfies the unbiased, linear and minimum-variance requirements:

$$\mathbf{M} = (\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{A})^{-1}\mathbf{A}^T\tilde{\mathbf{R}}^{-1} \tag{4.33}$$

Inserting equation 4.33 into eq. 4.23 yields the corresponding expression for $\mathbf{P}_k$:

$$\begin{aligned}\mathbf{P}_k &= \mathbf{M}\tilde{\mathbf{R}}\mathbf{M}^T = (\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{A})^{-1}\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{R}}\left[(\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{A})^{-1}\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\right]^T \\ &= (\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{A})^{-1}\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{A}(\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{A})^{-1} \\ &= (\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{A})^{-1} = (\mathbf{A}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{A})^{-1}\end{aligned} \tag{4.34}$$

An updated expression of the state estimate can be obtained by inserting equation 4.33 into eq. 4.12:

$$\hat{\mathbf{x}}_k = (\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{A})^{-1}\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{y} = \mathbf{P}_k\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{y} \tag{4.35}$$

$$= [\mathbf{A}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{A}]^{-1}\mathbf{A}^T(\mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{y} \tag{4.36}$$

Equations 4.34 and 4.36 represent the desired extension of the WLS algorithm for the case of process noise; however, they are not recommended for implementation in computer code, as $\mathbf{HQH}^T + \mathbf{R}$ is of dimension $m \times m$. Since m represents the number of observations and $m \gg n$, computing the inverse of $(\mathbf{HQH}^T + \mathbf{R})$ is extremely time-consuming. The classical weighted least squares algorithm also needs to compute $\mathbf{R}^{-1}$, however since $\mathbf{R}$ is a diagonal matrix (and in practice constant for a certain instrument calibration), the computation of the inverse is very efficient. $(\mathbf{HQH}^T + \mathbf{R})$ on the other hand is not diagonal. Hence an optimized expression of $\mathbf{P}_k$ which splits the inverse and reduces the computational burden is required. Also an equation of the form $\mathbf{P}_k = \tilde{\mathbf{P}}_k + \delta\mathbf{P}_k$, where $\tilde{\mathbf{P}}_k$ is the covariance of the classical weighted least squares algorithm, is desirable as it splits the updated covariance into the classical formulation and a delta covariance that contains the additional contribution due to the consideration of model uncertainties.

This can be achieved either by making use of the considerations in appendix A.6 or via direct application of the Sherman-Morrison-Woodbury identity (Golub and Van Loan, 1996, Section 2.1.3), which reads:

$$(\mathbf{R} - \mathbf{UH}^T)^{-1} = \mathbf{R}^{-1} + \mathbf{R}^{-1}\mathbf{U}(\mathbf{I} - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{U})^{-1}\mathbf{H}^T\mathbf{R}^{-1} \tag{4.37}$$

Defining $\mathbf{U} = -\mathbf{HQ}$ then results in:

$$(\mathbf{R} + \mathbf{HQH}^T)^{-1} = \mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{HQ}(\mathbf{I} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{HQ})^{-1}\mathbf{H}^T\mathbf{R}^{-1} \tag{4.38}$$

Compatibility with the derivation in appendix A.6 is obtained by applying the identity

$$\mathbf{Q}(\mathbf{I} + \mathbf{BQ})^{-1} = (\mathbf{I} + \mathbf{QB})^{-1}\mathbf{Q} \tag{4.39}$$

which is derived in appendix A.6 and yields the alternative form:

$$(\mathbf{HQH}^T + \mathbf{R})^{-1} = \mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{H}(\mathbf{I} + \mathbf{QH}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{QH}^T\mathbf{R}^{-1} \tag{4.40}$$

This step is not strictly required, however, as equation 4.38 already possesses the desired properties. $\mathbf{QH}^T\mathbf{R}^{-1}\mathbf{H}$ in equation 4.40 is $(n \times n) \times (n \times m) \times (m \times m) \times (m \times n) = (n \times n)$. Therefore the order of the inverse significantly reduced to $n \times n$ instead of $m \times m$. To get an impression of the difference between m and n, consider 10 observations of 3 observables each. Then $m = 30$, while n is still the dimension of the state vector, i.e. 6. In this case, equation 4.40 reduces the order of the inverse from 30 to 6 – a substantial performance improvement. Equations 4.40 and 4.34 lead to:

$$\mathbf{P}_k = (\mathbf{A}^T\mathbf{R}^{-1}\mathbf{A} - \mathbf{A}^T\mathbf{R}^{-1}\mathbf{H}(\mathbf{I} + \mathbf{QH}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{QH}^T\mathbf{R}^{-1}\mathbf{A})^{-1} \tag{4.41}$$

The first term in equation 4.41, $\mathbf{A}^T\mathbf{R}^{-1}\mathbf{A}$, is the inverse of the classical weighted least squares covariance matrix, which does not include any

propagation error. Therefore it is possible to define $\mathbf{A}^T\mathbf{R}^{-1}\mathbf{A} = \tilde{\mathbf{P}}_k^{-1}$, which allows to write:

$$\mathbf{P}_k = \left(\tilde{\mathbf{P}}_k^{-1} - \mathbf{A}^T\mathbf{R}^{-1}\mathbf{H}(\mathbf{I} + \mathbf{Q}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{Q}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{A}\right)^{-1} \tag{4.42}$$

Equation 4.42 represents the optimized description of the state vector error covariance that is suitable for implementation in computer code. It only contains inverses of order $n \times n$, except for $\mathbf{R}^{-1}$, which, albeit being $m \times m$, is easily evaluated since $\mathbf{R}$ is typically diagonal and constant.

Also an alternative form of equation 4.42 that allows to be written as $\mathbf{P}_k = \tilde{\mathbf{P}}_k + \delta\mathbf{P}_k$ can be obtained. To this end, define $\mathbf{S} \coloneqq \mathbf{A}^T\mathbf{R}^{-1}\mathbf{H}$, which yields:

$$\mathbf{P}_k^{-1} = \tilde{\mathbf{P}}_k^{-1} - \mathbf{S}(\mathbf{I} + \mathbf{Q}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{Q}\mathbf{S}^T \tag{4.43}$$

Applying identity 4.39 results in:

$$\mathbf{P}_k^{-1} = \tilde{\mathbf{P}}_k^{-1} - \mathbf{S}\mathbf{Q}(\mathbf{I} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{Q})^{-1}\mathbf{S}^T \tag{4.44}$$

Defining $\mathbf{N} = -\mathbf{Q}(\mathbf{I} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{Q})^{-1}$, which is $n \times n$, then leads to:

$$\mathbf{P}_k^{-1} = \tilde{\mathbf{P}}_k^{-1} + \mathbf{S}\mathbf{N}\mathbf{S}^T \tag{4.45}$$

Applying the matrix inverse leads to an expression compatible with the Sherman-Morrison-Woodbury identity:

$$\mathbf{P}_k = \left[\tilde{\mathbf{P}}_k^{-1} + \mathbf{S}\mathbf{N}\mathbf{S}^T\right]^{-1} \tag{4.46}$$

Equation 4.40 using $\mathbf{R} \to \tilde{\mathbf{P}}_k^{-1}$, $\mathbf{H} \to \mathbf{S}$ and $\mathbf{Q} \to \mathbf{N}$ yields:

$$\mathbf{P}_k = \tilde{\mathbf{P}}_k - \tilde{\mathbf{P}}_k\mathbf{S}(\mathbf{I} + \mathbf{N}\mathbf{S}^T\tilde{\mathbf{P}}_k\mathbf{S})^{-1}\mathbf{N}\mathbf{S}^T\tilde{\mathbf{P}}_k \tag{4.47}$$

Another application of identity 4.39 leads to the alternative form

$$\mathbf{P}_k = \tilde{\mathbf{P}}_k - \tilde{\mathbf{P}}_k\mathbf{S}\mathbf{N}(\mathbf{I} + \mathbf{S}^T\tilde{\mathbf{P}}_k\mathbf{S}\mathbf{N})^{-1}\mathbf{S}^T\tilde{\mathbf{P}}_k \tag{4.48}$$

which after full expansion reads:

$$\begin{aligned}\mathbf{P}_k = \tilde{\mathbf{P}}_k + \underbrace{\tilde{\mathbf{P}}_k \underbrace{\mathbf{A}^T\mathbf{R}^{-1}\mathbf{H}}_{\mathbf{S}} \underbrace{\mathbf{Q}(\mathbf{I} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{Q})^{-1}}_{-\mathbf{N}}}_{\cdots} \\ \underbrace{\left[\mathbf{I} - \underbrace{\mathbf{H}^T\mathbf{R}^{-1}\mathbf{A}}_{\mathbf{S}^T}\tilde{\mathbf{P}}_k \underbrace{\mathbf{A}^T\mathbf{R}^{-1}\mathbf{H}}_{\mathbf{S}} \underbrace{\mathbf{Q}(\mathbf{I} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{Q})^{-1}}_{-\mathbf{N}}\right]^{-1}}_{\delta\mathbf{P}_k} \cdots \\ \underbrace{\underbrace{\mathbf{H}^T\mathbf{R}^{-1}\mathbf{A}}_{\mathbf{S}^T}\tilde{\mathbf{P}}_k}_{\cdots}\end{aligned} \tag{4.49}$$

Equation 4.49 expresses the covariance of the propagation error considering weighted least squares algorithm as the sum of the classical non-propagation error WLS covariance and $\delta \mathbf{P}_k$. It requires the computation of three inverses, of which one is $\mathbf{R}^{-1}$ and the other two are of order $n \times n$. Despite nicely resolving the additional contribution of the propagation error to the state vector error covariance, equation 4.49 has no computational advantage with respect to eq. 4.42. Furthermore, since equation 4.42 expresses the covariance as the sum of $\tilde{\mathbf{P}}_k^{-1}$ and a secondary matrix, it can directly use the result of $\tilde{\mathbf{P}}_k^{-1}$, which is computed in the classical WLS for the calculation of $\tilde{\mathbf{P}}_k$. For this reason, and the fact that equation 4.42 contains less matrix multiplications than eq. 4.49, the preferred description of the covariance for implementation is formula 4.42.

4.2.2 *Sequential Batch Weighted Least Squares*

In case of the batch weighted least squares algorithm, all measurements are evaluated together in order to determine an estimate of the state and corresponding uncertainty at the time of a reference epoch, typically called t_0. *Sequential* batch weighted least squares orbit estimation is concerned with the update of a previously computed state and covariance. A practical example is the initial computation of a state and covariance using the WLS algorithm. Then, some time later, another batch of measurements (e.g. a new measurement track) arrives and shall be used to compute an updated state and covariance for the chosen epoch t_0. Obviously the success of such an update depends on the amount of previous measurements used to compute the epoch state estimate and the number of new measurements supplied to generate the update. The method typically works well only for two to three measurement batches and tends to diverge afterwards (Vallado, 2013, p. 774).

The basic solution to the state and covariance update problem is to consider the a priori information as an independent "observation" of the state at epoch, which is described by equation 4.50. The new measurement model is formulated in accordance with eq. 4.8 via equation 4.51:

$$\bar{\mathbf{x}}_0 = \mathbf{x}_0 + \boldsymbol{\eta}_k \tag{4.50}$$

$$\mathbf{y}_k = \mathbf{H}_k(\boldsymbol{\Phi}(t_k, t_0)\,\mathbf{x}_0 + \boldsymbol{\nu}_k) + \boldsymbol{\epsilon}_k = \mathbf{A}_k\mathbf{x}_0 + \mathbf{H}_k\boldsymbol{\nu}_k + \boldsymbol{\epsilon}_k \tag{4.51}$$

where $\boldsymbol{\epsilon}_k$ represents the measurement noise vector and $\boldsymbol{\nu}_k$ is the vector of the propagation errors from the epoch to the time of the new measurements. $\boldsymbol{\eta}_k$ in equation 4.50 is the error in the prior estimate. Therefore its covariance matrix is the one obtained from the previous run of the force model uncertainty considering WLS algorithm pre-

sented in section 4.2.1, i.e. $\mathrm{E}[\boldsymbol{\eta}_k \boldsymbol{\eta}_k^T] = \bar{\mathbf{P}}_0$. Combining equations 4.50 and 4.51 in a vectorized form yields:

$$\underbrace{\begin{bmatrix} \mathbf{y}_k \\ \bar{\mathbf{x}}_0 \end{bmatrix}}_{=:\mathbf{y}^*} = \underbrace{\begin{bmatrix} \mathbf{A}_k \\ \mathbf{I} \end{bmatrix}}_{=:\mathbf{A}^*} \mathbf{x}_0 + \underbrace{\begin{bmatrix} \mathbf{H}_k \boldsymbol{\nu}_k + \boldsymbol{\epsilon}_k \\ \boldsymbol{\eta}_k \end{bmatrix}}_{=:\boldsymbol{\epsilon}^*} \tag{4.52}$$

Invoking the assumption that the propagation and observation errors are independent of the error in the a priori information, the covariance of the error can be formulated as:

$$\begin{aligned} \mathrm{E}[\boldsymbol{\epsilon}^* \boldsymbol{\epsilon}^{*T}] &= \mathrm{E}\begin{bmatrix} (\mathbf{H}_k \boldsymbol{\nu}_k + \boldsymbol{\epsilon}_k)(\boldsymbol{\nu}_k^T \mathbf{H}_k^T + \boldsymbol{\epsilon}_k^T) & (\mathbf{H}_k \boldsymbol{\nu}_k + \boldsymbol{\epsilon}_k)\boldsymbol{\eta}_k \\ \boldsymbol{\eta}_k (\boldsymbol{\nu}_k^T \mathbf{H}_k^T + \boldsymbol{\epsilon}_k^T) & \boldsymbol{\eta}_k \boldsymbol{\eta}_k^T \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{P}}_0 \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{P}}_0 \end{bmatrix} = \mathbf{R}^* \end{aligned} \tag{4.53}$$

Equation 4.52 conforms to the classical WLS problem. It is therefore possible to directly apply its solution to eq. 4.52 in order to obtain expressions for the updated $\hat{\mathbf{x}}_0$ and $\mathbf{P}_0$:

$$\begin{aligned} \hat{\mathbf{x}}_0 &= \left(\mathbf{A}^{*T} \mathbf{R}^{*-1} \mathbf{A}^*\right)^{-1} \mathbf{A}^{*T} \mathbf{R}^{*-1} \mathbf{y}^* \\ &= \left[\begin{pmatrix} \mathbf{A}_k^T & \mathbf{I} \end{pmatrix} \begin{bmatrix} \tilde{\mathbf{R}}^{-1} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{P}}_0^{-1} \end{bmatrix} \begin{pmatrix} \mathbf{A}_k \\ \mathbf{I} \end{pmatrix} \right]^{-1} \begin{pmatrix} \mathbf{A}_k^T & \mathbf{I} \end{pmatrix} \begin{bmatrix} \tilde{\mathbf{R}}^{-1} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{P}}_0^{-1} \end{bmatrix} \begin{pmatrix} \mathbf{y}_k \\ \bar{\mathbf{x}}_0 \end{pmatrix} \\ &= \left(\mathbf{A}_k^T \tilde{\mathbf{R}}^{-1} \mathbf{A}_k + \bar{\mathbf{P}}_0^{-1}\right)^{-1} \left(\mathbf{A}_k^T \tilde{\mathbf{R}}^{-1} \mathbf{y}_k + \bar{\mathbf{P}}_0^{-1} \bar{\mathbf{x}}_0\right) \end{aligned} \tag{4.54}$$

The final equations can now directly be inferred from equation 4.54, considering that:

$$\bar{\mathbf{P}}_0^{-1} = \mathbf{A}_{\mathrm{prev}}^T \tilde{\mathbf{R}}_{\mathrm{prev}}^{-1} \mathbf{A}_{\mathrm{prev}} \tag{4.55}$$

and

$$\bar{\mathbf{x}}_0 = (\mathbf{A}_{\mathrm{prev}}^T \tilde{\mathbf{R}}_{\mathrm{prev}}^{-1} \mathbf{A}_{\mathrm{prev}})^{-1} \mathbf{A}^T \tilde{\mathbf{R}}_{\mathrm{prev}}^{-1} \mathbf{y}_{\mathrm{prev}} \tag{4.56}$$

according to equations 4.35 and 4.34. Using eqs. 4.55 and 4.56, the term $\bar{\mathbf{P}}_0^{-1} \bar{\mathbf{x}}_0$ in equation 4.54 can be simplified to:

$$\begin{aligned} \bar{\mathbf{P}}_0^{-1} \bar{\mathbf{x}}_0 &= \mathbf{A}_{\mathrm{prev}}^T \tilde{\mathbf{R}}_{\mathrm{prev}}^{-1} \mathbf{A}_{\mathrm{prev}} (\mathbf{A}_{\mathrm{prev}}^T \tilde{\mathbf{R}}_{\mathrm{prev}}^{-1} \mathbf{A}_{\mathrm{prev}})^{-1} \mathbf{A}_{\mathrm{prev}}^T \tilde{\mathbf{R}}_{\mathrm{prev}}^{-1} \mathbf{y}_{\mathrm{prev}} \\ &= \mathbf{A}_{\mathrm{prev}}^T \tilde{\mathbf{R}}^{-1} \mathbf{y}_{\mathrm{prev}} \end{aligned} \tag{4.57}$$

Inserting equations 4.55 and 4.57 into eq. 4.54 leads to:

$$\begin{aligned} \hat{\mathbf{x}}_0 &= \left(\mathbf{A}_k^T \tilde{\mathbf{R}}^{-1} \mathbf{A}_k + \mathbf{A}_{\mathrm{prev}}^T \tilde{\mathbf{R}}_{\mathrm{prev}}^{-1} \mathbf{A}_{\mathrm{prev}}\right)^{-1} \left(\mathbf{A}_k^T \tilde{\mathbf{R}}^{-1} \mathbf{y}_k + \mathbf{A}_{\mathrm{prev}}^T \tilde{\mathbf{R}}_{\mathrm{prev}}^{-1} \mathbf{y}_{\mathrm{prev}}\right) \\ &= \mathbf{P}_0 \left(\mathbf{A}_k^T \tilde{\mathbf{R}}^{-1} \mathbf{y}_k + \mathbf{A}_{\mathrm{prev}}^T \tilde{\mathbf{R}}_{\mathrm{prev}}^{-1} \mathbf{y}_{\mathrm{prev}}\right) \end{aligned} \tag{4.58}$$

where:

$$\mathbf{P}_0 = (\mathbf{A}_k^T \tilde{\mathbf{R}}^{-1} \mathbf{A}_k + \mathbf{A}_{\mathrm{prev}}^T \tilde{\mathbf{R}}_{\mathrm{prev}}^{-1} \mathbf{A}_{\mathrm{prev}})^{-1} = (\mathbf{A}_k^T \tilde{\mathbf{R}}^{-1} \mathbf{A}_k + \bar{\mathbf{P}}_0^{-1})^{-1} \tag{4.59}$$

Equations 4.58 and 4.59 represent the desired tools to update the state and covariance with propagation uncertainty consideration. The derivation shows that the algorithm reduces to the classical SBWLS, if $\tilde{\mathbf{R}} = \mathbf{H}\mathbf{Q}\mathbf{H}^T + \mathbf{R}$ is replaced by $\mathbf{R}$. It is hence enough to store $\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{A}$ and $\mathbf{A}^T\tilde{\mathbf{R}}^{-1}\mathbf{y}$ after each weighted least squares computation of the current batch of data. Once a new batch of measurements arrives, it can be updated with this prior knowledge via equations 4.58 and 4.59, where $\tilde{\mathbf{R}}^{-1}$ is computed according to equation 4.40. The framework therefore also fits the sequential batch processing seamlessly.

4.2.3 *Practical considerations*

The algorithms derived in sections 4.2.1 and 4.2.2 achieve their improved accuracy and more realistic covariances through additional calculations. They are therefore more time-consuming than their classical counterparts.

First of all, the combined process noise matrix $\mathbf{Q}$ has to be computed, which is achieved from the evaluation of equation 2.306 for the combined density covariance matrix, algorithm 3.1 for the orbital gravity error covariance and subsequent summing. This assumes that there is no or negligible correlation between density errors and gravity errors, which seems to be a justifiable assumption for potential field coefficient uncertainty (gravity errors of commission). The uncertainty attributed to gravity errors of omission is derived based on a globally averaged stochastic process model. Density errors predominantly influence the in-track position which, under the hypothesis of the applicability of the stochastic gravity error model, does not influence the gravity errors. This is by no means a rigorous proof, but taken together with the sequential filter simulation results presented in section 4.3, it provides an indication why no correlation analysis was pursued.

An additional effort is the evaluation of the $n \times n$ matrix inverse in equation 4.40, which needs to be computed for every measurement. Any means of reducing the operational duration is therefore desirable. As demonstrated in Woodburn and Coppola (2013), the selection of the coordinate frame used for the orbit estimation process, does not affect the final SBWLS solution, however the path of convergence. A secondary finding of Woodburn and Coppola (2013) is that it is equivalent to perform all operations or only the state updates in equinoctial elements. In fact, the realization of state updates in equinoctial elements can significantly reduce the number of SBWLS iterations required until convergence is achieved (Woodburn and Coppola, 2013). For practical applications this means that a single Jacobi matrix calculation (Cartesian to EOE), as well as a two state conversions (EOE to Cartesian and vice versa) are performed additionally in each iteration to increase the chance of saving some WLS/SBWLS iterations. This is generally a very attractive deal, since the extra workload of these calculations is small,

but the potential savings are large. It is hence strongly recommended to perform the least squares state updates according to the following strategy:

1. Compute the state correction according to equation 4.36 (WLS) or 4.58 (SBWLS), where eq. 4.40 is used to calculate $\tilde{\mathbf{R}}^{-1}$
2. Compute the Cartesian to equinoctial Jacobi matrix $\mathbf{J}_{\text{EOE,Cart}}$ using the current estimate of the epoch state and algorithm B.6
3. Convert the state correction to EOE via $\Delta\boldsymbol{x}_{\text{EOE}} = \mathbf{J}_{\text{EOE,Cart}}\Delta\boldsymbol{x}_{\text{Cart}}$
4. Convert the state from Cartesian to EOE
5. Perform the state update ($\hat{\mathbf{x}}_{0,\text{EOE}} = \hat{\mathbf{x}}_{0,\text{EOE}} + \Delta\boldsymbol{x}_{\text{EOE}}$)
6. Convert the updated state back to inertial Cartesian coordinates

The extended least squares algorithms perform well for the different kinds of density errors, as well as gravity errors of commission. The global WLS linearization however does not harmonize with gravity errors of omission, which are epistemic in nature. While formally the global averaging results in the assumed stochastic model to be zero-mean, this does not necessarily apply to any particular trajectory, which in practice exhibits a certain bias.

The updated estimators (eqs. 4.36, 4.40 and 4.42 for WLS, as well as 4.58, 4.59 and 4.40 for SBWLS) were derived under the assumption of the process noise being zero-mean. Since the least squares OD methods perform global linearization, the increasing violation of this requirement leads to inconsistent estimates when attempting to consider gravity errors of omission.

4.2.4 *Simulation results*

This section is devoted to the presentation of orbit determination results using the force model uncertainty quantification framework in combination with the process noise extension of the classical least squares estimators. Density uncertainty is first considered separately for solar flux input uncertainty (based on white noise or random walk error process models) and geomagnetic activity uncertainty (white noise error process model). In a second step, the full combination of grid-scale model uncertainty (Ornstein-Uhlenbeck error process), geomagnetic index uncertainty, solar flux uncertainty and gravity uncertainty due to errors of commission is presented. Semi-empirical models considered are DTM-2013, DTM-2012 and NRLMSISE-00. Also results of the classical least squares algorithms are depicted to demonstrate the improvement achieved through the physical process noise consideration.

Key simulation parameters or their changes (object, observer and scenario properties) are listed alongside the simulation results. Monte-Carlo simulations are used as state of the art validation technique. Due to the extensive number of possible combinations (epochs, er-

ror process models, uncertainty origins, density/gravity models) the presentation is limited to the position error components of the state vector. Further simulation results for the solar flux error case can be found in Schiemenz et al. (2019c), which also details the less accurate frame conversion techniques used to obtain the Cartesian covariance results in the early stages of the uncertainty quantification framework development.

SOLAR FLUX ERRORS

Error process model: White noise, density model: NRLMSISE-00

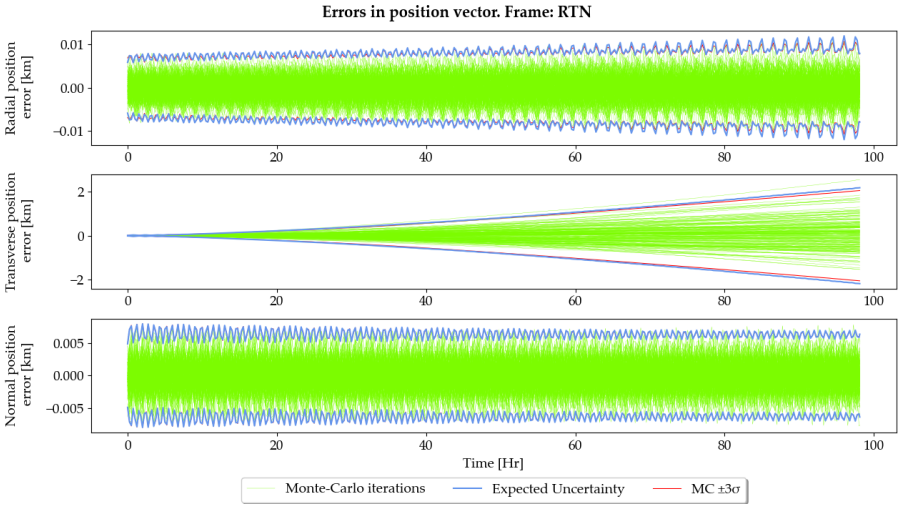


Figure 4.2: Extended WLS OD using NRLMSISE-00: Monte-Carlo position errors, 3σ bounds and estimated 3σ uncertainty due to solar flux errors

Error process model: White noise, density model: DTM-2012

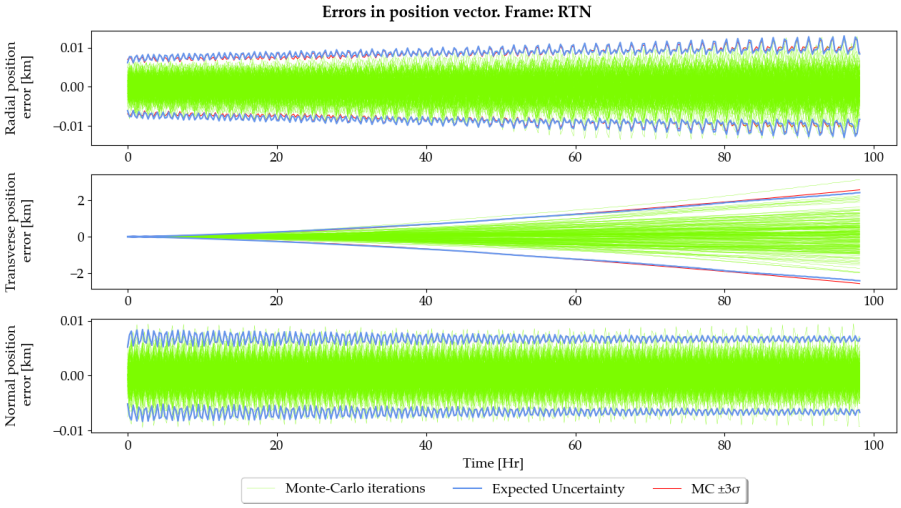


Figure 4.3: Extended WLS OD using DTM-2012: Monte-Carlo position errors, 3σ bounds and estimated 3σ uncertainty due to solar flux errors

Scenario parameters:

MC iterations: 200
t_0: *2015/Oct/02-00:00:00*
Duration: 4.5 days
Propagation step: 15 min

Perturbations:
30×30 *aspherical potential field (GGM02C)*
Atmospheric drag
Solar radiation pressure
Third body gravity (soli-lunar)

Solar flux errors:
Double-sided PSD of white noise process: 130 000 $\mathrm{sfu}^2\mathrm{s}$
Sampling: daily

Observer:
Type: Optical
Latitude: 49.7813 deg
Longitude: 9.973 94 deg
RaDec accuracy: 10^{-3} deg

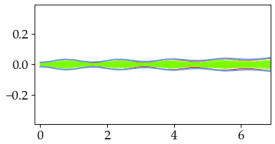
Object:

$$\mathbf{x}_0 : \begin{pmatrix} 1409.381\,740\ \mathrm{km} \\ -3738.321\,296\ \mathrm{km} \\ -5619.112\,527\ \mathrm{km} \\ 0.552\,548\ \mathrm{km/s} \\ -6.241\,291\ \mathrm{km/s} \\ 4.293\,374\ \mathrm{km/s} \end{pmatrix}$$

Frame: GCRF
Semi-major axis: 6880.2 km
BC: 0.06 m^2/kg

Error process model: Random Walk, density model: DTM-2012

Zoom on first hours of the transverse position error



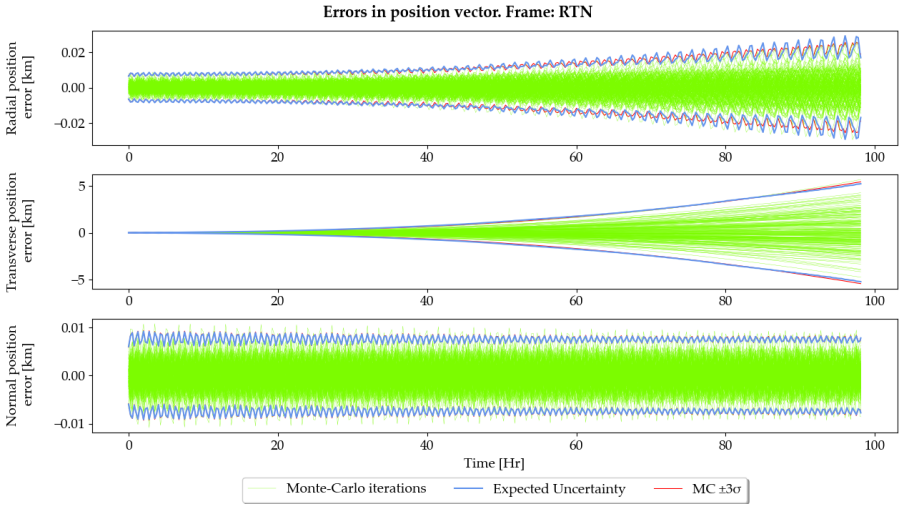


Figure 4.4: Extended WLS OD using DTM-2012: Monte-Carlo position errors, 3σ bounds and estimated 3σ uncertainty due to solar flux errors

Figures 4.2 to 4.4 represent exemplary simulation results obtained for white noise and Brownian motion error process models. In all cases the three sigma uncertainty (blue) is correctly estimated throughout the entire simulation time. The uncertainty predictions are alike for NRL-MSISE-00 and DTM-2012 in case of the white noise F10.7 measurement error process.

In contrast to the Brownian motion case in figure 4.4, it becomes apparent, that despite the differences in the error growth (t^3 for the white noise process and t^5 for the random walk model) the initial error growth is faster for the white noise process. Comparing equations 2.213 and 2.215, the origin of this observation is found in the dependence on the error process sampling time δt, which is reciprocal for the Brownian motion case. Therefore it takes the random walk case about 2.5 days to overtake the white noise error process model.

The results of a traditional WLS estimator (see e.g. algorithm 67 in Vallado, 2013) which only considers measurement uncertainty is depicted in figure 4.5 for the same scenario as in figure 4.4. In the case of the classical estimator it is at no point in time possible to attribute any realism to the estimated uncertainties. Likewise, the estimated uncertainties do not grow after the WLS computation of the initial state vector, since the estimated initial radial position variance is small and no propagation uncertainty is considered.

In summary, the solar flux uncertainty consideration makes a non-negligible contribution to improving the realism of the estimated uncertainty. The extension of the classical WLS estimator is able to process the physics-based analytic process noise matrices.

Traditional WLS estimator - no propagation error consideration
Error process model: Random Walk, density model: DTM-2012

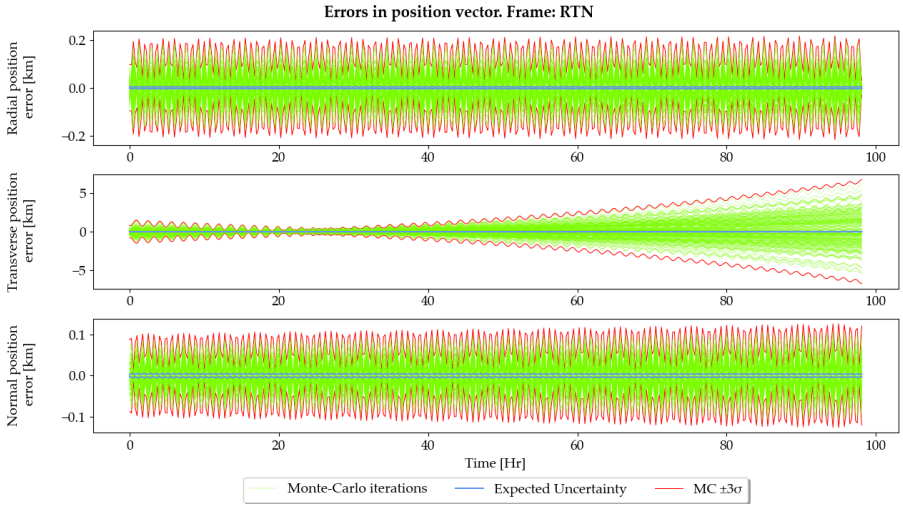


Figure 4.5: Traditional WLS OD using DTM-2012: Monte-Carlo position errors, 3σ bounds and estimated 3σ uncertainty due to solar flux errors

GEOMAGNETIC INDEX ERRORS

OD results using the extended least squares estimators and geomagnetic index uncertainty are presented in figures 4.6 to 4.10. In contrast to the solar flux scenario, the epochs are chosen based on periods of low geomagnetic activity.

Changes with respect to solar flux scenario:

t_0: *2011/Jun/18-00:00:00*
Gravity model: GGM05C

Geomagnetic index errors:

NRLMSISE-00:
Double-sided PSD of white noise process:
$21\,600\,\mathrm{AP}\,\mathrm{units}^2\mathrm{s}$
Sampling: daily

DTM:
Double-sided PSD of white noise process:
$2700\,\mathrm{AP}\,\mathrm{units}^2\mathrm{s}$
Sampling: 3 h

Object:

$$\mathbf{x}_0 : \begin{pmatrix} 3980.760\,745\,\mathrm{km} \\ -2950.088\,078\,\mathrm{km} \\ 4655.608\,963\,\mathrm{km} \\ 3.994\,088\,\mathrm{km/s} \\ -3.416\,397\,\mathrm{km/s} \\ -5.568\,957\,\mathrm{km/s} \end{pmatrix}$$

Semi-major axis: 6800 km

Extended WLS, density model: NRLMSISE-00

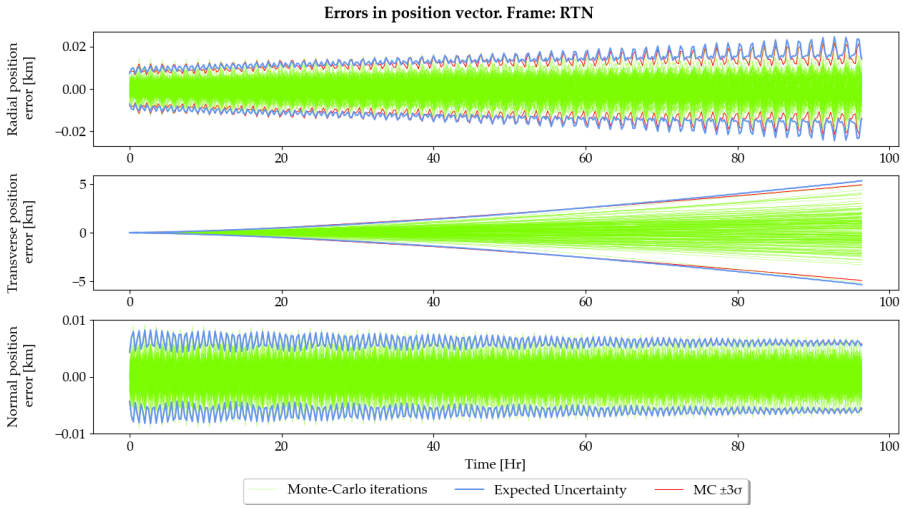
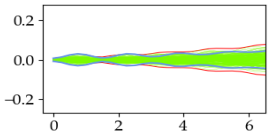


Figure 4.6: Extended WLS OD using NRLMSISE-00: Monte-Carlo position errors, 3σ bounds and estimated 3σ uncertainty during low geomagnetic activity

Extended WLS, density model: DTM-2013

Zoom on first hours of the transverse position error



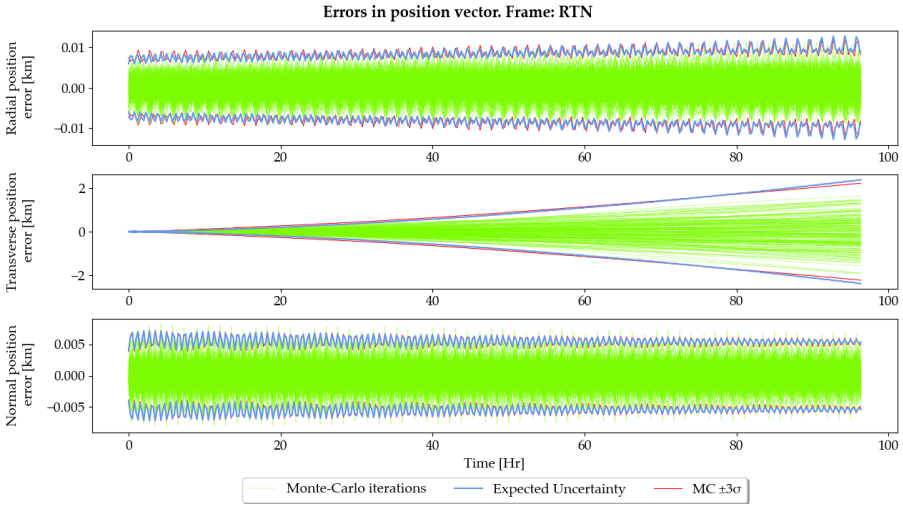


Figure 4.7: Extended WLS OD using DTM-13: Monte-Carlo position errors, 3σ bounds and estimated 3σ uncertainty due to geomagnetic index errors

Traditional WLS estimator - no propagation error consideration
Density model: DTM-2013

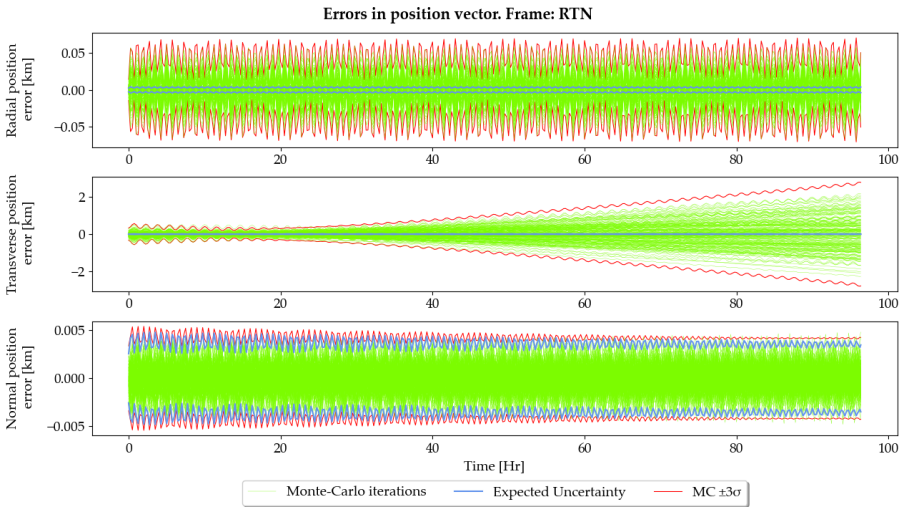


Figure 4.8: Traditional WLS OD using DTM-13: Monte-Carlo position errors, 3σ bounds and estimated 3σ uncertainty due to geomagnetic index errors

Figures 4.7 and 4.8 depict the same scenario, once using magnetic index uncertainty propagation quantification and once using a classical WLS estimator. The differences between the simulation results are qualitatively similar to the solar flux scenario insofar as the traditional WLS estimator is not able to generate a consistent state vector estimate at the epoch t_0. Likewise, the uncertainty-growth in the following propagation phase is not covered in the classical case, which is particularly noticeable in the dominant transverse direction.

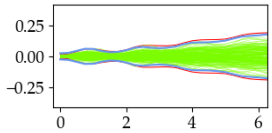
Another example using NRLMSISE-00 is shown in figure 4.9 for a period of very low geomagnetic activity, which is when the strongest orbital variability due to uncertain geomagnetic activity information can be expected. Indeed, the transverse 3σ boundary reaches 10 km after 4.5 days, which is twice as much as in figure 4.6.

Extended WLS, density model: NRLMSISE-00

Changes with respect to figure 4.6:

t_0: *2014/Nov/18-00:00:00*

Zoom on first hours of the transverse position error



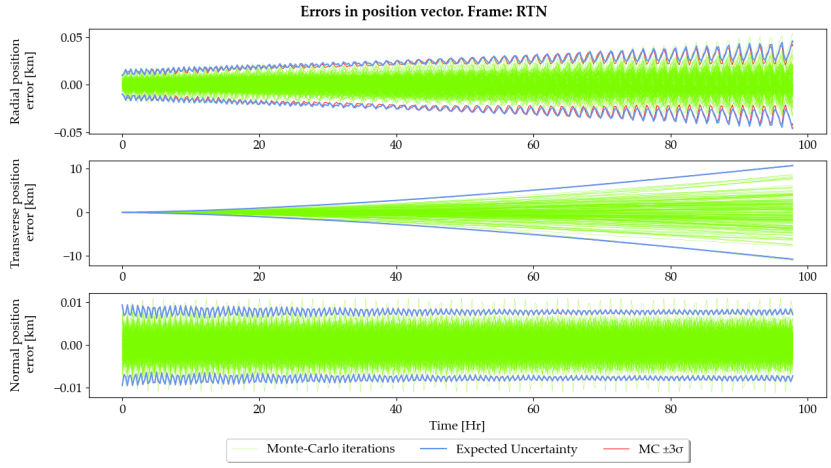


Figure 4.9: Extended WLS OD using NRLMSISE-00 (very low geomagnetic activity): Monte-Carlo position errors, 3σ bounds and expected 3σ uncertainty

Extended SBWLS, Density model: NRLMSISE-00

Changes with respect to solar flux scenario:

t_0: *2011/Oct/01-20:00:00*

Geomagnetic index errors:

NRLMSISE-00:
Double-sided PSD of white noise process:
21 600 AP units^2s
Sampling: daily

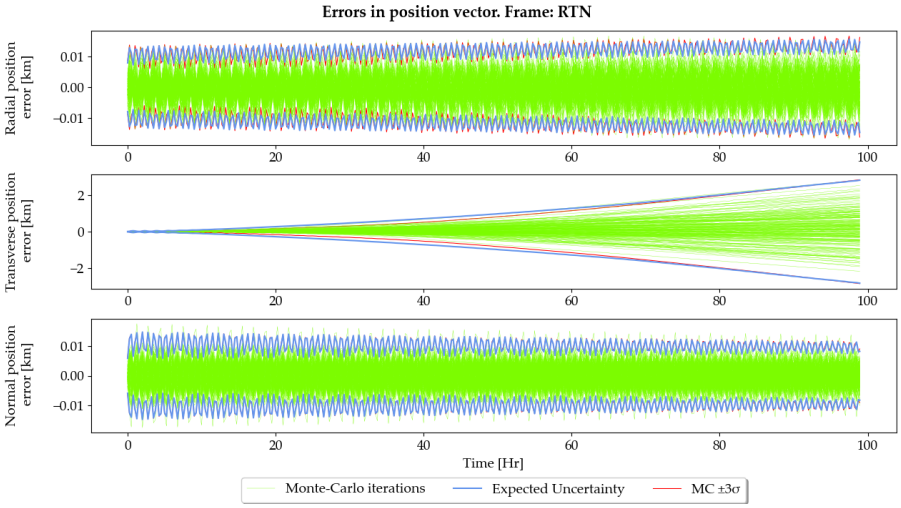


Figure 4.10: Extended SBWLS OD using NRLMSISE-00 for geomagnetic index errors: Monte-Carlo position errors, 3σ bounds and estimated 3σ uncertainty

Finally, also an example of the sequential batch weighted least squares algorithm is presented in figure 4.10. The SBWLS estimate is sensitive to the number of batches used for state updates. The process starts to diverge as soon as the new observations no longer contribute to reducing the uncertainty of the epoch estimate. This is the case

for short tracks and, especially considering propagation uncertainties, also for increasing periods between the epoch and the observation timestamps. To obtain consistent results, the simulation corresponding to figure 4.10 generates measurements only during the object visibility periods in the first 36 h after t_0 and groups them into three batches. These constraints complicate the practical application of the SBWLS estimator. It should therefore be used with caution.

SOLAR FLUX/GEOMAGNETIC INDEX/GRID-SCALE DENSITY ERRORS AND GRAVITY ERRORS OF COMMISSION

The complete part of the force model uncertainty quantification framework that is compatible with WLS orbit determination is applied in the simulations presented hereafter. Different model combinations are tested to demonstrate the applicability and importance of the physics-based process noise matrices for different altitudes and eccentricities within the LEO shell.

Figures 4.11 to 4.13 depict simulation results for increasing semi-major axes. A comparison of the graphs shows that the magnitude of the force model uncertainties decreases rapidly with orbital altitude. While in the first case ($a = 6880.2$ km, $e = 0.000145$) transverse errors of more than 30 km are anticipated after 4.5 days of propagation, only 3.75 km (with a slight overestimation of the transverse position uncertainty after ~60 h of propagation time) are expected for the third case ($a = 7100$ km, $e = 0.004$). Despite the circumstance that the error propagation theories assume circular orbits, also the covariances of the second scenario ($a = 7000$ km, $e = 0.01$) are correctly estimated.

Changes with respect to solar flux scenario:

t_0: *2015/Feb/18-09:00:00*

Additional error process model parameters:

Geomagnetic index errors: Double-sided PSD of white noise process: 2700 AP units^{2}s
Sampling: 3 h

Average model uncertainty: OUP with $\theta = 10^{-6}$ *and* $s_{Model} = 10\%$

Extended WLS, density model: DTM-2013, gravity model: GGM02C, solar flux proxy error process model: random walk

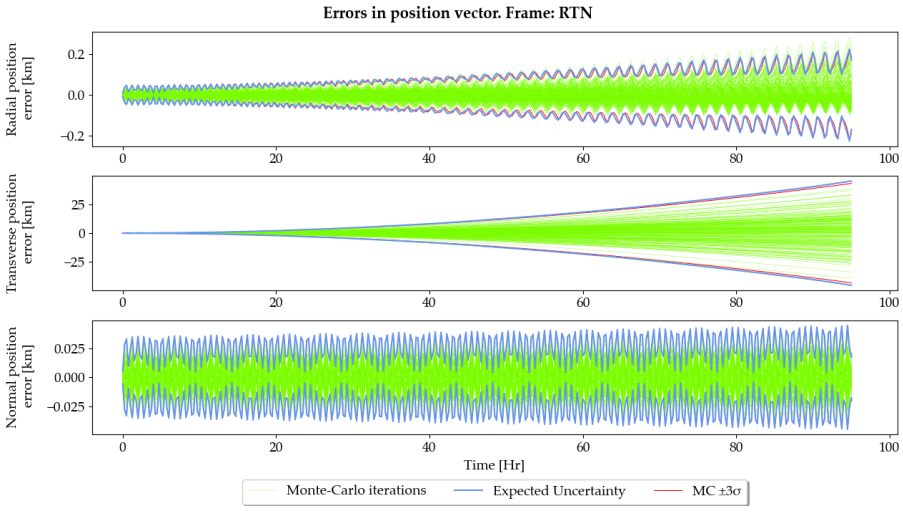


Figure 4.11: Extended WLS OD using the uncertainty quantification framework for DTM-2013 and GGM02C: Monte-Carlo position errors, 3σ bounds and estimated 3σ uncertainty

Extended WLS, density model: DTM-2012, gravity model: EGM96, solar flux proxy error process model: white noise

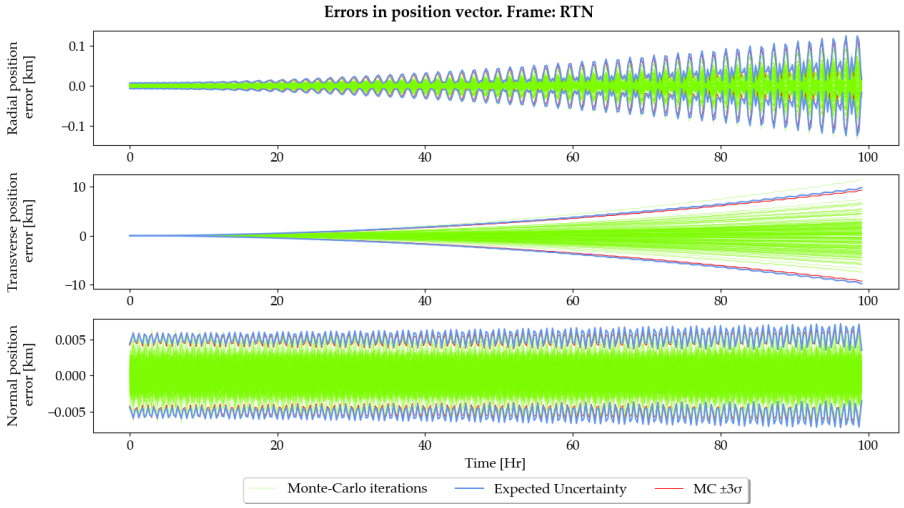


Figure 4.12: Extended WLS OD using the uncertainty quantification framework for DTM-2012 and EGM-96: Monte-Carlo position errors, 3σ bounds and estimated 3σ uncertainty

Changes with respect to solar flux scenario:
t_0*: 2014/May/02-00:00:00*

Additional error process model parameters:

Geomagnetic index errors: Double-sided PSD of white noise process:
$2700\,\mathrm{AP\,units^2 s}$
Sampling: 3 h

Average model uncertainty: OUP with $\theta = 10^{-6}$ *and* $s_{Model} = 10\%$

Object:

$$\mathbf{x}_0 : \begin{pmatrix} -397.388\,958\ \mathrm{km} \\ -1167.026\,235\ \mathrm{km} \\ 6854.493\,431\ \mathrm{km} \\ -3.778\,004\ \mathrm{km/s} \\ -6.456\,264\ \mathrm{km/s} \\ -1.251\,852\ \mathrm{km/s} \end{pmatrix}$$

Extended WLS, density model: NRLMSISE-00, gravity model: GGM05C, solar flux proxy error process model: white noise

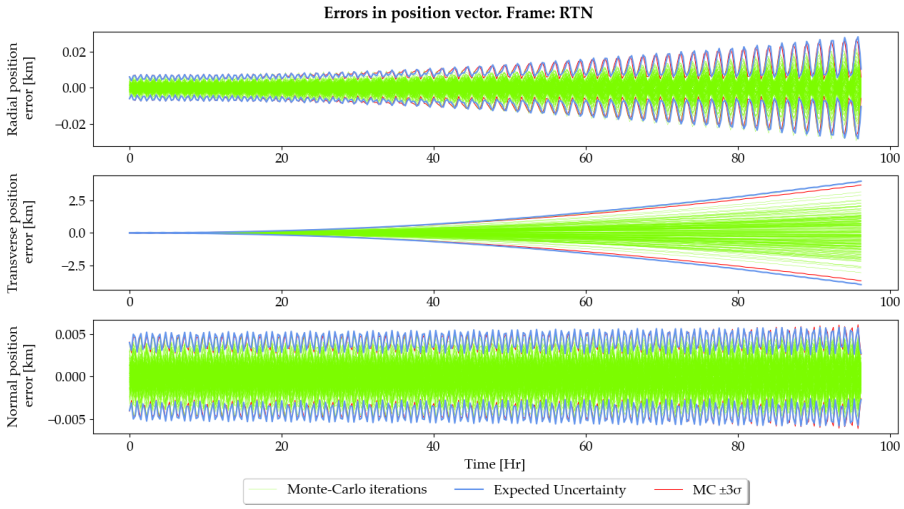


Figure 4.13: Extended WLS OD using the uncertainty quantification framework for NRLMSISE-00 and GGM05C: Monte-Carlo position errors, 3σ bounds and estimated 3σ uncertainty

Changes with respect to solar flux scenario:
t_0*: 2014/May/02-00:00:00*

Additional error process model parameters:

Geomagnetic index errors: Double-sided PSD of white noise process:
$21\,600\,\mathrm{AP\,units^2 s}$
Sampling: daily

Average model uncertainty: OUP with $\theta = 10^{-6}$ *and* $s_{Model} = 15\%$

Object:

$$\mathbf{x}_0 : \begin{pmatrix} -5600.663\,580\ \mathrm{km} \\ 277.964\,773\ \mathrm{km} \\ 4346.851\,744\ \mathrm{km} \\ -4.014\,563\ \mathrm{km/s} \\ -4.053\,795\ \mathrm{km/s} \\ -4.865\,098\ \mathrm{km/s} \end{pmatrix}$$

The radial position errors of figure 4.11 demonstrate a loss of the zero-mean Gaussian error property after ~2 days of propagation, indicating that for this type of low Earth orbit non-Gaussian estimation techniques may be beneficial. It has to be noted though that the radial position errors are two orders of magnitude smaller than the dominant transverse errors. Thus, no practical impact is expected in this case.

4.3 KALMAN FILTERS

In contrast to the least squares estimators, Kalman filters are not formulated as an optimization problem. Instead they make use of the Bayesian recursion relations, which build the heart of Bayesian dynamic state estimation. The working principle is depicted in figure 1.5, section 1.3.3. Whenever new observations are made available, a correction step is computed according to Bayes' theorem, which allows to compute the a posteriori (corrected) pdf from the a priori (predicted) pdf, the measurement model and the measurements. Since the integral in the prediction step usually cannot be solved in closed-form, analytical solutions to the time-evolution of the pdf are not available – with the exception of a linear system under the influence of Gaussian noise. In this case the Bayes estimator reduces to the well-known Kalman filter.

To apply the Kalman filter equations also for nonlinear problems, local linearization is applied at each step in the system and observation models to solve the inverse[2] uncertainty quantification problem. The measurements are processed sequentially to update the current estimate of the state, which then serves as the new baseline for the subsequent linearized propagation phase.

Unscented Kalman filters follow the same pattern, however they apply the technique of unscented transformation instead of local linearization via Jacobi matrices (cf. figure 2.21) to obtain an approximate solution of the Chapman-Kolmogorov prediction integral. Once a filter has reached convergence, it is a powerful recursive tool to solve the inverse uncertainty quantification problem within the boundaries of its assumptions (linear evolution of the state and Gaussian uncertainties).

Covariance growth is driven by radial position and force model uncertainties

In the absence of observations, the orbital state uncertainty increases as a consequence of the initial radial position uncertainty component[3] and additional uncertainties in the models used for propagation. Their interplay determines the shape and growth rate of the uncertainty volume. Measurement updates on the other hand reduce the state uncertainty, leading to characteristic "trumpet-like" uncertainty contours when depicting the estimated parameters versus time (cf. figure 3.19).

A downside of the Kalman filters is their dependence on an initial pdf estimate (covariance matrix in case of zero-mean Gaussian uncertainties), which crucially determines the evolution of the state and covariance during the filter initialization phase, the time required until steady-state and the tendency for divergence.

2 The *forward* direction is the relationship between the motion in state space and the corresponding observations. The *inverse* problem is hence to determine the (unknown) state parameters from noisy observations. Orbit determination is therefore an example of inverse uncertainty quantification

3 The orbital speed decreases with increasing altitude (distance from the center of the Earth). The radial position uncertainty at the beginning of a propagation phase is therefore associated with an in-track velocity uncertainty, which in turn drives the natural growth rate of the in-track position uncertainty

4.3.1 *Practical considerations*

The predictor-corrector working principle also has an influence on the incorporation of the force model uncertainty matrices. While covariance components that are formed incrementally between two arbitrary points in time (e.g. gravity uncertainty) do not require separate consideration, some additional effort is needed to integrate the density uncertainty framework with the sequential working principle. The necessary mathematical tools have been outlined in section 2.7.3.

The density uncertainty framework is of analytical nature and describes the development of the state uncertainties since the commencement of a density error process. Propagation steps of fixed time increments are realized differently for extended and unscented Kalman filters. In case of the EKF covariance propagation is achieved by making use of equation 2.300, which first removes the entire density uncertainty process noise accumulated up to time t_k from the a posteriori covariance $\hat{\mathbf{P}}_k$ and then propagates this difference linearly to t_{k+1} via the same STM which is also used for the classical covariance propagation. Finally, a new density error process noise matrix is added which describes the entire accumulated uncertainty since the start of the density error process. In case of the UKF no prior process noise subtraction is needed when using cloud-saving (cf. section 2.7.3). These techniques are able to incorporate the density uncertainty framework into the propagation phase of the filter. However, a look at figure 1.5 raises the question of how to proceed after the state estimate has been updated with new measurements.

The basic idea is to assume that measurement updates lead to a reduction in uncertainty sufficiently large to consider each measurement update as the starting time of a new "virtual" density error process. Due to the accuracy of modern sensors, a few observations are sufficient to meet this requirement, which is hence fulfilled for all sensor tracks encountered in practice. The overall principle is depicted in figure 4.14.

Special consideration is required for the random walk solar flux error process model, which represents the proxy error characteristics in case of orbit forecasting (Emmert et al., 2017). While in practice no measurement updates are expected when propagating into the future, there is nevertheless a theoretical interest in treating this case as well. The main difficulty with the random walk solar flux error process is a progressing mismatch of the solar flux error variability between the virtual density error processes and the actual random walk input error process. Following a measurement update that takes place after the first sampling step of the flux error process (e.g. after two days of orbit forecasting in case of daily F10.7 sampling) a new virtual density error process would assume a lower solar flux error variability than representative for the actual input error process at that time.

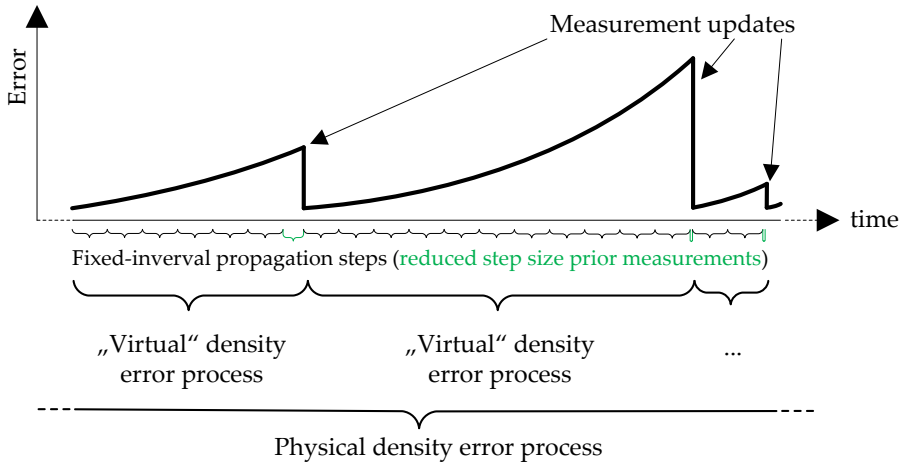


Figure 4.14: Density covariance modeling following a measurement update

In order to enable a uniform consideration of all density uncertainty sources between measurement updates, a workaround is proposed in the form that σ_{ρ_p} due to solar flux uncertainty is adjusted according to the time elapsed since the beginning of the physical density error process. This is not a mathematically rigorous approach, but an engineering decision rooted in the desire to avoid additional covariance propagations while maintaining a uniform treatment of the various density error sources – at the cost of a slight growth rate mismodeling for the solar flux uncertainty component. The impact of this approach is demonstrated in section 4.3.2 by means of numerical simulations for the scenarios of pure random walk solar flux errors, as well as for the combination of all considered density uncertainty drivers. Especially in the latter case the blending of the solar flux uncertainty with the other density uncertainty drivers masks the effect of the random walk error treatment, resulting in no noticeable impairment of the combined covariance estimates.

Also the Kalman filters benefit from measurement updates in equinoctial orbital elements. In contrast to the least squares estimators, this has no positive impact on the algorithm runtime, however it greatly extends the validity of the Gaussian assumption and therefore the quality of the estimates (Junkins et al., 1996; Sabol et al., 2010; Woodburn and Coppola, 2013; Alfriend and Park, 2016). Furthermore, the stability of the filter initialization phase in case of poor initial covariance estimates is improved, as will be demonstrated later.

The emulation of a Kalman filter operating natively in equinoctial elements works in a similar manner for the EKF and UKF by applying the following changes to the filters:

1. Convert the a priori state to equinoctial elements
2. UKF only: convert the propagated sigma points to EOE
3. UKF only: compute the matrix $\mathbf{P}_{xz}$ using the difference of the a priori state and the propagated sigma points in EOE

4. UKF only: obtain the Kalman gain for the measurement updates in equinoctial elements using $\mathbf{P}_{xz}$ from step 3
5. Define $\Delta\boldsymbol{x}_k$ as the correction between the a priori and the a posteriori state (EKF: Cartesian, UKF: EOE via the equinoctial Kalman Gain from step 4). This equals the Kalman gain matrix times the measurement residual vector
6. Define $\Delta\mathbf{P}_k$ as the correction between the a priori and the a posteriori covariance (EKF: Cartesian, UKF: EOE via the equinoctial Kalman Gain)
7. EKF only: obtain the Jacobian for the Cartesian to EOE conversion using the a priori state and algorithm B.6
8. EKF only: convert $\Delta\boldsymbol{x}_k$ and $\Delta\mathbf{P}_k$ to EOE via the a priori Jacobian
9. Perform the state and covariance updates in equinoctial elements
10. Convert the updated state back to Cartesian coordinates
11. EKF only: obtain the a posteriori Jacobian from EOE to Cartesian coordinates and use it to convert the state error covariance back to Cartesian coordinates
12. UKF only: use unscented transformation to convert the updated covariance back to Cartesian coordinates. Alternatively perform the back-conversion via step 11 if the use of a Jacobian in an otherwise linearization-free estimator is considered acceptable

In view of the superior estimation performance and the significantly prolonged validity of the Gaussian error assumption, the above computations represent only a minor additional effort. Further examples of the benefits of equinoctial measurement updates may be found in Woodburn and Coppola (2013).

4.3.2 *Simulation results*

In the following, orbit determination results obtained by applying the force model uncertainty framework in conjunction with the extended/unscented Kalman filter are presented. First, the effect of the random walk solar flux treatment outlined in section 4.3.1 is illustrated and its impact on the case where all modeled density error sources are considered is assessed. Subsequently, the altitude dependency of the force model uncertainties is analyzed in order to deduce conclusions about their importance and the applicability of a classical EKF.

Following the EKF results, the performance of the linearization-free UKF is presented. Cloud-saving and gravity uncertainty accumulation are used to account for the total force model uncertainty since the last measurement update. The influence of the ballistic coefficient on the magnitude of the density errors is highlighted. Finally, the complete force model uncertainty quantification framework (solar flux/magnetic index/average density model errors, as well as gravity errors of omission and commission) is applied and the importance of the initial covariance estimate is addressed.

Scenario parameters:

MC iterations: 200
t_0: 2011/Oct/01-20:00:00
Duration: 5.5 days
Propagation step: 10 min

Perturbations:
30×30 *aspherical potential field (GGM02C)*
Atmospheric drag (DTM-2012)
Solar radiation pressure
Third body gravity (soli-lunar)

Solar flux errors:
Random walk
Double-sided PSD of underlying white noise process: 130 000 $sfu^2 s$
Sampling: daily

Observer:
Type: Optical
Latitude: 49.7813 deg
Longitude: 9.973 94 deg
RaDec accuracy: 10^{-3} deg

Object:

$$x_0 : \begin{pmatrix} -307.504\,742\ \text{km} \\ -3068.587\,070\ \text{km} \\ -6158.385\,177\ \text{km} \\ 7.565\,267\ \text{km/s} \\ -0.797\,834\ \text{km/s} \\ 0.020\,935\ \text{km/s} \end{pmatrix}$$

Frame: GCRF
Semi-major axis: 6887.0 km
BC: 0.06 m^2/kg

Scenario 1: EKF, solar flux random walk density errors only

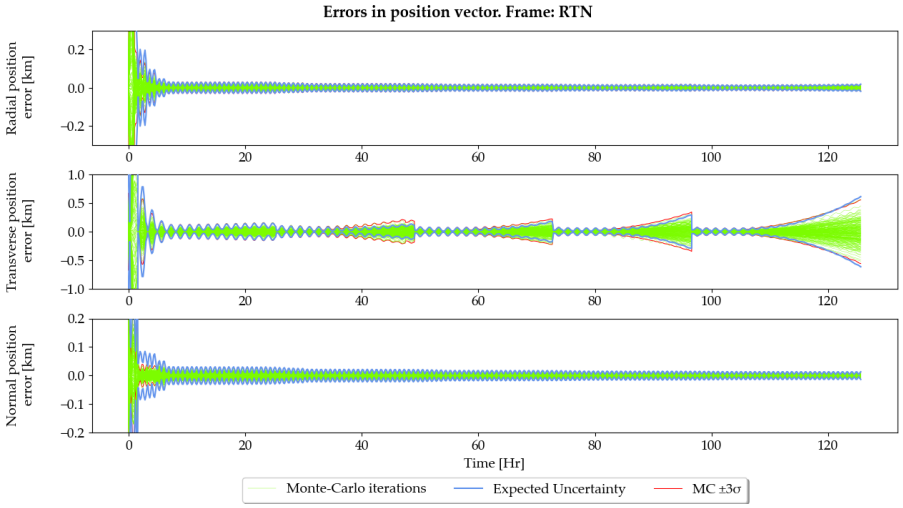


Figure 4.15: Expected and Monte-Carlo orbital uncertainty due to solar flux random walk errors

Figure 4.15 depicts the OD performance obtained using the orbital uncertainty quantification model for solar flux random walk errors as EKF process noise matrix. In contrast to previous cases, the simulation time has been extended by one day to evaluate the measurement update treatment in case of solar flux random walk errors for extended durations of the actual input error process.

As reported in section 4.3.1, the plot depicts slight uncertainty growth errors in the transverse direction for the propagation phases after 24 h of simulation time. Nevertheless, the essence of the transverse uncertainty growth during the propagation phases is captured and also the final and strongest transverse position error increase is correctly estimated. The radial and normal position errors do not show any deficiencies in the uncertainty modeling. In fact, the normal errors do not exhibit any growth at all, since density errors predominantly lead to in-plane kinematic state vector errors. The growth rate of the radial position error is almost insignificant, because the measurement updates are sufficiently frequent to keep the position errors in this direction small.

The extended scenario, in which also the uncertainty resulting from geomagnetic index input errors and grid-scale density model errors is considered in addition to the random walk solar flux input errors when computing the filter process noise, is depicted in figure 4.16. The blending of the individual density uncertainty drivers masks the random walk solar flux error treatment without any visual impact on the final covariance estimates, as already mentioned in section 4.3.1. The estimated three sigma standard deviations match the Monte-Carlo generated 3σ bounds in all axes throughout the entire simulation after the filter completed its initialization phase.

Scenario 2: EKF, all density error sources (solar flux input errors modeled as random walk)

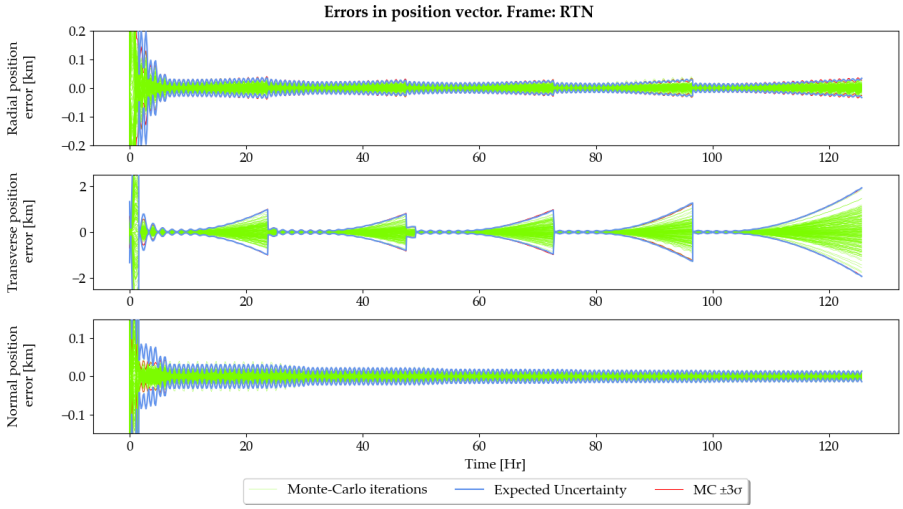


Figure 4.16: EKF OD using the complete density uncertainty quantification framework: Monte-Carlo position errors, 3σ bounds and estimated 3σ uncertainty

In the following, the semi-major axis is gradually increased in order to evaluate the impact of the density errors on the orbit. Since neutral thermospheric density is decreasing approximately exponentially with altitude, a rapid decline of the transverse position error bounds is expected (cf. figures 4.11-4.13). The question to be answered is therefore as of which altitude a classical EKF or UKF yields the same uncertainty consistency as its density uncertainty considering counterpart.

Scenario 3: EKF, all density error sources, semi-major axis: 7000 km

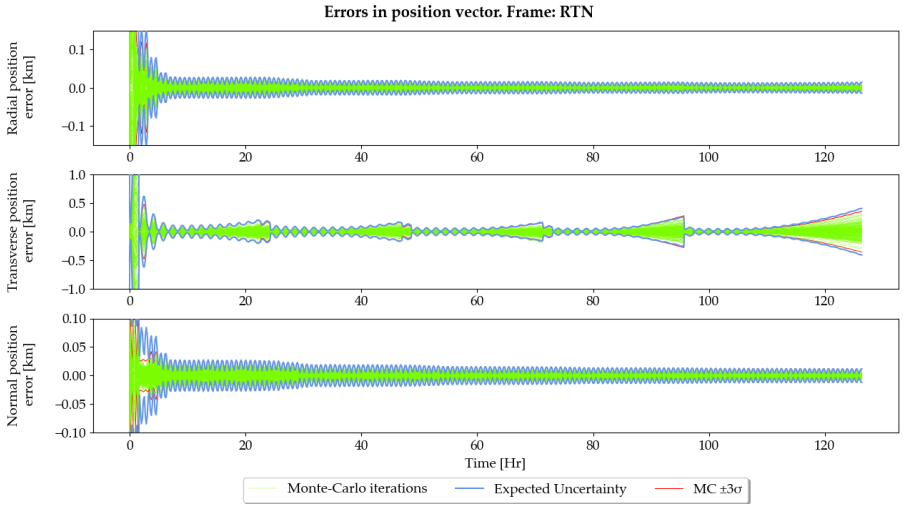


Figure 4.17: Expected and Monte-Carlo orbital uncertainty due solar flux, magnetic index and grid-scale model errors at $a = 7000$ km

Scenario 4: classical EKF, process noise neglected, all density error sources, semi-major axis: 7000 km

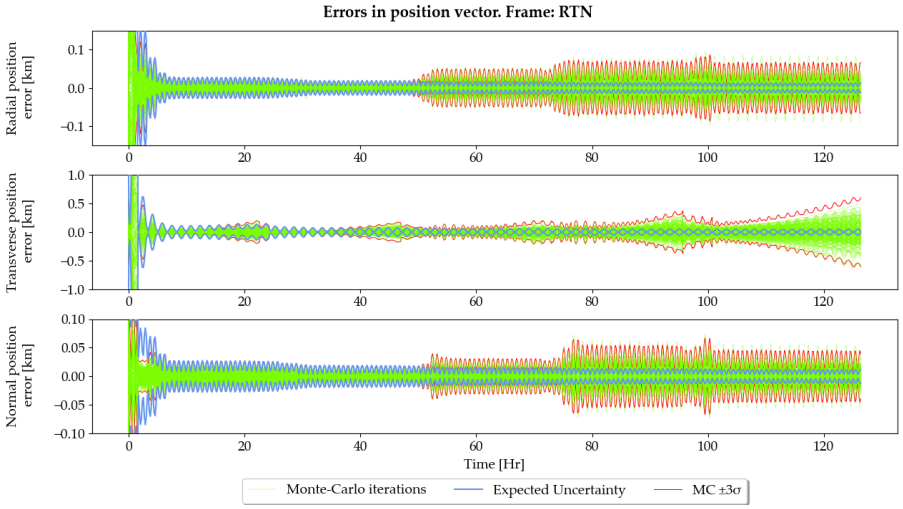


Figure 4.18: Expected and Monte-Carlo orbital uncertainty due solar flux, magnetic index and grid-scale model errors at $a = 7000$ km using a classical EKF

Scenario 5: EKF, all density error sources, semi-major axis: 7100 km

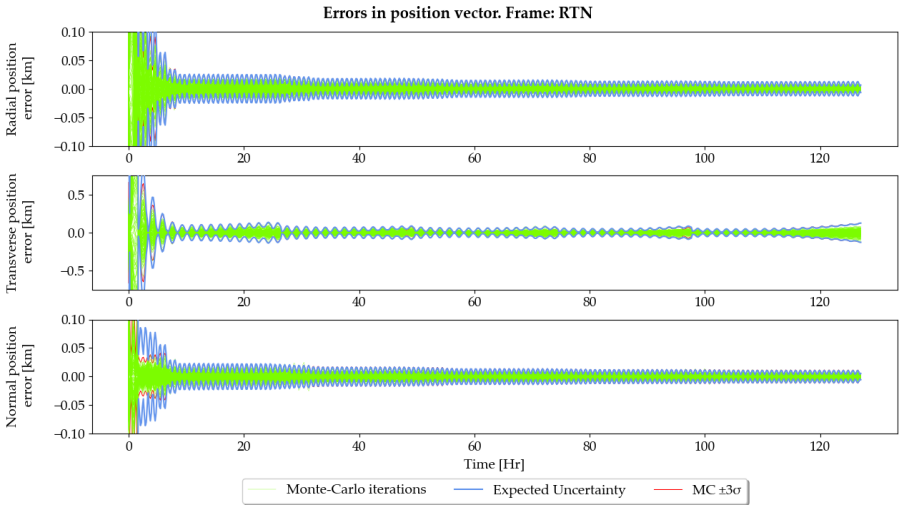


Figure 4.19: Expected and Monte-Carlo orbital uncertainty due solar flux, magnetic index and grid-scale model errors at $a = 7100$ km

Scenario 6: classical EKF process noise neglected, all density error sources, semi-major axis: 7100 km

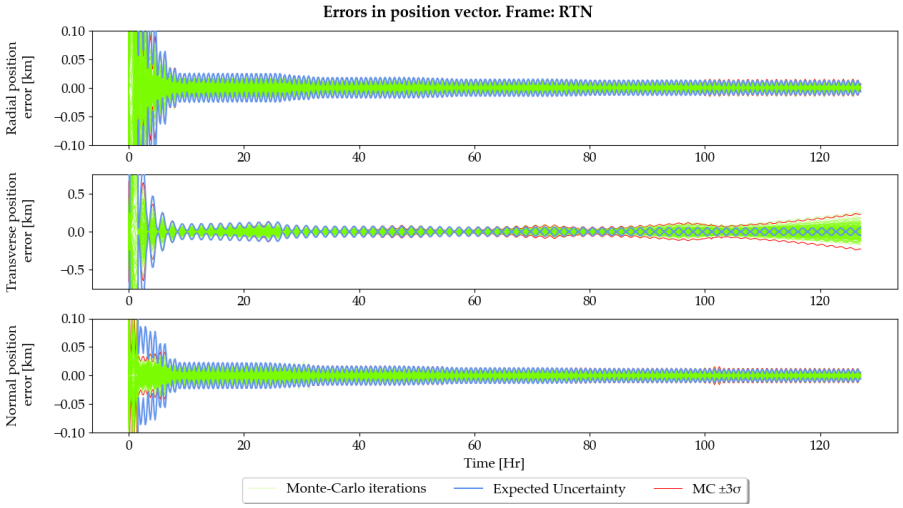
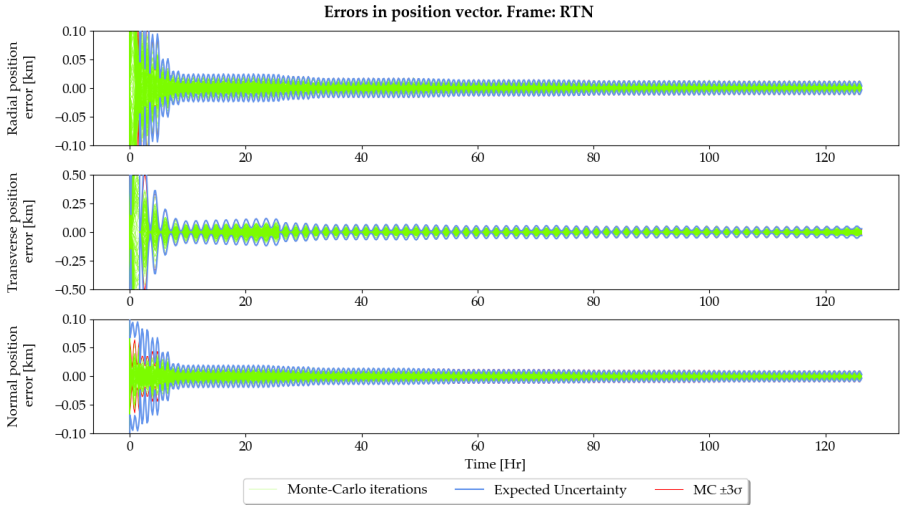
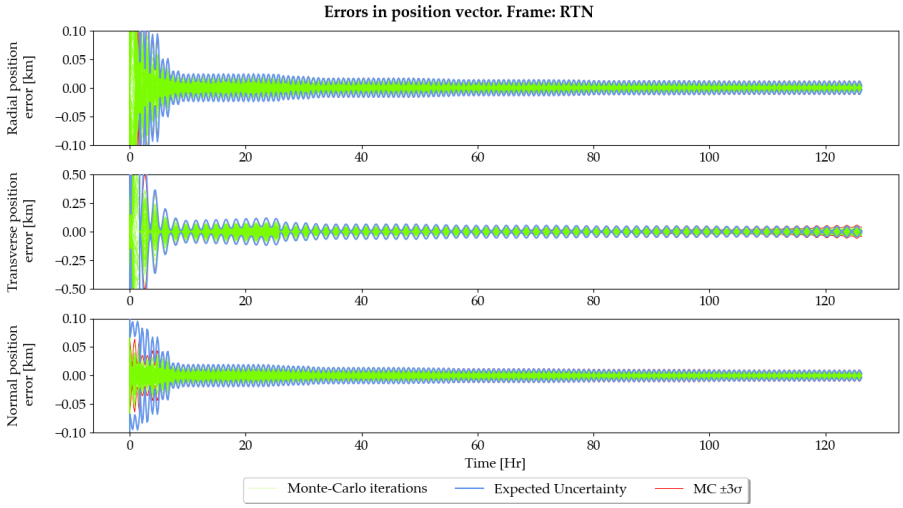


Figure 4.20: Expected and Monte-Carlo orbital uncertainty due solar flux, magnetic index and grid-scale model errors at $a = 7100$ km using a classical EKF



Scenario 7: EKF, all density error sources, semi-major axis: 7300 km

Figure 4.21: Expected and Monte-Carlo orbital uncertainty due solar flux, magnetic index and grid-scale model errors at $a = 7300\,\mathrm{km}$



Scenario 8: classical EKF, process noise neglected, all density error sources, semi-major axis: 7300 km

Figure 4.22: Expected and Monte-Carlo orbital uncertainty due solar flux, magnetic index and grid-scale model errors at $a = 7300\,\mathrm{km}$ using a classical EKF

Figures 4.17 to 4.22 confirm the expectation of position error covariances that swiftly decline with orbital altitude due to the exponentially decreasing neutral atmospheric density and the associated reduction in density errors. However, the observed position errors are much smaller than in the least squares case, as the frequent measurement updates are able to keep the duration of the virtual density error processes short.

While for $BC = 0.06\,\frac{m^2}{kg}$ and a semi-major axis of $a = 7000\,\mathrm{km}$ the classical EKF breaks down after 50 h of simulation time and fails to estimate correct position error covariances starting from the first propagation phase, the situation is much different for $a = 7100\,\mathrm{km}$. While a slight transverse propagation error covariance underestimation can be identified in figure 4.20 after 50 h of simulation time, the first noticeable underestimation takes place at $t = 70\,\mathrm{h}$. Filter smugness starts to develop at $t = 85\,\mathrm{h}$. If the semi-major axis is increased by

another 200 km, a classical EKF shows equal performance to its density uncertainty considering counterpart until 105 h of simulation time.

Without physics-based process noise matrices EKFs are unable to account for the covariance growth due to density errors. Hence, more frequent measurement updates are required to keep the orbit errors small and avoid filter smugness. For the current example, density errors can be judged to have an impact on operational orbit covariance consistency up to an altitude of approximately 800-850 km. However, as the boundary depends on the current thermospheric conditions and the ballistic coefficient of the satellite, it is impossible to specify a universal range.

So far only the density covariances of the force model uncertainty framework were applied, however also gravity errors should be taken into account. This is demonstrated in figure 4.23, where only density errors are considered, however also gravity errors of omission are simulated by using a 85×85 GGM02C potential field for the Monte-Carlo iterations and measurement generation, whereas only a 30×30 GGM02C field is used for the EKF-based orbit determination. The Monte-Carlo 3σ bounds have been removed in figure 4.23, because the aleatoric treatment of epistemic gravitational errors quantifies the expected uncertainty, but prevents its full characterization. The stochastic 3σ limits should thus be understood as error bounds.

Scenario parameters:
MC iterations: 200
t_0: 2015/02/18-09:00:00
Duration: 5.0 days
Perturbations:
Reference: 85×85 *GGM02C potential field*
OD: 30×30 *GGM02C*
Drag: DTM-2012
SRP, Third body gravity
Density errors:
Sol. flux: white noise,
PSD: 130 000 $\mathrm{sfu}^2\mathrm{s}$
Geomag. ind.: white noise,
PSD: 21 600 $\mathrm{AP\,units}^2\mathrm{s}$
Model OUP:
$\theta = 10^{-6}$, $s_{Model} = 10\%$
Object:
$$x_0 : \begin{pmatrix} 1624.727\,042\,\mathrm{km} \\ -30.426\,939\,\mathrm{km} \\ -6693.537\,803\,\mathrm{km} \\ 0.662\,754\,\mathrm{km/s} \\ 7.576\,662\,\mathrm{km/s} \\ 0.126\,079\,\mathrm{km/s} \end{pmatrix}$$
Frame: GCRF
Semi-major axis: 6887.0 km
BC: 0.006 m^2/kg

Scenario 9: EKF, density and gravity errors simulated, only density uncertainty consideration, $\mathrm{BC} = 0.006\ \frac{\mathrm{m}^2}{\mathrm{kg}}$

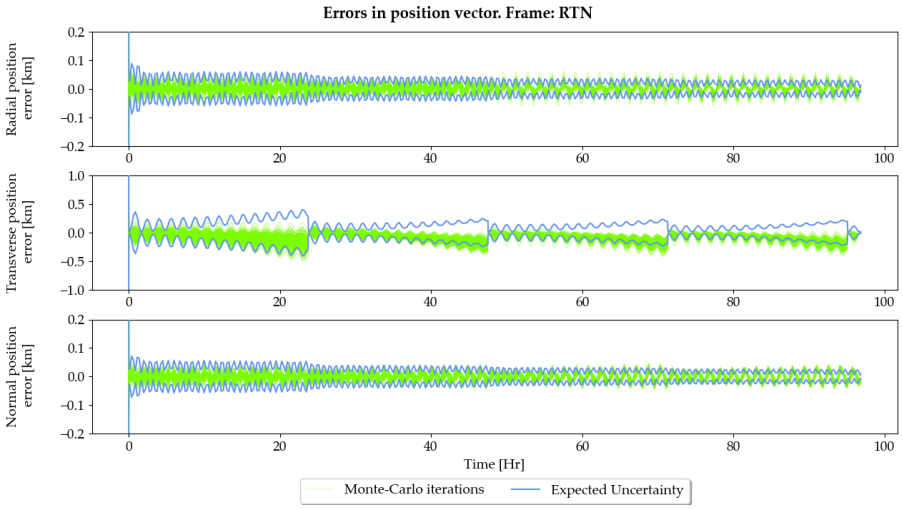


Figure 4.23: Monte-Carlo position errors and expected 3σ uncertainty. Density and gravity errors simulated, only density errors considered

The impact of gravity errors can be clearly identified in figure 4.23. The radial and normal position errors depict an oscillatory motion, while the dominant transverse position errors exhibit a bias. All axes exceed the expected error limits. Figure 4.24 illustrates the results when gravity errors are considered additionally to density errors using

algorithm 3.1 or 3.2. The radial and transverse errors remain within the error bounds. Also the normal error bounds improve, however a slight underestimation can be observed at $t = 70\,h$ and $90\,h$.

Scenario 10: EKF, density and gravity errors, $a = 6887\,km$, $BC = 0.006\,\frac{m^2}{kg}$

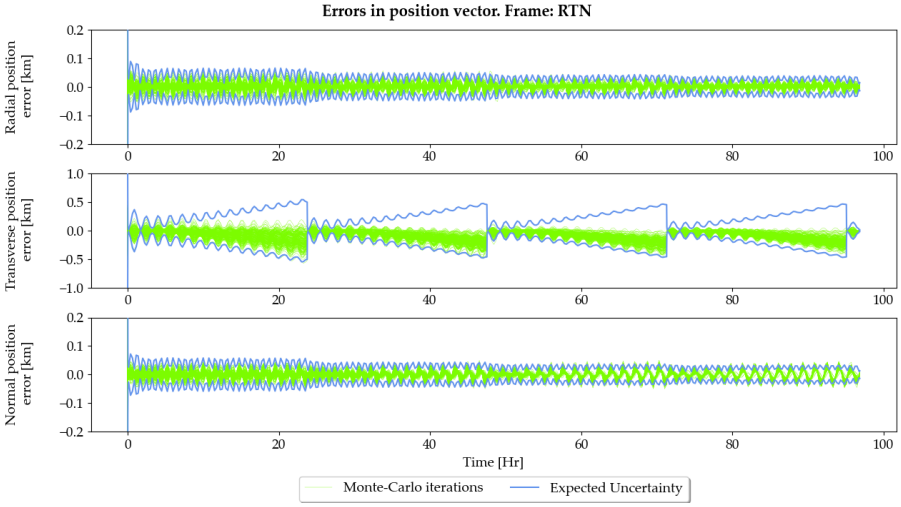


Figure 4.24: Monte-Carlo position errors and expected 3σ uncertainty. Density and gravity errors simulated and considered for the process noise generation

The consideration of both error sources is especially important below ~600 km. At higher altitudes or in case of larger ballistic coefficients it may be possible that the density uncertainty compensates the dominating transverse position errors of gravity modeling origin, however also in this case it is recommended to include the gravity error covariance in the filter process noise matrix, since the radial and normal components are not captured by the density error covariance (cf. figures 4.25 and 4.26).

Scenario 11: EKF, density and gravity errors simulated, only density uncertainty consideration, semi-major axis: $6887\,km$, $BC = 0.06\,\frac{m^2}{kg}$

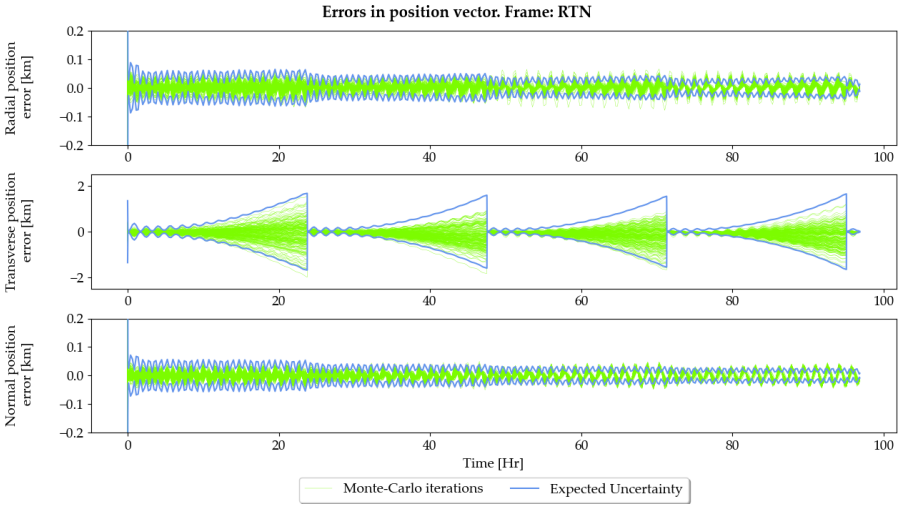


Figure 4.25: Monte-Carlo position errors and expected 3σ uncertainty. Density and gravity errors simulated, only density errors considered, increased BC

Scenario 12: EKF, density and gravity errors simulated, only density uncertainty consideration, semi-major axis: 7000 km, BC = 0.06 $\frac{m^2}{kg}$

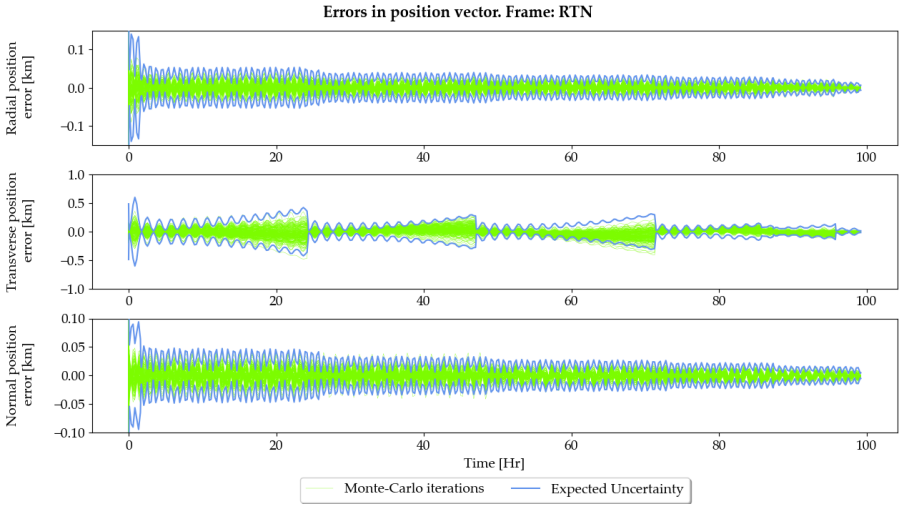


Figure 4.26: Monte-Carlo position errors and expected 3σ uncertainty. Density and gravity errors simulated, only density errors considered, increased BC

Figure 4.27 shows the result of the same scenario as depicted in fig. 4.25, however both density and gravity errors are considered in the filter process noise matrix. The improvement in the radial and normal components is evident.

Scenario 13: EKF, density and gravity errors, a = 6887 km, BC = 0.06 $\frac{m^2}{kg}$

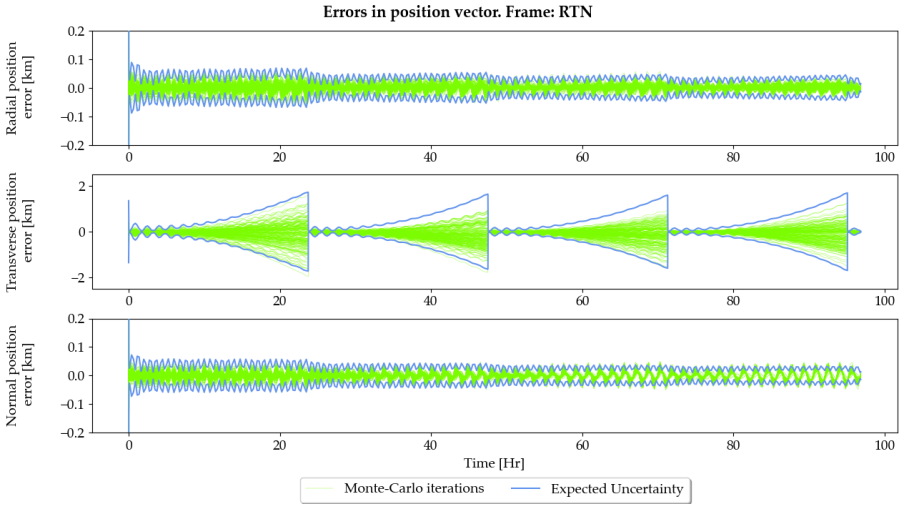


Figure 4.27: Monte-Carlo position errors and expected 3σ uncertainty, density and gravity errors considered, increased BC

Comparing figures 4.24 and 4.27 allows to infer the impact of the ballistic coefficient on the density uncertainty. Both scenarios are equal except for a ten times smaller ballistic coefficient in figure 4.24. At

each end of the four extended propagation phases, the 3σ uncertainty bounds differ by a factor of three to four. Also the slope of the expected uncertainty is significantly larger in figure 4.27, demonstrating the need for physics-based process noise matrices that are able to take the ballistic coefficient into account via its integrated effect on the fraction $\frac{\Delta\hat{n}}{\Delta t}$ which is central to the density covariance estimates.

The performance of a classical process noise neglecting EKF for $BC = 0.006\ \frac{m^2}{kg}$ and $BC = 0.06\ \frac{m^2}{kg}$ is contrasted in figures 4.28 and 4.29. It does not take a lot of words to assess the fundamental performance difference between figures 4.28 and 4.24, as well as 4.29 and 4.27.

Scenario 14: Classical EKF, density and gravity errors, $BC = 0.006\ \frac{m^2}{kg}$

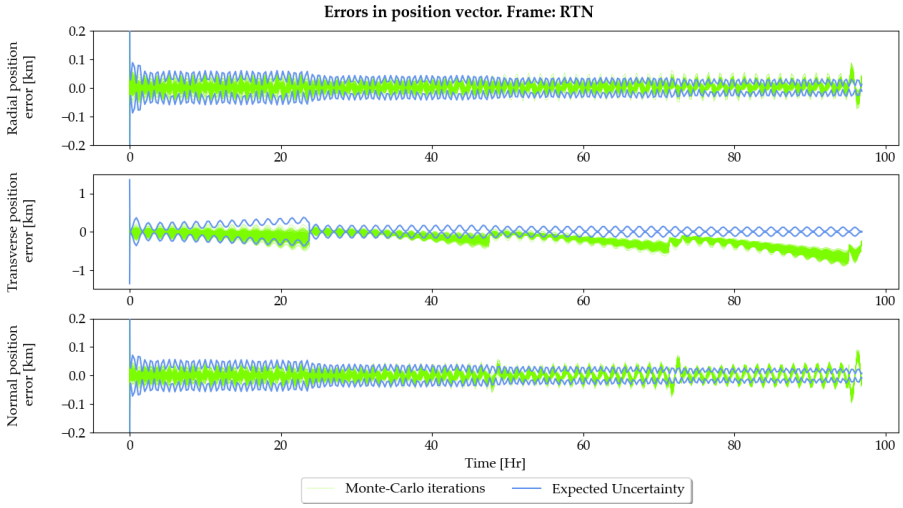


Figure 4.28: Monte-Carlo position errors and expected 3σ uncertainty. Density and gravity errors simulated but not considered

Scenario 15: Classical EKF, density and gravity errors, $BC = 0.06\ \frac{m^2}{kg}$

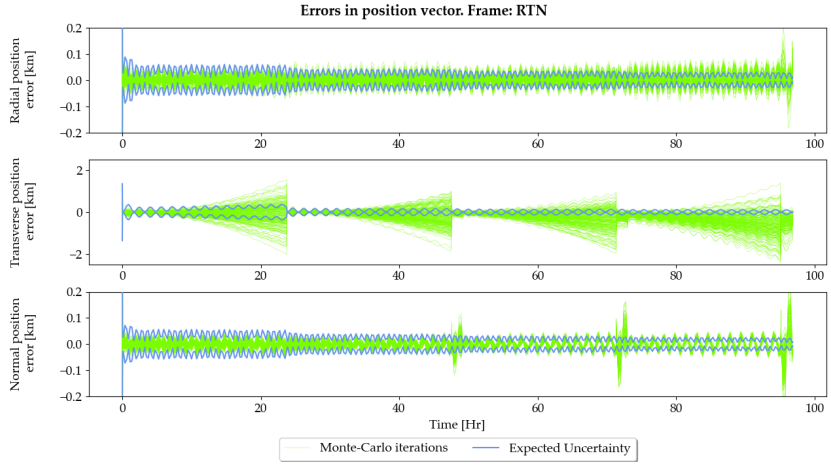


Figure 4.29: Monte-Carlo position errors and expected 3σ uncertainty. Density and gravity errors simulated but not considered

Finally, also the compatibility with the unscented Kalman filter shall be demonstrated. The impact of the initial process noise estimate, as well as the choice of the coordinate frame on the filter performance is analyzed as an example of practical relevance.

Scenario 16: UKF, density and gravity errors, poor choice of $\hat{\mathbf{P}}_0$. Coordinate frame: Cartesian

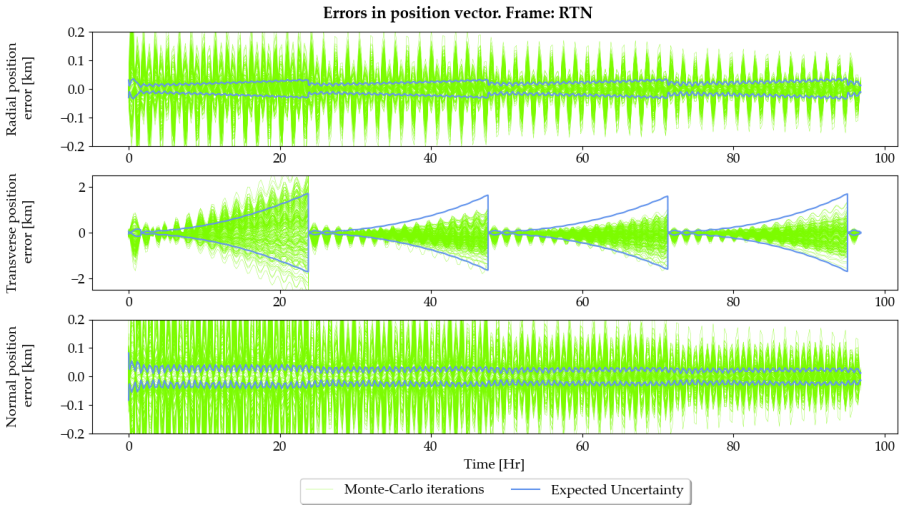


Figure 4.30: UKF OD with highly optimistic $\hat{\mathbf{P}}_0$ in GCRF coordinates using the complete force model uncertainty framework

Scenario 17: UKF, density and gravity errors, poor choice of $\hat{\mathbf{P}}_0$, Coordinate frame: EOE

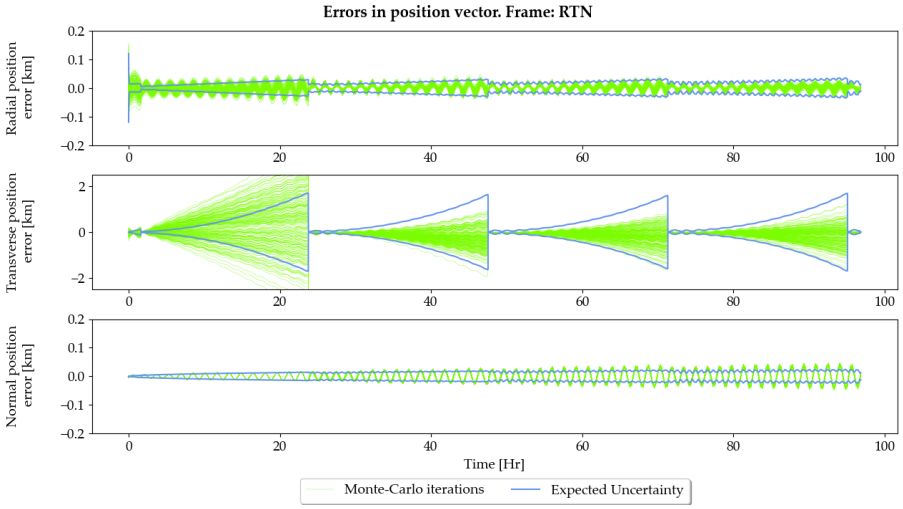


Figure 4.31: UKF OD with highly optimistic $\hat{\mathbf{P}}_0$ in EOE using the complete force model uncertainty framework

Figures 4.30 and 4.31 represent the same scenario as figure 4.27, except that the initial covariance estimate was chosen overly optimistic (and an UKF is used for the OD). The highly optimistic choice of $\hat{\mathbf{P}}_0$ has a tremendous impact on the OD estimates in figure 4.30. While

the dominant transverse position errors recover to a large extent after 30 h, they still regularly violate the error bounds after measurement updates. Also the radial and normal position errors are never within their expected limits throughout the simulation.

Using equinoctial orbit elements for the OD drastically improves the filter performance. The dominant transverse errors remain within their estimated bounds after the first extended propagation phase. The radial errors are correctly estimated after the second propagation phase. Only the normal errors remain out of limits, however they significantly improved in comparison to figure 4.30. Even though the initial covariance estimates are not expected to be of such bad quality in practice, performing the OD or at least the measurement updates in EOE is highly recommended, since orbital elements also maximize the validity of the Gaussian error assumption during propagation. The step by step conversion procedure on page 212 may be used for this purpose.

4.4 APPROXIMATE UNCERTAINTY QUANTIFICATION USING GP PROPAGATION THEORY

What is nowadays collectively known as the analytical GP propagation theory has been released 40 years ago by the United States Department of Defense (DOD) in SpaceTrack Report number 3 as Simplified General Perturbations Model 4 (SGP4) and Simplified Deep Space Perturbations Model 4 (SDP4) (Hoots and Roehrich, 1980). The report also contains further analytical propagation models, but the combination of SGP4 (orbital period less than 225 min) and SDP4 (orbital period greater than 225 min) became the centerpiece of the document, as they are the ones used most frequently to propagate the Two-Line Element (TLE) sets issued by NORAD.

For compatibility with the publicly available TLE catalog, it is important to use an implementation of the theory which is identical to the DOD version, or at least highly compatible with it. Also a seamless transition between the SGP4 and SDP4 models is desirable. Unfortunately, up to date source code is not available from DOD, such that an alternative, yet central source is required. The work of Vallado et al. (2006) is nowadays a widely accepted source which combines numerous corrections to the original implementation of SpaceTrack Report #3 in an up to date software package designed for maximum compatibility with the code presumably used by DOD today.

The GP theory has been developed without a counterpart for covariance propagation. Also the TLE file format contains no fields to indicate uncertainty information. A substantial amount of work has been performed to derive uncertainty estimates for the TLE catalog using various methods. The strategy used at the ESA's Space Debris Office (SDO) is outlined in Flohrer et al. (2008) and uses differential correction to estimate an epoch uncertainty. The main issue however

is to propagate the uncertainty, since no analytic counterpart for covariance propagation exists. It therefore became common practice to propagate the estimated epoch TLE covariances using SP methods, thereby accepting that different force models are used to perform the numerical propagation. It is however a known fact that different models lead to inconsistencies (Kaya et al., 2004; Escobar et al., 2011). The lack of an appropriate GP covariance propagation technique is therefore an unfortunate circumstance.

A presumably elegant solution lies in the usage of modern sampling-based sigma point algorithms (e.g. unscented transformation), since they use the state propagation model for uncertainty propagation. Despite its elegance, this approach does not solve the covariance propagation issue because the generation of sigma points is not possible. The orbital elements in TLE files represent a form of mean elements and must not be confused with the momentary osculating orbital elements (Vallado, 2013, section 9.9.1). However, initial covariance matrices (as e.g. obtained by the ESA Space Debris Office (SDO) method) cannot be reliably converted into a covariance for these mean elements (a discussion about the difficulties of the osculating to mean conversion may be found in section 9.9.1 of Vallado, 2013).

The TLE file format poses an additional obstacle for the sampling based algorithms. The GP state propagation codes internally use so-called "satellite records" which are generated from the two TLE character lines for propagation. The TLE file format limits the number of decimal places available for the orbital elements, such that even if there would be a suitable mean covariance for the sigma point generation, the corresponding sigma point mean elements would have to be truncated, thereby altering their statistical properties. Small changes due to small covariances could even get lost as a consequence of insufficient decimal places. Satellite records on the other hand contain over 100 data fields, of which most are required for propagation. Many of these would need to be manually updated for each sigma point to maintain all significant digits.

A third requirement for practical applications lies in the need to propagate covariances that correspond to a time different than the TLE epoch. A GP covariance propagation theory thus has to be capable of propagating $\mathbf{P}_i$ to $\mathbf{P}_j$, given a TLE with epoch t_0, where t_0, t_i and t_j may be in any chronological order.

GP COVARIANCE PROPAGATION

The solution to GP covariance propagation is based on the numerical approximation of the STM required for linear covariance propagation. The process has been briefly outlined in Escobar et al. (2011) and is supplemented with the necessary details for implementation in the following. The STM from time t_i to time t_j is defined by equation 4.60.

$$\boldsymbol{\Phi}(t_j, t_i) = \frac{\partial \mathbf{x}_j}{\partial \mathbf{x}_i} \tag{4.60}$$

Given an initial covariance matrix $\mathbf{P}_i$, the linear covariance propagation to time t_j is achieved according to equation 4.61:

$$\mathbf{P}_j = \boldsymbol{\Phi}(t_j, t_i)\mathbf{P}_i\ \boldsymbol{\Phi}(t_j, t_i)^T = \left(\frac{\partial \mathbf{x}_j}{\partial \mathbf{x}_i}\right) \mathbf{P}_i \left(\frac{\partial \mathbf{x}_j}{\partial \mathbf{x}_i}\right)^T \tag{4.61}$$

The GP theory is developed in the TEME pseudo-frame. The final step is a conversion to inertial Cartesian coordinates. Each propagation is always started from the two line character arrays or the corresponding set of satellite record vectors and returns a propagated Cartesian inertial state vector.

The propagation process from mean TLE orbital elements to propagated Cartesian kinematic states is a one-way street, meaning that it is not possible to use the result of a propagation as input for a subsequent propagation, as the process cannot be initialized using inertial Cartesian states or osculating orbital elements. However, since the GP theory is analytic, there is no impact on algorithm runtimes when using the TLE epoch data to initialize propagations.

The objective is to numerically compute $\partial \mathbf{x}_j / \partial \mathbf{x}_i$. Assuming that the GP propagation from mean orbital elements to inertial Cartesian coordinates can be expressed as some function $\mathrm{GP}(t_0, t_i, \mathbf{x}_{0,m})$, where t_0 is the TLE epoch time, t_i the final propagation time and $\mathbf{x}_{0,m}$ the epoch state vector constructed from the mean orbital elements of the second TLE line, it is possible to state:

$$\mathbf{x}_{i,\mathrm{Cart}} = \mathrm{GP}(t_0, t_i, \mathbf{x}_{0,m}) \tag{4.62}$$

The chain rule allows to split equation 4.60 as follows:

$$\frac{\partial \mathbf{x}_j}{\partial \mathbf{x}_i} = \frac{\partial \mathbf{x}_j}{\partial \mathbf{x}_{0,m}} \frac{\partial \mathbf{x}_{0,m}}{\partial \mathbf{x}_i} = \frac{\partial \mathbf{x}_j}{\partial \mathbf{x}_{0,m}} \left(\frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_{0,m}}\right)^{-1} \tag{4.63}$$

Equation 4.63 boils down to the numerical evaluation of $\partial \mathbf{x}_k / \partial \mathbf{x}_{0,m}$, where k is an arbitrary target time for the propagation. Using central differencing, two analytic GP propagations are required to obtain estimates of the Jacobian. Also higher order methods (e.g. the five point stencil method) are possible, however they require further propagations. Numerical experiments have not shown significant differences between both approaches, such that in practice the faster central differencing approach is preferable.

An important point for the success of this method lies in the choice of the deviations applied to the TLE mean elements. Four of the six mean orbital elements in $\mathbf{x}_{0,m}$ are angular quantities, such that the deviation added to the variables for the central differencing should be chosen independently of their actual value. The mean motion and the eccentricity on the other hand showed best results using relative differences.

Also this numerical approach to GP covariance propagation is constrained by the number of significant digits available through the TLE file format. In contrast to sampling-based approaches however the perturbations do not need to fulfill statistical properties and can therefore be chosen in accordance with the number of decimal places available. The pseudocode of the numerical Jacobi matrix derivation is given in algorithm 4.1. The algorithm needs to be invoked twice to compute the STM according to equation 4.63.

Algorithm 4.1: Computation of GP Jacobian via central differencing

```
function numJacGP(line1, line2, tEnd, numJac)
    ▷ State vector definition:
    ▷ 1: mean motion, 2: eccentricity, 3: inclination
    ▷ 4: RAAN, 5: argument of perigee, 6: mean anomaly
    ▷ Define the delta vector h
    h ← [1E-6, 1E-2, 1E-3, 1E-3, 1E-3, 1E-3]
    ▷ Get the mean elements from the second line of the TLE
    call line2ToxCoe(line2, xCoeMeanBck)
    for i ∈ [1, 6] do
        ▷ Restore the original TLE data
        line2tmp ← line2
        xCoeMean ← xCoeMeanBck
        ▷ Obtain the delta for the numerical differentiation
        if i ⩽ 2 then
            ▷ h(i) is relative for mean motion and eccentricity
            δ ← xCoeMean(i) · h(i)
        else
            δ ← h(i)
        end if
        ▷ Addition of the i^th component
        xCoeMean(i) ← xCoeMeanBck(i) + δ
        ▷ Update the second line of the TLE
        call xCoeToLine2(xCoeMean, line2tmp)
        ▷ Propagate
        call GP(line1, line2tmp, tEnd, xp)
        ▷ Same operations when subtracting delta
        xCoeMean(i) ← xCoeMeanBck(i) − δ
        call xCoeToLine2(xCoeMean, line2tmp)
        call GP(line1, line2tmp, tEnd, xm)
        ▷ Store the gradient as the i^th column of the Jacobian
        numJac(:,i) ← (xp − xm)/(2δ)
    end for
    return numJac
end function
```

Validation

Algorithm 4.1 has been validated via the following strategy:

- Randomly download a TLE from SpaceTrack
- Perform a zero time propagation to obtain the osculating epoch GCRF state from the TLE
- Define an initial GCRF covariance matrix (e.g. by definition in EOE with subsequent conversion to GCRF via algorithm B.5)
- Perform GP covariance propagation using algorithm 4.1
- Perform SP covariance propagation using the initial GCRF state and covariance
- Transform the SP covariance into the RTN frame
- Transform the GP covariance into the RTN frame of the SP propagation

Exemplary results for a LEO, MEO and GEO TLE are presented in figures 4.32-4.34. Each figure depicts the SP propagation in the top left subplot, the GP propagation in the top right subplot and the difference of the standard deviations in the lower subplot. The models used for the SP propagation are chosen to approximately match the GP theory: 4×4 EGM-96 potential field, MSISE-90 atmospheric drag and soli-lunar third body perturbations (using JPL ephemerides). The ballistic coefficient is extracted from the TLE[4] via the conversion factor $BC = \frac{2}{6378.135 \cdot 2.461 \cdot 10^{-5}} B^* \cong 0.078482951\, B^*$ (Vallado, 2013, p. 106). The input TLE, as well as the initial EOE epoch covariance are given below the title of each figure.

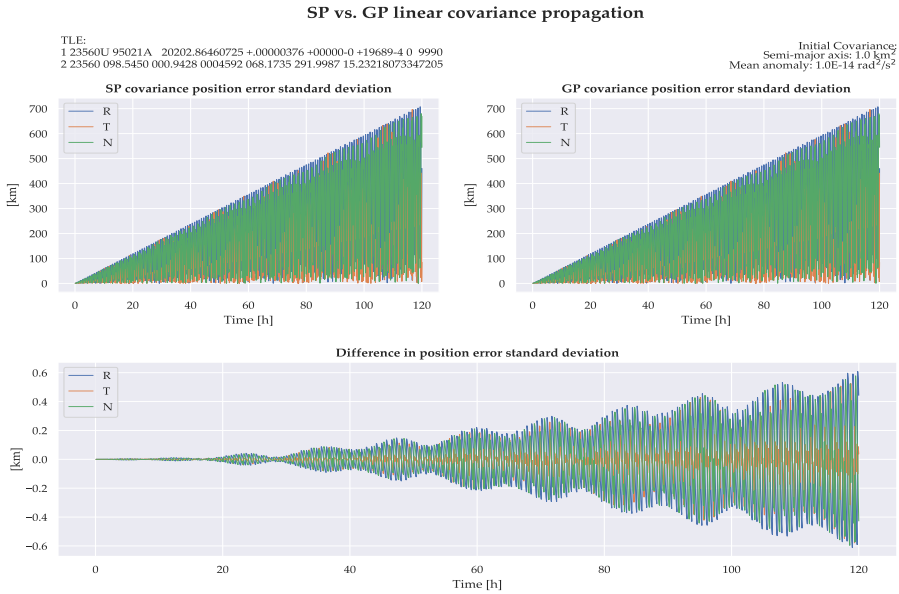


Figure 4.32: SP vs. GP covariance propagation for a randomly picked LEO RSO

4 The estimation of a ballistic coefficient from B* for LEO RSOs can only represent an approximate guess, since B* is a solve-for parameter in the TLE generation and therefore susceptible to error aliasing. In fact, some TLEs clearly reveal error aliasing in the form of negative B* values, which are not representative for the physical ballistic coefficient, however minimize the residuals during TLE generation

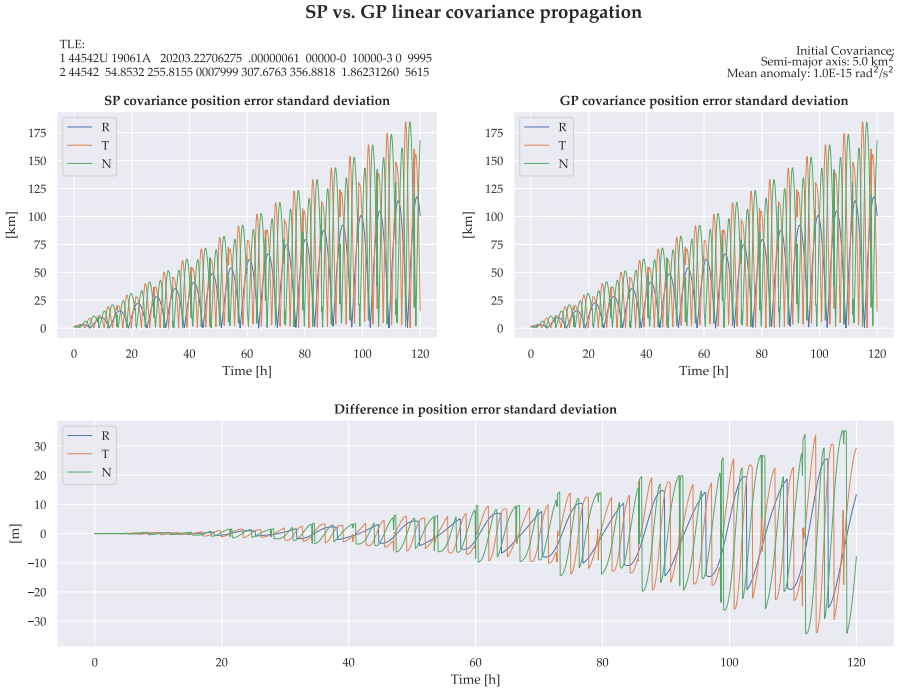


Figure 4.33: SP vs. GP covariance propagation for a randomly picked MEO RSO

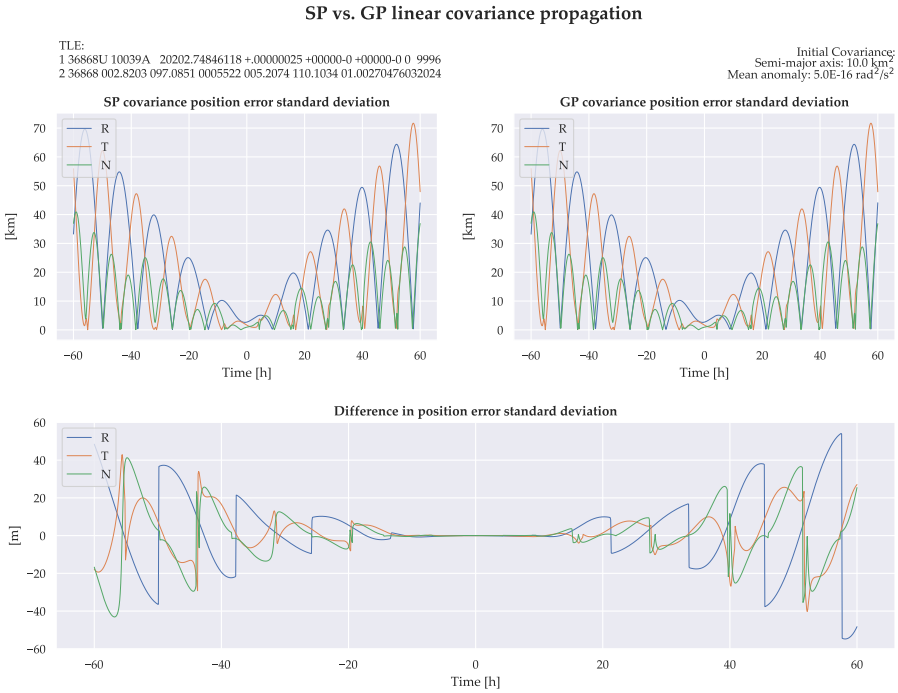


Figure 4.34: SP vs. GP covariance propagation for a randomly picked GEO RSO

All cases cover a propagation duration of five days. In figures 4.32 and 4.33 pure forward propagation is applied, whereas backward[5]

5 The ability to perform backward propagation is less relevant for real-time applications, however commonly used to detect close conjunctions between state vectors/TLE sets of differing epochs. Also some long-term encounter collision probability algorithms make use of backward propagation to accumulate the total collision risk over time

and forward propagation are simulated in figure 4.34. In all three examples the RTN covariances rise quickly up to multiple hundreds of kilometers, which in most cases is more than can be truthfully represented using a Gaussian uncertainty assumption. Therefore the initial covariance estimates allow to validate the covariance propagation for the full range of practically relevant covariance scales, up to worst-case conditions. After all, initial covariance estimates of TLE files are inaccurate, such that quick growth rates should be expected when using empirical approaches such as Flohrer et al. (2008).

The MEO and GEO standard deviations in figures 4.33 and 4.34 show differences of 30 m to 50 m after five days of propagation, which can be attributed to the model mismatch between the SP and GP theory and is negligible in case of one sigma standard deviations of 175 km and 70 km, respectively. Not surprisingly, the differences between the GP and SP propagations are greatest in LEO. The maximum standard deviation difference of 600 m is less than one tenth of a percent of the absolute standard deviation levels at this time, which is clearly negligible and in fact indicates very similar astrodynamic force modeling. The results presented in figures 4.32 to 4.34 demonstrate the effectiveness of the semi-analytic[6] GP covariance propagation and thus allow to avoid SP covariance propagation when using TLEs.

An additional question of interest is the standard deviation difference between GP and SP covariance propagation when all force models are taken into account for the SP propagation instead of aiming to match the TLE perturbations. Such results allow assessing to which extent (i.e. up to which propagation duration) SP covariance propagations can be approximated using the semi-analytic GP method. For this purpose, the scenario of figure 4.32 was recalculated, however this time using a 85×85 EGM-96 gravity field, additional HWM-93 thermospheric winds and solar radiation pressure. Figure 4.35 illustrates standard deviation differences of up to 4 km after 5 days of propagation, which is remarkable considering overall standard deviation levels of multiple hundreds of kilometers and the vast difference in the underlying force modeling theories.

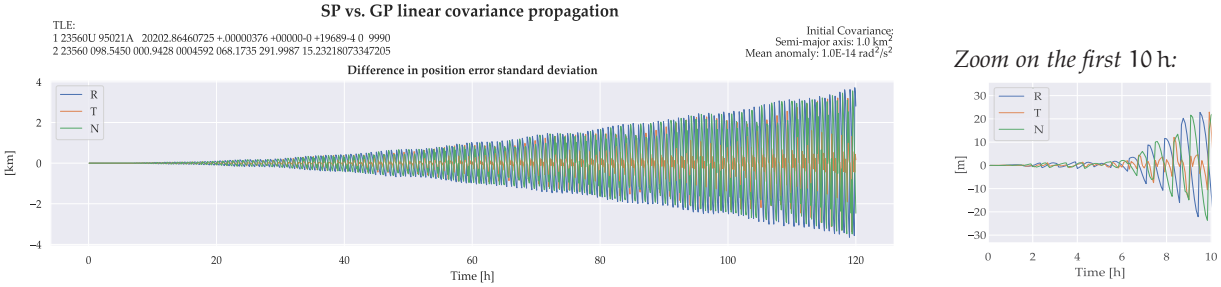


Figure 4.35: GP vs. SP covariance propagation difference for a randomly picked LEO object when considering all available force models for the SP propagation

6 Analytic GP theory and numeric STM computation

A limit on the propagation duration for which linear SP covariance propagations can be approximated using algorithm 4.1 depends on the maximum acceptable errors. A zoom on the first 10 h is depicted alongside figure 4.35 and reveals that the standard deviation errors remain below 10 m during the first 7 h of covariance propagation. Differences within 1 m are observed throughout the first 2.5 h. Similar results are also obtained for other LEO RSOs, with increasing errors for lower altitudes.

The ability to propagate covariances within the framework of GP theory is compelling, as it opens new doors for the applicability of the analytic theory. Major advantages of the method are the avoidance of model inconsistencies, while at the same time a tremendous runtime improvement over the SP theory is achieved by virtue of the analytical nature of the TLE propagation. The potential use of GP covariance propagation within the context of the force model uncertainty quantification framework is therefore an intriguing idea that is examined in the following.

FORCE MODEL UNCERTAINTY FRAMEWORK EVALUATION USING THE GP THEORY

The near Earth part of the GP theory, i.e. SGP4, has its root in the works of Brouwer and Kozai, who independently studied the motion of near Earth objects and published their results in the same month and journal (Brouwer, 1959; Kozai, 1959).

TLE sets contain eight variables required for orbit propagation, which include the epoch time, an atmospheric drag coefficient and six mean orbital elements. All orbital elements, except for the (mean) mean motion follow the Brouwer theory, whereas the latter is taken from Kozai. The development of the SGP4 model, including commonalities with alternative analytical models such as PPT3, is presented in Hoots et al. (2004) along with the equations that describe the initialization of the algorithm and the orbit propagation.

The SGP4 theory retains only the long and short periodic terms of the Brouwer theory that do not contain eccentricity (Hoots et al., 2004). In contrast to PPT3, which assumes a semi-empirical drag model, the treatment of atmospheric drag is based on the power function model proposed by Lane and Cranford (1969), with the difference that only the dominant terms for the secular effect of drag are included (Hoots et al., 2004). Secular Earth zonal gravity effects are included from Brouwer up to J_4.

In view of these fundamental model differences compared to the SP theory, it must be clear that the developments of chapters 2 and 3 cannot be transferred 1:1 to the GP theory. This is also not the objective of this section. The basic idea is rather to reassess the force model uncertainty framework regarding the extent to which the concepts can be related to the GP theory and which parts of the computations can be performed using SGP4 orbit propagation and semi-analytic

SGP4 covariance propagation. As only a subset of the considered SP force model uncertainty origins is conceptually transferable to the GP theory and since individual quantities of the transferable components cannot be obtained using pure SGP4, the extension of the force model covariances towards the GP theory is considered experimental. Nevertheless, it is believed that the resulting covariance estimates are more accurate than the operational techniques of empirical covariance scaling discussed in section 1.4.2.

Results of the GP force model uncertainty framework evaluation are presented in section 6.2, where the force model quantification is applied to the covariance propagation between the RSOs epoch times and the time of closest approach in order to assess its impact on the collision probabilities determined from space catalog snapshots[7].

Atmospheric density uncertainty

The atmospheric density uncertainty quantification framework is summarized in table 2.7. The GP model does not use solar flux or geomagnetic activity space weather files as input. Although the solar energy input and the current forcing of the magnetosphere also contribute to TLE accuracy, there is no connection between these model-specific density uncertainty drivers and the analytic GP force model. The meaningfulness of considering input density uncertainty is therefore questionable. Average model uncertainty however also applies to the GP theory. The parametrization of the Ornstein-Uhlenbeck error process model only requires an assumption for the average model accuracy (s_{Model}) and an estimate of $\frac{\Delta\hat{n}}{\Delta t}$.

Literature reports no estimates for the average accuracy of the SGP4 power law density model, however assuming the standard deviation to be at least twice that of modern semi-empirical density models, a broad range of $20\% \leqslant s_{\text{Model}} \leqslant 40\%$ may be guessed. The fraction $\frac{\Delta\hat{n}}{\Delta t}$ needs to be evaluated with density-only perturbations. The most consistent option to compute the fraction would be to separate the SGP4 force model into its components in order to consider only the drag contribution. However, since the power law density model is inherently coupled with the gravity model (Lane and Cranford, 1969), this would be an unreasonably complex approach.

A better solution is to use the analytical estimation for the term derived in Emmert et al. (2017, appendix A) which doesn't require an orbit propagator and assumes a spherically symmetric exponential atmosphere. Alternatively also an SP propagator may be used (ideally with averaged space weather input files for the time period of interest).

Sequential covariance propagation with density process noise consideration is based on equation 2.300, where $\mathbf{\Phi}(t_{k+1}, t_k)$ is to be com-

7 In fact, the idea of evaluating the force model covariances using the GP theory and comparing results to the SP case emerged after investigating the possibility of GP covariance propagation as part of the work carried out for the space catalog conjunction assessment impact analysis presented in chapter 6

puted with the full force model used for state propagation. Thus, equation 2.300 needs to be evaluated using algorithm 4.1 for $\mathbf{\Phi}(t_{k+1}, t_k)$.

If the decision is made to additionally compute the SP input density uncertainty despite its missing connection to the GP theory, the estimation of σ_{ρ_p} for these components via equation 2.83 is only possible by means of a semi-empirical density model, which inevitably deviates from the SGP4 density model.

In summary, it can be concluded that grid-scale density uncertainties are compatible with the GP propagation theory once the ability of GP covariance propagation according to algorithm 4.1 is implemented and the analytical estimation of $\frac{\Delta \hat{n}}{\Delta t}$ from Emmert et al. (2017) or alternatively drag-only averaged space weather SP propagation is used to calculate the corresponding covariance matrix according to equations 2.247, 2.248 and 2.288.

Gravity uncertainty

The core of the gravity uncertainty algorithm is the evaluation of equation 3.53 using precomputed inner integral data. Algorithm 4.1 allows to compute $\mathbf{\Phi}(t_{k+1}, \tau_j + \Delta/2)$ using the GP theory. Similarly, the nonzero components of the matrix $\mathbf{G}(\tau_j + \Delta/2)$ consist of the RTN to GCRF frame conversion, which can be evaluated with the help of a single GP state vector propagation. This leaves the evaluation of the matrix product $\mathbf{R}(0, \tau_j + \Delta/2)^{1/2}\ \mathbf{I1}(\tau_j + \Delta/2)\ \mathbf{R}(0, \tau_j + \Delta/2)^{1/2}$ according to equation 3.52. Both components of the matrix product require an estimate of the Earth central distance at time $\tau_j + \Delta/2$ for their evaluation. This estimate can be obtained using the same GP propagation performed for the evaluation of $\mathbf{G}(\tau_j + \Delta/2)$. Both parts of the matrix product however also depend on precomputed model data (inner integral coefficients and degree variances). The SGP4 theory accounts for the secular effects of Earth zonal harmonics up to and including J_4.

Modern gravity models agree fairly well within the potential field coefficient range required for LEO orbit determination. Thus, any gravity model may be used to approximate the orbital impact of GP gravity errors of omission by setting the full field degree/order truncation point as recommended by the IERS. Gravity errors of commission only apply up to J_4. As the standard deviation of the first potential field coefficients reported by modern gravity models is very similar, an arbitrary model can also be used to estimate the orbital impact of gravity errors of commission.

In summary, it can be concluded that the orbital uncertainty due to gravity errors can also be approximated by means of the GP theory. Similar to the density uncertainty case, the GP framework only has to be abandoned for the evaluation of a single quantity, which however is not expected to have a significant impact on the overall uncertainty estimates.

5

ADAPTIVE GAUSSIAN MIXTURE BASED ORBIT DETERMINATION

Corresponding publications: Schiemenz et al. (2020d)

Large initial uncertainties in the semi-major axis or force model uncertainties are key drivers of the along-track uncertainty growth. Long propagation times may therefore result in the need to use filtering algorithms for orbit determination that do not reside to the assumption of Gaussianity for state errors but estimate the entire probability density function (pdf). Adaptive Gaussian mixture based filters have demonstrated the capability of realistic pdf estimation in the past. Previous research in the field of orbit determination using Gaussian mixture filters however restricted its attention to initial uncertainties in the semi-major axis direction and neglected force model uncertainties.

Chapter 4 assumed the orbit errors to be Gaussian, such that classic orbit estimation algorithms were suitable for the OD when additionally including force model uncertainties. The Gaussian assumption shall be relaxed in this chapter, which presents the development of a novel Gaussian Mixture Model (GMM) orbit estimator capable of considering atmospheric density uncertainty. In principle, the additional consideration of gravity uncertainty would also be possible, however the aleatoric treatment of the epistemic gravity errors complicates the validation of the overall algorithm, as the global averaging renders the stochastic gravity model zero-mean, whereas the actually considered orbit errors are not (cf. chapter 3).

It is shown that the neglect of process noise, as has been customary for many years, can lead to undesired characteristics of the pdf estimates and that the inclusion of atmospheric density uncertainty process noise, even in cases where it is not the dominant driver of along-track uncertainty growth, is able to correct these deficiencies. For LEO satellites with increased ballistic coefficients or small initial uncertainties in the semi-major axis direction, density uncertainty is the dominant driver of the along-track uncertainty increase. Due to its growth that evolves at least cubic in time, situations may arise which require the usage of Gaussian mixtures also for the process noise when working in Cartesian coordinates. The theoretical foundation for this case is elaborated and an algorithm capable of dynamically switching between a single Gaussian and a Gaussian mixture for the density uncertainty process noise is presented.

5.1 INTRODUCTION TO NON-GAUSSIAN ORBIT DETERMINATION

A common and justified practice is to assume that the initial orbit determination (IOD) of an object conditioned on multiple days of

measurement tracks results in a Gaussian probability density function (Horwood and Poore, 2011, section VIII). The Gaussian assumption has the very appealing property that only the first and second moment of the pdf (the mean and the covariance) are required to fully describe the orbital uncertainty. The nonlinear dynamics however result in any initially Gaussian pdf to eventually become non-Gaussian (e.g. crescent or banana-shaped). The point in time when this happens depends on two factors: the error volume of the initial Gaussian and the degree of nonlinearity experienced in the transformation. It is therefore possible to maintain the Gaussian assumption for highly nonlinear transformations, given that the Gaussian volume is sufficiently small. Similarly also a weakly nonlinear transformation may render the Gaussian approximation invalid, if the volume of the Gaussian is sufficiently large.

A widely accepted technique to handle non-Gaussian error volumes are Gaussian Mixture Models (GMMs), also known as Gaussian sums[1]. GMMs are weighted additions of individual Gaussians that have been shown to converge to any pdf arbitrarily closely in the L^1-norm (Sorenson and Alspach, 1971). The theory of Gaussian mixture filtering dates back to 1972 when Alspach and Sorenson published the corresponding filtering equations and techniques (Alspach and Sorenson, 1972).

The main principle behind GMMs in the context of SSA is to split an initial Gaussian into a Gaussian mixture whenever the nonlinearity is sufficiently large for the transformation to result in a non-Gaussian error volume. Since a split decreases the variances of the components (also called kernels) along the splitting direction, the resulting kernel covariances are assumed to be sufficiently small to be truthfully propagated via the well-established propagation techniques, such as first order Taylor linearization, Gauss-Hermite (GH) quadrature (Genz and Keister, 1996) or unscented transformation (Julier and Uhlmann, 1997, 2004). Since nonlinearity acts upon the resulting kernels independently, the filter cycles can be perfectly parallelized on modern CPUs or GPUs, allowing for efficient implementation.

Successful application of Gaussian mixtures in orbit determination requires the mixtures to be adaptive. Some researchers have focused on making the kernel weights adaptive. While it can be shown that any linear transformation leaves the weights unchanged, any nonlinear transformation gives room to kernel weight optimization. Examples of this class of adaptivity are Terejanu et al. (2008a,b, 2011) and Vishwajeet et al. (2014). Others have focused on making the number of Gaussian mixture components adaptive, for example DeMars et al. (2013), Horwood and Poore (2011) or Horwood et al. (2012). While updating weights with the Fokker-Planck-Kolmogorov Equation (FPKE)-error as feedback is reported to improve the quality of the pdf estimates (at the cost of increased operational runtimes) when using first order Taylor-

1 Gaussian mixture model and Gaussian sum are synonyms

series based propagation (Terejanu et al., 2008b), Horwood et al. (2011) determined that when using higher-order nonlinear methods such as UT (or GH quadrature) no pdf quality improvement is observed (Vittaldev and Russell, 2016).

This chapter follows the second approach in that the orbit determination is started with a single Gaussian. Refinement is performed using a nonlinearity index, where the one developed in Vittaldev (2015) is recommended for this purpose, since in contrast to entropy-based approaches (e.g. DeMars et al., 2013) it is not only able to tell when to split, but also provides a means to choose the most promising splitting direction(s).

GMM-related studies generally assume that the uncertainty is sufficiently large to justify the use of Gaussian mixtures. In the past decades however also some research has been performed that investigated the operational impact of non-Gaussian error volumes and the question of when Gaussian error volumes become non-Gaussian. This type of research allows to evaluate the need for higher order filtering techniques. Relevant publications of this topic are Junkins et al. (1996), Sabol et al. (2010), Alfriend and Park (2016) and Ghrist and Plakalovic (2012), as already noted in section 1.3.3. Based on these investigations the following statements can be made:

- Orbit determination in orbital element space (e.g. COE or EOE) significantly increases the validity of the Gaussian error volume assumption. 10 days and more are not uncommon for well-cataloged LEO RSOs
- The primary reason for Cartesian representations of the orbit error distribution to become non-Gaussian much quicker than in orbit element space is the limitation of the linear reference frame and not the potential usage of linearized dynamics for covariance propagation
- Nonlinearity is easiest detected in the radial/transverse plane (Cartesian coordinates) or in the size-variable[2]/fast-variable[3] plane (orbital elements)
- Non-Gaussian error volumes have no operational impact on high interest events (HIEs) in spacecraft conjunction assessment
- Non-Gaussian error volumes are required for larger initial uncertainties in the radial position direction (a couple of hundred meters), which are common to the uncorrelated track (UCT) problem

As previously noted, a key aspect that has not been taken into account in all of the recent GMM-based studies is the force-model uncertainty encountered when performing orbit propagation. Grid-scale density

2 The semi-major axis or mean motion are typical size-variable choices in COE and EOE

3 Common fast variables are: mean anomaly (COE), true anomaly (COE), mean longitude (EOE) or true longitude (EOE), however also the eccentric anomaly/longitude are possible

uncertainty grows with time cubed (t^3), as has been shown in section 2.6. If only propagated model input data is available (which is the case for real-time applications), a Brownian motion error process model must be assumed for the density uncertainty caused by the uncertainty of the solar flux index/indices. In this case the error may even grow with t^5, as shown in Emmert et al. (2017) and section 2.5.3.

It is known that the growth of the along-track standard deviation depends linearly on the initial standard deviation in the radial position (semi-major axis) direction (Horwood et al., 2011). Force model uncertainty caused by atmospheric density uncertainty may hence become the dominant mechanism of uncertainty growth if either the initial uncertainty in the semi-major axis direction is small, or if the ballistic coefficient of the satellite is sufficiently large.

In the following Gaussian mixture-based orbit determination with the explicit consideration of atmospheric density uncertainty is investigated for the first time. Key questions to be answered are:

1. How can atmospheric density process noise be incorporated into Gaussian mixtures which undergo component splitting and merging?
2. Can process noise be safely neglected when performing orbit determination/uncertainty propagation using Gaussian mixtures, as claimed in Horwood and Poore (2011)? Or is the contrary the case, i.e. that it is also beneficial to be considered even if it is not the dominant process of the uncertainty growth?
3. What are remaining challenges and operational implications?

5.2 GAUSSIAN MIXTURES, COARSENING AND REFINEMENT

Prior to the discussion of the GMM design required to make the special nature of the atmospheric density process noise compatible with Gaussian mixtures, the relevant theory behind the GMM pdf representation, GMM splitting and GMM merging (also called refinement and coarsening/reduction) needs to be introduced.

5.2.1 *Gaussian mixture probability densities*

In the context of Gaussian mixture models the true pdf is approximated by a finite sum of weighted Gaussian kernels:

$$\hat{\mathbf{p}}(t, \mathbf{x}(t)|\mathbf{z}_k) = \sum_{i=1}^{N} w_i \, \mathcal{N}(\mathbf{x} - \boldsymbol{\mu}_i, \mathbf{P}_i) \tag{5.1}$$

where:

- $\hat{\mathbf{p}}(t, \mathbf{x}(t)|\mathbf{z}_k)$ denotes the pdf estimate at time t conditioned on the first k measurements $\mathbf{z}_k$
- N denotes the total number of Gaussian kernels in the mixture
- w_i is the kernel weight with the constraints of normalization $\left(\sum_{i=1}^{N} w_i = 1\right)$ and positivity $(w_i \geqslant 0)$ for $i = 1, \ldots, N$

Some authors also write $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_i, \mathbf{P}_i)$ or simply $\mathcal{N}(\boldsymbol{\mu}_i, \mathbf{P}_i)$ to express the normal distribution in the state $\boldsymbol{x}$ conditioned on the mean $\boldsymbol{\mu}_i$ and covariance matrix $\mathbf{P}_i$. The simplest GMM is the degenerate case of one component with weight $w_1 = 1$, which equals the assumption of Gaussianity. In this case the terms uncertainty and covariance can be used interchangeably.

After a measurement update the kernel weights are updated according to equation 5.2:

$$w_{i,k} = \frac{w'_{i,k}\beta_{i,k}}{\sum_{i=1}^{N} w'_{i,k}\beta_{i,k}} \tag{5.2}$$

where $\beta_{i,k}$ denotes the evaluation of the innovation normal distribution at the estimated a priori kernel mean $\boldsymbol{\mu}'_{i,k}$, $\mathbf{R}_k$ the sensor covariance matrix, h the measurement function and $\mathbf{H}$ its Jacobian:

$$\beta_{i,k} = \mathcal{N}(\boldsymbol{z}_k - h(t_k, \boldsymbol{\mu}'_{i,k}), \mathbf{P}_{zz}) \tag{5.3}$$

In case of the EKF it holds that $\mathbf{P}_{zz} = \mathbf{H}_{i,k}\mathbf{P}'_{i,k}\mathbf{H}^T_{i,k} + \mathbf{R}_k$. For the UKF the innovation covariance is obtained from the kernel sigma point set (index j) via $\mathbf{P}_{zz} = \left[\sum_{j=0}^{2n_{\text{sigma}}} w_j \left(\boldsymbol{z}_k - h(t_k, \boldsymbol{\mu}'_{i,k,j})\right)\left(\boldsymbol{z}_k - h(t_k, \boldsymbol{\mu}'_{i,k,j})\right)^T\right] + \mathbf{R}_k$.

A non-adaptive Gaussian mixture estimator can be obtained by splitting an initial covariance into a Gaussian mixture and then running a bank of nonlinear estimators in parallel on each component. This static Gaussian mixture however quickly degenerates to the case of a standard extended or unscented Kalman Filter, since initially the uncertainty is still Gaussian and therefore all kernels but one will obtain zero weight within a couple of filter cycles. A better approach is to start with one initial kernel, i.e. the common Gaussian assumption, which is then refined into a Gaussian mixture whenever a nonlinearity index recommends a split.

5.2.2 *Splitting of a multivariate Gaussian*

The splitting of multivariate Gaussians is achieved by applying a univariate splitting library along a certain splitting direction. Establishing a univariate splitting library requires solving a complicated nonlinear optimization problem (Horwood et al., 2011; Vittaldev and Russell, 2016). The most extensive univariate splitting library published to date has been worked out in Vittaldev and Russell (2016) and is publicly available for download. The Vittaldev-libraries, as well as the three to five component libraries given in Huber et al. (2008) and DeMars et al. (2013) are all homoscedastic, meaning that the kernels have the same finite variance. As has been shown in Horwood et al. (2011), homoscedasticity is by no means a restriction but an interesting property of the optimal solution. For the following developments the splitting library by Vittaldev which supports up to $N = 39$ kernels per splitting action was used.

The methodology behind splitting a Gaussian kernel along an eigenvector of the covariance has been elaborated in Huber et al. (2008). A first generalization of multivariate Gaussian splitting which allows an arbitrary splitting direction can be found in Aristoff et al. (2014). An optimized algorithm which does not involve Gram-Schmidt orthogonalization has been presented later in Vittaldev and Russell (2016). The respective generic splitting equations are:

$$\boldsymbol{\mu}_i = \boldsymbol{\mu} + \mu_i \mathbf{S}\hat{\mathbf{a}}^\star \tag{5.4}$$

for the kernel means and

$$\mathbf{P}_i = \mathbf{S}\left(\mathbf{I} + (\sigma_i^2 - 1)\hat{\mathbf{a}}^\star \hat{\mathbf{a}}^{\star\mathsf{T}}\right)\mathbf{S}^\mathsf{T} \tag{5.5}$$

for the kernel covariance matrices, where

$$\hat{\mathbf{a}}^\star = \frac{\mathbf{S}^{-1}\mathbf{u}}{\|\mathbf{S}^{-1}\mathbf{u}\|_2} \tag{5.6}$$

The scalar variables w_i, μ_i and σ_i are obtained from the chosen univariate standard Gaussian splitting library. $\mathbf{S}$ can be computed either as the principal matrix square root obtained e.g. via spectral decomposition or as the lower triangular Cholesky factor. For a homoscedastic splitting library it holds that $\sigma_i = \sigma$ and therefore $\mathbf{P_i} = \mathbf{P}$. Splitting in multiple directions is realized in a recursive manner. As long as the splitting directions are orthogonal their order has no influence on the result (Vittaldev and Russell, 2016).

Once the splitting routines are implemented, the next task is to develop the ability of choosing a suitable splitting direction $\mathbf{u}$. Two main factors influence the splitting decision: the error volume described by the covariance and the nonlinearity experienced in the transformation. Consequently it makes sense to think about an index which takes both arguments into consideration and translates them into a single number per splitting direction, such that at any time it is possible to decide in which direction(s) to split with how many kernels. Entropy-based splitting criteria don't fulfill this purpose, as they do not differentiate between splitting directions.

Probably the first published index for this purpose has been presented in Junkins et al. (1996) and reads:

$$\varphi_i = \frac{\|\boldsymbol{\Phi}_i(t, t_0) - \boldsymbol{\Phi}(t, t_0)\|_2}{\|\boldsymbol{\Phi}(t, t_0)\|_2} \tag{5.7}$$

where $\boldsymbol{\Phi}$ denotes the STM from time t_0 to t, evaluated at the kernel mean and $\boldsymbol{\Phi}_i$ the STM evaluated at any point on the initial three-sigma envelope. φ_i can then be computed for an arbitrary splitting direction $\mathbf{u}_i$ by computing the STM for the propagation from t_0 to t for the initial state $\boldsymbol{\mu}_i = \bar{\boldsymbol{\mu}} \pm 3\|\mathbf{S}^{-1}\mathbf{u}_i\|_2^{-1}\mathbf{u}_i$. In his original work Junkins

used 12 different points[4] for the evaluation of φ_i and then chose the supremum for plotting the overall nonlinearity index.

The nonlinearity measure defined by equation 5.7 has two distinct downsides. First, it works with linearized state error transition matrices, which are not required in case of the modern unscented Kalman filter. Second, the evaluation of an STM is computationally more expensive than the propagation of a satellite state vector. Due to this second argument the Junkins index is also not too suitable for an EKF, as 12 additional STMs would need to be evaluated for each GMM kernel.

Vittaldev (2015) proposed a different nonlinearity measure, which instead operates on the set of sigma points:

$$\varphi_i = \left\| \frac{f(\bar{\mu} + h\|S^{-1}u_i\|_2^{-1}u_i) + f(\bar{\mu} - h\|S^{-1}u_i\|_2^{-1}u_i) - 2f(\bar{\mu})}{2h^2} \right\|_p \tag{5.8}$$

where h is a scalar chosen in such a way that the nonlinear function evaluations correspond to the propagated kernel mean ($f(\bar{\mu})$) and the symmetric sigma-points along u_i. The p-norm may be taken as the 2-norm without restrictions. If the state vector elements have different units, the sigma point set first needs to be converted to canonical units via the respective canonical distance, time and velocity scales $\left(\mathtt{DU} = R_{\text{Earth}},\ \mathtt{TU} = \sqrt{R_{\text{Earth}}^3/(GM)} \text{ and } \mathtt{VU} = \mathtt{DU}/\mathtt{TU} = \sqrt{(GM)/R_{\text{Earth}}}\right)$ prior to the evaluation of equation 5.8 (Vittaldev and Russell, 2016). For orbit elements (COE/EOE) only the size variable is dimensional and needs to be converted (multiplication with TU for mean motion or division by DU for semi-major axis).

In case of sigma point Gaussian uncertainty propagators (such as e.g. UT) the evaluation of this nonlinearity measure is a simple algebraic operation which essentially boils down to the comparison of the combined deviation of the symmetric sigma point pairs per considered direction with respect to the propagated kernel mean. If the matrix root is realized using Cholesky decomposition, then the considered splitting directions are those parallel to the columns of the lower triangular Cholesky factor. For spectral decomposition they are parallel to the eigenvectors of the kernel covariance matrix.

Since propagating sigma points is computationally less expensive than propagating STMs, equation 5.8 is to be preferred over the Junkins nonlinearity measure also for adaptive Gaussian mixture based filters that use the EKF as workhorse. It has to be mentioned, however, that in this case the evaluation of equation 5.8 requires the uncertainty propagation to be performed twice: once using an STM for the kernel state and once using unscented transformation to evaluate the nonlinearity measure.

4 Junkins work dates back to a time prior to the UT, which makes the equality between his choice of 12 evaluation points to the commonly adopted set of 12 symmetric sigma points when performing UT even more interesting

Finally, it makes sense to translate the nonlinearity measure into a recommendation for the number of Gaussian kernels to be used for a split. There is no universal function for this purpose, however the selected one should make sure that small values of the nonlinearity index result in no splits, whereas for increasing values of φ_i the recommended splitting number should increase and stay in line with the maximum component number supported by the underlying splitting library. For a typical LEO satellite using Cartesian coordinates the Vittaldev-index was found to start at values in the order of 10^{-12}, which then increased with propagation time to orders of 10^{-4} and larger if no split was performed. Motivated by numerical experiments, an equation of the form $n_i = a \cdot \ln(\varphi_i) + b$ was chosen to translate between nonlinearity index and kernel splitting recommendation. Since the Vittaldev splitting libraries only support splits with uneven kernel numbers, the final function selected reads:

$$n_i = \text{odd}(0.404 \ln(\varphi_i) + 7.0) \tag{5.9}$$

with $\text{odd}(x) = 2\,\text{ceil}(x/2) - 1$. Equation 5.9 results in the mapping presented in table 5.1.

φ_i	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^{0}
n_i	1	3	3	5	5	7	7

Table 5.1: Translation between nonlinearity index and kernel splitting recommendation as provided by equation 5.9

For practical operations any value of n smaller than the minimum supported split of the splitting library ($n_{min} = 3$ for the Vittaldev-libraries) is set to zero. The mapping function can be made more aggressive by increasing b. Decreasing b results in splits being executed later, i.e. for larger values of the nonlinearity index.

5.2.3 *GMM coarsening*

After a measurement update Gaussian kernels can become sufficiently close to be replaced by fewer mixture components. Consequently the ability of Gaussian mixture reduction, also referred to as GMM coarsening or GMM merging is required to keep the number of kernels under control. Gaussian mixture reduction can be defined as the task of replacing a Gaussian mixture in N components with another Gaussian mixture in M components where $M < N$, while keeping the distance between the resulting pdf from the original probability density as small as possible for the distance metric under consideration.

Most of the merging algorithms found in literature make use of the moment-preserving merge, which is defined by equations 5.10 to 5.12, as their basic instrument.

$$w = \sum_{i=1}^{N} w_i \tag{5.10}$$

$$\boldsymbol{\mu} = \sum_{i=1}^{N} w_i \boldsymbol{\mu}_i \tag{5.11}$$

$$\mathbf{P} = \sum_{i=1}^{N} w_i \left[\mathbf{P}_i + (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T\right] \tag{5.12}$$

If the moment-preserving merge is computed for a subset of the Gaussian mixture ($N < N_{GMM}$), then the weights in equation 5.11 and 5.12 need to be normalized with the weight obtained from equation 5.10. In the special case of the two-component moment-preserving merge, the equations reduce to:

$$w_{ij} = w_i + w_j \tag{5.13}$$

$$\boldsymbol{\mu}_{ij} = w_{i|ij}\boldsymbol{\mu}_i + w_{j|ij}\boldsymbol{\mu}_j \tag{5.14}$$

$$\mathbf{P}_{ij} = w_{i|ij}\mathbf{P}_i + w_{j|ij}\mathbf{P}_j + w_{i|ij}w_{j|ij}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \tag{5.15}$$

where $w_{i|ij} = \frac{w_i}{w_{ij}}$ and $w_{j|ij} = \frac{w_j}{w_{ij}}$.

In 1988 Salmond proposed two distance criteria to determine pairs of GMM kernels which are suitable candidates for merging (Salmond et al., 1988). Some time later Jason Williams worked on Gaussian mixture reduction using the integral squared difference (ISD) of two pdfs (Williams, 2003; Williams and Maybeck, 2003). His ISD metrics have the appealing property of being solvable in closed-form for Gaussian distributions. A comparison of Salmond's and Williams' distance metrics is presented in Runnalls (2007) alongside the presentation of a novel metric based on the Kullback-Leibler (KL) distance. The KL-metric not only shows superior results for the example considered in Runnalls (2007), but can also be computed orders of magnitude faster than Salmond's and Williams' metrics, since neither matrix inversions nor double sums of multivariate Gaussian pdf evaluations are required to obtain the KL distance. The simplicity and performance of Runnalls' KL-distance has helped to attract a number of researchers to its application in the past decade. Given two kernel components, the KL-distance can be computed using equation 5.15 for $\mathbf{P}_{ij}$ and eq. 5.16 for the resulting distance measure:

$$\begin{aligned} B\big((w_i, \boldsymbol{\mu}_i, \mathbf{P}_i), (w_j, \boldsymbol{\mu}_j, \mathbf{P}_j)\big) = \frac{1}{2}\big[&(w_i + w_j)\log\big(\det(\mathbf{P}_{ij})\big) \\ &- w_i \log\big(\det(\mathbf{P}_i)\big) - w_j \log\big(\det(\mathbf{P}_j)\big)\big] \end{aligned} \tag{5.16}$$

All merging strategies that have just been reviewed work by merging pairs of Gaussian components until either a desired target size or a maximum distance threshold between the kernels is reached. As will be discussed in section 5.4, the performance of the merging algorithm is crucial for operational GMM-based orbit determination. The reason is that the computational burden of obtaining the distance metric grows with the square of the number of mixture components. Under certain circumstances (e.g. those discussed in section 5.4), the number of Gaussian kernels can easily go into the thousands. In this case it becomes very costly to repeatedly compute the distance metric for every possible combination of kernel pairs and to keep track of the minimum distances encountered.

5.3 GAUSSIAN MIXTURE MODEL DESIGN FOR ATMOSPHERIC DENSITY UNCERTAINTY CONSIDERATION

A design for a GMM-based OD algorithm which shall be adaptive in the number of kernels needs to accommodate the steps of nonlinearity detection, splitting of a Gaussian into a Gaussian mixture and merging of Gaussians. When process noise shall be considered, also the computation of the process noise matrices needs to be well thought of. Important points in the design are discussed topic by topic below after a short recap on covariance propagation with density uncertainty process noise.

DENSITY UNCERTAINTY COVARIANCE PROPAGATION

The content of sections 2.7.2 and 2.7.3 is imperative for the successful design of a GMM filter capable of density uncertainty consideration. This includes the computation of the overall density uncertainty covariance, the frame conversions between orbital elements and inertial GCRF coordinates, as well as the particularities of the stepwise propagation of the filter covariance matrices.

Atmospheric density uncertainty has the unique advantage that it can be formulated analytically. A difficulty however is that the theory develops from the start of a density error process, i.e. the time of the last measurement update. As such the combined covariance matrix cannot be simply evaluated in incremental steps of δt, which is typical for sequential orbit determination. For an EKF, covariance propagation including density uncertainty process noise can be achieved using equation 2.300 or alternatively equations 2.302 and 2.303. Both options operate in the same way: to propagate the covariance in time, any prior process noise contribution since the last measurement update needs to be removed from the covariance matrix of the state before this process-noise free covariance can be propagated forward in time using the STM. Finally, the process noise matrix which belongs to the time t_{k+1} is added to the result. Due to the analytic nature of the

density uncertainty framework, increased values of t do not lead to longer computation times.

If one wishes to account for additional sources of process noise, equation 2.300 must be used as well. In this case $\mathbf{Q}_k$ represents the total accumulated process noise since the last measurement update until time t_k. The propagation is then performed in a similar manner by first subtracting the so far accumulated process noise prior to propagation and a subsequent addition of the total process noise for time t_{k+1}. To avoid long computation times, the accumulated process noise contribution of sources formulated in a non-analytical manner should be stored in memory at any time-step. In case of an unscented Kalman filter the covariance propagation is performed using equation 2.304 and sigma point cloud-saving.

The general concept of obtaining a process noise free covariance prior propagation with subsequent addition of the updated propagation error covariance is essential for the extension of the concepts to Gaussian mixture models.

COMPUTATION OF NONLINEARITY INDEX AND KERNEL SPLITTING RECOMMENDATION

Equation 5.8 operates on the set of the propagated sigma points used for covariance propagation. Consequently the function needs to be evaluated directly after the covariance propagation step of every kernel filter cycle. This results in a $N \times 6$ matrix of splitting recommendations, since every kernel may be split in up to six directions. In practice however typically one or two components of the nonlinearity index vector dominate the others by orders of magnitude, such that equation 5.9 will only recommend splitting in one or two dimensions - even for Cartesian coordinates.

Covariance propagation (CP) is usually performed in equally spaced time intervals in the absence of measurements. If this time interval is sufficiently short, it can be justified to apply the splitting recommendations computed using the covariance propagation up to the current time prior to the next iteration of the CP-loop. If covariance propagation is implemented without interruption during measurement updates, it becomes necessary to partition the propagation into small intervals to allow for nonlinearity to be detected.

KERNEL MERGING

Reduction of the Gaussian mixture components is crucial to keep the overall number of kernels under control, as otherwise repeated splitting operations quickly lead to kernel numbers growing without bound. An intuitive point of time for merging is directly after a measurement update, since then the volume of the pdf is significantly reduced, such that less kernels are needed for propagation. However, even in the absence of measurements merging is required. It may

either be performed at the end of a filter cycle (after the CP) or at the beginning of the next iteration. In the latter case the simulation output data directly reflects the results of the individual kernel covariance propagation, without kernel merging and therefore data reduction.

In both cases it is important to be aware of the fact that the merge of two kernels results in the combined covariance to grow. Therefore the saved cloud of sigma-points must be recomputed from the merged covariance after a merge has taken place. The cloud of the removed kernel needs to be deleted. Similarly also the GMM data structures containing the weights, means and covariances need to be adapted accordingly to reflect the merge.

The steps required for kernel merging are summarized in figure 5.1 and form a building block of the overall filter design in figure 5.2.

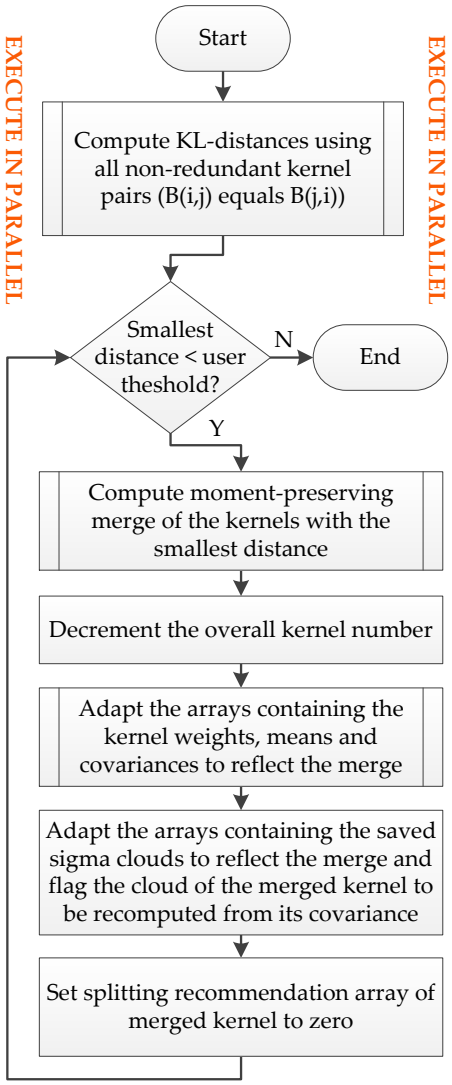


Figure 5.1: Kernel merging for adaptive GMM filtering

KERNEL SPLITTING

As discussed above, it is possible to apply the splitting recommendations prior to the next iteration of the covariance propagation. In the meantime however a merging process has taken place and may have changed the kernel covariance or even deleted the kernel. It is therefore a design decision to treat the case where the merging algorithm suggests that a kernel can be merged which has a nonzero entry in the $N \times 6$ matrix of splitting recommendations. A conservative choice would be to take the greater splitting recommendation of every possible splitting direction for the two kernels contained in the merge. This however quickly leads to an unbalanced filter, since the merged kernel would be split shortly after. A better choice is to clear the splitting recommendation in this case to avoid unnecessary merging with subsequent splitting. Eventually, this situation should

not arise too often, as it is a sign of unfavorably chosen splitting and merging thresholds (e.g. overly aggressive splitting and merging).

Since a split replaces the original kernel with multiple smaller ones, the saved sigma-point cloud of the original kernel needs to be deleted whenever a split is detected. The cloud of the new "children" is then to be computed from the reduced covariance and subsequently saved for propagation until the next merge or split.

INCLUSION OF PROCESS NOISE

Finally, the incorporation of atmospheric density uncertainty into a Gaussian mixture filter for orbit determination shall be described.

From a theoretical viewpoint the incorporation of process noise into a Gaussian mixture filter has already been treated in the original work by Alspach and Sorenson (1972). Having the EKF as workhorse in mind, they showed that the classical linearized covariance propagation equation is valid for each kernel (index i) individually, i.e. $\hat{\mathbf{P}}_{k+1,i} = \mathbf{\Phi}_i(t_{k+1}, t_k)\hat{\mathbf{P}}_{k,i}\mathbf{\Phi}_i(t_{k+1}, t_k)^T + \mathbf{Q}_k$. In Alspach and Sorenson (1972) the process noise[5] is assumed to be equal for each kernel. In fact, it would not even be possible to evaluate equation 2.306 for every kernel, since due to the merging and splitting a kernel is not guaranteed to "survive" the entire time between measurement updates. Even if the kernels existed for this duration, it would be unnecessarily time-consuming to evaluate equation 2.306 for every kernel of the mixture, since the mixture components are known to grow into the hundreds for Cartesian coordinates (Aristoff et al., 2014). Nevertheless it is also not possible to use the same *Cartesian* form of the process noise for every kernel, since the transformation from orbital elements (OE) to Cartesian coordinates is state vector dependent.

The solution to this dichotomy lies in the usage of the moment-preserving merge given by equations 5.10 to 5.12. The fully merged Gaussian can be defined independently of the number of mixture kernels and provides sufficient continuity to be suitable for the density uncertainty computations. Consequently, a GMM-based filter for OD can be designed which computes the process noise only once (using the merged Gaussian) and then performs the conversion to GCRF coordinates separately for each kernel:

$$\begin{array}{ccc} \text{Fully merged Gaussian} & \text{Kernel state} \boldsymbol{\mu}_{i,k} & \text{Kernel process noise} \\ \mathbf{Q}_{k_{OE}} & \longrightarrow & \mathbf{Q}_{k,i_{GCRF}} \end{array} \tag{5.17}$$

Furthermore it should be noted that due to the non-sequential nature of equation 2.306 there is no contribution of density uncertainty in the sigma points. Hence the splitting recommendation is based solely on the process-noise free covariance of the kernels and their experienced nonlinearity. This might sound like a deficiency at first, but for the

5 Alspach and Sorenson used the term "plant noise" for process noise

case of density uncertainty it is in fact a desirable property, since no contribution of the process noise actually enters the linear or UT-based covariance propagation. Instead, the propagation of the orbital uncertainty is an intrinsic part of the process noise matrix, which neither uses STMs nor unscented transformation for the propagation but models their time-dependence via equations 2.178 and 2.179.

Conversely the addition of a constantly growing density uncertainty process noise matrix to each kernel after the covariance propagation also gives rise to the situation where the density uncertainty becomes sufficiently large for the probability density function to show ragged outer contours due to an increasing overlap of the density uncertainty Gaussians. At some point it may therefore become necessary to split $\mathbf{Q}_k$ into a Gaussian mixture of its own, since otherwise the conversion from orbital elements to Cartesian coordinates cannot be truthfully performed via local linearization or unscented transformation. This case is discussed further in section 5.4.

Last but not least, special attention needs to be paid to the covariance propagation including density uncertainty. As reasoned earlier, the classical EKF requires any prior uncertainty to be subtracted from the covariance prior propagation. Consider the case where a kernel covariance is propagated using either equation 2.300 or eqs. 2.302/2.303. Also assume that the propagation takes place in orbital elements, so that each kernel receives the same process noise contribution $\mathbf{Q}_{k_{OE}}$. As long as there is no kernel split, the kernel covariance could be fruitfully propagated using these equations. With atmospheric density uncertainty included, component merging is uncritical as long as both kernels which are part of a merge contain the same process noise contribution[6]. However, once a split takes place, the process fails as the attempt to subtract the covariance contribution from the reduced covariance after the split may render the result indefinite. The situation gets even more complicated in Cartesian coordinates, since then every kernel contains a different process noise contribution, which would need to be removed at the beginning of the next iteration.

Consequently, subtracting prior contributions is not feasible for GMM-based filters. Instead the technique already used for the UKF must be extended to the Gaussian mixture case. It is therefore required to save the Gaussian mixture parameters of each kernel after the covariance propagation and prior to the addition of the process noise. If no measurement update is available, this state of the GMM is then restored in the next iteration, such that the merging and splitting operations can be executed prior to the next kernel filter cycles.

6 This can be quickly verified by setting $\mathbf{P}_i \to \mathbf{P}_i + \mathbf{Q}$ in equation 5.12, which results in an additional term of the form $\frac{1}{w_m}\sum_{i=1}^{N} w_i \mathbf{Q}$. Since $\mathbf{Q}$ is considered equal for both kernels, it can be taken out of the sum, which shows that it is possible to subtract the prior process noise to make the resulting merged kernel free of any process noise contribution

The requirements discussed above have led to the filter design summarized in figure 5.2.

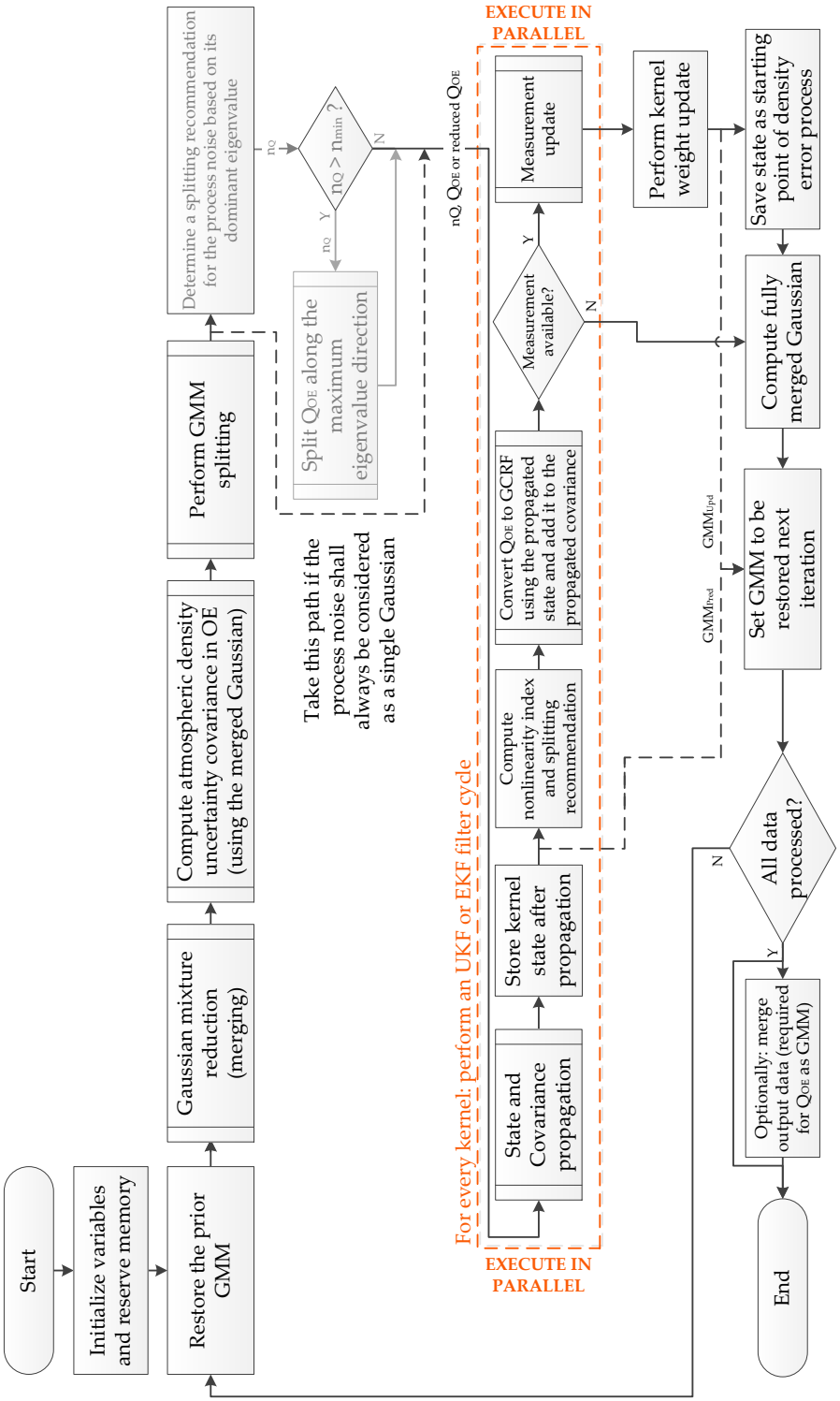


Figure 5.2: Flowchart of the GMM-based filter for orbit determination including combined atmospheric density uncertainty consideration. The design meets all requirements discussed in section 5.3

5.4 PROCESS NOISE AS SECONDARY GAUSSIAN MIXTURE

Non-Gaussian uncertainties are best identified in the radial/transverse plane in Cartesian coordinates (Alfriend and Park, 2016), where each evaluation of equation 2.306 (after subsequent conversion to GCRF coordinates via Jacobians or unscented transformation) corresponds to an ellipse with the principal direction in the transverse direction of the kernel.

If density uncertainty is the dominant driver of the uncertainty growth, then the situation may arise where the principal axes of the ellipses become sufficiently long to deviate from the orbital path. This can be imagined as a tangent starting to deviate from a curve with a large radius of curvature. Density uncertainty is developed in orbital elements, where nonlinearities arise much later than in Cartesian coordinates. As such the Gaussian assumption in orbital elements is valid for larger uncertainties than in Cartesian coordinates. Consequently it might be the case that after the transformation to GCRF the resulting Gaussian is no longer realistic.

An example of this case is the combined density uncertainty covariance given in equation 5.18 (classical/equinoctial orbital elements using mean motion as size-variable):

$$Q = \begin{pmatrix} 1.097744633925843 \cdot 10^{-15} & 0 & 0 & 0 & 0 & -5.568858527881267 \cdot 10^{-10} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -5.568858527881267 \cdot 10^{-10} & 0 & 0 & 0 & 0 & 3.001800530676636 \cdot 10^{-4} \end{pmatrix} \tag{5.18}$$

Transforming this matrix to GCRF coordinates via Jacobians or UT results in a single Gaussian which is not representative of the actual uncertainty in Cartesian coordinates. Instead it is necessary to perform the covariance conversion via a Gaussian mixture of its own.

For this purpose the covariance first needs to be split into a Gaussian mixture in orbital elements. The splitting needs to be performed in the direction of the eigenvector with the maximum associated eigenvalue, which is equal to the eigenvector with the maximum sixth component in absolute magnitude (Aristoff et al., 2014). Obviously this choice can also be verified using Vivek's nonlinearity index, which takes both the size of the covariance and the nonlinearity of the transformation into account, however computing the index would be an unnecessary overhead to the transformation since the optimum splitting direction is known a priori. Once the split is performed, each component can be converted to Cartesian coordinates using Jacobians or UT, since after the split the kernel covariance is assumed to be small enough for both techniques to be valid.

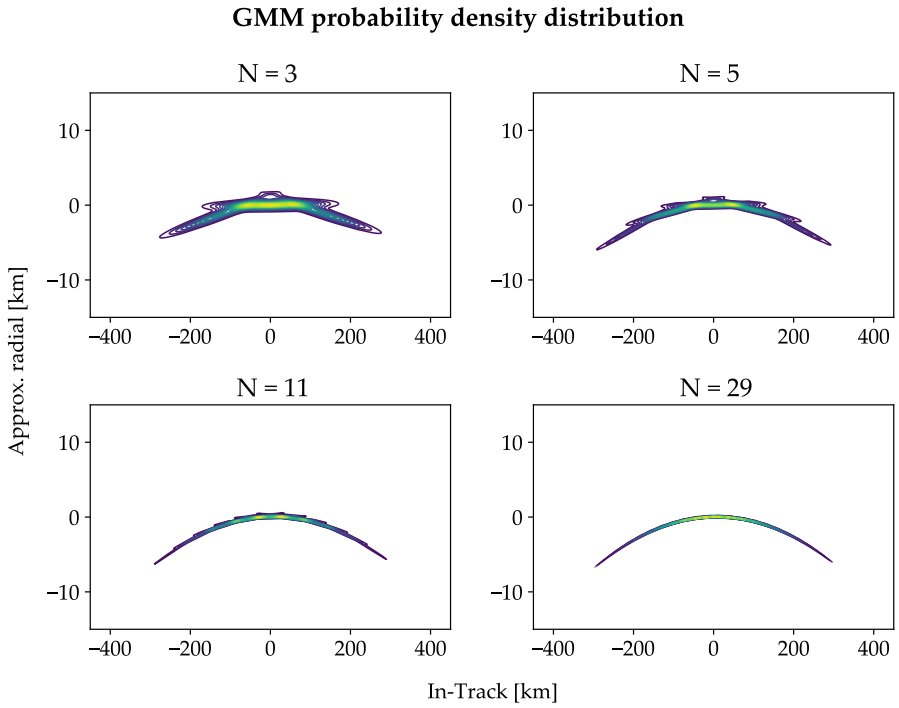


Figure 5.3: Conversion of the covariance given in eq. 5.18 to Cartesian coordinates via Gaussian mixtures of $N = 3, 5, 11$ and 29 components. The Cartesian pdfs are shown in the NTW-frame[7] of the merged mean

Figure 5.3 shows the result of this process for the matrix given in equation 5.18 when performing splits with $N = 3, 5, 11$ and 29 components. It can be clearly seen how for small numbers of N, the individual Gaussian kernels act as tangents to the orbital path. Once enough kernels are contained in the mixture, the ellipses become small enough to recreate the orbital curvature.

In the framework of the GMM filter the conversion steps are executed at different points in time, since the propagated kernel mean is only available within a kernel filter cycle, whereas the Gaussian mixture in orbital elements needs to be computed prior to the filter cycles for it to be made available to all kernels. Modeling the process noise as a secondary Gaussian mixture therefore works as follows: after the computation of equation 2.306 first a process noise splitting recommendation n_Q is derived based on the magnitude of its dominating eigenvalue, which can be quickly estimated using the power method (von Mises and Pollaczek-Geiringer, 1929) since the uncertainty in the in-track direction by far dominates the other directions. If n_Q is larger than the minimum splitting number supported by the underlying splitting library, the result of equation 2.306 is subsequently split

7 The NTW-frame is similar to the Radial/Transverse/Normal frame, however the T-direction is in fact the in-track direction and not the transverse, at the expense of the N-direction only being approximately radial. The exact definition of the frame may be found in Vallado (2013, p. 157)

using a zero-vector as the mean. This process noise Gaussian mixture (still in orbital elements) is then passed to the kernels filter cycles to perform the fusion process.

For each kernel the filter cycle starts with the state and covariance propagation. The result of this propagation phase is first backed up prior to the GMM fusion. This is because the predicted (process noise-free) GMM is needed for initialization purposes at the next time-step if no measurement update is performed. After the propagation phase the two Gaussian mixtures are fused.

The fusion of two Gaussian mixtures has first been discussed in the original work by Alspach and Sorenson (1972). The reference also provides the mathematical fusion equations. Put into simple words, the process can be summarized as follows: for each kernel in the first Gaussian mixture n_Q new kernels are spawned. This results in a total of $N = n_{prior} \cdot n_Q$ kernels. The parameters of the resulting GMM are based on the combinations of the weights, means and covariances of the original two GMMs via the following rules: the fused kernel weights are computed as the product of the original weights ($w_{fused} = w_i \cdot w_j$). The fused means and covariances are constructed via summation ($\boldsymbol{\mu}_{fused} = \boldsymbol{\mu}_i + \boldsymbol{\mu}_j$ and $\mathbf{P}_{fused} = \mathbf{P}_i + \mathbf{Q}_j$).

The addition of the means and covariances needs to take place in the same coordinate frame. Since the process noise Gaussian mixture is created in orbital elements, the predicted Cartesian state is first converted to orbital elements as well. Then the mean vector of the j^{th} process noise kernel is added. This fused state vector is then used to convert the reduced process noise of the j^{th} process noise kernel from orbital elements to Cartesian coordinates. After this conversion the fused process noise is obtained via direct addition. Finally, the fused kernel state vector is converted to Cartesian coordinates to complete the fusion process.

Note that the fusion of two Gaussian mixtures results in a new GMM with a significantly increased kernel number. If the original GMM had 200 kernels and the process noise sum contained 10 kernels, then the resulting mixture would consist of 2000 Gaussian components. Therefore caution is required for implementation. In the proposed layout (figure 5.2), the kernels are spawned within the parallel zone after propagation. The fusion step is executed prior to a possible measurement update, which would need to be executed for every kernel of the fused mixture. Consequently both tasks are put into a single loop with n_Q iterations. Each iteration ends after the GMM fusion if no measurement update is available. For this to work in parallel, the memory for the resulting Gaussian mixture must be shared between the threads and each kernel must know where in the shared memory area it is allowed to write.

The implementation in the Airbus tool SPOOK uses OpenMP® to facilitate the parallelization, memory management and thread handling. Since performing GMM fusion and measurement updates are computationally inexpensive tasks, there is no significant performance loss due to the kernel spawning. Also, if a measurement update takes place, the resulting pdf can be expressed with a significantly lower number of mixture components after the update, such that the next call to the merging routine reduces the mixture to a manageable number of kernels again. If no measurement update is performed, the predicted GMM prior to the addition of the process noise can be restored in the next overall filter cycle, as explained in section 5.3.

Therefore in practice the computationally expensive propagation step is not needed to be executed on an increased number of kernels with respect to the case of treating the process noise as a single Gaussian. The output GMMs on the other hand contain the fused mixtures at any time, such that an optional merging stage has been added as the last step of the design in figure 5.2. Merging the output GMMs is possible without significantly changing the pdf results, since some kernel weights become very small after fusion such that they can be replaced by fewer components (Alspach and Sorenson, 1972). Also potential covariance overlap allows for some of the kernels to be merged. In effect the output merge is often able to compress the kernel numbers by up to 80%. This significantly reduces the size of the simulation output files containing the time-evolution of the fused GMM pdf and the orbit estimation errors, which eases the analysis of the results.

These benefits of posterior mixture reduction however also come with a significant cost in runtime. As mentioned in section 5.2.3, merging is computationally expensive. Even using the KL-distance metric, which can be evaluated orders of magnitude faster than Salmond's and William's criteria, merging a Gaussian mixture of more than 2000 components down according to a distance threshold is very time-consuming, since $\sum_{k=1}^{N-1} k = \frac{1}{2}(N^2 - N)$, i.e. approximately two million distance computations are required to be performed at every time-step where such high kernel numbers were required in the OD.

The output merging stage may be implemented using parallelization for the distance computation and minimum distance search. This tremendously speeds up the process, however once the kernel numbers increase above a critical threshold, say 500, the final merging process notably slows down the overall performance and can even take longer than the orbit determination itself for very large kernel counts ($\gtrsim 1000$). The only way to reduce the required runtime is hence to properly tune the merging and splitting thresholds, such that the components of the original sum do not grow too fast but are just about large enough to properly map the nonlinearity encountered in the propagation phases.

5.5 SIMULATION RESULTS

To demonstrate all aspects discussed in the previous sections, two simulation scenarios which only differ by the accuracy of the radar measurements were created and are presented hereafter. Since unscented transformation allows to directly evaluate equation 5.8 on the propagated sigma-points, an UKF was chosen as the baseline of the adaptive Gaussian mixture filter. An EKF would also have been possible, but its linear covariance propagation results in the need to perform an additional UT to compute the nonlinearity index, which slows down the OD process.

In both scenarios range-only measurements of a single simulated radar observer are used to initialize the Adaptive Gaussian Mixture Unscented Kalman Filter (AGMUKF) with two tracks of measurement data (each of approximately 7 min duration). After filter initialization, subsequent measurement updates are suppressed for 10 days, resulting in an uncertainty propagation phase long enough for the developments to be validated.

In the first scenario the one sigma standard deviation of the radar measurements is set to 800 m. This measurement uncertainty results in sufficient initial semi-major axis uncertainty after filter initialization for it to dominate the along-track uncertainty growth over the entire simulation time. Scenario 1 therefore allows to demonstrate the positive effects of density uncertainty consideration even for scenarios where it is not the dominating driver of the overall uncertainty growth. For the second scenario the one sigma measurement accuracy is set to a tenth of the value of scenario 1, i.e. 80 m. In this case, density uncertainty is initially inferior but, due to its growth with t^3, eventually dominates over the uncertainty growth due to the initial semi-major axis uncertainty. Scenario 2 serves to demonstrate the importance of treating the process noise as a Gaussian mixture of its own, if it is dominating the along-track uncertainty growth. The states and covariances of both scenarios after the last measurement update prior to the 10 days propagation phase are given in Schiemenz et al. (2020d, appendix B). All cases were computed using a fictive low Earth orbiting satellite of the following characteristic parameters:

- Semi-major axis: 6887.0 km
- Eccentricity: $1.476 \cdot 10^{-4}$
- Ballistic coefficient: $\frac{A \cdot c_D}{m} = 0.06 \, \frac{m^2}{kg}$

In each Monte-Carlo iteration the true density is calculated using stochastic solar flux and magnetic index inputs. The output of the semi-empirical density model is then used to drive the Ornstein-Uhlenbeck process simulating the model uncertainty. The true density and density uncertainty computation processes are characterized according to the following assumptions on the input and model uncertainty:

- Density model: DTM-2013 (Bruinsma, 2015)
- Solar flux input uncertainty: white noise error process centered on published F10.7 data, characterized by a double-sided power spectral density of 130000 sfu^2s and daily sampling
- Magnetic index uncertainty: white noise error process centered on published global geomagnetic amplitude data, characterized by a double-sided power spectral density of 2700 AP $units^2s$ and three hour sampling cadence[8]
- 10% relative model uncertainty
- OUP rate of mean reversion: $\theta = 10^{-6}$

The Kullback-Leibler distance for kernel merging is set to $5.0 \cdot 10^{-4}$. The translation of the Vittaldev nonlinearity index into a splitting recommendation is performed via equation 5.9.

For all scenarios 200 Monte-Carlo iterations are computed to test the consistency of the pdf with the MC samples. This agreement may be evaluated using either the log likelihood (LL) metric (Vittaldev and Russell, 2016) or the likelihood agreement measure (LAM) (DeMars et al., 2013). Both statistics have been computed for each scenario, however the LL turned out to be more convenient, as the non-logarithmic nature of the LAM makes it difficult to be visualized over the entire simulation time. The following pdf plots have been chosen to facilitate the discussion of the effect of process noise consideration when performing orbit determination using adaptive Gaussian mixture filters. Therefore they depict the state of the pdf at selected instants of time. To ease the analysis of the simulation results, also videos of each presented case showing the entire course of the simulation period have been created. They are published as supplementary material to Schiemenz et al. (2020d).

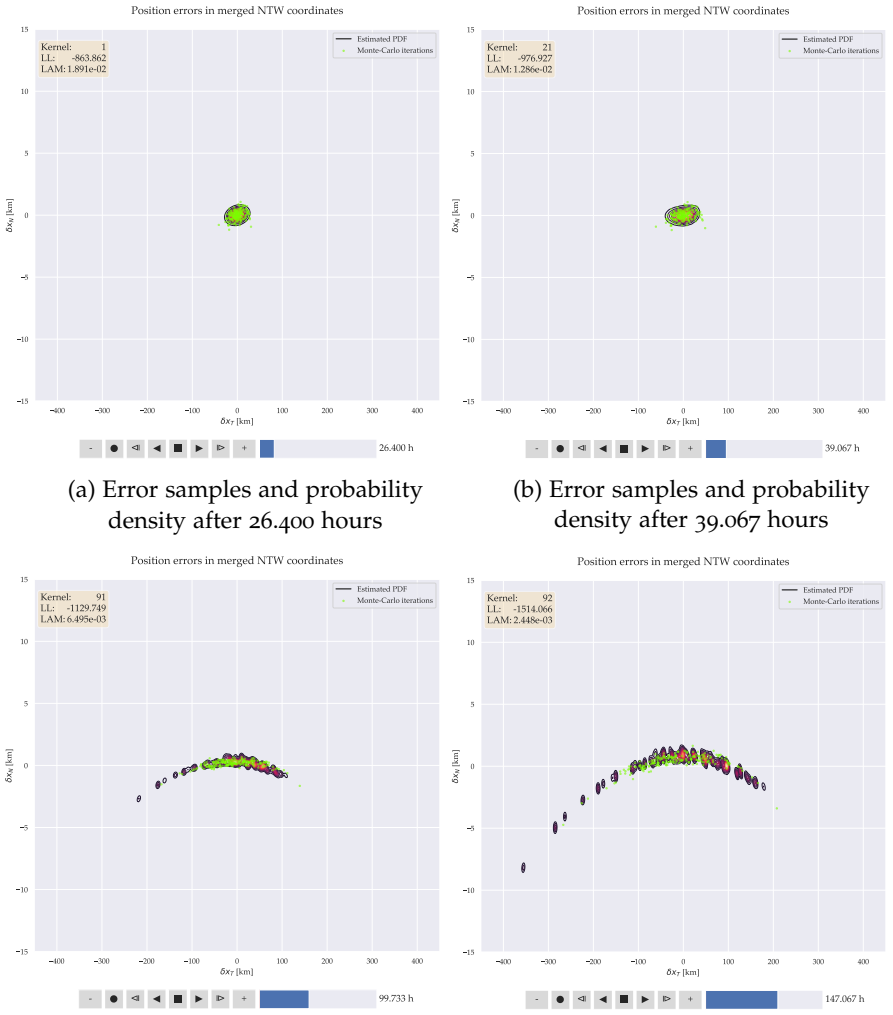
Scenario 1

Classical computation (density uncertainty not accounted for via process noise)

Figures 5.4, 5.5 and 5.6 depict the results obtained with the AGMUKF if process noise is neglected. As designed, the filter starts with a single Gaussian (figure 5.4a). At roughly 37 h of simulation time the assumption of Gaussianity is abandoned and the first kernel splits take place (figure 5.4b). Shortly after the filter reaches a maximum in the kernel number of approximately 100 kernels. The effect of kernel splitting and merging can be clearly identified by the noise-like behavior of the kernel numbers in figure 5.6. Figure 5.4 reveals that with progressing time the pdf starts to become multi-modal (it develops gaps of increasing magnitude). The gaps arise due to the highly nonlinear space-environment, which results in the kernels

8 Recall that DTM-2013 uses the planetary kp index as magnetic parameter, however the theory of computing the uncertainty in the kp index develops from an uncertainty in the planetary magnetic amplitude (cf. section 2.4.1)

to experience different propagation paths. Since density uncertainty is not considered, the gaps are not closed, which makes it difficult to infer the mode of the pdf (in fact it shows multiple modes) and its dimensions. Density uncertainty is not the dominant driver of along-track uncertainty growth in scenario 1, which can be verified by comparing the distribution of the MC samples with the spread of the pdf blobs.



(a) Error samples and probability density after 26.400 hours

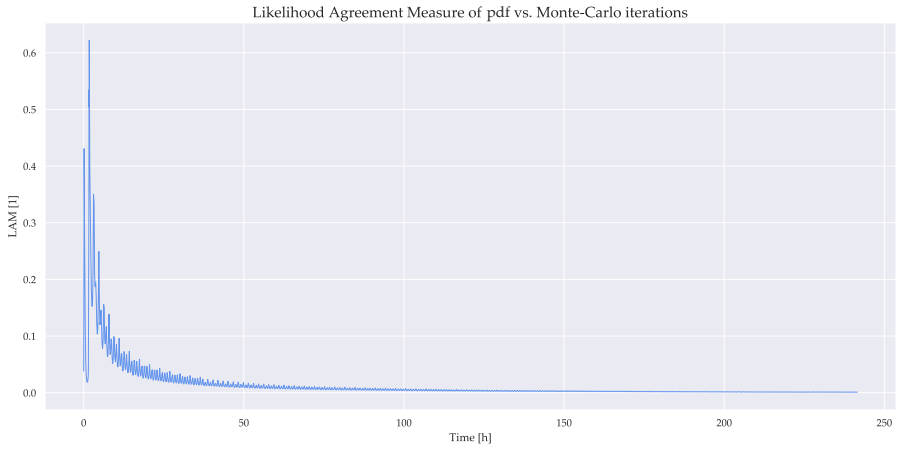
(b) Error samples and probability density after 39.067 hours

(c) Error samples and probability density after 99.733 hours

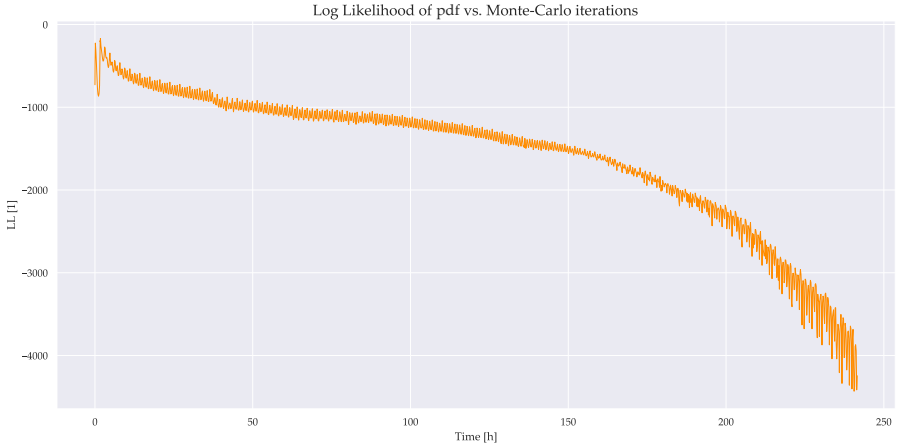
(d) Error samples and probability density after 147.067 hours

Figure 5.4: Evolution of estimated pdf and Monte-Carlo samples for the classical case of neglected process noise

The LL, LAM and kernel numbers are depicted in figures 5.5 and 5.6. As mentioned earlier, the LAM spans over multiple orders of magnitude and therefore cannot be visualized properly (cf. figure 5.5a). It is probably better suited for relative comparisons. The LL on the other hand reveals a significant reduction in the pdf agreement with the MC samples once the gaps become too broad.



(a) Likelihood agreement measure of AGMUKF with neglected process noise



(b) Log likelihood of AGMUKF with neglected process noise

Figure 5.5: LAM and LL of scenario 1 using the classical approach without process noise consideration

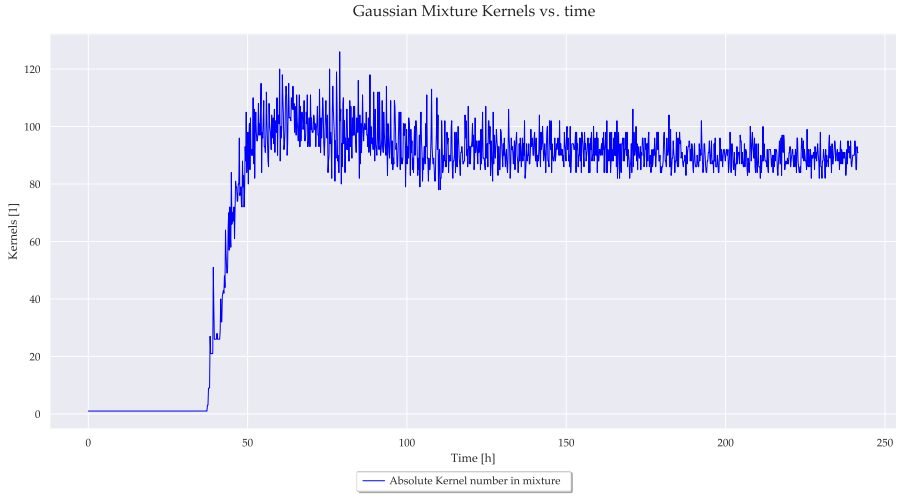
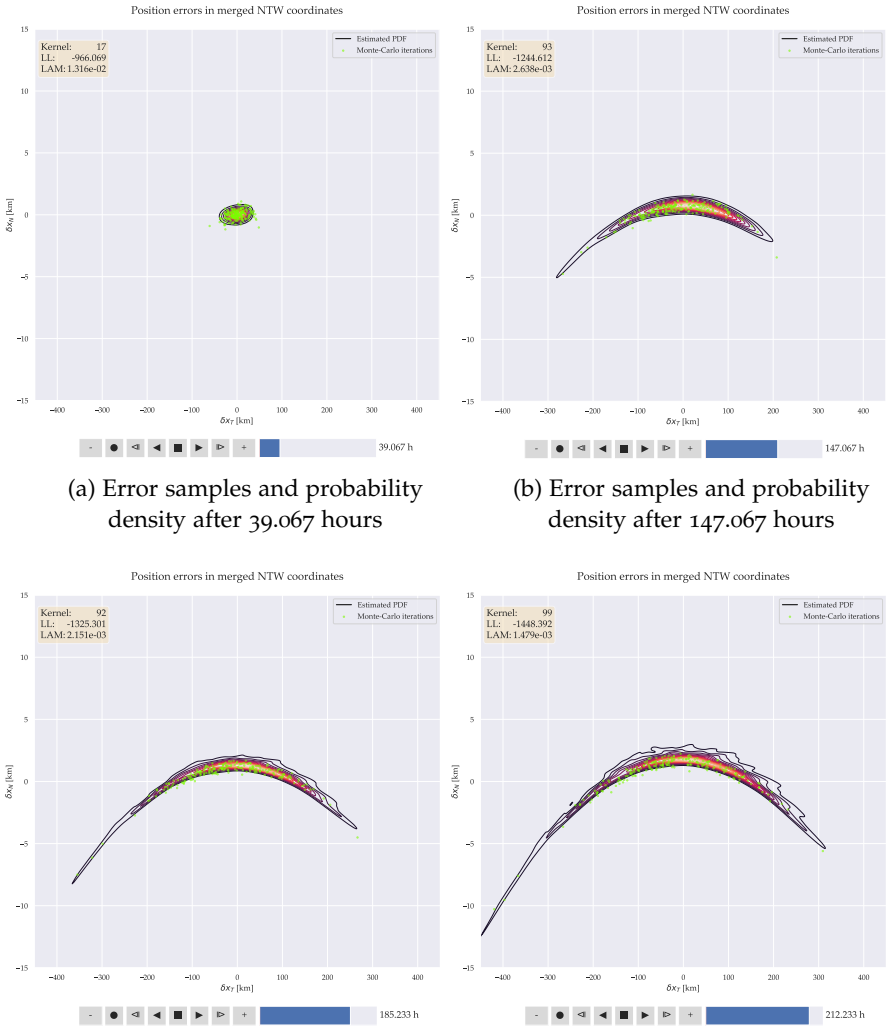


Figure 5.6: Number of Gaussian kernels in GMM versus simulation time

Density uncertainty consideration using a single Gaussian for the process noise

Once combined density uncertainty process noise is integrated into the kernel covariance propagations, the resulting pdf changes significantly as demonstrated in figures 5.7 and 5.8. While initially the pdf and LL remain similar to figures 5.4b and 5.5b, the realistic growth of each kernel's covariance is able to result in a gap-free, single mode estimated probability density function. Consequently, the MC samples match the estimated pdf better, which leads to a substantial increase of the log likelihood computations.

The kernel numbers are comparable to the case without process noise, however the initial growth is slightly slower than without process noise consideration (in contrast to figure 5.6 there is no kernel overshoot in figure 5.8b).



(a) Error samples and probability density after 39.067 hours

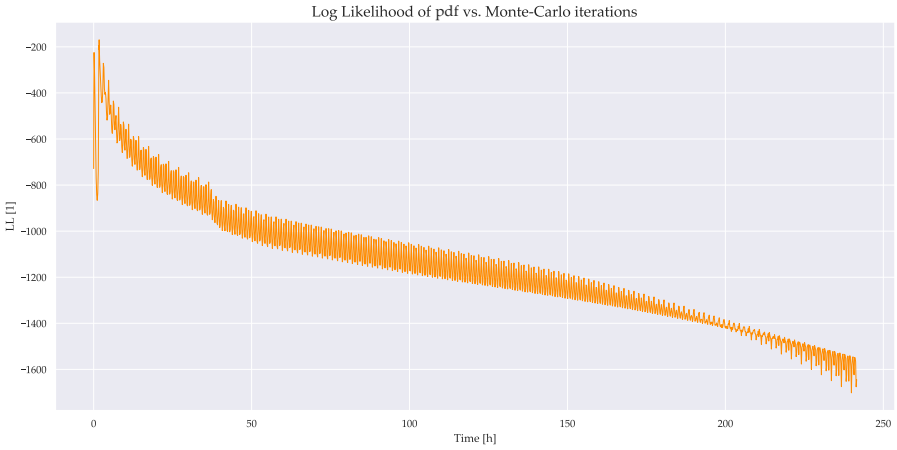
(b) Error samples and probability density after 147.067 hours

(c) Error samples and probability density after 185.233 hours

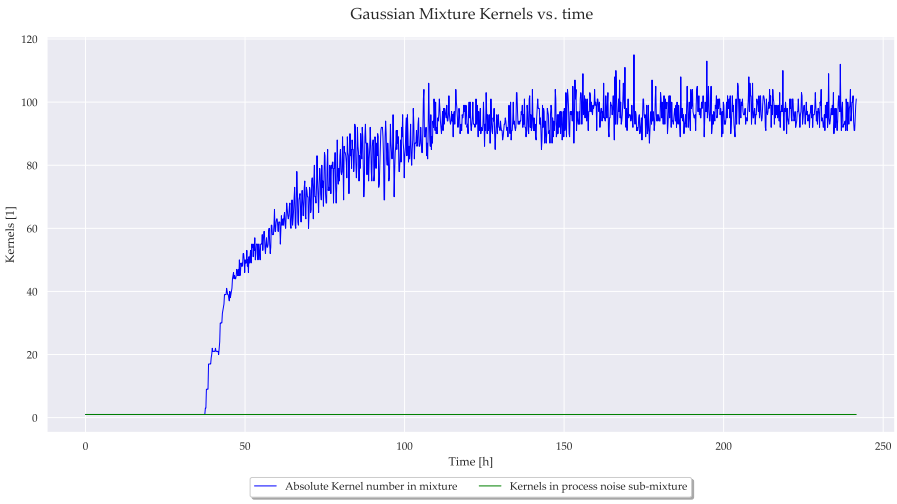
(d) Error samples and probability density after 212.233 hours

Figure 5.7: Evolution of estimated pdf and Monte-Carlo samples with process noise consideration as a single Gaussian

Treating the process noise as a single Gaussian seems appropriate until approximately 160 h of simulation time. At this point, two effects begin to become apparent. First, the pdf begins to develop the anticipated ragged outer contours (cf. figure 5.7c). Second, it can be seen that the curvature of the MC samples can no longer be reproduced on the inside of the pdf, which is due to the modeling of the process noise as single Gaussians (cf. figure 5.7d).



(a) Log likelihood of AGMUKF with process noise modeled as a Gaussian



(b) Number of Gaussian kernels in GMM versus simulation time

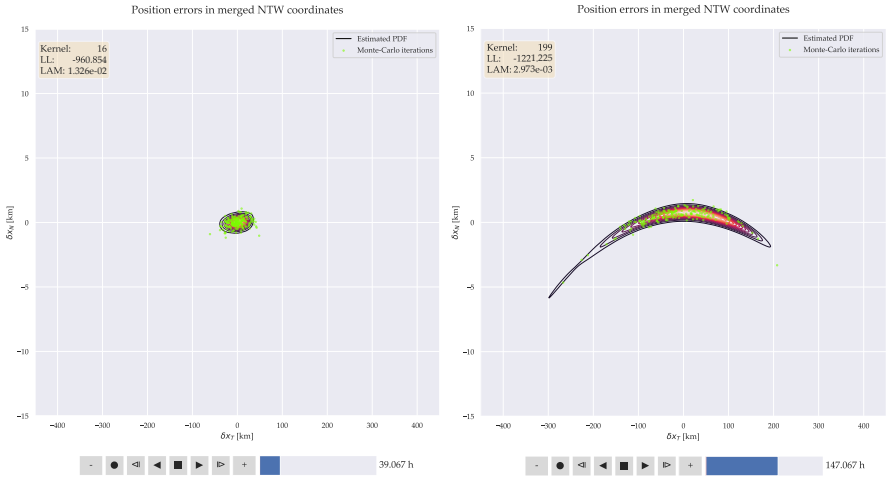
Figure 5.8: LL and kernel number of scenario 1 when modeling the process noise as single Gaussian

Density uncertainty consideration using a secondary GMM for the process noise if required

To further improve the agreement of the pdf with the MC samples, the restriction of treating the process noise as a single Gaussian is removed. As a result, the filter increases the kernels in the secondary process noise Gaussian mixture with progressing simulation time to keep the

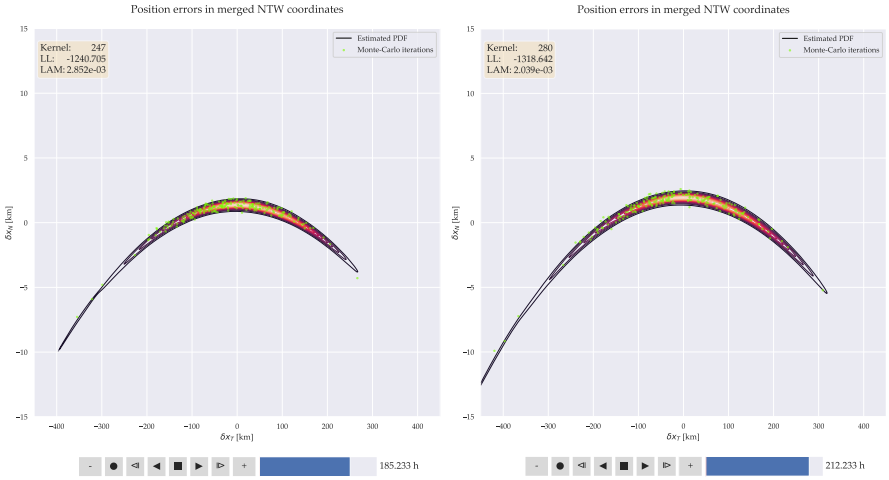
Cartesian representation of the uncertainty in agreement with the original Gaussian in orbital elements. The fusion of the two Gaussian mixtures results in up to ~1500 kernels, however, as explained in section 5.4, the output merging step is able to reduce the final kernel numbers to levels below 300.

Figures 5.9 and 5.10 clearly demonstrate that treating the process noise as a secondary Gaussian mixture is the single means required to avoid ragged outer pdf contours while at the same time correctly representing the curvature of the MC samples. Using this approach the pdf estimates remain realistic over the entire 10 days of propagation time.



(a) Error samples and probability density after 39.067 hours

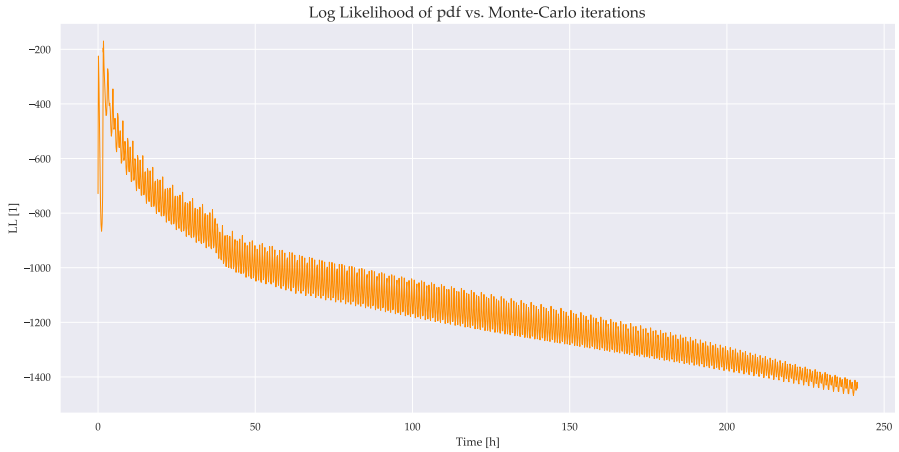
(b) Error samples and probability density after 147.067 hours



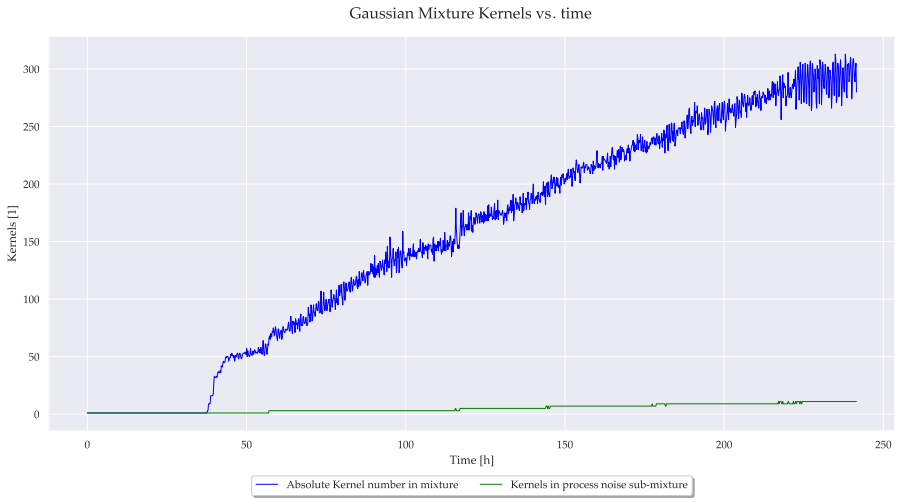
(c) Error samples and probability density after 185.233 hours

(d) Error samples and probability density after 212.233 hours

Figure 5.9: Evolution of estimated pdf and Monte-Carlo samples with process noise consideration as a secondary Gaussian mixture



(a) Log likelihood of AGMUKF with process noise modeled as a secondary Gaussian mixture



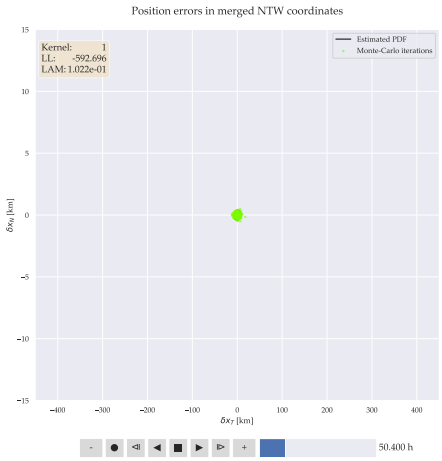
(b) Number of Gaussian kernels in GMM (after a posterior merging step) versus simulation time

Figure 5.10: LL and kernel number of scenario 1 when modeling the process noise as a secondary Gaussian mixture

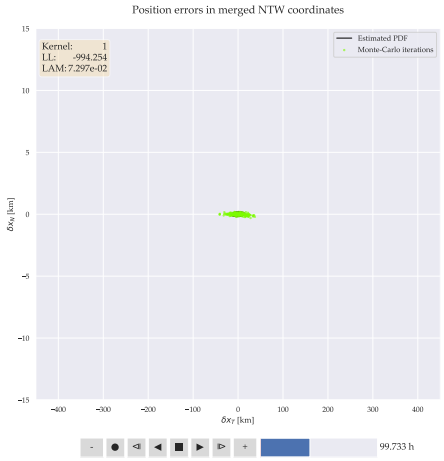
Scenario 2

Next the results of scenario 2, for which density uncertainty is initially inferior but then evolves to dominate the along-track uncertainty growth, are presented. The scenario is computed once without process noise consideration and once with the process noise modeled as a Gaussian mixture.

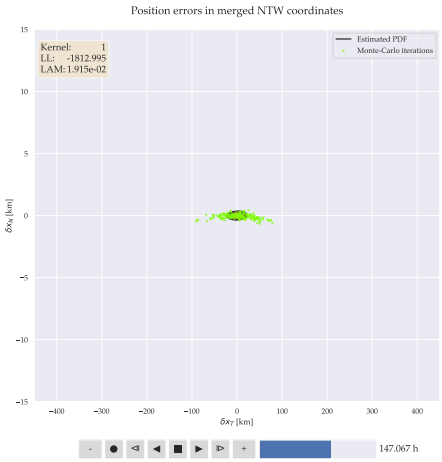
Figure 5.11a shows that the neglect of process noise is appropriate for the first 50 h of simulation time. After this point the growth of the pdf is not realistic anymore, since density uncertainty is dictating the along-track uncertainty growth (figures 5.11b to 5.11d). The need to model the process noise as a Gaussian mixture of its own is evident from the curvature of the MC samples. Since process noise is neglected, the filter never performs a splitting operation and operates with the assumption of Gaussianity over the entire simulation period.



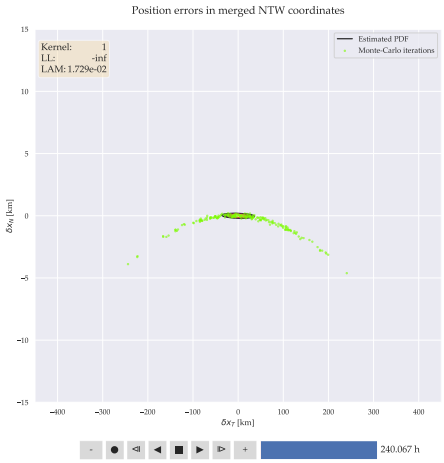
(a) Error samples and probability density after 50.400 hours



(b) Error samples and probability density after 99.733 hours



(c) Error samples and probability density after 147.067 hours

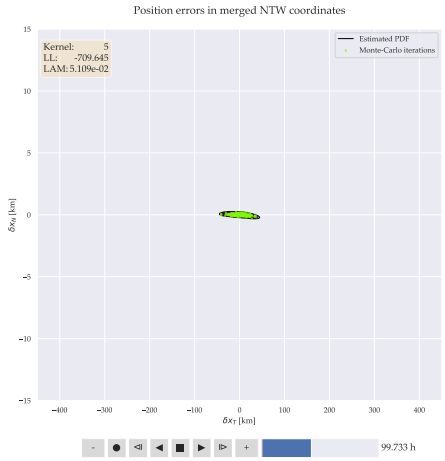


(d) Error samples and probability density after 240.067 hours

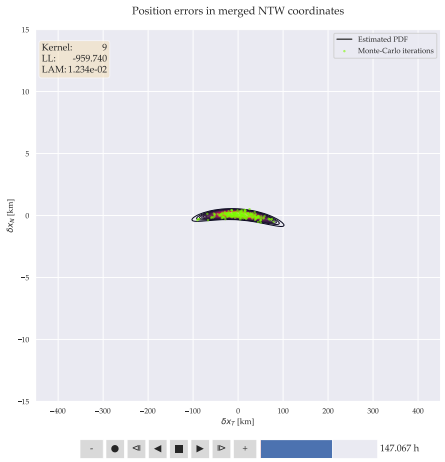
Figure 5.11: Evolution of estimated pdf and Monte-Carlo samples for the classical case of neglected process noise

Using the combined density uncertainty consideration of chapter 2 with subsequent transformation into a Gaussian mixture, realistic pdf estimates can be obtained (figures 5.12a to 5.12c). The distribution of the MC samples is identical to figure 5.11. The Gaussian mixture is able to properly model the transformation from orbital elements to Cartesian coordinates. Up to 13 kernels are used for the process noise Gaussian mixture, however the output merging stage starts to merge two process noise kernels towards the end of the simulation (cf. figure 5.12d).

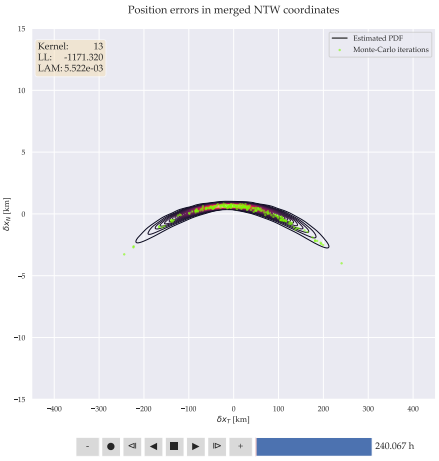
The results of scenario 2 clearly demonstrate the need for density uncertainty consideration of low Earth orbiting satellites. The effect increases with growing ballistic coefficients and decreasing orbital altitudes. Smaller ballistic coefficients and higher orbital altitudes on the other hand decrease its impact.



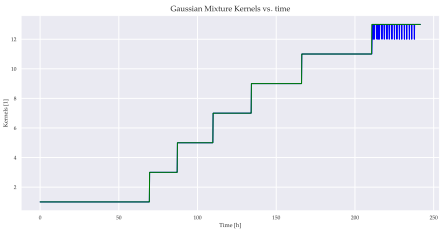
(a) Error samples and probability density after 99.733 hours



(b) Error samples and probability density after 147.067 hours



(c) Error samples and probability density after 240.067 hours



(d) Number of Gaussian kernels in GMM (blue: absolute after posterior merging, green: process noise)

Figure 5.12: Evolution of estimated pdf and Monte-Carlo samples when modeling density uncertainty as a Gaussian Mixture

Conclusion

The research presented in this chapter resulted in a highly adaptive Gaussian mixture filter framework which starts the orbit determination process using a classical UKF or EKF and splits the Gaussians in the state covariance and the process noise whenever the need to do so is detected. This approach is of great advantage for practical applications, since the algorithm runtime remains similar to the classical Kalman filters until the Gaussian uncertainty assumption is deemed inadequate. Therefore, runtime increases only occur if there is an actual need for higher order methods.

Numerical Monte-Carlo simulations have been conducted to validate the developments and find the answers to the research questions posed in section 5.1. Cartesian coordinates were used to exacerbate the effect of nonlinearities and to demonstrate that the curse of dimensionality can be counteracted by splitting in up to two dimensions per step, given that the splitting directions are evaluated dynamically using a nonlinearity index.

As demonstrated, the neglect of process noise may cause undesired gaps in the pdf estimates. The inclusion of process noise, also when it is not the dominating driver of the overall uncertainty growth, leads to an increased size of the kernel covariances. Since the process noise is propagated analytically using its own theory, the increased covariances do not enter subsequent propagation cycles, thereby avoiding situations in which the Gaussian assumption for each kernel may be violated. Instead, the increased covariances help to either bridge gaps in the pdf or they result in covariance overlap for closely spaced kernels, which then allows the merging process to reduce the overall kernel number in the mixture further than without process noise.

The numerical simulations also confirmed the occurrence of ragged outer pdf contours when modeling the process noise as a single Gaussian for extended propagation durations. In this regard the abilities of treating the process noise as a secondary Gaussian mixture and subsequent Gaussian mixture fusion have proven essential to obtaining realistic pdf contours over 10 days of simulation time. This also applies to small initial uncertainties, since in this case the curvature of the pdf is dominated by atmospheric density uncertainty, which is not captured at all if process noise is neglected and underrepresented if it is modeled as a single Gaussian. In the latter case the curvature of the pdf depends on the location of the kernel means, which may be too closely spaced in case of small initial radial uncertainties (and therefore few mixture components) for the tangential kernel covariance increase to result in a banana-shape of the pdf.

A subtask of this research was to study if process noise can be safely neglected when performing OD/uncertainty propagation using Gaussian mixtures, as claimed in Horwood and Poore (2011). Judging by the extensive simulation results and the preceding discussion, Horwood's claim has to be contradicted. Only for the UCT problem with significant initial uncertainties in the semi-major axis direction ($\gtrsim$1 km) and RSOs with perigees above ~800 km the neglect of atmospheric density uncertainty is acceptable. This however does not answer if process noise can generally be neglected under these circumstances, as at these altitudes other forces are dominating over atmospheric drag. Their uncertainty must first be quantified and propagated to satellite orbits before the neglect of process noise can be justified.

Finally, a need for increased efficiency in GMM reduction has been identified. When process noise is modeled as a secondary GMM, the total kernel number may increase significantly, such that a post-OD merging process becomes necessary. Despite a high degree of parallelization, this step has shown a certain potential for becoming the runtime bottleneck due to the quadratic dependence of the merging pairs on the kernel number. More advanced reduction algorithms are therefore desirable to reduce also GMMs with over a thousand components to manageable scales without notably decreasing the runtime.

Part IV

INFLUENCE OF FORCE MODEL UNCERTAINTIES ON THE COLLISION PROBABILITY

6

SPACE CATALOG CONJUNCTION ASSESSMENT INCLUDING FORCE MODEL PROCESS NOISE

Finally, the influence of force model uncertainties on the collision probability shall be evaluated. The analysis is intended to permit first statements on the extent to which a catalog collision risk distribution changes by taking physics-based propagation uncertainties into account.

The sensitivity of the collision probability with respect to covariance shape and size changes was examined in section 1.3.4 with the result of a strong dependence on the covariance size if the miss distance is significantly larger than the covariance ellipsoid. The orientational dependence was found to be of relevance only in case of high covariance aspect ratios. Literature also reports the assumption of Gaussianity to have little impact on high interest events (HIEs) collision probabilities greater than 10^{-4}(Ghrist and Plakalovic, 2012). Based on this information, the following framework has been chosen for this study:

- Perturbation theory: SP and GP
- Assumption of Gaussian uncertainty volumes
- Linear (two-dimensional) and nonlinear (three-dimensional) P_c computation based on the position covariance
- Assumption of spherical RSO shapes

The SP analysis is performed using the restricted SP catalog of February 25th, 2020 and the developments of chapters 2 and 3. The GP analysis is based on the public TLE catalog of the same day and the developments of section 4.4.

All catalog data was obtained from `https://space-track.org`. The GP catalog snapshot consists of a single TLE file of ~2.5 MB. The SP ephemerides on the other hand require ~5.5 GB, since they contain the state vector solutions of each non-military object in time-steps of some seconds over multiple days. The files are delivered in the itc-format, which is also known as 18 SPCS SP ephemeris format. A file format description is available at SpaceTrack.

The itc-format contains neither covariance information nor additional information on the RSOs. Truthful covariance propagation in special perturbations however also requires information on the models used, as well as the ballistic and solar radiation pressure coefficients. For this purpose also the SP vectors of February 25th, 2020 were obtained from SpaceTrack. The SP vectors are delivered in the VCM file format, however any covariance information is removed to avoid conclusions being drawn about the capabilities of the SSN. An example of a special perturbations vector file in the VCM format is given in

```
 1 <> SP VECTOR/COVARIANCE MESSAGE - V2.0
 2 <>
 3 <> MESSAGE TIME (UTC):                                    CENTER:
 4 <> SATELLITE NUMBER:     5                    INT. DES.: 1958-002B
 5 <> COMMON NAME: VANGUARD 1
 6 <> EPOCH TIME (UTC): 2013 336 (02 DEC) 04:01:32.274  EPOCH REV: 94590
 7 <> J2K POS (KM):      4182.85661723    5327.31383408    1975.71259599
 8 <> J2K VEL (KM/S):  -6.426755648471   3.196693759190   3.908474611282
 9 <> ECI POS (KM):      4163.54380568    5340.37985192    1981.22397634
10 <> ECI VEL (KM/S):  -6.442024042506   3.176817979228   3.899534165111
11 <> EFG POS (KM):      1235.21828193   -6658.00193606    1981.22397634
12 <> EFG VEL (KM/S):   6.164703708150   2.624058908291   3.899534165111
13 <> GEOPOTENTIAL: EGM-96 36Z,36T  DRAG: JAC70/MSIS90  LUNAR/SOLAR:  ON
14 <> SOLAR RAD PRESS:  ON  SOLID EARTH TIDES:  ON  IN-TRACK THRUST: OFF
15 <> BALLISTIC COEF (M2/KG):  0.206683E-01 BDOT (M2/KG-S): 0.000000E+00
16 <> SOLAR RAD PRESS COEFF (M2/KG):  0.258457E-01  EDR(W/KG):  0.45E-04
17 <> THRUST ACCEL (M/S2):  0.000000E+00  C.M. OFFSET (M):  0.000000E+00
18 <> SOLAR FLUX: F10: 176  AVERAGE F10: 152  AVERAGE AP:  10.0
19 <> TAI-UTC (S): 35  UT1-UTC (S): -0.06148  UT1 RATE (MS/DAY): -1.100
20 <> POLAR MOT X,Y (ARCSEC):  0.0532  0.2905 IAU 1980 NUTAT:   4 TERMS
21 <> TIME CONST LEAP SECOND TIME (UTC): 2049 365 (31 DEC) 23:59:59.999
22 <> INTEGRATOR MODE: ASW          COORD SYS: J2000  PARTIALS: FAST NUM
23 <> STEP MODE: AUTO  FIXED STEP: OFF  STEP SIZE SELECTION: MANUAL
24 <> INITIAL STEP SIZE (S):   20.000  ERROR CONTROL: 0.100E-13
25 <> VECTOR U,V,W SIGMAS (KM):           0.0124     0.0530     0.0109
26 <> VECTOR UD,VD,WD SIGMAS (KM/S):      0.0001     0.0000     0.0000
27 <> COVARIANCE MATRIX (EQUINOCTIAL ELS): ( 9x 9) WTD RMS:  0.11031E+01
28 <>  0.42255E-11  0.41132E-12  0.76041E-12 -0.99354E-11 -0.89253E-12
29 <>  0.40469E-10 -0.71798E-13 -0.33776E-14  0.30115E-12  0.27939E-14
30 <>  0.12358E-11  0.22524E-12 -0.14392E-11 -0.12571E-13  0.23944E-11
31 <>  0.36651E-13 -0.15150E-12 -0.13586E-12  0.40621E-14  0.59381E-12
32 <>  0.59100E-12 -0.62234E-07  0.35086E-08  0.22947E-06  0.22472E-08
33 <> -0.11983E-07  0.16224E-08  0.22216E-02  0.00000E+00  0.00000E+00
34 <>  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
35 <>  0.00000E+00  0.22312E-06 -0.26106E-07 -0.55914E-06 -0.53576E-08
36 <>  0.46806E-07 -0.61319E-08 -0.53200E-02  0.00000E+00  0.20231E-01
```

Figure 6.1: Example of a special perturbations vector file in the VCM format. Source: Lidtke (2016)

figure 6.1. The line numbers are not part of the actual messages. The files made available to Airbus only contain lines 4, 6, 9-10 and 13-24.

This study assumes that the information provided in the SP vectors also applies to the itc ephemerides, however there is no official confirmation if the perturbation models and coefficients actually match for both SP datasets[1]. While in principle also the vector files contain a state vector solution that could be used for CA (lines 9 and 10 in figure 6.1), a single SP snapshot is not suitable for conjunction assessment, since it requires propagation using the full SP force models, whereas the itc files can be interpolated[2], which reduces the risk of model mismatch and speeds up the computations. For a single RSO there

1 SPOOK reads the models from the vector files and performs the subsequent propagations and collision probability computations using the same algorithms to avoid model mismatch. If a model is detected in the vector files which is not available in SPOOK, the subsequent processing is conducted with the "closest" available one, which is selected based on the model family and age

2 Given that the encounter region lies within the available data

is little difference between propagation and interpolation, however for the entire non-military catalog of nearly 20 000 objects the runtime impact is tremendous.

Any method for the computation of the collision probability of two objects requires information about their shape. ESA's Database and Information System Characterising Objects in Space (DISCOS) is a well-known and trusted source for obtaining RSO properties. Unfortunately, a direct export of the object radii is not available, however a snapshot can be generated by the SDO in a file format compatible with ESA's PROOF tool[3]. The effective hard body radius (HBR) generation logic for the RSO population is based on the following procedure:

- take the diameter of the object – if not available,
- calculate the diameter from the mean cross-sectional area (which is calculated in DISCOS for all objects of different geometries, e.g. box, cylinder, etc. under the assumption of a randomly tumbling object) – if not available,
- calculate the diameter from the radar cross-section (RCS) information in DISCOS – if not available,
- take the data from the PROOF-2009 file – if not available,
- take the size based on the Satellite Situation Report (SSR), which contains the categories small, medium and large for the RCS instead of a numerical value. The diameter is then set to 0.35 m, 1.13 m or 3.57 m, respectively.

A crucial, yet missing piece of information are the RSO epoch covariance matrices. Rough estimates in tabulated form are provided by Flohrer et al. (2008) for TLE data based on the perigee altitude, inclination and orbital eccentricity. No covariance source is available for the itc files, which is why scaled variants of the lookup table in Flohrer et al. (2008) are used for the RTN epoch covariances. The assessment concerning the impact of the force model uncertainty consideration is therefore linked to the (scale of the) initial covariance matrix used. A schematic high-level representation of the simulation setup is shown in figure 6.2. Tasks marked by a dashed orange frame benefit greatly from parallelization and should be implemented accordingly.

The algorithms and techniques that were chosen to perform the computations are discussed in section 6.1 alongside alternative choices. The section also contains the derivation of an extension to Alfano's adjoining tube 3D collision probability algorithm (Alfano, 2006), which was developed during this doctoral research, however not published so far. In contrast to the original algorithm the extension does not require covariance symmetrization to decouple the relative velocity direction from the conjunction coordinate frame, which removes the need for additional frame transformations, simplifies the implementation and speeds up the computations. The results of the force model impact

3 For further information on PROOF, see https://sdup.esoc.esa.int/proof (free Space Debris User Portal account required)

analysis are presented in section 6.2 after a short description of the tool that was developed as part of this thesis to perform CARA.

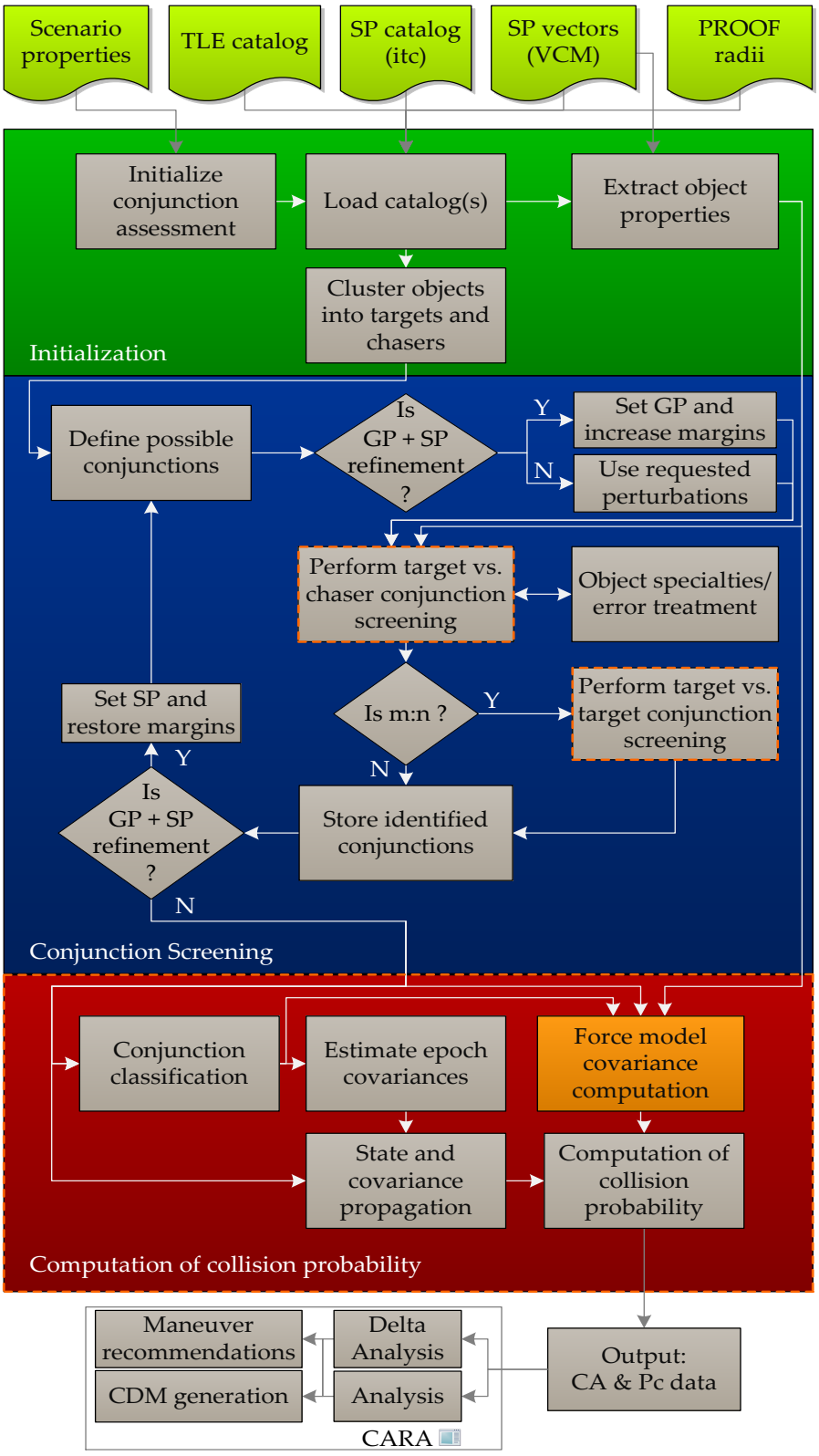


Figure 6.2: High level simulation setup of the space catalog study

6.1 PREREQUISITES

Figure 6.2 identifies the logical flow of the steps required to compute space catalog collision probabilities. The theory and design of the most important building blocks is discussed in sections 6.1.1 and 6.1.2. Covariance information is required only in the third processing stage (red background) after the conjunctions have been identified. Thus, the effort needed for extending existing CA solutions to account for force model uncertainties is minimal.

6.1.1 *Conjunction screening*

Conjunction screening algorithms aim to detect when a safety (or screening) volume[4], which is flown with a protected target RSO, is penetrated by a secondary chaser object (cf. figure 6.3). If all potentially dangerous conjunctions shall be found, then each catalog RSO serves as target and chaser. This is known as the all vs. all or n:n scenario. If only a subset of a catalog is to be tested for conjunctions, the problem reduces to m:n screening ($m < n$).

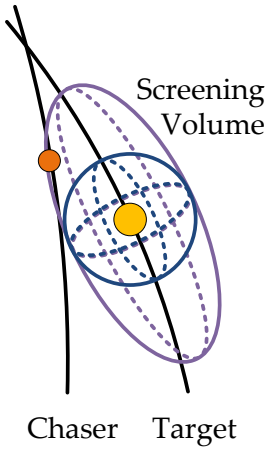


Figure 6.3: Identification of close conjunctions using screening volumes

The identification of possibly dangerous conjunctions is a straightforward task in case of two objects. Simple orbit propagation can be used to determine if the objects come closer to each other than acceptable. For an entire space catalog however this brute-force approach is impractical due to the amount of orbit propagations and distance checks required. Instead, smarter methods are needed to reduce the computational complexity.

Practice-proven conjunction screening algorithms can be divided into filter-based and sieving-based strategies. The filter concept dates back to Hoots et al. (1984) who define two geometrical prefilters (the apogee-perigee filter and an orbital path filter), as well as a time prefilter with the objective of ruling out pairs of objects that cannot come close enough to pose an actual risk. The geometrical prefilters discard pairs of objects based only on the geometry of their orbits, whereas the time filter makes use of known orbital characteristics to determine time ranges in which two objects may come close to each other. Hoots et al. (1984) first formulate the filter equations for two-body Keplarian orbits and then present extensions to account for the secular changes caused by perturbations.

Despite considering secular changes, it was repeatedly observed in the literature that the original filters lack a certain kind of robustness

4 An ellipsoidal shape is common and at times called "warning football" (Alfano, 1994)

(Alarcon Rodriguez et al., 2002; Woodburn et al., 2009; George and Harvey, 2011). Improvements to the original filters are presented in Woodburn et al. (2009), who recommend the addition of buffers (also called "pads") to the apogee-perigee prefilter and a "trust but verify" strategy for the time filter in order to increase their robustness. The orbit path filter however was found to be unstable and is therefore not recommended for practical applications (Woodburn et al., 2009). An improved toroidal path filter has been worked out in Alfano (2012b) and relaxes some of the original path filter assumptions, however still requires padding. A practical guide to the selection of filter parameters is available in Alfano and Finkleman (2014).

The second class of conjunction screeners comprises the so-called sieve-algorithms. These methods are sampling based and sift out pairs of objects for the current sampling step based on simple relative distance tests. The concept dates back to Healy (1995), who recognized the advantage of this method for parallel processing on modern computers and emphasized its robustness which arises from the fact that no assumptions regarding the orbital motion have to be made.

A modern, reliable and efficient sieve approach is presented in Alarcon Rodriguez et al. (2002), Klinkrad (2006, section 8.2) and Escobar et al. (2011). The proposed "smart sieve" method combines the sieving concept with Hoots' apogee-perigee prefilter (which is known to be reliable using pads). Multiple sieves based on simple flight dynamics principles are used in order of increasing computational complexity to rule out pairs of objects. All conjunctions that make it through the sieves are passed to a numerical root finder to determine the nominal time of closest approach (TCA) and the associated relative distance.

Alternative concepts for close encounter identification using genetic algorithms have been presented in Faulds and Spencer (2003). Highly performant results are reported in Novak et al. (2013) who use the initial steps of the smart sieve method and then apply a spatial binning algorithm to further minimize the number of object pairs that are analyzed with the numerical root finder. A technique for choosing the screening volume distance thresholds based on a maximum probability metric is worked out in Alfano (2013).

SMART SIEVE CONJUNCTION SCREENING

For the catalog study in section 6.2 the smart sieve method was selected due to its robustness and parallelizability. Details regarding each individual sieve can be found in Alarcon Rodriguez et al. (2002) and Klinkrad (2006, section 8.2). The following discussion concentrates on aspects which are not addressed in the original publications.

Epoch filter

TLE catalogs sometimes contain RSOs which are older than 30 days and thus unsuitable for CA. Hence, an epoch filter should be applied prior to the sieves to discard TLEs older than a threshold.

Sieving

Due to the huge amount of sieving operations (several hundred billion in case of multiple days of n:n screenings), even small changes to the code can result in runtime differences in the order of minutes.

To obtain the maximum performance, it is preferable to implement the sieves using squared instead of metric distances to avoid the computation of square roots. The squared version of the sieves is given in Alarcon Rodriguez et al. (2002), whereas Klinkrad (2006, section 8.2) presents the metric distance equivalents.

The apogee-perigee filter is listed as the first "sieve" of the algorithm. It is the only filter which operates purely on the geometry of the orbits and does not necessarily require sampling. The apogee-perigee filter can be implemented in two ways. One possibility is to calculate the apogees and perigees of all objects only once. In this case large pads are required to make sure that conjunctions are not erroneously discarded. Alternatively, very small pads can be used if the apogees and perigees are calculated at each sampling step. Due to the large number of sampling steps between t_0 and t_{end}, the first approach is almost always more efficient.

TLE catalog snapshots sometimes contain objects of identical orbit parameters but differing identifiers (e.g. in case of vehicles that are docked to the ISS). If any of these objects is part of a close encounter, the conjunction is reported twice in the output files. Although this is technically correct, it does not provide any added value. Hence, one of the two objects may be put into an exclusion list if this kind of situation is detected. Since equal TLEs lead to equal propagations, identical orbit parameters can be identified in the first sampling step of the algorithm by checking for zero relative distance and zero relative velocity.

Propagation vs. Interpolation

The ability to propagate the trajectories of all objects is required at two locations in the algorithm: the initial sampling of all objects in steps of δt_1 between t_0 and t_{End}, as well as the numerical root finder used for the fine conjunction detection process. Since the sieves are amenable to parallelization, thread-safe orbit propagators (GP and SP) are required. If the SP screening is based on itc-files, the ephemerides should be interpolated instead of using propagation. Only if the ephemerides do not cover the screening period, propagation needs to be applied. To avoid unnecessary propagations and repeated failing interpolations, it is recommended to save the last successful technique for each object to ensure that it will be the method of first choice in the subsequent sampling step.

Fine conjunction detection

After the initial sampling phase state vector arrays ($\mathbf{x}_{i,k}$) are available for each object (index i) and sampling time (index k). All conjunctions that pass the sieves are processed in a two-stage fine conjunction detection process consisting of a linear search algorithm and a numerical root finder. The purpose of the linear search algorithm is to perform additional sampling steps in order to detect points in time for which the relative velocity has opposite signs, thus indicating that a conjunction must take place. The linear search algorithm is operated in steps of δt_2 ($\delta t_2 < \delta t_1$). It is computationally less efficient than the numerical root finder, however it ensures that also multiple conjunctions within δt_1 are detected and supports the convergence of the root finder towards the local distance minimum.

For the algorithm to produce correct results, it is pertinent to center the fine conjunction detection on each sampling step ($\left[t_k - \frac{\delta t_1}{2}, t_k + \frac{\delta t_1}{2}\right]$), instead of only looking for zero relative velocity between t_k and t_{k+1}. The correct way for conducting the fine conjunction detection is depicted in figure 6.4.

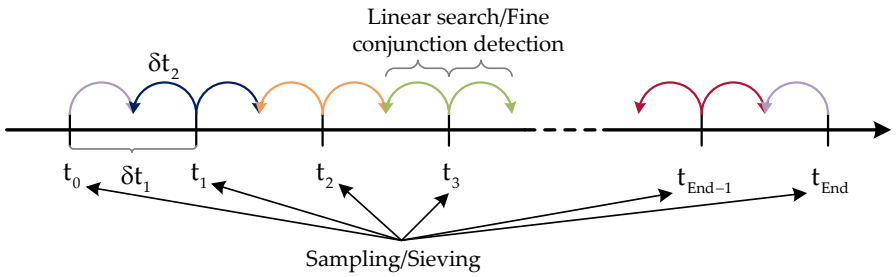


Figure 6.4: Smart sieve: centered fine conjunction detection

An example that yields incorrect results by looking for zero relative velocity between t_k and t_{k+1} is given in figure 6.5. At t_0 the absolute distance between the target and the chaser is 1020 km, which is larger than $R_{c,1}$. The conjunction is therefore discarded by the absolute (squared) distance sieve at t_0. At t_1 the conjunction is passing the r^2 sieve, since $r_1 < R_{c,1}$. Also all other sieves are passed. Thus the pair is handed over to the fine conjunction detection. If this step were to only look ahead, i.e. from t_1 to t_2, the pair would be

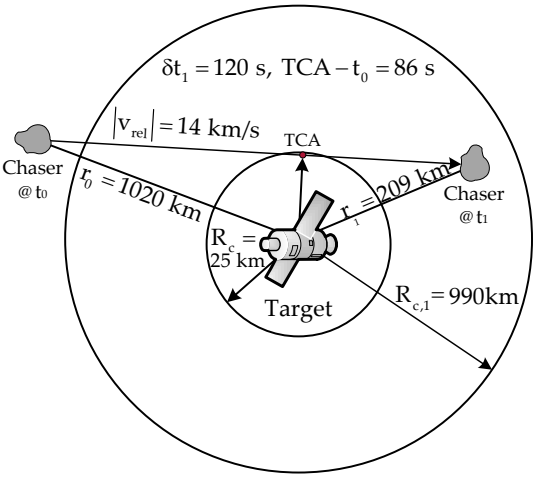


Figure 6.5: Exemplary case that requires centered fine conjunction detection

erroneously filtered out. Due to the centered fine conjunction detection however, the search is performed over $\left[t_k - \frac{\delta t_1}{2}, t_k + \frac{\delta t_1}{2}\right]$, which evaluates to $[60\,s, 180\,s]$. Looking forward and backward therefore makes sure that the close conjunction at $t = 86\,s$ is detected.

Escobar et al. (2011) recommend a regula falsi ("false position") root finder to determine the TCA by searching for zero relative velocity. The standard version of the algorithm (Hamming, 1971, section 2.5) is not suitable for determining the TCAs. Instead, several improvements must be considered to create a robust and efficient algorithm.

The problem with the standard regula falsi method is addressed in Hamming (1971, section 2.6) as its "one-sided approach to zero". Each iteration of the numerical root finder requires two orbit propagations (or interpolations). Thus, fast convergence is desirable. Hamming's improved method accelerates convergence by reducing the function value that is retained for the subsequent iteration. This results in a two-sided approach towards the root. Hamming (1971, section 2.6) notes that his decision of halving the function values is arbitrary. As a matter of fact, a too rapid reduction of the retained function value can lead to unfavorably many iterations being performed from the direction of the artificially reduced values, which in turn slows down convergence and may lead to the maximum number of allowed iterations being exceeded. In extreme cases (e.g. for GEO conjunctions with encounters of multiple hours duration and very small relative velocity slopes) it is also possible for an overly aggressive reduction of the retained function value to create artificial roots which are then considered as being sufficiently close to zero to stop the algorithm, thereby leading to incorrect results (see also the discussion about exit criteria below).

Numerical experiments have shown fastest convergence when multiplying the retained function value by 0.8, which is more conservative than Hamming's multiplication with 0.5 and therefore recommended for practical applications.

Another point to be considered is the representation of time. Most astrodynamic programs use Julian dates (JD) for their continuity. In double precision representation the floating point relative accuracy is limited to approximately $2^{-52} \cong 2.2 \cdot 10^{-16}$ (machine epsilon). Considering an arbitrary TCA of $t_{JD} = 2\,458\,880.5$ days, the machine epsilon amounts to an algorithmic time uncertainty in the order of 45 µs, which when multiplied with a relative velocity of e.g. 15 km/s results in a position uncertainty of ~0.7 m. This is certainly appropriate for the GP theory and also acceptable for SP conjunction screening, however the absolute time uncertainty can be reduced at no effort by transitioning to modified Julian dates for which the machine epsilon corresponds to merely ~1 µs of time uncertainty (or ~1.5 cm in position uncertainty) for a relative velocity of 15 km/s.

Finally also the exit criteria of the algorithm need to be adapted. The original regula falsi method and also Hamming's improved version use

a simple zero-value test to stop the algorithm. However, an exact zero value can never be found in computer implementations of the problem, considering the limited absolute time accuracy. Thus, a two-stage test should be implemented, which checks absolute changes in time additionally to the relative velocity for effectively zero values using different thresholds. The time change test is particularly important in case of conjunctions for which the root finder cannot locate a point in time at which the relative velocity is less than the specified zero relative velocity tolerance. To avoid the algorithm stopping only after the maximum number of permitted iterations, a simple test of the previous iterations' time deltas can be used. If there is evidence of stagnation (e.g. because the Julian date accuracy limit has been reached), the algorithm should be ended.

All vs. all or some versus all?
Most papers focus on the all vs. all problem, which is common to space surveillance centers. Practical applications however also include the m:n case. Prominent examples are screenings of all active satellites against the space catalog or the screening of a single satellite or a satellite fleet against the catalog.

If both n:n and m:n screening shall be supported, it is worthwhile to take a closer look at how the data can be partitioned to avoid passing many pairs of objects to the sieves, which are then immediately discarded because they do not belong to the target set. To reduce the conjunction screening runtime, it is thus in both cases important to examine only non-redundant pairs of targets and chasers.

At each time-step of the n:n case there are $n(n-1)$ possible pairs of targets and chasers. However, half of this set consists of redundant combinations, resulting in $n(n-1)/2$ pairs of objects to be checked for close conjunctions. For a moderate catalog size of 20 000 objects, this amounts to 199 990 000, i.e. roughly 200 million instead of 400 million checks at each sampling time, which is tremendous considering that thousands of sampling times are required for multiple days of conjunction screening.

One way to make sure that the sieves are only invoked for non-redundant pairs of objects, is to formulate the checks in a single loop with $n(n-1)/2$ elements that uses 1D to upper-triangular 2D index unwrapping. The conversion from a linear index i_{conj} to upper-triangular indices (i, j) which can be used for the target and chaser data is achieved via equations 6.1 and 6.2 when indexing from zero.

$$i = n_{conj} - 2 - \left\lfloor \frac{\sqrt{-8i_{conj} + 4n_{conj} \cdot (n_{conj} - 1) - 7}}{2} - 0.5 \right\rfloor \tag{6.1}$$

$$j = i_{conj} + i + 1 - \frac{n_{conj}(n_{conj} - 1)}{2} + \frac{(n_{conj} - i)((n_{conj} - i) - 1)}{2} \tag{6.2}$$

If indexing is started from one, optimized equations can be found by considering that $i_{conj} \to i_{conj} - 1$ and $i \to i + 1$ in eq. 6.1, as well as $i \to i - 1$ and $j \to j + 1$ in equation 6.2. An additional advantage of a single-loop formulation is that all iterations are easily parallelized without resorting to nested parallelism or inner loop only parallelization.

There are multiple approaches to enable m:n screening. Considering that in practice m is usually much smaller than n (e.g. 3000 vs. 20 000 or even 10 vs. 20 000), the following data partitioning was adopted for SPOOK: first an m:(n-m) screening is performed, i.e. all targets are removed from the chasers. For these sets all possible combinations are guaranteed to be unique. The remaining m:m screening is carried out subsequently and also uses upper-triangular indices to make sure that redundant combinations are only checked once. The data partitioning approach is well suited for recursive implementation.

SP sieving using GP as prefilter

SP orbit propagation requires significantly more computing time than its analytical GP counterpart. For direct SP screenings to be solved in a reasonable timeframe, a major part of the ephemerides needs to support interpolation. A significant acceleration of the problem is however possible by first conducting a complete GP conjunction cycle and then using the identified conjunctions for SP refinement. This ensures that time-consuming SP computations are only applied to conjunctions of interest. To avoid boundary cases being missed, the GP screening volume should be enlarged with respect to the SP safety volume.

A seven-day all vs. all GP conjunction screening run of the complete TLE catalog requires something between a few minutes and an hour of runtime, depending on the implementation, degree of parallelization and hardware. This is a very small investment, considering that it reduces the number of conjunctions to be analyzed in SP from a few hundred billion to a few hundred thousand.

If this "GP prefilter" is combined with the maximum supported degree of interpolation and information about the last successfully used propagation method, the time required for complete catalog SP conjunction screening is reduced to something in the order of two hours, which is well in line with the eight-hour catalog update cycle at CSpOC.

Mixed theory propagation

Depending on the data sources used, situations may arise where mixed theory screenings become necessary. As a real world example, consider a satellite operator who possesses exact ephemerides (e.g. GPS-based) of his satellites and wants to use the public TLE catalog to perform conjunction screening. Another example arises from the application

of the GP prefilter for SP screenings. In contrast to the SP catalog, SpaceTrack's TLE catalog also contains so-called "Analyst Objects" (satellite numbers 80 000-89 999), which are of unknown origin (they cannot be associated with a specific launch). Hence, conjunctions between SP catalog RSOs and analyst objects are also of SP vs. GP nature. To support the conjunction assessment and collision probability computation for the aforementioned scenarios, conjunction screeners need to be capable of mixed theory orbit and covariance propagation.

A performance recommendation for GP vs. GP screenings is to conduct the sieving and the fine conjunction detection in the TEME frame (Escobar et al., 2011), which is possible because the conversion between TEME and GCRF depends only on time, such that state vector differences and the orbital shape remain unchanged. For CA using TLEs, the conversion to GCRF is therefore only required to evaluate a post–root finder RTN ellipsoid criteria (cf. Klinkrad, 2006, section 8.2) or to output the screening results in inertial coordinates.

In contrast to pure GP conjunction screening, a crucial point for mixed theory propagation is the consistent application of GCRF coordinates, which is the natural frame for SP propagations. The GP propagations therefore directly need to be converted to GCRF when sampling the orbits. Only then it is made sure that the quantities of interest (apogee/perigee altitudes and relative distances/velocities) are evaluated in a common coordinate frame.

Finally, operational software should be universal and maintainable at low cost. It is therefore advisable to develop "unified propagation" routines which accept the information required for orbit/covariance propagation of both theories (e.g. via optional arguments), respect the aforementioned frame recommendations, and decide whether to perform GP propagation, SP propagation or interpolation based on the available inputs. Since CARA always requires pairs of objects to be propagated, it is also recommended to create wrapper routines which receive target and chaser data, propagate both using the unified propagation routines and optionally also return the position difference and the combined covariance.

Validation

Before using identified close encounters to calculate a collision probability, it is necessary to validate the screening results against practice-proven solutions. A publicly available source that lists upcoming conjunctions is the Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space (SOCRATES) service of the Center for Space Standards & Innovation (CSSI) (Kelso and Alfano, 2005), which is available at `https://celestrak.com/SOCRATES/`. SOCRATES uses the commercial AGI STK Advanced CAT add-on module to screen all payloads against the entire TLE catalog using a 5 km spherical safety volume. STK Advanced CAT employs the Hoots prefilters

for conjunction screening. For SOCRATES 30 km pads, as well as an epoch filter with a maximum age of 30 days are applied. The top 10 conjunctions (with respect to a sorting criterion such as minimum range) of the next seven days are uploaded twice a day. SOCRATES is a great source to test the basic functionality of a conjunction screener, however 10 conjunctions are not sufficient for validation. Thus, the complete 7-day screening results of a SOCRATES run were requested from CSSI. Thanks to the provided dataset, a detailed validation of the smart sieve implementation could be carried out.

The screening results are compared with respect to conjunction detection, the time difference at TCA and the associated position/velocity differences. Table 6.1 summarizes the key figures. Exemplary results of the first 10 conjunctions are reported in table 6.2 and demonstrate the millisecond-level agreement.

Total conjunctions SOCRATES	29 872
Total conjunctions SPOOK	30 920
Matched conjunctions	29 705
Missed conjunctions SOCRATES	1215
Missed conjunctions SPOOK	167
Missed conjunctions < 4.75 km SOCRATES	1119
Missed conjunctions < 4.75 km SPOOK	9
Average TCA difference	8.0972 ms
Average TCA difference without outliers[5]	-0.8064 ms
Maximum TCA difference	70.3780 s
Average position difference at TCA	0.1117 m
Average position difference at TCA without outliers	0.1069 m
Maximum position difference at TCA	-3.348 km
Average speed difference at TCA	0.0020 m/s
Average speed difference at TCA without outliers	0.0019 m/s
Maximum speed difference at TCA	0.6372 m/s

Table 6.1: Validation of SPOOK smart sieve conjunction screener versus SOCRATES dataset

Multiple interesting points can be inferred from table 6.1. Considering the different implementations and conjunction screening approaches, an average TCA difference of less than a millisecond (neglecting outliers) is remarkable. To minimize boundary effects, the focus has been set on conjunctions with distances smaller than 4.75 km at TCA.

SPOOK detected 1119 conjunctions that were not part of the SOCRATES outputs. On the contrary, SOCRATES reported 9 conjunc-

5 When sorting both datasets with respect to TCA, outliers are defined as matched conjunctions which do not share the same index, i.e. sorting order. This situation sometimes arises for GEO conjunctions with very small relative velocities

Conjunction	Target	Chaser	TCA SOCRATES (UTC)	TCA SPOOK (UTC)	Relative distance at TCA SOCRATES [km]	Relative distance at TCA SPOOK [km]	Relative speed at TCA SOCRATES [km]	Relative speed at TCA SPOOK [km]	TCA difference [ms]	Rel. distance difference at TCA [m]	Rel. speed difference at TCA [m/s]
1	2828	14458	2020-02-08 12:00:09.875	2020-02-08 12:00:09.872	4.087	4.084	3.788	3.788	3	2.511	0.363
2	5553	7057	2020-02-08 12:00:45.297	2020-02-08 12:00:45.299	2.822	2.820	12.752	12.752	-2	1.908	0.487
3	35782	43886	2020-02-08 12:00:45.297	2020-02-08 12:00:45.300	4.190	4.200	14.990	14.990	-3	-10.335	0.258
4	20232	39074	2020-02-08 12:01:06.809	2020-02-08 12:01:06.809	4.949	4.919	12.365	12.365	0	30.407	-0.104
5	34005	43257	2020-02-08 12:01:40.393	2020-02-08 12:01:40.389	2.656	2.653	1.937	1.937	4	2.769	-0.031
6	1512	13502	2020-02-08 12:01:40.789	2020-02-08 12:01:40.788	4.306	4.302	6.636	6.636	1	3.539	-0.420
7	43669	44097	2020-02-08 12:02:00.655	2020-02-08 12:02:00.656	1.849	1.850	4.713	4.713	-1	-0.919	-0.415
8	39441	41093	2020-02-08 12:02:18.641	2020-02-08 12:02:18.643	0.047	0.048	14.258	14.258	-2	-1.118	0.372
9	33492	44342	2020-02-08 12:02:20.380	2020-02-08 12:02:20.379	4.435	4.448	9.416	9.416	1	-12.534	-0.076
10	7209	35211	2020-02-08 12:02:21.113	2020-02-08 12:02:21.116	3.295	3.294	12.977	12.977	-3	0.651	0.285

Table 6.2: SOCRATES and SPOOK screening results of the first 10 conjunctions starting from 2020-02-08 12:00:00.000 UTC

tions that were not in the SPOOK results. It is believed that many of the conjunctions missed by SOCRATES are caused by the application of the path filter, which is known to erroneously exclude some conjunctions (Woodburn et al., 2009; George and Harvey, 2011; Alfano and Finkleman, 2014). T.S. Kelso from CSSI furthermore pointed out that STK's method for conjunction identification is searching for the closest point within a distance threshold for which only a single conjunction is reported in case of multiple close approaches.

The nine conjunctions not reported by SPOOK were investigated in detail. It was found that four of the conjunctions involved RSOs

of over 24 days age, which cannot be truthfully propagated over such long timeframes and are thus not reliable for conjunction assessment. The remaining five conjunctions were not reported by SPOOK, since they were slightly above the 5 km safety volume.

TCA solutions of matched conjunctions can vary for several reasons, such as propagator differences[6], numerical root finder convergence criteria or differing conjunction definitions[7]. In summary, the smart sieve conjunction screener which has been integrated into SPOOK can be considered validated, since all differences to SOCRATES were found to be either numerical boundary cases or caused by TLE data of multiple weeks age. In addition, over 1100 conjunctions could be detected which were not recognized by SOCRATES.

6.1.2 *Computation of a collision probability*

Once conjunctions between RSOs have been identified, their risk assessment is the next logical step. In earlier times only the expected minimum distance was used for this purpose. It however turned out that a purely distance-based assessment is not representative of the actual risk, but that knowledge of the orbit uncertainties must be included as well. If the orbits of both objects are known very accurately, even small relative distances do not pose a risk. Conversely, also large relative distances can be critical if they are about the size of the combined uncertainty. For this reason, a transition from minimum distance to collision probabilities took place in the 1990s and early 2000s.

The meaningfulness of the probability concept in this context can certainly be disputed and is closely linked to the Bayesians vs. frequentists debate (see section 1.3.4 for a detailed discussion on these topics). The Bayesian school represents the current state of the art, in which the collision probability is defined as the integral of the pdf describing the uncertainty in the relative state. However, also the Monte-Carlo technique, in which the target and chaser epoch trajectories are first sampled according to their pdfs and subsequently propagated to test for collisions, is regularly used as a reference and contains features of both positions. MC collision probabilities also show the dilution effect, an artifact of pdfs and the Bayesian school, however the frequentists long-term frequency definition is applied to determine the collision probability as the ratio of detected collisions over repetitions.

Several assumptions have been introduced in the past to simplify non-MC collision probability computations. Modeling the objects as

6 It is believed that SPOOK and STK share the same baseline SGP4 implementation, however different programming languages, compiler optimization flags, parallelization instructions, etc. still lead to diverse results

7 SGP4 produces position and velocity, however the velocity solution deviates from the result of a numerical differentiation of neighboring position solutions, which leads to slightly different TCA results when searching for zero crossings in the relative velocity or a minimum in the relative position

spheres eliminates the need for attitude information of the conjunction partners, which in most cases is not available. A typical but not mandatory assumption is that uncertainties in relative velocity can be neglected. This enables the calculation of a collision probability only on the basis of the relative position pdf. Furthermore, the target and chaser position errors are commonly assumed to be zero-mean, Gaussian and uncorrelated, which allows to compute the collision probability according to equation 6.3:

$$P_c = \iiint_V \frac{1}{\sqrt{(2\pi)^3|\mathbf{C}|}} e^{-\frac{1}{2}\mathbf{r}^T\mathbf{C}^{-1}\mathbf{r}} \, dx \, dy \, dz \tag{6.3}$$

where $\mathbf{r} = \mathbf{r}_T - \mathbf{r}_C$ is the relative distance vector (target minus chaser) and V defines the limits of integration as the encounter region (or "encounter shell") in which the conjunction is taking place. Chan (2008, section 2.1) has shown that under the assumption of uncorrelated position errors the covariance of $\mathbf{r}$ is found as the addition of the individual position error covariances[8], i.e. $\mathbf{C} = \mathbf{C}_T + \mathbf{C}_C$.

Research in the field of collision probability algorithms is broad and excellent, which has led to several publications detailing techniques to relax some of the constraints underlying equation 6.3. The application of GMMs for example allows to extend the concept of Gaussian errors to arbitrary pdf shapes (Vittaldev, 2015, chapter 5). The inclusion of velocity error uncertainties has been demonstrated by Coppola (2012). Arbitrary object shapes have been elaborated for example in Patera (2005), Chan (2008, chapter 6), Alfano (2012a) or Krier (2017). Error correlation was addressed in Coppola et al. (2004), Chan (2008, section 2.6), Casali et al. (2019) and Gondelach and Linares (2020).

However, many of these modern techniques have not been implemented in operational environments as the necessary additional information for their application is not available or because their impact is assumed negligible (Schiemenz et al., 2019a).

Ultimately, collision probabilities are calculated to compare their values against predefined thresholds, thus aiding the decision-making process for future actions, such as the execution of CAMs or requesting further observations. Only changes of at least an order of magnitude (OOM) in P_c are considered operationally relevant to CARA (Ghrist and Plakalovic, 2012). For this reason, a number of simplifying assumptions can be justified operationally, whereas some modern algorithms may not meet the OOM criterion and are thus not of operational significance (the improved P_c estimates do not alter the decision making process but require further data and/or computing time). This in turn leads to a proliferation of equation 6.3 and the corresponding collision probability algorithms.

8 In the context of collision probabilities, covariances are typically denoted using a capital $\mathbf{C}$ to avoid confusion between the collision probability P_c and the capital $\mathbf{P}$ used to express covariance matrices in the context of orbit determination

ENCOUNTER CATEGORIZATION

Close encounters in space can be divided into two categories based on their duration: short-term encounters (STE) and long-term encounters (LTE). In contrast to long-term encounters, the encounter shell of STEs is traversed very quickly, which justifies a rectilinear motion assumption and thus a reduction of the problem from 3D to 2D. Short-term encounters are therefore characterized by substantially simpler and less computationally demanding P_c calculation algorithms.

Two completely different approaches for STE detection have been published in the literature. Chan (2008, sections 3.3-3.4) uses purely geometric considerations to test the deviation of the orbital paths from straight lines and the time to transverse the encounter region. The test starts by defining a minimum rectilinear path length requirement S, which is based on maximum differences between the 3D and reduced 2D collision probability integrals. The distance S is then used together with the target/chaser approach angle ϕ to approximate the deviation of the orbital paths from straight lines in terms of a deflection angle α and the ratio of the encounter region transit time over the orbital period. Boundary thresholds on both quantities are used to assess if a conjunction qualifies for the linear motion approximation.

Alfano (2007a) defines a two-stage propagation-based test, which aims to find the minimum velocity required at TCA such that a user-specified probability ratio $f = |P_{c,2D} - P_{c,3D}|/P_{c,2D}$ is not exceeded by the two-dimensional short-term encounter computation of the collision probability. Alfano first performs a two-body coarse assessment to approximate the limits of linear motion based on the specified value for f. The coarse evaluation is followed by a refined assessment which uses the complete force models and the coarse evaluation data to determine the minimum relative velocity v_{min} for the conjunction. If $v_{conj} > v_{min}$, the 2D computation is justified.

Alfano's test is potentially more accurate, however significantly more complex and time-consuming. It involves quadratic curve fitting, state vector and covariance propagation, whereas Chan's geometric considerations only require square root and trigonometric function evaluations. Unfortunately, Chan does not describe his test in full detail. Instead, examples are formulated in an inverse manner by specifying deflections from rectilinear paths and comparing the resulting values for S against the rectilinear path requirements, whereas in practice a requirement for S is given and ϕ needs to be computed to find the deflections from the rectilinear paths. Chan suggests interpolation of table data to tackle the operational direction of the problem, however also notes that closed-form equations for the operational problem can be derived from those of the inverse direction. Strict boundary values for the deflection angle and ratio of the encounter shell transit time over the orbital period are not provided, however two examples

are presented together with an assessment of their results regarding the validity of the rectilinear motion approximation.

Based on the fragmented information in Chan (2008), an encounter classification subroutine was developed for usage in the force model uncertainty catalog study. The code has shown very robust performance for the categorization of millions of conjunctions. The Fortran sources are given in listing B.7, found in appendix B.3.

REVIEW AND SELECTION OF CONJUNCTION PROBABILITY ALGORITHMS

Equation 6.3 requires the relative position vector and the combined covariance of the target and chaser RSOs. The computation of conjunction collision probabilities is not performed in GCRF coordinates, but in the "encounter frame", which is aligned with the relative velocity vector and depicted in figure 6.6. Literature does not always agree on the axes labeling. Works by Patera typically associate the relative velocity direction with the z-axis of the encounter frame. Chan on the other hand assigns the y-axis to the relative velocity direction and Alfano sometimes uses the z or the x-axis to denote the relative velocity direction in the encounter frame (which he calls velocity normal co-normal (VNC) frame). In this work, Chan's convention is applied, i.e. the relative velocity direction is taken as the encounter frame y-axis.

At precisely the time of closest approach, the relative position vector has to be perpendicular to the relative velocity vector. In practice however the relative position and velocity vectors at TCA do not yield an exact zero dot-product[9] – a fact, which is usually ignored in textbooks (e.g. Klinkrad, 2006, eq. 8.25). The omission of this information is critical however, as the inaccurate frame definition which directly uses the relative position vector to define the encounter frame x-axis can only be tolerated in case of 2D collision probability calculations, but even then this results in the character of the 2D transformation as a special case of the 3D transformation to be lost. Thus, the fully orthogonal encounter frame definition should also be used for two-dimensional collision probabilities.

Computations of the 3D collision probability require to calculate the encounter frame at many different points in time prior and after the TCA for which the angle between the relative position and relative velocity vectors is far from orthogonal. For this reason it is essential to formulate the GCRF to encounter frame transformation in a universally applicable manner that ensures the encounter frame axes to be orthogonal to each other. The 2D encounter plane (sometimes also called "b-plane") is at every instant defined as the plane perpendicular to the relative velocity direction.

9 Recall the discussion about the conjunction definition as minimum position or zero relative velocity, as well as the convergence criteria of the numerical root finder and the time accuracy of Julian date representations

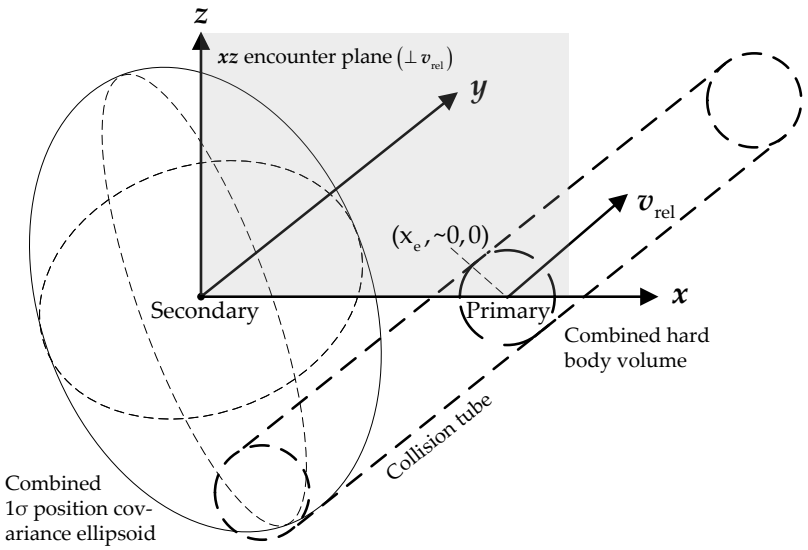


Figure 6.6: Encounter frame definition. The encounter plane is defined perpendicular to the relative velocity direction (shaded gray). At TCA the y-coordinate of the miss vector is zero (in practice close to zero), at any other instant it is nonzero

Equations 6.4-6.10 summarize the generic transformation:

Generic transformation equations to convert the relative GCRF state and combined position covariance into the encounter frame. The opposite transformation is achieved by taking the transpose of equation 6.7

$$\mathbf{y}_b = \frac{\boldsymbol{\nu}_{rel}}{|\boldsymbol{\nu}_{rel}|} \tag{6.4}$$

$$\mathbf{z}_b = \frac{\mathbf{r}_{rel} \times \boldsymbol{\nu}_{rel}}{|\mathbf{r}_{rel} \times \boldsymbol{\nu}_{rel}|} \tag{6.5}$$

$$\mathbf{x}_b = \mathbf{y}_b \times \mathbf{z}_b \tag{6.6}$$

$$\mathbf{T}_{Enc,\,GCRF} = \begin{bmatrix} \mathbf{x}_b^T \\ \mathbf{y}_b^T \\ \mathbf{z}_b^T \end{bmatrix} \tag{6.7}$$

$$\mathbf{r}_{Enc} = \mathbf{T}_{Enc,\,GCRF}\ \mathbf{r}_{GCRF} \tag{6.8}$$

$$\boldsymbol{\nu}_{Enc} = \mathbf{T}_{Enc,\,GCRF}\ \boldsymbol{\nu}_{GCRF} \tag{6.9}$$

$$\mathbf{C}_{Enc} = \mathbf{T}_{Enc,\,GCRF}\ \mathbf{C}_{GCRF}\ \mathbf{T}_{Enc,\,GCRF}^T \tag{6.10}$$

The transformation into the encounter plane is achieved either by performing the 3D transformation and subsequently dropping the y-components of the position/velocity vectors and the position covariance, or by applying the 2×3 transformation matrix $\mathbf{T}_{Enc,\,GCRF} = [\mathbf{x}_b\ \mathbf{z}_b]^T$ instead of equation 6.7.

Within the encounter plane, the lines of constant probability density are ellipses and the hard body sphere reduces to a hard body circle, which is defined by the hard body radius ($r_{HB} = r_p + r_s$). The integral in equation 6.3 can be split into an integration within the encounter plane and an integral over the relative velocity direction. For STEs the encounter duration is sufficiently short to approximate the combined covariance with its value at TCA. Furthermore, the curvature effect

of gravity is neglected, resulting in the volume being swept out by the primary on its passage through the encounter region to form a "collision tube" (cf. figure 6.6).

Taken together, these conditions allow to decouple the integration along the relative velocity direction y from the encounter plane integration in x/z. Many authors restrict themselves to this consideration of STEs and directly provide the reduced 2D collision probability formula, without correctly motivating its origin. The only valid mathematical derivation of the reduced 2D collision formula is the integration of equation 6.3 with respect to the relative velocity direction y. The calculations are published in Chan (2008, section 3.5). The reduced 2D collision probability integral (which does not contain a dependency on the relative velocity direction) is given in equation 6.11 or alternatively eq. 6.12:

$$P_{c,2D} = \frac{1}{2\pi\sigma_x\sigma_z\sqrt{1-\rho_{xz}^2}} \iint_A e^{-\frac{\left[\left(\frac{x}{\sigma_x}\right)^2 - 2\rho_{xz}\left(\frac{x}{\sigma_x}\right)\left(\frac{z}{\sigma_z}\right) + \left(\frac{z}{\sigma_z}\right)^2\right]}{2(1-\rho_{xz}^2)}} \, dx\, dz \quad (6.11)$$

$$= \frac{1}{2\pi\sqrt{|\mathbf{C}_2|}} \iint_A e^{-\frac{1}{2}\left[\begin{pmatrix}x\\z\end{pmatrix}^T \mathbf{C}_2^{-1} \begin{pmatrix}x\\z\end{pmatrix}\right]} \, dx\, dz \quad (6.12)$$

where $\mathbf{C}_2$ denotes the 2D encounter plane covariance matrix. The expression in the square brackets of the exponential function argument in equation 6.12 represents the squared encounter plane Mahalanobis miss distance. The 3D equivalent, i.e. the squared actual Mahalanobis miss distance, can also be identified in equation 6.3.

The cross-correlation term in equation 6.11 can be avoided by rotating the encounter frame about the relative velocity direction to align the covariance principal axes with the encounter plane axes. In this principal axis encounter plane, equation 6.11 becomes:

$$P_{c,2D} = \frac{1}{2\pi\sigma'_x\sigma'_z} \iint_{A'} e^{-\frac{1}{2}\left[\left(\frac{x'}{\sigma'_x}\right)^2 + \left(\frac{z'}{\sigma'_z}\right)^2\right]} \, dx'\, dz' \quad (6.13)$$

Authors omitting the correlation term are implicitly assuming principal axis encounter plane coordinates.

Short-term encounter collision probability algorithms

Equations 6.11-6.13 are typically not used directly to compute the two-dimensional encounter plane collision probability. Instead, they serve as the baseline for more optimized algorithms.

Four classes of 2D algorithms with broad acceptance have emerged over the past 30 years. These are founded on Foster and Estes (1992), Patera (2001, 2005), Alfano (2002, 2005) and Chan (1997, 2003). Chan's method is a well-known analytical algorithm, whereas Foster, Patera and Alfano rely on numerical integration.

Foster's technical report started the transition from miss ranges as risk metric to collision probabilities. His method transforms equation 6.11 into polar coordinates, which rephrases the Cartesian double integral into an angular and a radial integration. Patera's algorithms on the other hand express eq. 6.11 as a one-dimensional contour integral, which in Patera (2001) is centered at the origin of the encounter plane, whereas in Patera (2005) the origin is changed to the center of the hard body contour, which requires less integration steps and allows to convert the contour integral into a standard definite integral. Motivated by earlier results of Chan (1997) and Alfriend et al. (1999), Alfano develops an error function series expression of equation 6.13, which exploits the intrinsic availability of the error function in many programming languages.

A relative accuracy review of the methods is available in Chan (2008, section 5.5). An excellent extended analysis that evaluates the absolute accuracies by comparison against a direct numerical evaluation of equation 6.13 is presented in Alfano (2007b).

In order to establish a library of 2D collision probability methods, all algorithms, including different variants, were implemented in Fortran and integrated into SPOOK. Such a library allows to perform own comparisons and helps gaining a deeper understanding of runtime and accuracy differences, which eventually lays the foundation for the choice of a baseline algorithm for catalog conjunctions that fulfill the STE criteria. The following lessons have been learned:

- Chan's analytical formula is by far the fastest, however a number of catalog conjunctions have been observed for which the equivalent area approximation leads to errors of multiple orders of magnitude in comparison to the numerical methods
- Alfano (2007b) and Chan (2008, section 5.5) recommend integration steps of 0.5 deg and $r_{HB}/12$ for Foster's method, however some conjunctions were found that required a finer angular resolution (0.25 deg) to agree with the Alfano and Patera results
- Patera recommends 400 steps for his contour integral approach. For high aspect ratios of the combined covariance, some conjunctions were found that required further integration steps, which is in line with findings reported by Alfano (2007b)
- Patera uses 200 integration steps for his optimized object-centered integral formulation. Alfano (2007b) found that already 50 iterations are sufficient to avoid errors within operational decision-making thresholds. Catalog tests with the SPOOK implementation generally confirm Alfano's results, however some high aspect-ratio conjunctions were found for which only $\geqslant 200$ steps yielded results that matched Alfano's error function method. This is in line with Balch et al. (2019), who propose $10 \max(\sigma'_x/\sigma'_z, \sigma'_z/\sigma'_x)$ iterations for Patera's method

- The formulation of Alfano's method which separates even and odd parts of the argument yields accurate results with only 10 to 50 integration steps
- A direct formulation of Alfano's method which does not separate even and odd terms (Chan, 2008, eq. 5.35) requires approximately as many integration steps as Patera's updated method
- Simpson's $\frac{1}{3}$-rule (Hoffman and Frankel, 2001, section 6.8.2) yields slightly superior numerical integration performance in comparison to the trapezoidal integration rule
- Integrals over angular quantities (as found in the algorithms of Foster and Patera) can be accelerated by calculating $\sin(d\phi)$ and $\cos(d\phi)$ only once and then using Chebyshev polynomials to obtain $\sin(n\,d\phi)$ and $\cos(n\,d\phi)$

Based on above findings, the following usage recommendations can be established:

Algorithm	Performance	Accuracy	Application
Chan	Very high	Low to Medium	Trend analysis, dependency testing
Foster	Low	Medium to High	Validation of other algorithms
Patera (Contour)	Medium	High	Operational application, validation of other algorithms, non-spherical objects
Alfano (direct)	Medium to High	High	Operational application, validation of other algorithms
Patera (Updated)	Medium to High	High	Operational application, validation of other algorithms, non-spherical objects
Alfano (even/odd separation)	High	High	Operational application, validation of other algorithms

Table 6.3: Assessment of classical encounter plane collision probability algorithms

As the catalog study presented in section 6.2 was conducted using the HBR file provided by the ESA SDO, the capability to consider non-spherical object shapes (as provided by Patera's algorithms) was not required. Hence, Alfano's method was selected as the baseline for two-dimensional collision probabilities. The algorithm demonstrated accurate results for a wide range of catalog conjunctions and requires the fewest number of integration steps among the numerical approaches.

Long-term encounter collision probability algorithms

For long-term encounters it is not sufficient to consider the conjunction geometry only at the time of closest approach. Since the target and chaser are in the encounter region for a considerable amount of time (possibly many hours), the collision probability must be accumulated over this period to calculate the total P_c. In addition to the actual algorithms used to calculate the collision probability, also the limits of integration are important to be chosen appropriately. If the integration is done only in the vicinity of the TCA, the probability of collision may be too small, because not all parts of the motion through the encounter-shell are taken into account. However, if the range is chosen too large, significantly more runtime is required without actually accumulating any collision probability. The maximum accumulation limit should be set to one orbital revolution (Alfano, 2006).

In principle, the encounter region can be limited by time or relative position Mahalanobis distance. Both checks should be implemented in software routines. Patera suggests a three sigma Mahalanobis limit, but Alfano points out that this may not be sufficient and suggests ten sigma instead (Alfano, 2006). If only a time limit shall be considered (e.g. for testing purposes), the Mahalanobis limit can be raised to a very large number (3000 is used in Alfano, 2009). The inverse however is not always true, since for some conjunctions the Mahalanobis limit may never be exceeded (e.g. in case of traveling partners).

Field-tested Gaussian approaches for the calculation of the 3D collision probability can be classified into three categories according to the granularity applied in partitioning the path swept out by the primary on its way through the encounter region. The first group is formed by the "adjoining tube" or "adjacent cylinder" algorithms. Representatives of this class are Patera (2003), McKinley (2006) and Alfano (2006). These algorithms model the path of the primary through the encounter shell via many short adjacent tubes (cf. figure 6.7). The assumptions of a constant combined covariance and rectilinear motion are applied to each cylinder section. The final collision probability is calculated as the sum of the individual tube collision probabilities. Patera and Alfano calculate the tube P_c contribution as the product of a 1D sectional collision probability and a 2D cross sectional P_c. The 2D face-component of the tube P_c can in principle be calculated using any STE collision probability method (a discussion on this point ensues), although Patera (2003) naturally limits the discussion to his 2D contour integral method. In essence however Alfano's and Patera's methods differ only slightly in the calculation of the sectional 1D tube P_c component.

Consider the single tube section in figure 6.8. The 1D sectional collision probability is computed as the integral of the one-dimensional Gaussian pdf in the y-direction from the initial to the final tube face:

$$P_{1D} = \frac{1}{\sqrt{2\pi}\sigma} \int_{y_i}^{y_f} \exp\left(-\frac{y^2}{2\sigma^2}\right) dy \tag{6.14}$$

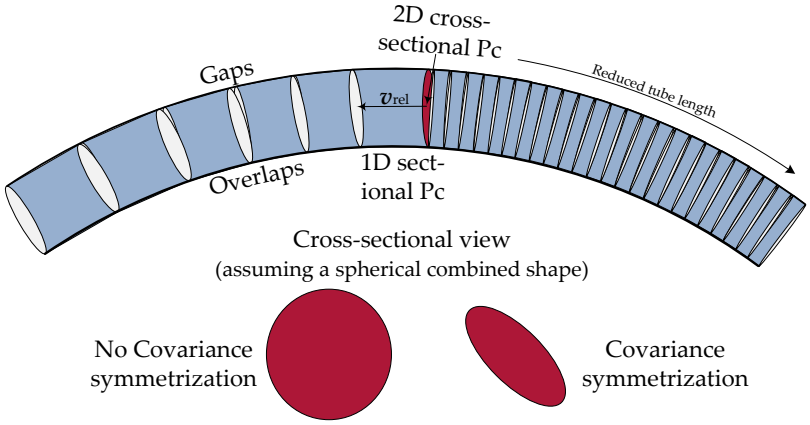


Figure 6.7: Nonlinear relative motion adjoining tube collision probability computation

Assuming a constant pdf, the integrand can be taken out of the integral, which results in:

$$\begin{aligned} P_{1D} &= \frac{y_f - y_i}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2}\right) \\ &= \frac{\Delta y}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2}\right) \end{aligned} \tag{6.15}$$

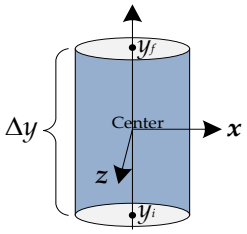


Figure 6.8: Sectional tube geometry

Defining the one-dimensional Mahalanobis distances as M_f and M_i, it is possible to rewrite equation 6.15 in the form of eq. 6.16 by considering that the tube section value of y is evaluated at the beginning of the tube.

$$P_{1D} = \frac{M_f - M_i}{\sqrt{2\pi}} \exp\left(-0.5\, M_i^2\right) \tag{6.16}$$

In contrast, Alfano's one-dimensional collision probability equation is obtained by splitting equation 6.14 and exploiting the symmetry of the problem and using the error function for each part of the integral:

$$\begin{aligned} P_{1D} &= \frac{1}{\sqrt{2\pi}\sigma}\left(\int_{y_i}^{0} \exp\left(-\frac{y^2}{2\sigma^2}\right) dy + \int_{0}^{y_f} \exp\left(-\frac{y^2}{2\sigma^2}\right) dy\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma}\sqrt{\frac{\pi}{2}}\sigma\left(\operatorname{erf}\left(\frac{y_f}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{y_i}{\sqrt{2}\sigma}\right)\right) \\ &= \frac{1}{2}\left(\operatorname{erf}\left(\frac{M_f}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{M_i}{\sqrt{2}}\right)\right) \end{aligned} \tag{6.17}$$

To obtain positive sectional collision probabilities, independently of the propagation direction, it is recommended to take the absolute value of equation 6.17, which yields:

$$P_{1D} = \left|\frac{1}{2}\left(\operatorname{erf}\left(\frac{M_f}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{M_i}{\sqrt{2}}\right)\right)\right| \tag{6.18}$$

Equations 6.15 and 6.18 are the expressions used by Patera and Alfano to obtain the decoupled one-dimensional collision probability component. To obtain correct results, it is important to evaluate all components of the equations in the same tube–encounter frame. It is therefore not possible to take the M_f of the previous iteration as the M_i of the current iteration, since it is required to be evaluated in the encounter frame of the next tube section.

In order to decouple the x/z cross-sectional P_c from the sectional P_c in the relative velocity direction, Patera and Alfano apply a transformation which symmetrizes the combined covariance in three dimensions. In this way Patera reduces the covariance to a single standard deviation σ. Alfano on the other hand scales the covariance in all axes to obtain an identity matrix pdf. This causes metric distances to match Mahalanobis distances, which is why he also refers to the target frame of the transformation as "Mahalanobis space". The operations required for this transformation involve rotations and scaling. For this reason, the relative path of the RSOs and the hard body volume are rotated/stretched/compressed accordingly. This is indicated in the lower part of figure 6.7 by a corresponding distortion of the spherical hard body x/z contour.

Interestingly, the complete transformation of the covariance and the hard body volume has not been published in full detail yet. Patera (2003) specifies a transformation of the combined covariance, however only a pointwise transformation of each infinitesimal hard body volume element is given, which is not very helpful to determine M_i and M_f. Alfano (2006) notes that an eigendecomposition of the combined covariance is necessary to determine the rotational and scaling components, however does not describe the hard body transformation process. The complete transformation for arbitrary covariances and spherical hard body volumes has been derived as part of this work. The Fortran sources can be found in listing B.8, appendix B.5.

Since the pdf is isotropic after the transformation (it contains no cross-correlations and equal standard deviations in all axes) each direction can easily be decoupled form the others. Thus, equations 6.15 and 6.18 do not depend on quantities in x or z. To evaluate the cross-sectional 2D collision probability directly in the Mahalanobis space, adapted formulations of the 2D collision probability are necessary. Such Mahalanobis space formulations of Chan's and Patera's object-centered approaches have been worked out in Alfano (2007a).

McKinley (2006) does not require covariance symmetrization, since in his method the tube P_c is not decoupled into 2D and 1D collision probabilities. In fact, this separation is neither possible nor desired in his algorithm, which aims at obtaining more accurate results by reducing the problem of gaps and overlaps. To this end, McKinley calculates the in-plane alignment angles of adjacent cylinder elements and adapts the 3D integration limits to connect the cylinders seam-

lessly in the orbital plane. Out-of-plane nonlinearity is not considered. The adaptation of the integral limits necessitates a time-consuming three-dimensional numeric integration. McKinley's algorithm is more accurate than Alfano's and Patera's, but it also requires substantially more runtime. The difference in the collision probability estimates is only significant if the gaps and overlaps are non-compensating and located in areas of high probability density.

In the context of this work, a novel extension of the adjoining tube algorithm was developed, which does not require covariance symmetrization. The extension solves a problem of practical relevance, i.e. the application of the original STE encounter plane collision probability algorithms in the adjoining tube methods.

Since the classical two-dimensional collision probability algorithms are formulated in the encounter plane, but not in the Mahalanobis frame, the algorithms either need to be recast into Mahalanobis space, or the transformation only serves the purpose of decoupling the y-component, such that the one-dimensional sectional tube collision probability can be computed according to equation 6.15 or 6.18. Obviously, the transformation into Mahalanobis space just to compute one component of the collision probability is not an efficient solution. However, if it were possible to compute M_f and M_i without explicitly decoupling the y and x/z-components, Mahalanobis formulations of the encounter plane algorithms would become obsolete and at the same time the algorithm would be accelerated, since frame transformations would no longer be necessary. This is exactly what is achieved with the new extension of the algorithm. All calculations take place in the encounter frame, which means that all encounter plane formulations for the calculation of the cross-sectional collision probability are directly applicable without changing frames. The derivation is presented after the introduction of the remaining nonlinear collision probability algorithms.

Alfano's adjoining parallelepiped algorithm (Alfano, 2007a) forms the second granularity category of the three-dimensional collision probability algorithms. By dividing the tube into bundles of abutting parallelepipeds, the problem of gaps and overlaps can be solved in the in-plane and out-of-plane directions. Each parallelepiped end is adjusted to create a smooth transition layer, which further increases the accuracy with respect to McKinley (2006) who only accounts for the in-plane adaption. The calculation of each parallelepiped collision probability contribution no longer uses the classic 2D formulae, but the product of equation 6.18 for each spatial direction ($P_{elem} = P_{1D,x} \cdot P_{1D,z} \cdot P_{1D,y}$). The total collision probability is obtained as the sum of all individual parallelepiped collision probabilities during the passage of the primary through the encounter region. At each sampling time, the (arbitrary) hard-body x/z cross-section is assembled via the face sides of the parallelepipeds. The accuracy of the collision probability

computations increases if more parallelepipeds are used and if the parallelepiped length (analogous to the length of the tubes in the case of the adjoining tube algorithms) is not chosen too long for the constant covariance assumption to become violated (a mechanism for selecting the tube length is described in Alfano, 2006). Further details can be found in Alfano (2007a).

The third granularity category is obtained by shortening the parallelepipeds to small volume elements (voxels). The path of the primary through the encounter region is no longer modeled by tubes or parallelepipeds, but sampled in Mahalanobis space using voxels. At each sampling instant the P_c contribution of any new voxel is determined using the same equation as for the parallelepiped approach. The subsequent sampling instant starts by assessing which voxel P_c contribution was already considered in the previous sampling step. These are excluded from further consideration. The additional P_c contribution due to all new voxels is added to the so-far accumulated overall P_c. Increased sampling frequencies lead to a higher number of voxels to be part of successive sampling steps. This prolongs the runtime, but also enhances the accuracy of the results. Sampling too seldom on the other hand leads to an underestimation of the collision probability, since not all volume elements which have been swept out by the primary are guaranteed to have been taken into account for the collision probability accumulation. The algorithm supports arbitrary object shapes. Further details are explained in Alfano (2006, 2012a).

Usage recommendations for the nonlinear collision probability algorithms are collected in table 6.4.

Algorithm	Performance	Accuracy	Application
Adjoining tube (Patera/Alfano)	High	Medium	Operational application
Adjoining tube (McKinley)	Medium	Medium+	Operational application
Parallelepipeds	Low	Medium to High	Numerical studies
Voxels	Very Low	High	Numerical studies, Validation of other algorithms
Monte-Carlo[10]	Ultra Low (runtime depends on P_c)	Extremely high (simulated truth)	Validation of other algorithms

Table 6.4: Assessment of nonlinear collision probability algorithms

10 An auxiliary development of this doctoral research project was the development of a GP Monte-Carlo collision probability algorithm, which represents a new approach for the validation of pure GP collision probability computations. So far, no comparable technique has been published. The developments are summarized in appendix B.5

Alfano's/Patera's adjoining tube algorithm (with and without the option to perform the P_c computation in Mahalanobis space), McKinley's method, and the most accurate P_c computation according to the Monte-Carlo approach (Alfano, 2009) were implemented as options for nonlinear collision probability computations in SPOOK. As for any new software implementation, it is essential to perform a final validation of the algorithms. A unique and comprehensive source for nonlinear collision probability algorithm validation is Alfano (2009), which not only examines the MC method from the perspective of statistically significant repetition numbers, but also provides 12 two-body test cases against which own implementations/algorithms can be tested[11].

ADJOINING TUBE THREE-DIMENSIONAL COLLISION PROBABILITY WITHOUT COVARIANCE SYMMETRIZATION

In the following the extension of the adjoining tube algorithm is presented which does not require a transformation into Mahalanobis space for the computation of the sectional P_{1D}. To this end, the y-direction in Mahalanobis space is transformed into an encounter frame direction η which, besides y and σ_y, also depends on x, z, σ_x and σ_z.

The first step matches eq. 3.25 in Chan (2008), which is part of the derivation of the reduced two-dimensional STE collision probability by integrating the three-dimensional collision probability with respect to the y-direction:

$$P_c = \underbrace{\int_{-\infty}^{\infty} e^{-\frac{a}{2R}\left(y-\frac{b}{a}\right)^2} dy}_{B_\infty} \underbrace{\iint_{A_{xz}} A e^{-\frac{\frac{x^2}{\sigma_x^2}-\frac{2\rho_{xz}xz}{\sigma_x\sigma_z}+\frac{z^2}{\sigma_z^2}}{2(1-\rho_{xz}^2)}} dx\,dz}_{P_{2Dr}} \tag{6.19}$$

where according to Chan (2008, eq. 3.21):

$$A = 1/\left(\sigma_x\sigma_y\sigma_z\sqrt{(2\pi)^3 R}\right) \tag{6.20}$$

$$R = 1 + 2\rho_{xy}\rho_{yz}\rho_{xz} - \rho_{xy}^2 - \rho_{yz}^2 - \rho_{xz}^2 \tag{6.21}$$

$$a = (1-\rho_{xz}^2)/\sigma_y^2 \tag{6.22}$$

$$b = \frac{(\rho_{xy}-\rho_{xz}\rho_{yz})\,x}{\sigma_x\sigma_y} + \frac{(\rho_{yz}-\rho_{xy}\rho_{xz})\,z}{\sigma_y\sigma_z} \tag{6.23}$$

11 During the validation campaign a couple of unresolved issues were identified in some of the reference cases reported in Alfano (2009). For example, only 5 columns are given for the TCA covariances of scenarios 5, 6, 7, 8 and 11. Likewise, cases 1 and 2 are reported to have equal orbital motion and covariances in the introduction of case 2, however different epoch and TCA state vectors/covariances are given for both cases. Furthermore, the MC result of case 4 could not be validated. In this scenario a significantly smaller MC P_c was determined after 10^6 iterations. Finally, a general slight mismatch was observed for the two-body covariance propagation. An analytic equinoctial two-body covariance propagator and a numeric Shampine-Gordon integrator implementation agreed on the computed Cartesian TCA covariances when starting from the given epoch data, however the results showed numerical differences with respect to those reported by Alfano for the times of closest approach

To rephrase the pdf in B_∞ in the form of a classical Gaussian, it is necessary to define:

$$\eta = \sqrt{\frac{a}{R}}\sigma_y\left(y - \frac{b}{a}\right) \tag{6.24}$$

from which it follows that:

$$dy = \sqrt{\frac{R}{a}}\frac{d\eta}{\sigma_y} \tag{6.25}$$

Equations 6.24 and 6.25 allow to rewrite B_∞ as:

$$B_\infty = \sqrt{\frac{2\pi R}{a}}\left[\frac{1}{\sqrt{2\pi}\sigma_y}\int_{-\infty}^{\infty} e^{-\eta^2/(2\sigma_y^2)}\,d\eta\right] \tag{6.26}$$

The first part of equation 6.26 (before the brackets) completes P_{2Dr} in equation 6.19 to result in the two-dimensional encounter plane collision probability. Thus, the three-dimensional collision probability can be written as:

$$P_c = P_{c,2D}\left[\frac{1}{\sqrt{2\pi}\sigma_y}\int_{-\infty}^{\infty} e^{-\eta^2/(2\sigma_y^2)}\,d\eta\right] \tag{6.27}$$

In the context of the adjoining tube algorithm, the integration over the relative velocity direction is not performed from $-\infty$ to ∞, but from the beginning to the end of a tube section. Hence:

$$P_{1D} = \frac{1}{\sqrt{2\pi}\sigma_y}\int_{\eta_i}^{\eta_f} e^{-\frac{1}{2}\left(\frac{\eta}{\sigma_y}\right)^2}\,d\eta \tag{6.28}$$

which in analogy to equation 6.14 results in:

$$P_{1D} = \frac{1}{2}\left(\operatorname{erf}\left(\frac{\eta_f}{\sqrt{2}\sigma_y}\right) - \operatorname{erf}\left(\frac{\eta_i}{\sqrt{2}\sigma_y}\right)\right) \tag{6.29}$$

Defining $M_f = \eta_f/\sigma_y$ and $M_i = \eta_i/\sigma_y$, equation 6.29 reduces to eq. 6.17, which is therefore also valid in the encounter frame, given the different definitions of M_f and M_i in the encounter and Mahalanobis frames.

To complete the evaluation of the sectional one-dimensional collision probability in the relative velocity direction, a comprehensive description of η shall be derived. To this end, $\frac{b}{a}$ can be computed from equations 6.22 and 6.23:

$$\begin{aligned}\frac{b}{a} &= \left(\frac{(\rho_{xy} - \rho_{xz}\rho_{yz})\,x}{\sigma_x\sigma_y} + \frac{(\rho_{yz} - \rho_{xy}\rho_{xz})\,z}{\sigma_y\sigma_z}\right)\frac{\sigma_y^2}{1-\rho_{xz}^2} \\ &= \left(\frac{\rho_{xy} - \rho_{xz}\rho_{yz}}{1-\rho_{xz}^2}\sigma_y\right)\underbrace{\frac{x}{\sigma_x}}_{M_x} + \left(\frac{\rho_{yz} - \rho_{xy}\rho_{xz}}{1-\rho_{xz}^2}\sigma_y\right)\underbrace{\frac{z}{\sigma_z}}_{M_z}\end{aligned} \tag{6.30}$$

Furthermore, equations 6.21 and 6.22 yield:

$$\sqrt{\frac{a}{R}} = \sqrt{\frac{1-\rho_{xz}^2}{1+2\rho_{xy}\rho_{yz}\rho_{xz}-\rho_{xy}^2-\rho_{yz}^2-\rho_{xz}^2}}\frac{1}{\sigma_y} \tag{6.31}$$

Inserting equations 6.30 and 6.31 into eq. 6.24 and defining $M_y = y/\sigma_y$, the complete form of η is obtained:

$$\eta = \sqrt{\frac{1-\rho_{xz}^2}{1+2\rho_{xy}\rho_{yz}\rho_{xz}-\rho_{xy}^2-\rho_{yz}^2-\rho_{xz}^2}}\,\sigma_y \left(M_y - \frac{\rho_{xy}-\rho_{xz}\rho_{yz}}{1-\rho_{xz}^2}M_x - \frac{\rho_{yz}-\rho_{xy}\rho_{xz}}{1-\rho_{xz}^2}M_z\right) \tag{6.32}$$

Equation 6.32 can be evaluated using the encounter frame relative position and combined covariance information. If all cross-correlation coefficients are set to zero, eq. 6.32 reduces to $\eta = \sigma_y M_y = y$. The variable η can thus be understood as a generalized relative velocity integration direction.

In the context of the adjoining tube algorithm, M_i can thus be computed by evaluating equation 6.32 with the encounter frame quantities of the tube start and dividing by σ_y. The tube end M_f is found by converting the propagated state and covariance into the encounter frame of the tube start and performing the same calculations. The sequential P_{1D} follows from eq. 6.17 or 6.29. The propagated relative GCRF state and combined covariance must also be used to compute the new M_i which, despite corresponding to the same time instant, differs from the previously computed M_f due to the updated encounter frame definition of the new tube section.

The updated algorithm yields results that are numerically consistent with the Mahalanobis frame approach (which uses equation 6.18 to evaluate the sectional collision probability in the Mahalanobis space and either the two-dimensional Mahalanobis formulations of Chan's and Patera's algorithms found in Alfano (2007a) or any other encounter frame STE collision probability algorithm to compute the face contribution of each tube). Since Mahalanobis space transformations are no longer necessary, less time is required to compute the collision probability estimates.

INTEGRATION OF FORCE MODEL UNCERTAINTIES

Finally, the integration of the force model uncertainties into the collision probability algorithm needs to be discussed. The process starts by determining if an object is actually a LEO RSO. Only if the mean motion is found to be larger than 11.25 orbital revolutions per day and the eccentricity smaller than 0.25 (criteria used by SpaceTrack), the LEO flag is activated, which enables the computation of the density and gravity uncertainty for each conjunction partner between the respective epoch time and the determined TCA.

If the nonlinear motion check determines that the conjunction is a long-term encounter, a target and chaser drag-only propagation is performed between TCA $\pm \delta t_{max}$ prior to starting the desired adjoining tube algorithm. The data is used to compute $\Delta\hat{n}/\Delta t$, which is required for the combined density uncertainty computation within the nonlinear collision probability algorithms.

Different propagation sequences are used by McKinley and Alfano to transverse the encounter shell when accumulating the tube collision probabilities. While the adjustment of the integration limits for the gap/overlap minimization causes McKinley to propagate only forward in time, Alfano suggests to propagate from the TCA to the positive and negative time limits of the encounter shell. Alfano's approach has the advantage of a guaranteed start within the encounter region, whereas McKinley's pure forward propagation has to be started at the earliest possible point of the encounter shell, since this is when the P_c rates may be starting to rise from zero. However, if in these early iterations only the time limit is met but not the accompanying sigma-criterion, the tube will not generate an appreciable P_c contribution, such that it is required to directly proceed to the next tube. The approach often leads to many initial propagation steps which exceed the sigma limit, are therefore not necessary and result in prolonged runtimes.

Force model uncertainties can be respected for both strategies in the same manner. A few considerations are however necessary to minimize the workload of the force model consideration. Both encounter region transit strategies require covariance information at many sampling times, which may be shortly before an epoch state or many days after. For this reason, the same process noise calculation strategy as has been used for the Kalman filters has to be applied, i.e. it must be ensured that all force model covariance contributions are removed from the state error covariance before performing covariance propagation. The propagation endpoint force model covariance is evaluated subsequently and the final covariance is determined as the sum of both contributions.

At each time-step the initial drag-only propagation is interpolated to enable a fast determination of $\Delta\hat{n}/\Delta t$ for the combined density uncertainty covariance. The total gravity covariance is formed via accumulation ($Q_{GrvAcc} = CovProp(Q_{GrvAcc}) + Q_{Grv}$), whereby the result of the Higham algorithm is stored in a copy of the accumulated uncertainty ($Q_{GrvUsed} = nearPSD(Q_{GrvAcc})$) to ensure that always at most one nearPSD-iteration is contained within $Q_{GrvUsed}$.

GP covariances are computed as described in section 4.4. The only difference to the SP force model process noise calculation is that due to the analytical nature of the GP theory, the accumulation of the gravity process noise to reduce the workload is no longer necessary. Instead, the gravity process noise can directly be calculated from the RSO epoch states without a negative runtime impact.

6.2 REAL WORLD CATALOG STUDY

This section presents the results of the real world catalog force model uncertainty impact study. The uncertainty quantification methodology using physics-based process noise/covariance matrices is a new and unique approach to assess the impact on collision probabilities, however a hand full of publications which use different methodologies to analyze several aspects of the density uncertainty impact on close encounters already exist. These are briefly reviewed in the following.

Emmert et al. (2014) were the first to study the effect of solar EUV proxy uncertainties on LEO RSO conjunction rates. The work uses a precursor of equation 2.211 to estimate the in-track position error standard deviations of the LEO TLE catalog, which were found to range between a couple of centimeters up to more than a thousand kilometers after seven days of propagation when considering Brownian motion F10.7 input errors. Perigee altitudes above 800 km were found to result in meter-level standard deviations after seven days of propagation, which is in accordance with the altitude dependency analysis conducted in section 4.3.2. Many simplifying assumptions (unperturbed orbits for identifying collisions, estimation of ballistic coefficients, arbitrary choice of radial position error, neglecting the mean motion uncertainty of the primary object, etc.) are made to approximate the corresponding impact on the average conjunction rates. It is demonstrated that the conjunction frequencies increase approximately quadratically with the F10.7 solar background activity. Also an equation for the inverse problem was derived and allows to assess the maximum EUV forecast uncertainty for a given target conjunction frequency and solar background activity.

A computationally very intensive analysis was performed by Bussy-Virat et al. (2018) who use pure Monte-Carlo techniques to analyze the effect of uncertainties in the solar flux and magnetic index model input data on the final collision probability estimates. The study is a good example for the possibility to perform completely numerical analyses of complex relationships due to the availability of increasing computing power. Also Bussy-Virat et al. (2018) validated their MC collision probability algorithm against the Alfano test cases and subsequently investigate 49 conjunctions for changes in the associated collision probability. The input uncertainty pdfs were derived empirically from the comparison of F10.7/Ap forecasts and final data. They turned out to be uncorrelated and bias-free, which justifies the independent perturbation of both parameters with respect to the nominal values obtained from space weather predictions (the same approach is also chosen for SPOOK). While there is nothing to be criticized in terms of the applied methodology, a few points in the study deserve further discussion. It is claimed that analytical uncertainty estimates lack the accuracy required to conduct CARA. This statement must be contradicted. The results from chapters 2-5 demonstrate clearly

that even highly non-Gaussian pdfs can be determined accurately and considerably faster than with MC techniques. It is noted further that Emmert et al. (2017) consider drag as the only perturbation, which is deemed non-representative of real world orbits. Section 2.7.3 however has shown that the consideration of analytical density uncertainties can be extended to arbitrary disturbances by minimal adjustments to the covariance propagation. Finally, the study uses overly optimistic epoch covariances (less than 10 m position error standard deviation), which is not representative of actual conditions and emphasizes the relative impact of the density uncertainties (cf. chapter 5).

The key result of Bussy-Virat et al. (2018) is that for the 49 cases examined, the changes in the collision probability were approximately within an order of magnitude and thus, in contrast to the interpretation in the paper, just above the operational relevance threshold. The role of solar bursts was also investigated. The maximum P_c change was found to be 30.4%, which is significantly below an OOM. The operational impact of the storms on the presented cases is therefore not of practical relevance, despite the fact that for one of the cases the collision probability decreases just below the conventional HIE threshold[12].

A very interesting study which examines the density error treatment of the CSpOC system was authored by Hejduk and Snow (2018) who reexamine archived records of actual collisions by artificially injecting density errors into the events and subsequently recalculating the collision probability.

Their main finding is that the quantification of density errors greatly increases the stability and durability of the CARA results. It was also concluded that more accurate density models help to improve CA, although they do not eliminate the need for uncertainty quantification. The study uses the NASA three-stage encounter classification scheme (cf. table 6.5) to assess the collision risk which, in a slightly modified form, is also used in SPOOK.

Category	NASA	SPOOK
Red (HIE)	$\geqslant 4.4 \cdot 10^{-4}$	$\geqslant 10^{-4}$
Yellow (Potential HIE)	$\geqslant 10^{-7}$	$\geqslant 10^{-7}$
Green (Low Risk)	$< 10^{-7}$	$\geqslant 10^{-10}$
Gray (Very Low Risk)	N/A	$< 10^{-10}$

Table 6.5: Encounter risk classification

The most recent study on the topic was conducted by Gondelach and Linares (2020). It deals with the effect of grid-scale density modeling errors on the collision probability (space weather input errors are not considered). An interesting claim of the paper relates to the importance of target/chaser covariance correlation in case of similar ballistic coefficients. Previous studies have so far attributed little importance

12 The 10^{-4} threshold is not a hard operational limit which directly triggers the maneuvering decision. Some missions at NASA also use $4.4 \cdot 10^{-4}$ to *consider* a CAM (Newman et al., 2014)

to this aspect (cf. section 2.6 in Chan, 2008 or the 7% P_c change in Coppola et al., 2004). However, the latter study also emphasizes that further research may reveal very different results. Recently, Casali et al. (2019) examined hundreds of archived NASA conjunctions for the effect of cross-correlation. The study confirms the findings of the earlier works in that the consideration of cross-correlations leads to covariance deflation, which in general can result in smaller or larger collision probabilities (changes by a factor of 0.5 to 2.0 were found in one third of the studied cases). Another important result of Casali et al. (2019) is that the collision probabilities typically decrease and rarely experience a more than twofold increase due to cross-correlation, which is operationally insignificant. Neglecting cross-correlations is therefore conservative in terms of the risk metric. However, the study also mentions that special geometries can be identified or constructed for which the effect is greater. Scenario S0 in Gondelach and Linares (2020) seems to be such a case, as it demonstrates a P_c increase above one OOM.

While it is undisputed that cross-correlation is negligible for near head-on geometries (Chan, 2008; Casali et al., 2019), the influence of drag-sensitivity has not yet been fully understood. Gondelach and Linares (2020) highlight the importance of equal ballistic coefficients for the impact of cross-correlation and show negligible contribution for differing ballistic coefficients, whereas Casali et al. (2019) found that "if the drag sensitivities are comparable or identical for the two satellites, through some combination of energy dissipation rate and propagation time, the effect again vanishes, even for significant drag cases".

Shelton and Junkins (2019) do not specifically address density uncertainty, but are interested in the impact of parametric model uncertainty on the collision probability. Instead of using process noise, they extend the state vector with the uncertain parameters of interest and model their impact on the propagation. The study is limited to temporally and spatially constant parameter uncertainty, which restricts the practical applicability. The P_c calculation is carried out using a modern approach that has its roots in Coppola (2012) and can take velocity uncertainty and non-Gaussian uncertainty volumes into account. A fundamental result of the work is that the consideration of the model uncertainties does not affect the P_c integral, but is only relevant for covariance propagation. Depending on the considered ballistic coefficient uncertainty, a P_c variability of up to two OOM is demonstrated for a HIE case with a Monte-Carlo collision probability larger than 1%. Further tests would be desirable to examine if this P_c sensitivity also applies to non-HIE conjunctions. The presented 15 test cases do not contain any HIE for which the difference between the non-Gaussian and single Gaussian P_c computations differ by more than an order of magnitude, which is in line with Ghrist and Plakalovic (2012). Some of the presented non-HIE cases however show differences of many OOM, which suggests a non-negligible impact of non-Gaussian P_c computations for medium to low risk conjunctions.

Force model uncertainties can both increase or mitigate the probability of collision. This can be understood by considering the possible effects on the time of closest approach and the associated distance at closest approach (DCA) or in terms of the covariance inflation, which increases the probability of collision if both RSOs are rather distant, or lowers it if both conjunction partners are close (cf. also figure 1.9).

6.2.1 *Conjunction analysis risk assessment*

Based on the processing of the catalogs, SPOOK generates output files similar to the one shown in figure 6.9. Descriptive information is contained in the file header (e.g. the selection of the smart sieve filters and the screening volume or the chosen algorithms to compute the STE and LTE collision probabilities). Following the header, each row contains the desired information of the conjunctions, such as the conjuncting RSOs, the data set record age at TCA or the perturbation theories used.

```
CONJUNCTION SCREENING RESULTS
-----------------------------------------------
GENERATED BY:  AIRBUS DEFENCE AND SPACE
CREATED VIA:   SPOOK
CREATION DATE: 2020-04-22T07:14:29 +0200

SCREENING PARAMETERS:
Apogee/Perigee Filter: YES (Pad: 250.0km)
X-Sieve: YES
Y-Sieve: YES
Z-Sieve: YES
Rel. Distance-Sieve: YES
Min. Distance-Sieve: YES
Fine Distance-Sieve: YES
Ellipsoid-Sieve: YES (N2.0 x T5.0 x W2.0km)
Pc computation (Short-Term Encounters): YES -> ALFANO (M-SERIES)
Pc computation (Long-Term Encounters):  YES -> ALFANO ADJOINING TUBES (COV SYMMETRIZATION)

Screening start: 2020-02-25T03:00:00.000
Screening duration: 4.500 days
Theory used for computation: SP + ITC
Objects loaded from target catalog: 18846
Objects loaded from chaser catalog: 18846
-----------------------------------------------
Detected conjunctions: 11854
-----------------------------------------------

  NORAD_T    NORAD_C              SAT_NAME_T             SAT_NAME_C       PERT_T     PERT_C   REC_AGE_AT_TCA_T [d] REC_AGE_AT_TCA_C [d]
     3743      19174              SL-8 DEB               SL-14 DEB           SP         SP          0.03393900394        0.12473959429
    35954      41066  DMSP 5D-3 F18 DEB              NOAA 16 DEB           SP         SP          0.08202196285       -0.05793418270
    18958      33909           COSMOS 1933          COSMOS 2251 DEB         SP         SP          0.12531375419        0.09137096489
    17624      38483    COSMOS 1275 DEB            COSMOS 2251 DEB         SP         SP          0.12616741098        0.12616498070
    21358      38361         DELTA 1 DEB               OPS 4682 DEB         SP         SP         -0.01814446039        0.12584895408
    37460      82500      FENGYUN 1C DEB  *NAME_NOT_IN_CATALOG*            SP         GP          0.12675206782        2.36107867630
    41973      42023          FLOCK 3P-9               FLOCK 3P-32          SP         SP         -0.02091154549        0.12702113995
    30972      35243      FENGYUN 1C DEB            FENGYUN 1C DEB         SP         SP          0.01889516972        0.12747603795
    28654      30572             NOAA 18            FENGYUN 1C DEB         SP         SP          0.01231086254        0.08263286622
     6148      39300          COSMOS 514                SL-16 DEB          SP         SP          0.02417591913        0.12885716930
    21544      37788         DELTA 1 DEB                   EDUSAT          SP         SP         -0.02878353372        0.12988072587

          TCA (JD)                TCA (UTC)      DAYS_SINCE_EPOCH      DELTA_R_TCA_I [km]      DELTA_R_TCA_J [km]  DELTA_R_TCA_K [km]
2458904.62502851803  2020/02/25T03:00:02.464        0.00002851803             1.47766660             -0.52337348         -0.99523201
2458904.62507128017  2020/02/25T03:00:06.159        0.00007128017             0.54301211             -2.11334419         -0.99985904
2458904.62546033971  2020/02/25T03:00:39.773        0.00046033971            -0.23447132             -1.10284498         -1.29796464
2458904.62633457547  2020/02/25T03:01:55.307        0.00133457547            -1.86778390              0.15739721          1.03547209
2458904.62641963689  2020/02/25T03:02:02.657        0.00141963689            -1.32140614             -0.21546657         -0.95633975
2458904.62679058639  2020/02/25T03:02:34.707        0.00179058639             0.02956572             -0.47470542         -0.49378737
2458904.62723067682  2020/02/25T03:03:12.730        0.00223067682             0.10973307             -0.11069401          0.95641247
2458904.62753018131  2020/02/25T03:03:38.608        0.00253018131             0.73245332              0.83207244         -0.35481715
2458904.62768070027  2020/02/25T03:03:51.612        0.00268070027             0.49932309             -2.87033731         -0.24923520
2458904.62906933343  2020/02/25T03:05:51.590        0.00406933343            -1.27079488              0.66283296         -0.24510822
2458904.62998327194  2020/02/25T03:07:10.555        0.00498327194             0.20960299              0.51373520          1.37021294

DELTA_V_TCA_I [km/s]      DELTA_V_TCA_J [km/s]   DELTA_V_TCA_K [km/s]   ABS_REL_DIST_TCA [km]  ABS_REL_VEL_TCA [km/s]    K_Sq_Ellipse [1]
         2.66543547               -8.66977593             8.51354066             1.85685356             12.43985271          0.71932676
        -6.96895783               -2.41072522             1.31054108             2.40016748              7.48969206          0.76172433
       -10.07082713               -6.34809623             7.20956742             1.71928940             13.91753380          0.72423332
        -0.59838469               -3.24729508            -0.58549099             2.14139978              3.35347421          0.75265745
        -7.17605347                9.56430638             7.75854458             1.64533454             14.25365616          0.66985032
         1.78076856               10.69126940           -10.17981548             0.68559852             14.86953333          0.11747275
        -0.00116710               -0.00007500             0.00012523             0.96903010              0.00117619          0.23393376
         6.42365545               -9.38991462            -8.74340012             1.16392767             14.34855016          0.32294517
         2.63575984                0.49708658            -0.44495335             2.92408584              2.71888001          0.41279151
         2.80913900                0.18126748           -14.07079292             1.45407882             14.34961090          0.51044891
         1.60614114               -1.96938687             0.49279148             1.47828981              2.58863235          0.47452981

          Pc [1]      Pc_max (Alfriend) [1]      Pc_max (Chan) [1]
      4.2070E-13                2.5343E-09                6.2365E-09
      9.0559E-10                3.8458E-09                4.9857E-09
      2.4553E-11                6.8215E-08                3.1069E-07
      2.0720E-11                1.9630E-10                9.7072E-10
      1.2610E-09                9.3113E-09                6.4708E-05
      9.7442E-07                1.0055E-06                2.0117E-05
      1.7885E-08          N/A (long enc.)           N/A (long enc.)
      1.2016E-09                1.4193E-09                1.5583E-09
      5.9547E-07                7.1681E-07                3.3597E-04
      2.7949E-07                3.7336E-07                3.6277E-04
      7.7331E-11                1.4877E-09                1.1380E-08
```

Figure 6.9: Exemplary SPOOK output for an all vs. all catalog screening run. For better readability the columns have been positioned below each other

To conduct a comprehensive operational risk assessment, not only the actual collision probability is calculated, but also the maximum collision probabilities according to Alfriend et al. (1999) and Chan (2008, chapter 11), which allows to evaluate the impact of the force model uncertainties on these risk indicators.

Alfriend's maximum collision probability scales the entire combined covariance with a scalar k^2 to obtain the corresponding maximum P_c. Chan on the other hand does not explicitly use scaling parameters, but derives an expression of the maximum collision probability which is independent of the covariance size and only based on the covariance orientation angle. Assuming that the combined covariance orientation of the estimated covariance is appropriate, Chan's approach is equivalent to using individual scaling parameters for each of the encounter plane x/z covariance components. Chan's $P_{c,max}$ results are thus usually some orders of magnitude above those of Alfriend. Hence, if even Chan's maximum collision probability does not classify a conjunction as HIE, it can be excluded from further analysis with a high degree of confidence.

A Python tool was developed for the collision assessment risk analysis. It allows to import a single or multiple collision screening output files and supports the following analyses:

- Temporal evolution of conjunction
- Encounter classification according to table 6.5 (P_c, Alfriend's $P_{c,max}$, Chan's $P_{c,max}$)
- Absolute comparison of collision probability histograms
- In-depth histogram delta analysis (conjunction risk classification changes)

These instruments support the risk analysis and allow interesting comparisons, such as SP vs. SP against GP vs. GP catalog screenings, or examining the impact of the force model uncertainties on the catalog scenarios. The tool also supports the generation of CDMs for all conjunctions or a desired subset, which can then be sent to the respective Os/Os (where at least one active spacecraft is involved).

Two practical applications of the CARA tool are illustrated in figures 6.10 and 6.11, which depict the CDM export functionality of the data browser[13] and the in-depth risk classification changes functionality. The results shown in figure 6.11 are to be interpreted as follows: due to the consideration of the force model uncertainties 419 out of 14754 conjunctions changed their risk classification. Five conjunctions which were considered as low risk prior to accounting for force model uncertainties became very low risk conjunctions (gray color code) afterwards. On the contrary, 241 formerly gray conjunctions and 26 formerly yellow conjunctions are classified as low risk (green category) after taking density and gravity uncertainties into account. The

13 SPOOK uses the pandastable library (Farrell, 2016) as baseline for individual conjunction analyses and data browsing

figure also depicts the rare case of a non-neighboring category change for 82 formerly gray conjunctions, which experienced changes of at least three OOM to be classified as potential HIEs when including force model uncertainties. Similarly, 43 green and 20 red conjunctions switched their color to yellow. The results correspond to a pure GP all vs. all category screening (not limited to LEOs, for which the possibility of category changes arises) using the methodology explained in section 4.4 and the epoch covariance lookup tables (LUTs) from Flohrer et al. (2008), scaled by a factor of $k^2 = 0.1^2$ to assess the force model uncertainty impact if precise epoch covariances were available.

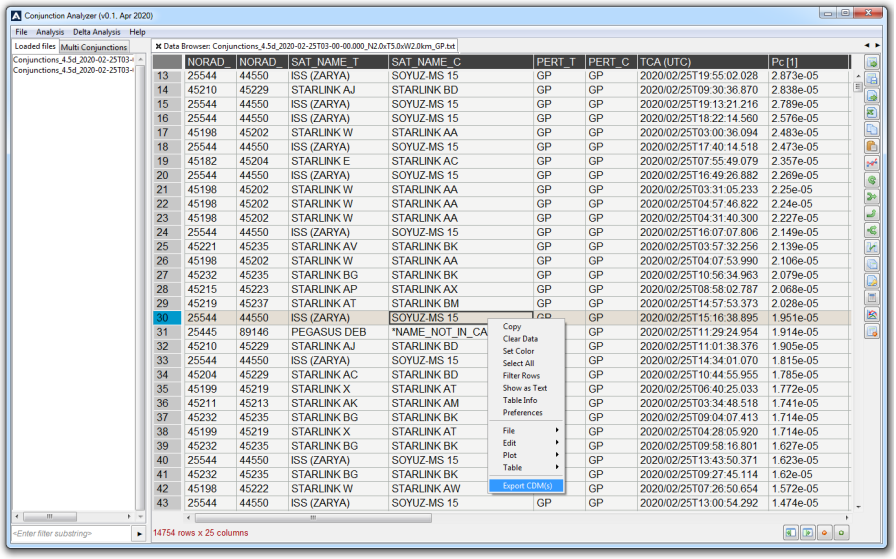


Figure 6.10: CDM generation capability of the conjunction risk analysis tool

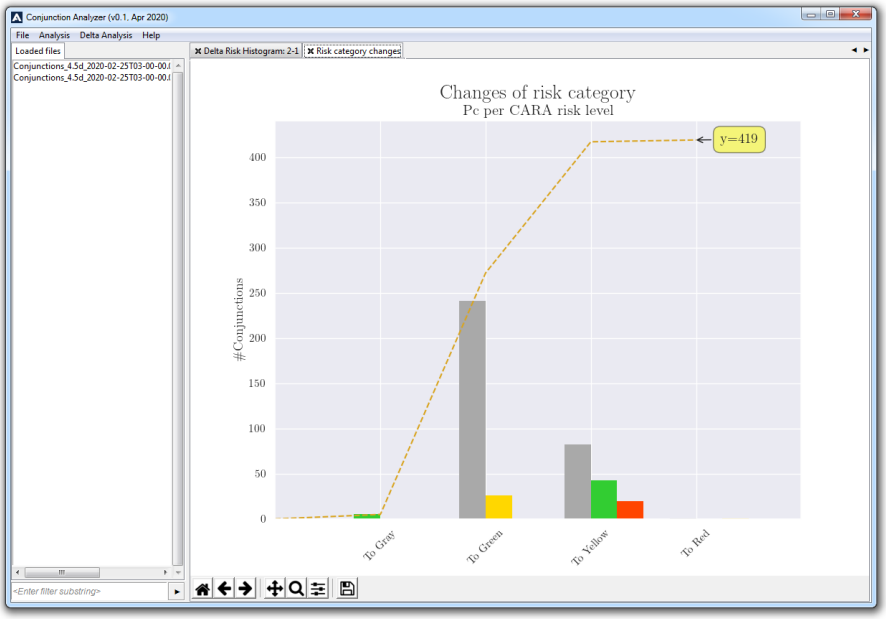


Figure 6.11: Risk category changes when considering force model uncertainty

Further interesting analyses can be performed using the temporal conjunction evolution feature of the conjunction analyzer tool. An example is depicted in figure 6.12 and demonstrates the evolution of the collision risk for the twin 1.5U CubeSats AeroCube 10A (JimSat) and 10B (DougSat), which have been developed by the Aerospace Corporation of El Segundo as technology demonstrators. Repeated conjunctions were identified as part of the SP catalog screenings (purple markers) and the GP screenings using the public TLE catalog (cyan markers). The tooltips highlight the differences for a conjunction which has been identified using SP and GP perturbations. The TCA difference is only 11 ms, however the more accurate SP dataset yields an over four times larger collision probability than the GP conjunction analysis. The P_c results converge towards the gray classification (very low risk) with progressing time.

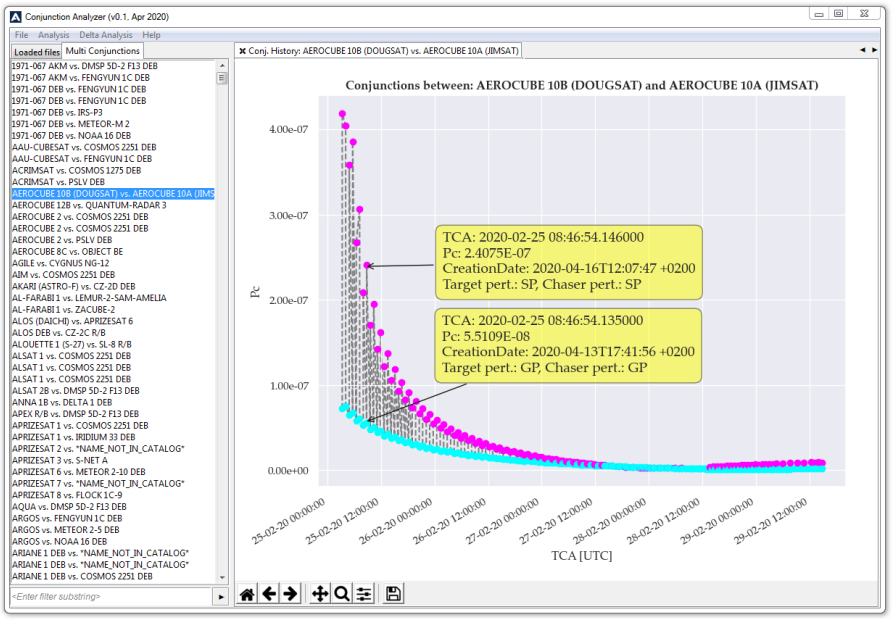


Figure 6.12: SP vs. GP screening: temporal evolution of conjunction between AeroCube 10A and AeroCube 10B

6.2.2 *Impact of force model uncertainties on the catalog collision probability distribution*

The force model uncertainty impact on the SP and GP catalogs of February 25th, 2020 was examined in relation to the size of the epoch covariance matrix $\hat{\mathbf{P}}_0$, which allows an assessment of the operational impact in terms of the quality of the available data, as well as potentially more accurate future catalog information.

The epoch covariance $\hat{\mathbf{P}}_0$ is taken as a scaled version of table 6.6 ($\hat{\mathbf{P}}_0 = k^2\mathbf{P}_{ini}$). A change of the collision probability is only possible for LEO RSOs, nevertheless the screening was always conducted using the entire catalog as possible chasers. The following subsets of operational

$e < 0.1$		$i < 30$ deg	30 deg < 60 deg	$i < 60$ deg
$h_p < 800$ km	R	67	107	115
	T	118	308	517
	N	75	169	137
800 km $< h_p$ $< 25\,000$ km	R	191	71	91
	T	256	228	428
	N	203	95	114
$h_p > 25\,000$ km	R	357	-	-
	T	432	-	-
	N	83	-	-
$e \geqslant 0.1$		$i < 30$ deg	30 deg < 60 deg	$i < 60$ deg
$h_p < 800$ km	R	2252	629	494
	T	4270	909	814
	N	1421	2057	1337
800 km $< h_p$ $< 25\,000$ km	R	1748	1832	529
	T	3119	1878	817
	N	971	1454	1570
$h_p > 25\,000$ km	R	402	4712	-
	T	418	6223	-
	N	83	1208	-

Table 6.6: Prior scaling $\mathbf{P}_{ini}$ RTN epoch position covariance standard deviation [m] lookup table according to Flohrer et al. (2008, table 2). The velocity variances are chosen as: $\sigma^2_{v_R}, \sigma^2_{v_N} = 10^{-3}\ m^2/s^2$ and $\sigma^2_{v_T} = 5 \cdot 10^{-3}\ m^2/s^2$

interest were analyzed over a screening period of 4.5 days: all vs. all, LEO vs. all and payload vs. all. The results are presented in tables 6.7-6.48 and grouped according to perturbation theory and the type of collision probability (P_c, Alfriend $P_{c,max}$, Chan $P_{c,max}$). Each table displays the information produced by a category change analysis of the CARA tool (cf. figure 6.11).

ALL VS. ALL SCREENING

Perturbation theory: SP	**Initial covariance**: $k^2 = 0.5 \cdot 0.01^2$		**Collision probability**: P_c	
#Conjunctions: 11854	**#Category changes**: 343		**Reclassification ratio**: 2.89%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	11792	30	28	4
Combined density and gravity covariance	11463	271	117	3

Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
250	79	0	0	11	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	2	0	0	0	1

Table 6.7: Impact of force model uncertainties on SP catalog P_c: $k^2 = 0.5 \cdot 0.01^2$

Perturbation theory: SP	**Initial covariance**: $k^2 = 0.1 \cdot 0.1^2$		**Collision probability**: P_c	
#Conjunctions: 11854	**#Category changes**: 337		**Reclassification ratio**: 2.84%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	11342	266	241	5
Combined density and gravity covariance	11042	506	301	5
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
263	37	0	0	30	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	7	0	0	0	0

Table 6.8: Impact of force model uncertainties on SP catalog P_c: $k^2 = 0.1 \cdot 0.1^2$

Perturbation theory: SP	**Initial covariance**: $k^2 = 0.5 \cdot 0.1^2$		**Collision probability**: P_c	
#Conjunctions: 11854	**#Category changes**: 235		**Reclassification ratio**: 1.98%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	10712	768	368	6
Combined density and gravity covariance	10519	941	389	5
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
178	15	0	0	23	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	18	0	0	0	1

Table 6.9: Impact of force model uncertainties on SP catalog P_c: $k^2 = 0.5 \cdot 0.1^2$

Perturbation theory: SP	**Initial covariance**: $k^2 = 0.5 \cdot 0.5^2$		**Collision probability**: P_c	
#Conjunctions: 11854	**#Category changes**: 238		**Reclassification ratio**: 2.01%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	7499	3534	820	1
Combined density and gravity covariance	7311	3740	802	1
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
181	8	0	1	11	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	37	0	0	0	0

Table 6.10: Impact of force model uncertainties on SP catalog P_c: $k^2 = 0.5 \cdot 0.5^2$

Perturbation theory: SP	**Initial covariance**: $k^2 = 0.1 \cdot 1.0^2$		**Collision probability**: P_c	
#Conjunctions: 11854	**#Category changes**: 218		**Reclassification ratio**: 1.84%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	7666	3300	888	0
Combined density and gravity covariance	7494	3494	866	0
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
170	4	0	2	8	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	34	0	0	0	0

Table 6.11: Impact of force model uncertainties on SP catalog P_c: $k^2 = 0.1 \cdot 1.0^2$

Perturbation theory: SP		**Initial covariance**: $k^2 = 0.5 \cdot 1.0^2$		**Collision probability**: P_c	
#Conjunctions: 11854		**#Category changes**: 165		**Reclassification ratio**: 1.39%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		5014	5975	865	0
Combined density and gravity covariance		4909	6104	841	0
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
110	2	0	7	10	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	36	0	0	0	0

Table 6.12: Impact of force model uncertainties on SP catalog P_c: $k^2 = 0.5 \cdot 1.0^2$

Perturbation theory: SP		**Initial covariance**: $k^2 = 1.0 \cdot 1.0^2$		**Collision probability**: P_c	
#Conjunctions: 11854		**#Category changes**: 53		**Reclassification ratio**: 0.45%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		3607	7593	654	0
Combined density and gravity covariance		3599	7614	641	0
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
22	0	0	14	2	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	15	0	0	0	0

Table 6.13: Impact of force model uncertainties on SP catalog P_c: $k^2 = 1.0 \cdot 1.0^2$

The results of the SP all vs. all screening are summarized by ascending covariance scaling factors in tables 6.7-6.13. To ensure that the initial covariance of SP RSOs is smaller than for GP RSOs in case of mixed SP vs. GP conjunctions, the value of k^2 is always chosen as $a \cdot k_{GP}^2$, where a represents the SP vs. GP reduction factor.

The majority of the results agree with intuitive expectations. The force model uncertainty consideration causes category changes towards more and less critical risk classifications. For small initial covariances most conjunctions are of very low risk (gray category) since the miss distances are significantly larger than the standard deviations. The impact of the physics-based process noise consideration increases for decreasing epoch state error covariances, as the dominant covariance growth factor is shifting from the initial radial uncertainty to the force model uncertainty consideration.

The current epoch SP catalog data is probably close to the case of table 6.10, which depicts a moderate overall impact, since most category changes are from very low risk to low risk (+206 low risk events, -188 very low risk events, -18 potential HIEs, unchanged number of HIEs). There was no reclassification from potential HIE (yellow) to HIE (red) for the SP catalog of February 25th, 2020. Both observations are not fully in line with initial expectations, as a larger HIE impact was suspected. Nevertheless, non-neighboring category changes (P_c changes larger than three OOM) became apparent in all cases except for table 6.13.

In the following, the GP screening results are presented for comparison. The absolute number of identified conjunctions is about 25% larger, however the reclassification ratios are almost identical to the SP screening from $k^2 = 0.5^2$ onwards. Reclassifications as HIE are observed for $k^2 \leqslant 0.1^2$. A significant increase of category changes can be observed for $k^2 = 0.01^2$ (table 6.14), which in comparison to table 6.7, even displays 25 conjunctions that are reclassified from very low risk to HIE (P_c change of at least six OOM). Table 6.15 corresponds to figure 6.11, which depicts a graphical visualization of the category changes. Tables 6.14 to 6.18 clearly highlight the criticality of the initial epoch covariances when interpreting studies that aim to assess the impact of force model uncertainties on the collision probabilities.

Perturbation theory: GP	**Initial covariance**: $k^2 = 0.01^2$		**Collision probability**: P_c	
#Conjunctions: 14754	**#Category changes**: 1115		**Reclassification ratio**: 7.56%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	14528	102	106	18
Combined density and gravity covariance	13499	567	616	72
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
510	494	25	0	40	10
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	5	25	0	0	6

Table 6.14: Impact of force model uncertainties on GP catalog P_c: $k^2 = 0.01^2$

Perturbation theory: GP	**Initial covariance**: $k^2 = 0.1^2$		**Collision probability**: P_c	
#Conjunctions: 14754	**#Category changes**: 419		**Reclassification ratio**: 2.84%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	12142	1501	1018	93
Combined density and gravity covariance	11823	1720	1136	75
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
241	82	1	5	43	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	26	1	0	0	20

Table 6.15: Impact of force model uncertainties on GP catalog P_c: $k^2 = 0.1^2$

Perturbation theory: GP	**Initial covariance**: $k^2 = 0.5^2$		**Collision probability**: P_c	
#Conjunctions: 14754	**#Category changes**: 296		**Reclassification ratio**: 2.01%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	7303	5817	1614	20
Combined density and gravity covariance	7127	5993	1614	20
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
196	4	0	24	34	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	38	0	0	0	0

Table 6.16: Impact of force model uncertainties on GP catalog P_c: $k^2 = 0.5^2$

Perturbation theory: GP	**Initial covariance**: $k^2 = 1.0^2$		**Collision probability**: P_c	
#Conjunctions: 14754	**#Category changes**: 126		**Reclassification ratio**: 0.85%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	4921	8259	1574	0
Combined density and gravity covariance	4899	8293	1592	0
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
57	0	0	35	11	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	23	0	0	0	0

Table 6.17: Impact of force model uncertainties on GP catalog P_c: $k^2 = 1.0^2$

Perturbation theory: GP	**Initial covariance**: $k^2 = 2.0^2$		**Collision probability**: P_c	
#Conjunctions: 14754	**#Category changes**: 69		**Reclassification ratio**: 0.47%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	5180	8484	1090	0
Combined density and gravity covariance	5207	8465	1082	0
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
11	0	0	38	6	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	14	0	0	0	0

Table 6.18: Impact of force model uncertainties on GP catalog P_c: $k^2 = 2.0^2$

Next, the scenarios are analyzed for the impact of the force model consideration on the maximum collision probability according to Alfriend et al. (1999), who determine the $P_{c,max}$ when scaling the covariance using an arbitrary scalar k^2. Since the same dependency is also used to study the importance of the epoch covariances, the risk classification without process noise consideration is identical for all cases of a perturbation theory. Likewise, most conjunctions are classified as low risk instead of very low risk when examining the maximum single scalar-scaling collision probability.

The overall impact of the physics-based covariance matrices on this metric is reduced, since the k^2 scaling invariance is able to reflect the dominant in-track covariance growth due to force modeling errors. Different risk classifications thus arise mainly in cases that are poorly represented by a single scaling parameter, i.e. when the covariance growth is strongly dominated by the force model uncertainty consideration, as then the different radial and transverse position error growth rates are significant (cf. table 6.19). Nevertheless also individual HIE reclassifications are possible (cf. table 6.22). Overall, the GP results are very similar to the respective SP screenings.

Since maximum collision calculations are not directly transferable to long-term encounters (Alfano, 2006, section 2.C), the absolute conjunction number is reduced by the amount of identified LTEs (358 conjunctions).

Perturbation theory: SP		**Initial covariance**: $k^2 = 0.5 \cdot 0.01^2$		**Collision probability**: Alfriend $P_{c,max}$	
#Conjunctions: 11496		**#Category changes**: 1174		**Reclassification ratio**: 10.21%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		2264	7726	1493	13
Combined density and gravity covariance		1604	8163	1709	20
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
722	0	0	62	303	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	80	7	0	0	0

Table 6.19: Impact of force model uncertainties on SP Alfriend $P_{c,max}$: $k^2 = 0.5 \cdot 0.01^2$

Perturbation theory: SP		**Initial covariance**: $k^2 = 0.5 \cdot 0.1^2$		**Collision probability**: Alfriend $P_{c,max}$	
#Conjunctions: 11496		**#Category changes**: 185		**Reclassification ratio**: 1.61%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		2264	7726	1493	13
Combined density and gravity covariance		2239	7780	1463	14
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
67	0	0	42	23	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	52	1	0	0	0

Table 6.20: Impact of force model uncertainties on SP Alfriend $P_{c,max}$: $k^2 = 0.5 \cdot 0.1^2$

Perturbation theory: SP		**Initial covariance**: $k^2 = 0.5 \cdot 0.5^2$		**Collision probability**: Alfriend $P_{c,max}$	
#Conjunctions: 11496		**#Category changes**: 60		**Reclassification ratio**: 0.52%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		2264	7726	1493	13
Combined density and gravity covariance		2275	7721	1486	14
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
4	0	0	15	17	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	23	1	0	0	0

Table 6.21: Impact of force model uncertainties on SP Alfriend $P_{c,max}$: $k^2 = 0.5 \cdot 0.5^2$

Perturbation theory: SP		**Initial covariance**: $k^2 = 0.5 \cdot 1.0^2$		**Collision probability**: Alfriend $P_{c,max}$	
#Conjunctions: 11496		**#Category changes**: 38		**Reclassification ratio**: 0.38%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		2264	7726	1493	13
Combined density and gravity covariance		2264	7721	1497	14
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
5	0	0	5	16	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	11	1	0	0	0

Table 6.22: Impact of force model uncertainties on SP Alfriend $P_{c,max}$: $k^2 = 0.5 \cdot 1.0^2$

Perturbation theory: SP	**Initial covariance**: $k^2 = 0.5 \cdot 2.0^2$		**Collision probability**: Alfriend $P_{c,max}$	
#Conjunctions: 11496	**#Category changes**: 24		**Reclassification ratio**: 0.21%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	2264	7726	1493	13
Combined density and gravity covariance	2263	7718	1502	13
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
3	0	0	2	14	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	5	0	0	0	0

Table 6.23: Impact of force model uncertainties on SP Alfriend $P_{c,max}$: $k^2 = 0.5 \cdot 2.0^2$

Perturbation theory: GP	**Initial covariance**: $k^2 = 0.01^2$		**Collision probability**: Alfriend $P_{c,max}$	
#Conjunctions: 13401	**#Category changes**: 1161		**Reclassification ratio**: 8.66%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	3098	8719	1567	15
Combined density and gravity covariance	2502	9082	1804	13
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
686	0	0	90	307	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	74	1	0	0	3

Table 6.24: Impact of force model uncertainties on GP Alfriend $P_{c,max}$: $k^2 = 0.01^2$

Perturbation theory: GP	**Initial covariance**: $k^2 = 0.1^2$		**Collision probability**: Alfriend $P_{c,max}$	
#Conjunctions: 13401	**#Category changes**: 175		**Reclassification ratio**: 1.31%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	3098	8719	1567	15
Combined density and gravity covariance	3102	8747	1538	14
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
49	0	0	53	20	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	52	0	0	0	1

Table 6.25: Impact of force model uncertainties on GP Alfriend $P_{c,max}$: $k^2 = 0.1^2$

Perturbation theory: GP	**Initial covariance**: $k^2 = 0.5^2$		**Collision probability**: Alfriend $P_{c,max}$	
#Conjunctions: 13401	**#Category changes**: 68		**Reclassification ratio**: 0.51%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	3098	8719	1567	15
Combined density and gravity covariance	3110	8721	1555	15
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
12	0	0	24	9	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	23	0	0	0	0

Table 6.26: Impact of force model uncertainties on GP Alfriend $P_{c,max}$: $k^2 = 0.5^2$

Perturbation theory: GP	**Initial covariance**: $k^2 = 1.0^2$		**Collision probability**: Alfriend $P_{c,max}$	
#Conjunctions: 13401	**#Category changes**: 45		**Reclassification ratio**: 0.34%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	3098	8719	1567	15
Combined density and gravity covariance	3104	8714	1568	15

Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
6	0	0	12	13	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	14	0	0	0	0

Table 6.27: Impact of force model uncertainties on GP Alfriend $P_{c,max}$: $k^2 = 1.0^2$

Perturbation theory: GP	**Initial covariance**: $k^2 = 2.0^2$		**Collision probability**: Alfriend $P_{c,max}$	
#Conjunctions: 13401	**#Category changes**: 21		**Reclassification ratio**: 0.16%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	3098	8719	1567	15
Combined density and gravity covariance	3098	8712	1576	15

Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
3	0	0	3	11	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	4	0	0	0	0

Table 6.28: Impact of force model uncertainties on GP Alfriend $P_{c,max}$: $k^2 = 2.0^2$

General properties of the Chan $P_{c,max}$ are depicted in figure 1.11. It is insensitive to the covariance size and only depends on the covariance orientation. Thus, force model uncertainties are expected to have very little impact on this absolute maximum collision probability. Category changes only arise if the force model consideration results in an appreciable covariance orientation change. The GP results are again very similar to the respective SP screenings.

There are thousands of potential high risk events and hundreds of HIEs for Chan's maximum collision probability. It is therefore not advisable to base the general risk assessment on this metric, however, conjunctions with a low risk Chan $P_{c,max}$ can be excluded from a detailed analysis with a high confidence level.

Perturbation theory: SP	**Initial covariance**: $k^2 = 0.5 \cdot 0.01^2$		**Collision probability**: Chan $P_{c,max}$	
#Conjunctions: 11496	**#Category changes**: 109		**Reclassification ratio**: 0.95%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	0	6834	4170	492
Combined density and gravity covariance	0	6828	4185	483

Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
0	0	0	0	32	1
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	24	21	0	3	8

Table 6.29: Impact of force model uncertainties on SP Chan $P_{c,max}$: $k^2 = 0.5 \cdot 0.01^2$

Perturbation theory: SP		**Initial covariance**: $k^2 = 0.5 \cdot 0.1^2$		**Collision probability**: Chan $P_{c,max}$	
#Conjunctions: 11496		**#Category changes**: 31		**Reclassification ratio**: 0.27%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		0	6834	4170	492
Combined density and gravity covariance		0	6838	4166	492
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
0	0	0	0	2	1
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	5	11	0	2	10

Table 6.30: Impact of force model uncertainties on SP Chan $P_{c,max}$: $k^2 = 0.5 \cdot 0.1^2$

Perturbation theory: SP		**Initial covariance**: $k^2 = 0.5 \cdot 0.5^2$		**Collision probability**: Chan $P_{c,max}$	
#Conjunctions: 11496		**#Category changes**: 20		**Reclassification ratio**: 0.17%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		0	6834	4170	492
Combined density and gravity covariance		0	6836	4168	492
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
0	0	0	0	1	1
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	3	7	0	1	7

Table 6.31: Impact of force model uncertainties on SP Chan $P_{c,max}$: $k^2 = 0.5 \cdot 0.5^2$

Perturbation theory: SP		**Initial covariance**: $k^2 = 0.5 \cdot 1.0^2$		**Collision probability**: Chan $P_{c,max}$	
#Conjunctions: 11496		**#Category changes**: 15		**Reclassification ratio**: 0.13%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		0	6834	4170	492
Combined density and gravity covariance		0	6832	4169	495
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
0	0	0	0	2	1
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	0	7	0	1	4

Table 6.32: Impact of force model uncertainties on SP Chan $P_{c,max}$: $k^2 = 0.5 \cdot 1.0^2$

Perturbation theory: SP		**Initial covariance**: $k^2 = 0.5 \cdot 2.0^2$		**Collision probability**: Chan $P_{c,max}$	
#Conjunctions: 11496		**#Category changes**: 8		**Reclassification ratio**: 0.07%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		0	6834	4170	492
Combined density and gravity covariance		0	6833	4168	495
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
0	0	0	0	1	1
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	0	4	0	1	1

Table 6.33: Impact of force model uncertainties on SP Chan $P_{c,max}$: $k^2 = 0.5 \cdot 2.0^2$

Perturbation theory: GP		**Initial covariance**: $k^2 = 0.01^2$		**Collision probability**: Chan $P_{c,max}$	
#Conjunctions: 13401		**#Category changes**: 101		**Reclassification ratio**: 0.75%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		1	8015	4813	572
Combined density and gravity covariance		1	8023	4799	587
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
0	0	0	0	23	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	30	27	0	1	20

Table 6.34: Impact of force model uncertainties on GP Chan $P_{c,max}$: $k^2 = 0.01^2$

Perturbation theory: GP		**Initial covariance**: $k^2 = 0.1^2$		**Collision probability**: Chan $P_{c,max}$	
#Conjunctions: 13401		**#Category changes**: 28		**Reclassification ratio**: 0.21%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		1	8015	4813	572
Combined density and gravity covariance		1	8023	4796	581
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
0	0	0	0	0	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	7	15	0	1	5

Table 6.35: Impact of force model uncertainties on GP Chan $P_{c,max}$: $k^2 = 0.1^2$

Perturbation theory: GP		**Initial covariance**: $k^2 = 0.5^2$		**Collision probability**: Chan $P_{c,max}$	
#Conjunctions: 13401		**#Category changes**: 19		**Reclassification ratio**: 0.14%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		1	8015	4813	572
Combined density and gravity covariance		1	8020	4803	577
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
0	0	0	0	0	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	4	10	0	1	4

Table 6.36: Impact of force model uncertainties on GP Chan $P_{c,max}$: $k^2 = 0.5^2$

Perturbation theory: GP		**Initial covariance**: $k^2 = 1.0^2$		**Collision probability**: Chan $P_{c,max}$	
#Conjunctions: 13401		**#Category changes**: 13		**Reclassification ratio**: 0.10%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		1	8015	4813	572
Combined density and gravity covariance		1	8019	4805	576
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
0	0	0	0	0	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	3	7	0	1	2

Table 6.37: Impact of force model uncertainties on GP Chan $P_{c,max}$: $k^2 = 1.0^2$

Perturbation theory: GP	**Initial covariance**: $k^2 = 2.0^2$		**Collision probability**: Chan $P_{c,max}$	
#Conjunctions: 13401	**#Category changes**: 9		**Reclassification ratio**: 0.07%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	1	8015	4813	572
Combined density and gravity covariance	1	8017	4810	573

Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
0	0	0	0	1	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	3	3	0	0	2

Table 6.38: Impact of force model uncertainties on GP Chan $P_{c,max}$: $k^2 = 2.0^2$

The complete catalog screening results indicate a moderate operational impact of the force model uncertainty consideration when taking the presumably applicable dimensions of the epoch covariances into account. However, once these become more precise, also the operational relevance of physics-based process noise will increase. Especially the GP cases show that for precise epoch information there may be hundreds of potential HIEs and dozens of HIEs not being recognized as such when neglecting process noise. The SP screenings appear to be more robust with regard to the HIE classification, which opposes initial expectations and indicates the need for further investigations using catalogs of different days to gain a better understanding about the nature of this observation (special feature of the respective SP catalog vs. general tendency). Collision probability changes of several OOM can be observed for all but one of the SP cases, especially for gray to yellow risk reclassifications.

The maximum collision probabilities have demonstrated their operationally conservative character, however they are not able to replace the force model uncertainty consideration. Comparing e.g. cases 6.14 and 6.24 shows the possibility of actual HIEs not being recognized as such when using the Alfriend $P_{c,max}$ instead of accounting for force model uncertainties. In contrast, there is no risk of underestimation for Chan's $P_{c,max}$ (cf. tables 6.14 and 6.34). However, due to its independence on the covariance size, it is unsuitable for regular risk assessment but rather serves to evaluate the worst case risk. The consideration of force model errors therefore classifies as operationally relevant.

The LEO vs. all screenings essentially lead to the same conclusions. Pure GEO conjunctions (unaffected by force model uncertainty consideration) are excluded from this analysis. However, it turned out that non-LEO conjunctions only amount to less than 1% of the total identified conjunctions (11815 vs. 11854 conjunctions for the SP catalog and 14639 vs. 14754 conjunctions for the GP catalog). Hence, the influence on the all vs. all screening statistics is negligible. The counterparts to tables 6.10 and 6.17 are listed hereafter as examples. All other results are omitted for the sake of brevity, as they do not lead to new findings compared to the respective all vs. all screenings.

Perturbation theory: SP	**Initial covariance**: $k^2 = 0.5 \cdot 0.5^2$		**Collision probability**: P_c	
#Conjunctions: 11815	**#Category changes**: 238		**Reclassification ratio**: 2.01%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	7484	3512	818	1
Combined density and gravity covariance	7296	3718	800	1
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
181	8	0	1	11	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	37	0	0	0	0

Table 6.39: Impact of force model uncertainties on SP catalog P_c: $k^2 = 0.5 \cdot 0.5^2$, LEO vs. all

Perturbation theory: GP	**Initial covariance**: $k^2 = 1.0^2$		**Collision probability**: P_c	
#Conjunctions: 14639	**#Category changes**: 126		**Reclassification ratio**: 0.86%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	4887	8188	1564	0
Combined density and gravity covariance	4865	8222	1552	0
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
57	0	0	35	11	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	23	0	0	0	0

Table 6.40: Impact of force model uncertainties on GP catalog P_c: $k^2 = 1.0^2$, LEO vs. all

SpaceTrack uses a simple taxonomy to categorize RSOs into three classes: payloads, rocket bodies and debris. The payload subset is often used as catalog for the targets (e.g. in case of SOCRATES), since it includes all active satellites and is only approximately one quarter of the full catalog size (5000 vs. 20 000 objects, cf. figure 1.2), which results in faster conjunction screenings.

SP screenings of this subset can easily be realized using the GP payload catalog as conjunction prefilter. The reclassification ratios of this screening type are expected to be larger than for the respective all vs. all cases, since payloads typically feature greater hard body radii, leading to a more pronounced effect of atmospheric drag and therefore increased density uncertainties.

Perturbation theory: SP	**Initial covariance**: $k^2 = 0.5 \cdot 0.01^2$		**Collision probability**: P_c	
#Conjunctions: 3556	**#Category changes**: 185		**Reclassification ratio**: 5.20%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	3539	8	7	2
Combined density and gravity covariance	3360	127	68	1
Category changes due to process noise consideration				
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
124	55	0	0	5	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	2	0	0	0	1

Table 6.41: Impact of force model uncertainties on SP catalog P_c: $k^2 = 0.5 \cdot 0.01^2$, payload vs. all

Perturbation theory: SP		**Initial covariance**: $k^2 = 0.5 \cdot 0.1^2$		**Collision probability**: P_c	
#Conjunctions: 3556		**#Category changes**: 117		**Reclassification ratio**: 3.29%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		3092	277	185	2
Combined density and gravity covariance		3003	346	206	1
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
82	7	0	0	20	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	7	0	0	0	1

Table 6.42: Impact of force model uncertainties on SP catalog P_c: $k^2 = 0.5 \cdot 0.1^2$, payload vs. all

Perturbation theory: SP		**Initial covariance**: $k^2 = 0.5 \cdot 0.5^2$		**Collision probability**: P_c	
#Conjunctions: 3556		**#Category changes**: 115		**Reclassification ratio**: 3.23%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		2011	1076	469	0
Combined density and gravity covariance		1923	1173	460	0
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
84	4	0	0	7	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	20	0	0	0	0

Table 6.43: Impact of force model uncertainties on SP catalog P_c: $k^2 = 0.5 \cdot 0.5^2$, payload vs. all

Perturbation theory: SP		**Initial covariance**: $k^2 = 0.5 \cdot 1.0^2$		**Collision probability**: P_c	
#Conjunctions: 3556		**#Category changes**: 72		**Reclassification ratio**: 2.02%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		1093	2016	447	0
Combined density and gravity covariance		1046	2074	436	0
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
45	2	0	0	6	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	19	0	0	0	0

Table 6.44: Impact of force model uncertainties on SP catalog P_c: $k^2 = 0.5 \cdot 1.0^2$, payload vs. all

Perturbation theory: SP		**Initial covariance**: $k^2 = 0.5 \cdot 2.0^2$		**Collision probability**: P_c	
#Conjunctions: 3556		**#Category changes**: 72		**Reclassification ratio**: 2.02%	
Risk Histogram		Gray	Green	Yellow	Red
Process noise neglected		459	2758	339	0
Combined density and gravity covariance		447	2775	334	0
Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
12	0	0	0	2	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	7	0	0	0	0

Table 6.45: Impact of force model uncertainties on SP catalog P_c: $k^2 = 0.5 \cdot 2.0^2$, payload vs. all

Perturbation theory: GP	**Initial covariance**: $k^2 = 0.01^2$		**Collision probability**: P_c	
#Conjunctions: 4641	**#Category changes**: 499		**Reclassification ratio**: 10.75%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	4553	33	45	10
Combined density and gravity covariance	4089	276	250	26

Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
259	204	1	0	17	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	1	16	0	0	1

Table 6.46: Impact of force model uncertainties on GP catalog P_c: $k^2 = 0.01^2$, payload vs. all

Perturbation theory: GP	**Initial covariance**: $k^2 = 0.1^2$		**Collision probability**: P_c	
#Conjunctions: 4641	**#Category changes**: 141		**Reclassification ratio**: 3.04%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	3624	530	447	40
Combined density and gravity covariance	3520	612	469	40

Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
89	15	0	0	22	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	15	0	0	0	0

Table 6.47: Impact of force model uncertainties on GP catalog P_c: $k^2 = 0.1^2$, payload vs. all

Perturbation theory: GP	**Initial covariance**: $k^2 = 1.0^2$		**Collision probability**: P_c	
#Conjunctions: 4641	**#Category changes**: 40		**Reclassification ratio**: 2.67%	
Risk Histogram	Gray	Green	Yellow	Red
Process noise neglected	745	3259	637	0
Combined density and gravity covariance	722	3295	624	0

Category changes due to process noise consideration					
Gray to Green	Gray to Yellow	Gray to Red	Green to Gray	Green to Yellow	Green to Red
24	0	0	1	1	0
Yellow to Gray	Yellow to Green	Yellow to Red	Red to Gray	Red to Green	Red to Yellow
0	14	0	0	0	0

Table 6.48: Impact of force model uncertainties on GP catalog P_c: $k^2 = 1.0^2$, payload vs. all

Using only the payload subsets as targets, the screenings confirm an increase of the reclassification ratios (cf. e.g. tables 6.7 and 6.41, 6.10 and 6.43 or 6.17 and 6.48). The screening results for $k^2 = 0.5^2$ and $k^2 = 2.0^2$ when using GP perturbations, as well as the maximum collision probability tables have been omitted for brevity, since they do not lead to new insights in comparison to the respective all vs. all cases. All in all, the main results of this chapter can be summarized as follows:

- Overall operational significance: moderate (increased for smaller epoch covariances)
- Reclassification towards greater and reduced risk possible
- Multiple OOM P_c changes regularly observed
- $P_{c,max}$ metrics more robust regarding force model errors, however they do not remedy their consideration
- HIEs less affected than expected (further studies required)
- Similar reclassification ratios for SP and GP catalogs

Part V

RETROSPECTION AND OUTLOOK

7

FUTURE WORK

Despite the very thorough analysis and quantification of the dominant force model uncertainties in low Earth altitudes, as well as their propagation to satellite orbits and collision probabilities, a number of aspects have been identified which call for further research. Most of them represent extensions of the presented techniques and concepts, however also opportunities for enhancement have been located. The items proposed for further investigation are discussed below in order of the thesis' main parts.

Quantification of dominant force model uncertainties in low Earth orbits

The atmospheric density model input uncertainty theory is rigorous up to the derivation of the relative density error at the perigee altitude (ϵ_{ρ_p}) and its variance ($\sigma^2_{\rho_p}$). Similarly, also the propagation from relative density errors to satellite orbits is derived thoroughly.

The interface of both steps however leaves room for improvement, since the propagation from density errors to satellite orbits is based on the assumption of the relative density errors to follow a certain stochastic process. This is mostly appropriate in practice, however the average stochastic properties are superimposed by inter-orbit variations, which is why Emmert et al. (2017) propose the application of a filter mechanism for reducing $\sigma^2_{\rho_p}(t)$ to a single value for an uncertainty propagation phase. The method has shown adequate performance in the simulation studies carried out within this doctoral research, however an extended formulation of the propagation from relative density errors to orbital mean motion and mean anomaly, which is capable of considering inter-orbit relative density error variance fluctuations is desirable to further enhance the accuracy of long-term propagations.

Concerning geomagnetic index uncertainty, the nonlinear propagation of geomagnetic amplitude variances to quasi-logarithmic kp index variances was realized by Gaussian moment matching and assuming near-linear error propagation. As suggested in section 2.4.2, the assumption of near-linear uncertainty propagation can be relaxed when considering GMMs for this step. A dedicated investigation on possible benefits is recommended.

Eventually, also the question arises if the density and gravitational uncertainty quantification can be transferred to other celestial bodies. Such extensions are possible. A first application of the orbit gravity error covariance theory for the moon was previously considered in Wright et al. (2008a,b,c). To transfer the atmospheric density uncertainty quantification to another central body, the planet must possess

an atmosphere that can be modeled using the concept of hydrostatic balance and a Bates temperature profile. In our solar system Venus seems to be a possible candidate for this endeavor. The development of a semi-empirical global thermospheric density model for Venus based on data supplied by the Pioneer Venus Orbiter Neutral Gas Mass Spectrometer (PVONMS) is described in Hedin et al. (1983). As Hedin is also the principal developer of MSIS-86, the predecessor of the terrestrial semi-empirical density models MSIS-90 and NRLMSISE-00, his VTS3 Venus model exhibits many parallels to the MSIS-class models. VTS3 also uses the F10.7 index to model solar EUV heating. A (slightly modified) Bates profile describes the altitude dependence of the thermospheric temperature. Furthermore, the concept of "diffuse equilibrium" is used and thermal diffusion coefficients are introduced to express the distribution of neutral species density profiles. The VTS3 model is therefore a suitable candidate to transfer the analytical density uncertainty theory to Venus.

Orbit determination with physics-based process noise

The approximate uncertainty quantification using the GP propagation theory was developed based on the desire to avoid SP covariance propagation when performing GP collision probability studies (cf. section 4.4, note 7). Therefore, these developments are currently only available within SPOOK in the context of covariance propagation. An extension of the respective program codes for the purpose of orbit determination is therefore a logical next step towards GP OD with approximate force model uncertainty quantification.

In the scope of the novel force-model uncertainty considering Gaussian mixture orbit determination algorithm, a need for increased efficiency in GMM reduction was identified. This particularly applies to the case where the process noise is modeled as secondary Gaussian mixture that is subsequently fused with the state error GMM and thus leads to very large output GMMs. Due to the quadratic nature of state of the art merging algorithms, also high levels of parallelization do not represent an optimal solution. Further research with respect to more advanced Gaussian mixture reduction is therefore desirable.

Influence of force model uncertainties on the collision probability

The analyses presented in chapter 6 should be carried out for further dates/catalog snapshots in order to maximize the information gain. In particular, the force model uncertainty impact on HIEs, which has been smaller than expected, calls for further studies. It would also be worthwhile to compare the results of catalog encounters that classify as high risk when considering density and gravity uncertainty against Monte-Carlo P_c computations. Furthermore, the assumption of Gaussianity was utilized to compute the catalog collision probabilities. Future analyses should consider an extension towards the GMM

framework, which has been implemented in SPOOK as part of the research presented in chapter 5. The methodology behind GMM P_c calculations is described in Vittaldev (2015). Ideally, the correlation between the target and chaser covariances should also be taken into account when forming the combined covariance.

ESA recently expressed interest in machine learning based computations of the collision probability. The results of ESA's 2019 Kelvins collision avoidance challenge[1] demonstrate that approaches based purely on artificial intelligence (AI) are not yet mature enough to become the new baseline, however hybrid solutions of classical and AI algorithms could form an innovative concept for additional improvements of collision probability calculations.

Further studies and research tasks

Future work involving SPOOK and the force model uncertainty framework

At the time of writing, the Airbus robotic telescope (ART) in Extremadura, Spain, is fully calibrated and features 1σ accuracies of less than one arcsecond in right ascension and declination. Observation planning can be performed based on user inputs or is conducted independently to maintain a space catalog. LEO, MEO and GEO measurements are derived from the optical observations.

The availability of a robotic telescope capable of generating LEO measurements, as well as a software framework capable of processing in-house and externally provided measurements to conduct various space surveillance tasks, provides the possibility for further future studies featuring the force model uncertainty framework, such as:

- OD with force model uncertainty consideration and ART tracking data
- Usage of proprietary space catalog and independently generated OD results for conjunction screening and CARA
- Impact analysis with respect to the MHT measurement correlation algorithm performance
- Impact analysis on the ART tasking algorithm performance

Solar radiation pressure uncertainty

Future research should also address physics-based process noise formulations for the solar radiation pressure perturbation. The availability of attitude information should not be assumed in order to maximize the practical applicability. As in near-Earth orbits solar radiation pressure is approximately constant with altitude, SRP uncertainty is expected to be significant in particular for MEO and GEO RSOs, however also upper LEO objects may potentially benefit from such developments.

1 Available at https://web.archive.org/web/20200814112938/https://kelvins.esa.int/collision-avoidance-challenge

8

CONCLUSION

Today's operational orbit determination and space catalog maintenance systems suffer from overly optimistic covariance estimates. In practice, empirical covariance scaling and other proprietary correction approaches are therefore often used to mitigate the issue and improve covariance realism.

With regard to current commercial space activities, which have long refuted the "big sky" theory and are progressing much faster than the corresponding laws and political regulations, an immediate need for increased uncertainty realism emerges, since satellites are currently being brought into orbit more quickly than scientists are able to predict the long-term consequences of this commercially-driven race to space.

Space is a data-starved environment. The number of available sensors for taking measurements of resident space objects is limited. As a result of the associated costs, the sensors are usually in military possession. The maintenance of the space catalog is therefore often a secondary task. This circumstance contradicts the increased space activities, which makes the ability to realistically predict uncertainty volumes over several days even more important. In recent years, many advanced uncertainty propagation methods have been developed which efficiently propagate also non-Gaussian uncertainty volumes. These advancements however always disregarded the fact that the models used for propagation are themselves subject to errors and thus contribute significantly to the size and shape of the propagated uncertainty volumes.

In the state of the art analysis of this work, the fundamental origins of space surveillance and tracking uncertainties were analyzed and classified according to their nature (aleatoric vs. epistemic) or type (structural, parametric, input, observation, algorithmic or cross-tagging uncertainty). It turned out that atmospheric density uncertainty is the main contributor to LEO force model uncertainties, with potential field truncation uncertainty following in second place. So far, the consideration of force model errors has been limited to the extension of the state vector by some model input parameters of uncertain nature. In rare cases also a consider parameter for atmospheric density uncertainty was implemented as an adjustment to the ballistic coefficient covariance (e.g. at CSpOC). This technique is however not able to predict the actual growth of orbital uncertainty, as the entire time history of density errors determines the resulting orbit error and the corresponding evolution of the state uncertainty volume.

Under these aspects, the main theme of this work was defined as the improvement of LEO uncertainty realism via the quantification of dominant force model uncertainties. All actors in space, from space surveillance centers and satellite builders, over companies that establish their own catalogs or develop SST software, to the space agencies which operate large fleets of spacecraft, may benefit directly or indirectly from the contributions of this work, whose principal research tasks included the development of:

- physics-based uncertainty quantification algorithms for density uncertainty and gravity uncertainty
- the framework for integrating the resulting process noise matrices into classical orbit estimators
- a novel GMM orbit determination algorithm which extends the state of the art by considering force model uncertainties in addition to non-Gaussian uncertainty volumes
- a software suite to perform CARA and real world catalog impact analyses

These objectives have been broken down into smaller building blocks and were in most cases published in advance as journal research papers after their completion. All results, including those that go beyond the scope of the publications, have been presented in chapters 2-6.

Three types of density uncertainty root-causes were considered: solar flux model inputs (of measurement error and forecasting error origin), geomagnetic index inputs and grid-scale average model uncertainty. The integration into classical OD methods was designed and elaborated for batch estimators (WLS and SBWLS), as well as Kalman filters (EKF and UKF). The importance of the force model uncertainties in relation to the orbit altitude was investigated, as was the impact of the coordinate system and the role of the initial (prior) probability density function (pdf).

The novel GMM orbit determination algorithm presented in chapter 5 represents a special achievement, as for the first time it is now possible to model not only the non-Gaussian uncertainty growth due to initial radial uncertainties, but also the contribution of the second growth factor, i.e. force model errors, on the evolution of the pdf. An additional scientific novelty of the algorithm is its capability to autonomously model the process noise as a secondary Gaussian mixture which is subsequently merged with the state error GMM, should this become necessary to preserve the pdf realism. The Gaussian mixture orbit estimator is designed to start with a single kernel. Hence, it yields the same performance as a classical EKF/UKF with physics-based process noise consideration up to the point where further Gaussian mixture components are required to maintain a realistic pdf. This technique is particularly important for operational applications, since additional runtime is only invested when necessary.

The impact of the physics-based covariance formulations on operational CARA risk levels has been examined by means of complete space catalog studies using prior epoch covariance estimates of various sizes in combination with the US restricted SP and the public GP catalogs of February 25th, 2020.

Overall, the following scientific contributions were achieved:

- Analytical framework for the quantification of density uncertainties due to arbitrary model input uncertainties
- Instantiation of the generic density uncertainty framework for EUV solar flux and geomagnetic index uncertainties
- Analytical description of grid-scale density model uncertainties based on an Ornstein-Uhlenbeck error process model
- Derivation of the associated complete time-dependent orbit error covariance matrix in orbital elements, including the associated correlation coefficients (transformation to GCRF via Jacobians or unscented transformation)
- Improvement of an existing real-time capable algorithm for orbit gravity error covariance quantification with respect to:
  - in-track position error variance underestimation
  - radial/normal position error variance robustness
  - support for arbitrary perturbations
  - support for arbitrary temporal propagation directions
  - compatibility with batch estimators
  - overall algorithm performance
- WLS/SBWLS algorithms with support for process noise matrices
- Approximate GP density/gravity force model uncertainty quantification based on semi-analytical GP covariance propagation
- State of the art GMM Kalman filter, which for the first time allows the combined consideration of non-Gaussian error volumes and physics-based force model uncertainties
- Optimization of nonlinear collision probability calculations:
  - formulation of the encounter frame $\rightarrow$ Mahalanobis space transformation which also covers the closed-form conversion of spherical hard body volumes
  - algorithm for the realization of Monte-Carlo collision probabilities using the GP theory
  - extension of the nonlinear adjoining tube collision probability algorithm, such that it no longer requires a transformation into Mahalanobis space to separate sectional from cross-sectional collision probability components
- Realization of first studies applying force model uncertainty consideration to the full GP and SP catalogs to evaluate the operational impact of the developments
- Derivation of the first moment of the generic NLNM distribution

Based on these contributions, the following scientific findings were obtained:

- Force model uncertainties can dominate radial uncertainties as the main driver of uncertainty growth, given accurate knowledge of the pdfs at the start of a propagation phase and/or a sufficiently large ballistic coefficient
- Below ~800 km uncertainty realism can only be achieved by explicitly taking force model uncertainties into account
- Maximum uncertainty realism requires a physics-based model of the uncertainty growth caused by astrodynamic force modeling errors
- Density and gravity force model uncertainties can be neglected for most RSOs and applications above ~800 km orbital altitude
- If carefully designed, it is also possible to incorporate force model uncertainties into modern non-Gaussian error volume Kalman filters. The proper choice of the coordinate system (orbit elements vs. Cartesian GCRF) and the modeling of the process noise based on the merged Gaussian state are of particular importance in this context
- The possibility of fusing two GMMs permits realistic uncertainty propagations of more than 10 days even in Cartesian coordinates, which covers all operational needs
- When using GMM Kalman filters for orbit determination, force model uncertainties also need to be taken into account if they are not the dominant driver of the uncertainty growth to keep the state error pdf unimodal and reduce the overall kernel counts
- The consideration of force model uncertainties can lead to collision probability changes of more than six orders of magnitude. However, in practice the greatest changes have been observed for very low risk, low risk and potential HIE conjunctions. Only few reclassifications were observed above the HIE collision probability threshold
- Maximum collision probability metrics, which are sometimes used operationally, cannot replace the need to consider force model uncertainties, since their effect on the overall state vector error covariance is only insufficiently represented by a scalar scaling operation

Further studies are needed to gain a deeper understanding of the implications for operational CARA. Drawing conclusions is however hampered by the fact that the final results depend heavily on the initial epoch covariance information, which unfortunately is not made publicly available due to its militarily sensitive nature. Additional recommendations for future research are collected in chapter 7.

Finally, it can be concluded that the work at hand has made significant contributions and provided important insights into the un-

derstanding of physics-based process noise formulations and their relevance for the realism of orbital uncertainty volumes. However, in the end the developments are only as good as their underlying assumptions match reality (e.g. the eligibility of the stochastic noise processes), which can only be explored through broad and long-term operational use. The state of the art analysis and consideration of potential field truncation errors have shown that many uncertainties are of epistemic nature and can therefore only insufficiently be handled aleatorically. Further pioneering research is needed to harmonize the treatment of both types of uncertainty, without rendering recent achievements in aleatoric uncertainty modeling obsolete. The recently introduced OPMs seem promising in this regard.

This work can therefore be nicely closed with a quote from Frank Knight, which will also in future continue to be a wise guide for maintaining security in space:

You cannot be certain about uncertainty.

— Frank Knight

Part VI

APPENDIX

A

AUXILIARY CALCULATIONS AND MATHEMATICAL OPERATIONS

Appendix A contains auxiliary calculations referenced throughout the main body of this thesis.

A.1 STOCHASTIC MOMENTS OF A NORMAL DISTRIBUTION

DEFINITION OF EXPECTED VALUE

The expected value or mean of a probability distribution is its probability-weighted long-term average. In the discrete, real-valued case, the expected value of a random variable X can be computed as the sum of the event-probability products for each possible outcome x_i:

$$\mathrm{E}[X] = \mu = \sum_{i=1}^{n} x_i \, p(X = x_i) = \sum_{i=1}^{n} x_i p_i \tag{A.1}$$

In the continuous case the sum turns into an integral and the discrete probabilities are replaced by evaluations of the pdf:

$$\mathrm{E}[X] = \int_{\mathcal{R}} x \, f(x) \, dx \tag{A.2}$$

An important property of the expected value is linearity: $\mathrm{E}[X+Y] = \mathrm{E}[X] + \mathrm{E}[Y]$ and $\mathrm{E}[a \cdot X] = a \cdot \mathrm{E}[X]$. For the product of two random variables it holds that: $\mathrm{E}[X \cdot Y] = \mathrm{E}[X] \cdot \mathrm{E}[Y] + \mathrm{Cov}(X, Y)$, where $\mathrm{Cov}(X, Y)$ is defined in equation A.8 and represents a measure of inter-variable dependence. In case of two independent random variables the covariance is zero. For two fully correlated variables the absolute value of the covariance equals the variance. In case of a normal distribution the expected value coincides with the mode and the median.

DEFINITION OF VARIANCE

The variance is the second order moment/cumulant of a probability distribution. It also has an integral definition for continuous random variables, however in many cases the definition in terms of the expected value is of greater avail:

$$\mathrm{Var}(X) = \mathrm{E}\left[(X - \mathrm{E}[X])^2\right] = \mathrm{E}\left[X^2\right] - \mathrm{E}[X]^2 \tag{A.3}$$

Important properties of the variance are:

$$\mathrm{Var}(X + a) = \mathrm{Var}(X) \tag{A.4}$$

$$\mathrm{Var}(aX) = a^2 \mathrm{Var}(X) \tag{A.5}$$

$$\mathrm{Var}(aX \pm bY) = a^2 \mathrm{Var}(X) + b^2 \mathrm{Var}(Y) \pm 2ab \ \mathrm{Cov}(X, Y) \tag{A.6}$$

DEFINITION OF COVARIANCE

The covariance of two random variables is defined similarly to each variables' variance:

$$\begin{aligned} \mathrm{Cov}(X,Y) &= \mathrm{E}\big[(X-\mathrm{E}[X])(Y-\mathrm{E}[Y])\big] && \text{(A.7)}\\ &= \mathrm{E}[XY]-\mathrm{E}[X]\mathrm{E}[Y] && \text{(A.8)} \end{aligned}$$

If X or Y is zero-mean, the covariance equals the expected value of the random variable product:

$$\mathrm{Cov}(X,Y) = \mathrm{E}[XY] \tag{A.9}$$

If a correlation coefficient is available, the covariance may also be computed according to eq. A.11.

$$\rho_{XY} = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)}\sqrt{\mathrm{Var}(Y)}} \tag{A.10}$$

$$\Leftrightarrow \mathrm{Cov}(X,Y) = \rho_{XY}\sqrt{\mathrm{Var}(X)}\sqrt{\mathrm{Var}(Y)} \tag{A.11}$$

The correlation coefficient is bound by $-1 \leqslant \rho_{XY} \leqslant 1$. Positive correlations indicate similar growth tendency, whereas a negative correlation coefficient specifies opposite tendencies.

MOMENTS OF NORMAL DISTRIBUTION

The central moments of the normal distribution are used repeatedly in chapter 2. They are therefore gathered in table A.1 alongside the non-central moments for quick lookup.

Order	Non-central moment	Central moment
1	μ	0
2	$\mu^2 + \sigma^2$	σ^2
3	$\mu^3 + 3\mu\sigma^2$	0
4	$\mu^4 + 6\mu^2\sigma^2 + 3\sigma^4$	$3\sigma^4$
5	$\mu^5 + 10\mu^3\sigma^2 + 15\mu\sigma^4$	0
6	$\mu^6 + 15\mu^4\sigma^2 + 45\mu^2\sigma^4 + 15\sigma^6$	$15\sigma^6$
7	$\mu^7 + 21\mu^5\sigma^2 + 105\mu^3\sigma^4 + 105\mu\sigma^6$	0
8	$\mu^8 + 28\mu^6\sigma^2 + 210\mu^4\sigma^4 + 420\mu^2\sigma^6 + 105\sigma^8$	$105\sigma^8$

Table A.1: Moments of normal distribution

A.2 DERIVATIVE OF $a\,f(x)^{g(x)}\exp(c(x))$ WITH RESPECT TO x

The derivative in equation 2.38 belongs to the class $\frac{d}{dx}af(x)^{g(x)}\exp(c(x))$. Hence it is necessary to derive the appropriate rule for the differentiation. First, define $y(x) = f(x)^{g(x)}$ and apply the natural logarithm to both sides:

$$\begin{aligned}
& \ln(y(x)) = g(x)\ln(f(x)) \\
\Leftrightarrow \quad & \frac{d}{dx}\ln(y(x)) = \frac{d}{dx}g(x)\ln(f(x)) + g(x)\frac{\frac{d}{dx}f(x)}{f(x)} \\
\Leftrightarrow \quad & \frac{\frac{d}{dx}y(x)}{y(x)} = \frac{d}{dx}g(x)\ln(f(x)) + g(x)\frac{\frac{d}{dx}f(x)}{f(x)} \\
\Leftrightarrow \quad & \frac{d}{dx}f(x)^{g(x)} = f(x)^{g(x)}\left(\frac{d}{dx}g(x)\ln(f(x)) + g(x)\frac{\frac{d}{dx}f(x)}{f(x)}\right) \qquad \text{(A.12)}
\end{aligned}$$

The desired derivative can now be evaluated by a simple application of the product rule:

$$\frac{d}{dx}af(x)^{g(x)}\exp(c(x)) = a\frac{d}{dx}\left(f(x)^{g(x)}\right)\exp(c(x)) + af(x)^{g(x)}\frac{d}{dx}\exp(c(x)) \qquad \text{(A.13)}$$

Introducing equation A.12 results in:

$$\begin{aligned}
& \frac{d}{dx}af(x)^{g(x)}\exp(c(x)) \\
& = a\left[f(x)^{g(x)}\left(\frac{d}{dx}g(x)\ln(f(x)) + g(x)\frac{\frac{d}{dx}f(x)}{f(x)}\right)\exp(c(x)) + f(x)^{g(x)}\frac{d}{dx}\exp(c(x))\right] \\
& = af(x)^{g(x)}\left[\left(\frac{d}{dx}g(x)\ln(f(x)) + g(x)\frac{\frac{d}{dx}f(x)}{f(x)}\right)\exp(c(x)) + \left(\frac{d}{dx}c(x)\right)\exp(c(x))\right] \qquad \text{(A.14)}
\end{aligned}$$

Hence, the required equation of the derivative is:

$$\begin{aligned}
& \frac{d}{dx}af(x)^{g(x)}\exp(c(x)) \\
& = af(x)^{g(x)}\exp(c(x))\left[\frac{d}{dx}c(x) + \ln(f(x))\frac{d}{dx}g(x) + g(x)\frac{\frac{d}{dx}f(x)}{f(x)}\right] \qquad \text{(A.15)}
\end{aligned}$$

A.3 DEPENDENCE OF EXOSPHERIC AND BASELINE TEMPERATURE ERRORS/VARIANCES ON ABSOLUTE ERRORS IN THE SPHERICAL HARMONICS FUNCTION

The MSIS and DTM-models express exospheric temperature according to equation A.16,

$$T_{ex} = T_{ex_0} + T_{ex_0}G_{T_{ex}}(L) \qquad \text{(A.16)}$$

where T_{ex_0} is a model-coefficient parameter and represents the global and temporal average exospheric temperature if all dependencies are neglected. An error in $G(L)$ is therefore linearly related to an error in exospheric temperature:

$$\delta T_{ex} = T_{ex_0}\delta G_{T_{ex}}(L) \qquad \text{(A.17)}$$

In the MSIS-class models the same dependency is also used to compute the temperature at the baseline altitude:

$$\delta T_0 = T_{0_0} \delta G_0(L) \tag{A.18}$$

Equations A.17 and A.18 are of identical form. The only difference is the scaling factor (T_{0_0} vs. T_{ex_0}). The relationships of the variances read:

$$Var(\delta T_{ex}) = T_{ex_0}^2 Var(\delta G_i(L)) \tag{A.19}$$

and

$$Var(\delta T_0) = T_{0_0}^2 Var(\delta G_i(L)) \tag{A.20}$$

where the index i is used to denote the "temperature species" and determines the relevant model-coefficients to be used.

For NRLMSISE-00, $T_{0_0} = 364.7105\,K$ and $T_{ex_0} = 1027.318\,K$. The DTM-models assume T_0 to be fixed at $T_0 = T_{0_0} = 380.0\,K$, independently of the value of $G_0(L)$. Hence, the variance of the baseline temperature is only nonzero for NRLMSISE-00. The parametrization of T_{ex_0} for DTM-2012 and DTM-2013 is $T_{ex_0} = 1029.77\,K$.

A.4 FIRST MOMENT OF GENERIC NORMAL LOGNORMAL MIXTURE

Define the distributions X and Y as:

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}, \begin{bmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{bmatrix} \right) \tag{A.21}$$

and the resulting mixture as:

$$U = Xe^Y = XZ \tag{A.22}$$

To derive the expected value of U, the following tools are required:

- Conditional distribution function of bivariate correlated Gaussians: $Y|X = \mathcal{N}\left(\mu_Y + \frac{\sigma_Y}{\sigma_X}\rho(X - \mu_X), (1-\rho^2)\,\sigma_Y^2\right)$
- First moment of the lognormal distribution: $E[e^Z] = e^{\mu_Z + \sigma_Z^2/2}$
- Law of total expectation: $E[XY] = E[X\,E[Y|X]]$

Applying the conditional distribution function results in:

$$Y|X = \mathcal{N}\left(\frac{\sigma_Y}{\sigma_X}\rho X, (1-\rho^2)\,\sigma_Y^2\right) \tag{A.23}$$

The law of total expectation then yields:

$$E[XZ] = E[X\,E[Z|X]] = E\left[X\,E[e^Y|X]\right] = E\left[X\,E\left[e^{Y|X}\right]\right] \tag{A.24}$$

Applying the first moment of the lognormal distribution for $Y|X$ leads to:

$$E[U] = E\left[X\, e^{\frac{\sigma_Y}{\sigma_X}\rho X} e^{\frac{1}{2}(1-\rho^2)\sigma_Y^2}\right] = e^{\frac{1}{2}(1-\rho^2)\sigma_Y^2}\, E\left[X\, e^{\frac{\sigma_Y}{\sigma_X}\rho X}\right] \tag{A.25}$$

Since

$$\frac{d}{d\sigma_Y} e^{\frac{\sigma_Y}{\sigma_X}\rho X} = \frac{\rho}{\sigma_X} X e^{\frac{\sigma_Y}{\sigma_X}\rho X} \tag{A.26}$$

it holds that:

$$X e^{\frac{\sigma_Y}{\sigma_X}\rho X} = \frac{\sigma_X}{\rho}\frac{d}{d\sigma_Y} e^{\frac{\sigma_Y}{\sigma_X}\rho X} \tag{A.27}$$

Inserting equation A.27 into eq. A.25 it follows that:

$$E[U] = \frac{\sigma_X}{\rho} e^{\frac{1}{2}(1-\rho^2)\sigma_Y^2} \frac{d}{d\sigma_Y} E\left[e^{\frac{\sigma_Y}{\sigma_X}\rho X}\right] \tag{A.28}$$

$\frac{\sigma_Y}{\sigma_X}\rho X$ is normal with mean zero and variance $\left(\frac{\sigma_Y}{\sigma_X}\rho\right)^2 \sigma_X^2 = \sigma_Y^2\rho^2$. Applying the first moment of the lognormal distribution a second time yields:

$$E[U] = \frac{\sigma_X}{\rho} e^{\frac{1}{2}(1-\rho^2)\sigma_Y^2} \frac{d}{d\sigma_Y} e^{\frac{1}{2}\sigma_Y^2\rho^2} \tag{A.29}$$

Therefore:

$$E[X e^Y] = \sigma_X \sigma_Y \rho e^{\frac{1}{2}\sigma_Y^2} \tag{A.30}$$

Equation A.30 describes the first moment of the generic NLNM.

A.5 MINIMUM VARIANCE WLS PROCESS NOISE EXTENSION

The extension of the classical WLS algorithm for the case of propagation error uncertainty is shown in section 4.2.1. The derivation assumes that the requirement enforced in equation 4.28 results in minimum variance estimates, however in principle also maximum variance estimates may be found via equation 4.28. Consequently, it remains to be shown that the results of section 4.2.1 indeed lead to minimum variance estimates.

A sufficient approach for this purpose is to show that $\mathbf{P}_k^*$ obtained by any other estimator $\mathbf{x}_k^*$ is greater or equal to $\mathbf{P}_k$ in the sense of the matrix trace[1], meaning that $\mathrm{tr}(\mathbf{P}_k^*) \geqslant \mathrm{tr}(\mathbf{P}_k)$. The following steps are similar to those of the classical minimum variance algorithm which, for example, are shown in Schutz et al. (2004, Section 4.4).

1 The trace corresponds to the sum of all covariance eigenvalues. The condition $\mathrm{tr}(\mathbf{P}_k^*) \geqslant \mathrm{tr}(\mathbf{P}_k)$ is therefore equal to showing that $\mathbf{P}_k^* - \mathbf{P}_k$ is positive semidefinite

The estimator to be analyzed is given by equation 4.12. Any other linear, unbiased estimator can be written as:

$$\mathbf{x}_k^* = \mathbf{N}\mathbf{y} = (\mathbf{M}+\mathbf{B})\mathbf{y} = \hat{\mathbf{x}}_k + \mathbf{B}\mathbf{y} = \hat{\mathbf{x}}_k + \mathbf{B}(\mathbf{A}\mathbf{x}_k + \mathbf{H}\boldsymbol{\nu} + \boldsymbol{\epsilon}) \quad \text{(A.31)}$$

The expected value of equation A.31 needs to match $\mathbf{x}_k$ for the estimator to be unbiased which, using eq. 4.6, results in:

$$\begin{aligned} \mathrm{E}[\mathbf{x}_k^*] &= \mathrm{E}[\hat{\mathbf{x}}_k + \mathbf{B}\mathbf{y}] \\ &= \mathbf{x}_k + \mathrm{E}\left[\mathbf{B}(\mathbf{A}\mathbf{x}_k + \mathbf{H}\boldsymbol{\nu} + \boldsymbol{\epsilon})\right] \\ &= \mathbf{x}_k + \mathbf{B}\mathbf{A}\mathbf{x}_k \quad \text{(A.32)} \\ &\Leftrightarrow \mathbf{B}\mathbf{A} = \mathbf{0} \quad \text{(A.33)} \end{aligned}$$

Ignoring the trivial solution $\mathbf{B} = \mathbf{0}$, which corresponds to the estimator of section 4.2.1, it is found that $\mathbf{B}$ cannot have full rank to satisfy eq. A.33. Hence:

$$\operatorname{rank}(\mathbf{B}) < n \quad \text{(A.34)}$$

which will be needed in the discussion following eq. A.44. Based on the definition of the covariance matrix it can be deduced that:

$$\mathbf{P}_k^* = \mathrm{E}\left[\left[\mathbf{x}^* - \mathrm{E}(\mathbf{x}^*)\right]\left[\mathbf{x}^* - \mathrm{E}(\mathbf{x}^*)\right]^T\right] \quad \text{(A.35)}$$

Inserting equation A.32 results in:

$$\mathbf{P}_k^* = \mathrm{E}\left[\left[\mathbf{x}^* - (\mathbf{x}_k + \mathbf{B}\mathbf{A}\mathbf{x}_k)\right]\left[\mathbf{x}^* - (\mathbf{x}_k + \mathbf{B}\mathbf{A}\mathbf{x}_k)\right]^T\right] \quad \text{(A.36)}$$

Using equation A.31 for $\mathbf{x}^*$ leads to:

$$\begin{aligned} \mathbf{P}_k^* &= \mathrm{E}\Big[\left[\hat{\mathbf{x}}_k + \mathbf{B}(\mathbf{A}\mathbf{x}_k + \mathbf{H}\boldsymbol{\nu} + \boldsymbol{\epsilon}) - (\mathbf{x}_k + \mathbf{B}\mathbf{A}\mathbf{x}_k)\right] \\ &\qquad \left[\hat{\mathbf{x}}_k + \mathbf{B}(\mathbf{A}\mathbf{x}_k + \mathbf{H}\boldsymbol{\nu} + \boldsymbol{\epsilon}) - (\mathbf{x}_k + \mathbf{B}\mathbf{A}\mathbf{x}_k)\right]^T\Big] \\ &= \mathrm{E}\left[\left[(\hat{\mathbf{x}}_k - \mathbf{x}_k) + \mathbf{B}\mathbf{H}\boldsymbol{\nu} + \mathbf{B}\boldsymbol{\epsilon}\right]\left[(\hat{\mathbf{x}}_k - \mathbf{x}_k) + \mathbf{B}\mathbf{H}\boldsymbol{\nu} + \mathbf{B}\boldsymbol{\epsilon}\right]^T\right] \\ &= \mathrm{E}\big[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T + (\hat{\mathbf{x}}_k - \mathbf{x}_k)\boldsymbol{\nu}^T\mathbf{H}^T\mathbf{B}^T + (\hat{\mathbf{x}}_k - \mathbf{x}_k)\boldsymbol{\epsilon}^T\mathbf{B}^T \\ &\qquad + \mathbf{B}\mathbf{H}\boldsymbol{\nu}\boldsymbol{\nu}^T\mathbf{H}^T\mathbf{B}^T + \mathbf{B}\mathbf{H}\boldsymbol{\nu}\boldsymbol{\epsilon}^T\mathbf{B}^T + \mathbf{B}\mathbf{H}\boldsymbol{\nu}(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T \\ &\qquad + \mathbf{B}\boldsymbol{\epsilon}(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T + \mathbf{B}\boldsymbol{\epsilon}\boldsymbol{\nu}^T\mathbf{H}^T\mathbf{B}^T + \mathbf{B}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T\mathbf{B}^T\big] \quad \text{(A.37)} \end{aligned}$$

After applying the linearity of expectation, the first term equates to $\mathbf{P}_k$. The term $\mathbf{B}\mathbf{H}\mathrm{E}[\boldsymbol{\nu}\boldsymbol{\epsilon}^T]\mathbf{B}^T$ and its transpose equate to zero, since the propagation errors and measurement errors are assumed to be uncorrelated. Therefore, equation A.37 reduces to:

$$\begin{aligned} \mathbf{P}_k^* = \mathbf{P}_k &+ \mathbf{B}\mathbf{H}\mathbf{Q}\mathbf{H}^T\mathbf{B}^T + \mathbf{B}\mathbf{R}\mathbf{B}^T \\ &+ \mathbf{B}\mathbf{H}\mathrm{E}\left[\boldsymbol{\nu}(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T\right] + \mathrm{E}\left[(\hat{\mathbf{x}}_k - \mathbf{x}_k)\boldsymbol{\nu}^T\right]\mathbf{H}^T\mathbf{B}^T \\ &+ \mathbf{B}\mathrm{E}\left[\boldsymbol{\epsilon}(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T\right] + \mathrm{E}\left[(\hat{\mathbf{x}}_k - \mathbf{x}_k)\boldsymbol{\epsilon}^T\right]\mathbf{B}^T \quad \text{(A.38)} \end{aligned}$$

All terms of equation A.38 that still contain expected values require further analysis.

The expression for $\mathbf{BH}\mathrm{E}[\boldsymbol{\nu}(\hat{\mathbf{x}}_k - \mathbf{x}_k)^\mathsf{T}]$ is found using equations 4.35 and 4.8:

$$\begin{aligned}\mathbf{BH}\ \mathrm{E}[\boldsymbol{\nu}(\hat{\mathbf{x}}_k - \mathbf{x}_k)^\mathsf{T}] &= \mathbf{BH}\ \mathrm{E}\left[\boldsymbol{\nu}\left((\mathbf{A}\mathbf{x}_k + \mathbf{H}\boldsymbol{\nu} + \boldsymbol{\epsilon})^\mathsf{T}\tilde{\mathbf{R}}^{-1}\mathbf{A}\mathbf{P}_k - \mathbf{x}_k^\mathsf{T}\right)\right] \\ &= \mathbf{BH}\ \mathrm{E}[\boldsymbol{\nu}]\,\mathbf{x}^\mathsf{T}\mathbf{A}^\mathsf{T}\tilde{\mathbf{R}}^{-1}\mathbf{A}\mathbf{P}_k + \mathbf{BH}\ \mathrm{E}\left[\boldsymbol{\nu}\boldsymbol{\epsilon}^\mathsf{T}\right]\tilde{\mathbf{R}}^{-1}\mathbf{A}\mathbf{P}_k \\ &\quad + \mathbf{BH}\ \mathrm{E}\left[\boldsymbol{\nu}\boldsymbol{\nu}^\mathsf{T}\right]\mathbf{H}^\mathsf{T}\tilde{\mathbf{R}}^{-1}\mathbf{A}\mathbf{P}_k - \mathbf{BH}\ \mathrm{E}[\boldsymbol{\nu}]\,\mathbf{x}_k^\mathsf{T} \quad \text{(A.39)}\end{aligned}$$

The first, second and fourth term in equation A.39 vanishes, since $\mathrm{E}[\boldsymbol{\nu}] = \mathbf{0}$ and $\mathrm{E}\left[\boldsymbol{\nu}\boldsymbol{\epsilon}^\mathsf{T}\right] = \mathbf{0}$. Therefore:

$$\mathbf{BH}\ \mathrm{E}[\boldsymbol{\nu}(\hat{\mathbf{x}}_k - \mathbf{x}_k)^\mathsf{T}] = \mathbf{BHQH}^\mathsf{T}\tilde{\mathbf{R}}^{-1}\mathbf{A}\mathbf{P}_k \quad \text{(A.40)}$$

Similarly,

$$\begin{aligned}\mathbf{B}\ \mathrm{E}\left[\boldsymbol{\epsilon}(\hat{\mathbf{x}}_k - \mathbf{x}_k)^\mathsf{T}\right] &= \mathrm{E}\left[\mathbf{B}\boldsymbol{\epsilon}(\mathbf{P}_k\mathbf{A}^\mathsf{T}\tilde{\mathbf{R}}^{-1}(\mathbf{A}\mathbf{x}_k + \mathbf{H}\boldsymbol{\nu} + \boldsymbol{\epsilon}) - \mathbf{x}_k)^\mathsf{T}\right] \\ &= \mathbf{B}\ \mathrm{E}[\boldsymbol{\epsilon}]\mathbf{x}_k^\mathsf{T}\mathbf{A}^\mathsf{T}\tilde{\mathbf{R}}^{-1}\mathbf{A}\mathbf{P}_k + \mathbf{B}\ \mathrm{E}\left[\boldsymbol{\epsilon}\boldsymbol{\nu}^\mathsf{T}\right]\mathbf{H}^\mathsf{T}\tilde{\mathbf{R}}^{-1}\mathbf{A}\mathbf{P}_k \\ &\quad + \mathbf{B}\ \mathrm{E}\left[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\mathsf{T}\right]\tilde{\mathbf{R}}^{-1}\mathbf{A}\mathbf{P}_k - \mathbf{B}\ \mathrm{E}[\boldsymbol{\epsilon}]\mathbf{x}_k^\mathsf{T}\right] \quad \text{(A.41)}\end{aligned}$$

where the first, second and fourth term vanishes as a consequence of $\mathrm{E}[\boldsymbol{\epsilon}] = \mathbf{0}$ and $\mathrm{E}\left[\boldsymbol{\epsilon}\boldsymbol{\nu}^\mathsf{T}\right] = \mathbf{0}$. Therefore:

$$\mathbf{B}\ \mathrm{E}\left[\boldsymbol{\epsilon}(\hat{\mathbf{x}}_k - \mathbf{x}_k)^\mathsf{T}\right] = \mathbf{BR}\tilde{\mathbf{R}}^{-1}\mathbf{A}\mathbf{P}_k \quad \text{(A.42)}$$

Summing equations A.40 and A.42 yields:

$$\begin{aligned}\mathrm{E}\left[\mathbf{BH}\boldsymbol{\nu}(\hat{\mathbf{x}}_k - \mathbf{x}_k)^\mathsf{T} + \mathbf{B}\boldsymbol{\epsilon}(\hat{\mathbf{x}}_k - \mathbf{x}_k)^\mathsf{T}\right] &= \mathbf{B}\left[\mathbf{HQH}^\mathsf{T} + \mathbf{R}\right]\tilde{\mathbf{R}}^{-1}\mathbf{A}\mathbf{P}_k \\ &= \mathbf{B}\tilde{\mathbf{R}}\tilde{\mathbf{R}}^{-1}\mathbf{A}\mathbf{P}_k = \mathbf{BA}\mathbf{P}_k \\ &= \mathbf{0} \quad \text{(A.43)}\end{aligned}$$

which is the central part of equation A.38. The last step follows from equation A.33. Consequently, equation A.38 reduces to:

$$\begin{aligned}&\mathbf{P}_k^* = \mathbf{P}_k + \mathbf{BHQH}^\mathsf{T}\mathbf{B}^\mathsf{T} + \mathbf{BRB}^\mathsf{T} = \mathbf{P}_k + \mathbf{B}\tilde{\mathbf{R}}\mathbf{B}^\mathsf{T} \\ \Leftrightarrow\ &\mathbf{P}_k^* - \mathbf{P}_k = \mathbf{B}\tilde{\mathbf{R}}\mathbf{B}^\mathsf{T} \quad \text{(A.44)}\end{aligned}$$

For equation 4.35 to represent a minimum variance estimator, $\mathbf{B}\tilde{\mathbf{R}}\mathbf{B}^\mathsf{T}$ must be positive semidefinite. Since $\tilde{\mathbf{R}}$ is assumed to be an invertible covariance matrix, it has to be positive definite, which implies that $\mathbf{z}^\mathsf{T}\tilde{\mathbf{R}}\mathbf{z} > 0$ for any nonzero $\mathbf{z} \in \mathbb{R}^m$. Setting $\mathbf{z} = \mathbf{B}^\mathsf{T}\mathbf{w}$ implies $\mathbf{0} = \mathbf{B}^\mathsf{T}\mathbf{w}$ for some nonzero $\mathbf{w} \in \mathbb{R}^n$ due to the rank deficiency of $\mathbf{B}$ (cf. equation A.34). In total, it hence holds that $\mathbf{w}^\mathsf{T}\mathbf{B}\tilde{\mathbf{R}}\mathbf{B}^\mathsf{T}\mathbf{w} \geqslant 0$ for any $\mathbf{w} \in \mathbb{R}^n$, which is the definition of being PSD and therefore fulfills the minimum variance requirement of the estimators derived in sections 4.2.1 and 4.2.2.

A.6 MANUAL SEPARATION OF PERTURBED MATRIX INVERSE

Equation 4.37 makes use of the Sherman-Morrison-Woodbury identity to reduce the order of the matrix inverse. An intuitive alternative derivation is given in the following.

The task is to simplify $(\mathbf{R}+\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}})^{-1}$ and express it as:

$$(\mathbf{R}+\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}})^{-1} = \mathbf{R}^{-1}+\mathbf{X} \tag{A.45}$$

$\mathbf{X}$ in equation A.45 can be derived via regular matrix operations. First, define $\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}} = \mathbf{B}$, noting that $\mathbf{B}$ has nothing to do with the definition used in section A.5. From eq. A.45 it follows that:

$$\begin{aligned}
&(\mathbf{R}^{-1}+\mathbf{X})(\mathbf{R}+\mathbf{B}) = \mathbf{I} \\
\Leftrightarrow\ &\mathbf{I}+\mathbf{R}^{-1}\mathbf{B}+\mathbf{X}\mathbf{R}+\mathbf{X}\mathbf{B} = \mathbf{I} \\
\Leftrightarrow\ &\mathbf{X}(\mathbf{R}+\mathbf{B}) = -\mathbf{R}^{-1}\mathbf{B} \\
\Leftrightarrow\ &\mathbf{X} = -\mathbf{R}^{-1}\mathbf{B}(\mathbf{R}+\mathbf{B})^{-1}
\end{aligned} \tag{A.46}$$

Equation A.46 contains the same inverse that shall be expressed using eq. A.45. Therefore, equation A.45 can be inserted into eq. A.46 to obtain:

$$\begin{aligned}
&\mathbf{X} = -\mathbf{R}^{-1}\mathbf{B}(\mathbf{R}^{-1}+\mathbf{X}) \\
\Leftrightarrow\ &\mathbf{X}+\mathbf{R}^{-1}\mathbf{B}\mathbf{X} = -\mathbf{R}^{-1}\mathbf{B}\mathbf{R}^{-1} \\
\Leftrightarrow\ &(\mathbf{I}+\mathbf{R}^{-1}\mathbf{B})\mathbf{X} = -\mathbf{R}^{-1}\mathbf{B}\mathbf{R}^{-1} \\
\Leftrightarrow\ &\mathbf{X} = -(\mathbf{I}+\mathbf{R}^{-1}\mathbf{B})^{-1}\mathbf{R}^{-1}\mathbf{B}\mathbf{R}^{-1}
\end{aligned} \tag{A.47}$$

Inserting equation A.47 into eq. A.45 yields:

$$(\mathbf{R}+\mathbf{B})^{-1} = \mathbf{R}^{-1}-(\mathbf{I}+\mathbf{R}^{-1}\mathbf{B})^{-1}\mathbf{R}^{-1}\mathbf{B}\mathbf{R}^{-1} \tag{A.48}$$

The inverse $(\mathbf{I}+\mathbf{R}^{-1}\mathbf{B})^{-1}$ in equation A.48 is still of order $m \times m$. Therefore the goal of having only inverses of order $n \times n$, except for $\mathbf{R}^{-1}$, is not yet fulfilled. It also remains to be shown that $(\mathbf{I}+\mathbf{R}^{-1}\mathbf{B})^{-1}$ actually exists.

For these purposes, a helpful identity needs to be derived first. Consider the following equality, where $\mathbf{A}$ and $\mathbf{D}$ are non-singular:

$$\mathbf{V}-\mathbf{V}\mathbf{A}^{-1}\mathbf{U}\mathbf{D}^{-1}\mathbf{V} = \mathbf{V}-\mathbf{V}\mathbf{A}^{-1}\mathbf{U}\mathbf{D}^{-1}\mathbf{V} \tag{A.49}$$

Both sides of equation A.49 are equal, hence the statement is true. Multiplying $\mathbf{A}^{-1}\mathbf{A}$ to the first element on the left-hand side and $\mathbf{D}\mathbf{D}^{-1}$ to the first element on the right-hand side yields:

$$\begin{aligned}
&\mathbf{V}\mathbf{A}^{-1}\mathbf{A}-\mathbf{V}\mathbf{A}^{-1}\mathbf{U}\mathbf{D}^{-1}\mathbf{V} = \mathbf{D}\mathbf{D}^{-1}\mathbf{V}-\mathbf{V}\mathbf{A}^{-1}\mathbf{U}\mathbf{D}^{-1}\mathbf{V} \\
\Leftrightarrow\ &\mathbf{V}\mathbf{A}^{-1}(\mathbf{A}-\mathbf{U}\mathbf{D}^{-1}\mathbf{V}) = (\mathbf{D}-\mathbf{V}\mathbf{A}^{-1}\mathbf{U})\mathbf{D}^{-1}\mathbf{V} \\
\Leftrightarrow\ &(\mathbf{D}-\mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1} = \mathbf{D}^{-1}\mathbf{V}(\mathbf{A}-\mathbf{U}\mathbf{D}^{-1}\mathbf{V})^{-1}
\end{aligned} \tag{A.50}$$

A special form of identity A.50 is obtained by setting $\mathbf{A} = \mathbf{D} = \mathbf{I}$ and $\mathbf{U} = -\mathbf{P}$:

$$(\mathbf{I}+\mathbf{V}\mathbf{P})^{-1}\mathbf{V} = \mathbf{V}(\mathbf{I}+\mathbf{P}\mathbf{V})^{-1} \tag{A.51}$$

Equation A.51 is the identity required to obtain the reduced order inverse. For this purpose, equation A.47 is first expressed in its fully expanded form:

$$\mathbf{X} = -(\mathbf{I} + \mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}})^{-1}\mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1} \tag{A.52}$$

Applying identity A.51 to eq. A.52 using $\mathbf{V} \to \mathbf{R}^{-1}$ and $\mathbf{P} \to \mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}}$ yields:

$$\mathbf{X} = -\mathbf{R}^{-1}(\mathbf{I} + \mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1})^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1} \tag{A.53}$$

The inverse in equation A.53 is still of order $m \times m$, however identity A.51 can be applied a second time using $\mathbf{V} \to \mathbf{H}$ and $\mathbf{P} \to \mathbf{Q}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}$:

$$\mathbf{X} = -\mathbf{R}^{-1}\mathbf{H}(\mathbf{I} + \mathbf{Q}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{Q}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1} \tag{A.54}$$

The matrix product $\mathbf{Q}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}$ is $n \times n$. Hence the order of the inverse is also $n \times n$.

If the existence of $(\mathbf{I} + \mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}})^{-1}$ in equation A.52 can be proven, then $(\mathbf{I} + \mathbf{Q}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H})^{-1}$ in equation A.54 is guaranteed to exist by equality of equations A.52 and A.54. The existence of $(\mathbf{I} + \mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}})^{-1}$ can be shown by computing the determinant of $\mathbf{I} + \mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}}$, which needs to be nonzero for the inverse is to exist. It holds that:

$$|\mathbf{I} + \mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}}| = |\mathbf{R}^{-1}(\mathbf{R} + \mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}})| \tag{A.55}$$

Using the identity $\det(\mathbf{A}\mathbf{B}) = \det(\mathbf{A})\det(\mathbf{B})$, results in:

$$|\mathbf{I} + \mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}}| = |\mathbf{R}^{-1}| \cdot |(\mathbf{R} + \mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}})| \tag{A.56}$$

The observation error covariance matrix $\mathbf{R}$ is invertible, hence its determinant is nonzero and also $|\mathbf{R}^{-1}| \neq 0$. Thus, for $(\mathbf{I} + \mathbf{R}^{-1}\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}})^{-1}$ to exist, it must hold that $|(\mathbf{R} + \mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}})| \neq 0$. Since $|(\mathbf{R} + \mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}})| = |\tilde{\mathbf{R}}|$, the inverse exists if $\tilde{\mathbf{R}}^{-1}$ exists. The existence of $\tilde{\mathbf{R}}^{-1}$ however is required from the very beginning, as it is the expression that was sought to be simplified (eq. A.45). Equation A.54 is therefore well-defined. Inserting equation A.54 into eq. A.45 yields equation 4.40.

B

AUXILIARY DEVELOPMENTS

B.1 ap to kp polynomial coefficients

Geomagnetic index conversion from magnetic amplitude to quasi-logarithmic kp is only defined for discrete values of ap. Uncertainty quantification on the other hand requires continuous functions.

In section 2.4.1 a clamped cubic splining approach is postulated as an improvement to Vallado's cubic splining conversion algorithm. The ap to kp third-order polynomial coefficients are given in table B.1.

	a_3	a_2	a_1	a_0
$0 \leqslant ap < 2$	0.0128212855706683 00	0.0076907621919966 90	0.1000000000000000	0.0000000000000000
$2 \leqslant ap < 3$	-0.0359036178986799 00	0.3000401830080860 00	-0.4846988416321790	0.3897992277754786 0
$3 \leqslant ap < 4$	0.0102811382613862 00	-0.1156226224325090 00	0.7622895746896060 0	-0.8571891885669 99
$4 \leqslant ap < 5$	-0.0052209351468650 60	0.0704022584665063 00	0.0181900510935453 0	0.1349435095610820
$5 \leqslant ap < 6$	0.0106026023260742 00	-0.1669508036275830 00	1.2049553615639900 0	-1.8429986745563 30
$6 \leqslant ap < 7$	-0.0371894741574326 00	0.6933065730755400 00	-3.9565888986547500	8.4800898458811500
$7 \leqslant ap < 9$	0.0147902529668836 00	-0.3982676965350990 00	3.6844309886197300 0	-9.3489565577592 90
$9 \leqslant ap < 12$	-0.0001933568379784 27	0.0062897681961744 10	0.0434138060382646 0	1.5740949899851000
$12 \leqslant ap < 15$	0.0002540284760542 32	-0.0098161031090013 10	0.2366842617003730 0	0.8010131673366630
$15 \leqslant ap < 18$	-0.0008227570662385 19	0.0386392462941725 00	-0.4901459793472340	4.4351643725747000
$18 \leqslant ap < 22$	0.0003340534753142 61	-0.0238285229496776 00	0.6342738670420680 0	-2.3113547057611 10
$22 \leqslant ap < 27$	0.0001882852459515 70	-0.0142078198117401 00	0.4226183980074420 0	-0.7592145995071 87
$27 \leqslant ap < 32$	-0.0002290287981832 91	0.0195946177631837 00	-0.4900474165154990	7.4547777311992800
$32 \leqslant ap < 39$	8.02535376886814E-05	-0.0100964864805256 00	0.4600679192831980 0	-2.6797858506534 90
$39 \leqslant ap < 48$	5.75256037028785E-05	-0.0074373182041866 90	0.3563603565059800 0	-1.3315875345496 50
$48 \leqslant ap < 56$	-7.96728884985953E-05	0.0123192646728255 00	-0.5919556215906070	13.8414681149957 00
$56 \leqslant ap < 67$	3.12985956717728E-05	-0.0063239446677963 00	0.4520641014842160 0	-5.6469000490676 20
$67 \leqslant ap < 80$	-5.00757865332135E-07	6.77253931617976E-05	0.0238222074000232 0	3.9171689188126900
$80 \leqslant ap < 94$	-2.55002325897578E-06	0.0005595490876362 73	-0.0155236881579348	4.9663928003582400
$94 \leqslant ap < 111$	8.47046305237753E-07	-0.0003984245294719 43	0.0745258318502373 0	2.1448411734355200
$111 \leqslant ap < 132$	2.11242016197263E-06	-0.0008197940237646 58	0.1212978457167290 0	0.4142766603753310
$132 \leqslant ap < 154$	-1.05179943442250E-06	0.0004332369364078 17	-0.0441022410260379	7.6918804770570600
$154 \leqslant ap < 179$	2.39014661504994E-07	-0.0001631191759106 88	0.0477366002710119 0	2.9774866238085100
$179 \leqslant ap < 207$	7.18791807451520E-07	-0.0004207595032839 73	0.0938542188708298 0	0.2258020473527070
$207 \leqslant ap < 236$	-1.55381471501983E-06	0.0009905291471707 36	-0.1982825317732950	20.383237841797300
$236 \leqslant ap < 300$	6.34152268878335E-07	-0.0005585514774291 65	0.1673004956322820 0	-8.3759603141080 50
$300 \leqslant ap < 400+$	-1.27594489473349E-07	0.0001270206050873 51	-0.0383711291226729	12.191202161387400

Table B.1: ap to kp coefficients obtained by performing a clamped cubic splining of the discrete Bartels ap to kp grid

The coefficients are to be used with equation 2.104 in order to perform the conversion from magnetic amplitude to the kp-index. The resulting curve closely follows Vallado's algorithm (Vallado, 2013, algorithm 63), however it is free of abrupt slope changes at the outer ap-knots.

B.2 COVARIANCE COORDINATE FRAME CONVERSION

Covariance frame conversions are required throughout various parts of the work conducted within this thesis. One possibility is to use unscented transformation for the nonlinear transformations, however also conversions using Jacobi matrices are possible. Defining the six-dimensional GCRF Cartesian state vector as $\mathbf{x}_{GCRF} = [x, y, z, v_x, v_y, v_z]$, the six-dimensional classical orbital elements state vector as $\mathbf{x}_{COE} = [a|n, e, i, \Omega, \omega, \nu|M]$ and the six-dimensional equinoctial elements state vector as $\mathbf{x}_{EOE} = [a|n, h, k, p, q, \lambda_\nu|\lambda_M]$ allows to formulate the corresponding sets of partial derivatives. The first component of the orbital elements vectors is also known as "size variable", since it refers to the size of the orbit (semi-major axis a or mean motion n). The sixth variable is also known as "fast variable" (mean or true anomaly/longitude[1]), as it is the only variable in $\mathbf{x}_{COE}$ and $\mathbf{x}_{EOE}$ that changes quickly throughout the course of an orbit.

This appendix is devoted to the Fortran codes that facilitate the covariance frame conversions. It builds upon the equations listed in Vallado (2004), Vallado and Alfano (2015), as well as the references therein. Multiple typographical errors are corrected and new conversions between mean and true fast variables have been derived. Only the covariance conversion routines are listed in the following. The parametrization of constants, as well as the implementation of the state vector transformations is left to the interested reader. Additional invoked functions whose implementation is not specified can be obtained from https://celestrak.com/software/vallado/fortran.zip.

Listing B.1: covCoe2Eoe

```
!=================================================================!
Subroutine covCoe2Eoe(Pcoe, xcoe, elemType, fr, xeoe, Peoe, Jac)
!-----------------------------------------------------------------!
!! Author: Fabian Schiemenz
!! Date: 09/2019
!-----------------------------------------------------------------!
!! Covariance conversion: COE to EOE
!!
!! References:
!! Vallado and Alfano 2015: Updated Analytical Partials for
!! Covariance Transformations and Optimization (Rev 2)
!-----------------------------------------------------------------!
!> Keplarian covariance
real(dp), dimension(6,6), intent(in)            :: Pcoe
!> Input element set
real(dp), dimension(6), intent(in)              :: xcoe
!> Element set description type
! 0=Peoe sizevar: mean motion, Pcoe fastvar: mean anomaly
! 1=Peoe sizevar: semi-major axis, Pcoe fastvar: mean anomaly
```

1 In principle it is also possible to derive the partial derivatives with respect to the eccentric anomaly/longitude, however their usage is uncommon in this context

```
! 2=Peoe sizevar: mean motion, Pcoe fastvar: true anomaly
! 3=Peoe sizevar: semi-major axis, Pcoe fastvar: true anomaly
! 4=Peoe sizevar: semi-major axis, Pcoe: mean motion/mean anomaly
integer(i4), intent(in)                          :: elemType
!> Type of equinoctial element set (-1 =retrograde, 1=direct)
! If the input is 1 or -1, the specific setting is taken,
! otherwise it is automatically determined
integer(i4), intent(inout)                       :: fr
!> Cartesian covariance
real(dp), dimension(6,6), intent(out)            :: Peoe
!> Converted state vector
real(dp), dimension(6), intent(out), optional    :: xeoe
!> Optional output: Jacobian of the transformation
real(dp), dimension(6,6), intent(out), optional  :: Jac
!-----------------------------------------------------------------!
!Locals:
real(dp)                    :: aUsed, nUsed
real(dp)                    :: mu, muSi
real(dp)                    :: e, i, omega, argp
real(dp)                    :: trueAnom, meanAnom, eccAnom
real(dp)                    :: k, h, p, q, trueLon, meanLon
real(dp)                    :: omegaBar, sinOb, cosOb, cosOmega
real(dp)                    :: sinOmega, iHalf, taniHalf
real(dp), dimension(6,6)    :: Jacobian
!-----------------------------------------------------------------!

!Set local gravitational constant mu
mu   = muEarth
muSi = mu * 1.0E9_dp

!Extract the input state vector
e     = xcoe(2)
i     = xcoe(3)
omega = xcoe(4)
argp  = xcoe(5)

SELECT CASE(elemType)
CASE (0,1,4)
    !Extract the mean anomaly
    meanAnom = xcoe(6)

    !Compute the true anomaly from the mean anomaly
    call NewtonM(xcoe(2), xcoe(6), eccAnom, trueAnom)
CASE(2,3)
    !Extract the true anomaly
    trueAnom = xcoe(6)

    !Compute the mean anomaly from the true anomaly
    call NewtonNu(e, trueAnom, eccAnom, meanAnom)
END SELECT
```

```
if (elemType == 4) then
    !Compute the semi-major axis from the mean motion
    aUsed = (mu**(1.0_dp/3.0_dp))/(xcoe(1)**(2.0_dp/3.0_dp))
else
    !Semi-major axis is already given
    aUsed = xcoe(1)
end if

!Step 1: convert the state vector from COE to EOE and get mean
!       and true fast variables (km-units)
call coe2eoe(ac=aUsed, Ecc=e, Incl=i, Omega=omega, Argp=argp, &
             meanAnom=meanAnom, trueAnom=trueAnom, a=aUsed,   &
             n=nUsed, k=k, h=h, p=p, q=q, meanLon=meanLon,    &
             trueLon=trueLon, fr=fr)

!Return the converted state vector
if (present(xeoe)) then
    SELECT CASE(elemType)
    CASE(0,2)
        !First element = mean motion in EOE
        xeoe(1) = nUsed
    CASE(1,3,4)
        !First element = semi-major axis
        xeoe(1) = aUsed
    END SELECT

    xeoe(2) = h
    xeoe(3) = k
    xeoe(4) = p
    xeoe(5) = q

    !Fast variable type
    SELECT CASE(elemType)
    CASE (0,1,4)
        xeoe(6) = meanLon
    CASE(2,3)
        xeoe(6) = trueLon
    END SELECT
end if

!Auxiliary variables
omegaBar = fr*omega + argp
sinOb    = sin(omegaBar)
cosOb    = cos(omegaBar)

!Transition to SI units
aUsed = aUsed * kilo

!Fill the Jacobian
!Init with zero and then only fill the nonzero elements
Jacobian = 0.0_dp
```

```
!First row: sensitivity with respect to a/n
if (elemType .NE. 4) then
    !Size variable remains equal
    Jacobian(1,1) = 1.0_dp
else
    !Semi-major axis to mean motion sensitivity
    !nUsed=sqrt(muSi/aUsed**3.0_dp)
    Jacobian(1,1) = (-3.0_dp/(2.0_dp*aUsed))*nUsed
end if

!Second row: sensitivity with respect to h
Jacobian(2,2) = sinOb
Jacobian(2,4) = e*fr*cosOb
Jacobian(2,5) = e*cosOb

!Third row: sensitivity with respekt to k
Jacobian(3,2) = cosOb
Jacobian(3,4) = -e*fr*sinOb
Jacobian(3,5) = -e*sinOb

!Fourth row: sensitivity with respect to p
iHalf         = i/2.0_dp
cosOmega      = cos(omega)
sinOmega      = sin(omega)
taniHalf      = tan(iHalf)
Jacobian(4,3) = fr * taniHalf**(fr - 1.0_dp) * sinOmega * &
                (0.5_dp*taniHalf*taniHalf + 0.5_dp)
Jacobian(4,4) = taniHalf**fr * cosOmega

!Fifth row: sensitivity with respect to q
Jacobian(5,3) = fr * taniHalf**(fr - 1.0_dp) * cosOmega * &
                (0.5_dp*taniHalf*taniHalf + 0.5_dp)
Jacobian(5,4) = -taniHalf**fr * sinOmega

!Sixth row: sensitivity with respect to fast variable
Jacobian(6,4) = fr
Jacobian(6,5) = 1.0_dp

!No sizevar-changes, hence always 1.0
Jacobian(6,6) = 1.0_dp

!Assign optional output
if (present(Jac)) Jac = Jacobian

!Compute the transformed matrix via the similarity transformation
Peoe = Matmul(Jacobian, Matmul(Pcoe, transpose(Jacobian)))

!Symmetrize the result to compensate for numerical differences
Peoe = (Peoe + transpose(Peoe))/2.0_dp

End Subroutine covCoe2Eoe
!=============================================================!
```

Listing B.2: covEoe2Coe

```
!=============================================================!
Subroutine covEoe2Coe(Peoe, xeoe, elemType, fr, xcoe, Pcoe, Jac)
!-------------------------------------------------------------!
!! Author: Fabian Schiemenz
!! Date: 09/2019
!-------------------------------------------------------------!
!! Covariance conversion: EOE to COE
!!
!! References:
!! Vallado and Alfano 2015: Updated Analytical Partials for
!! Covariance Transformations and Optimization (Rev 2)
!-------------------------------------------------------------!
!> Keplarian covariance
real(dp), dimension(6,6), intent(in)            :: Peoe
!> Input element set
real(dp), dimension(6), intent(in)              :: xeoe
!> Indicate type of equinoctial elements
integer(i4), intent(in)                         :: elemType
!> Indicate equinoctial element set (direct=1 vs. retrograde=-1)
integer(i4), intent(in)                         :: fr
!> Cartesian covariance
real(dp), dimension(6,6), intent(out)           :: Pcoe
!> Converted state vector
real(dp), dimension(6), intent(out), optional   :: xcoe
!> Optional output: Jacobian of the transformation
real(dp), dimension(6,6), intent(out), optional :: Jac
!-------------------------------------------------------------!
!Locals:
real(dp), dimension(3)      :: r, v
real(dp)                    :: aUsed, nUsed, mu, muSi
real(dp)                    :: e, i, omega, argp, trueAnom
real(dp)                    :: meanAnom, eccAnom, kh2, pq2
real(dp)                    :: k, h, p, q, meanLon, trueLon
real(dp), dimension(6,6)    :: Jacobian
!-------------------------------------------------------------!

!Set local gravitational constant mu
mu   = muEarth
muSi = mu * 1.0E9_dp

!Extract the state vector
SELECT CASE(elemType)
CASE(0,2)
    !Mean motion as first element
    nUsed = xeoe(1)
    aUsed = (mu**(1.0_dp/3.0_dp))/(nUsed**(2.0_dp/3.0_dp))
CASE(1,3)
    !Semi-major axis as first element
    aUsed = xeoe(1)
    nUsed = sqrt(mu/aUsed**3.0_dp)
END SELECT
```

```
h     = xeoe(2)
k     = xeoe(3)
p     = xeoe(4)
q     = xeoe(5)

SELECT CASE(elemType)
CASE(0,1)
    !Mean longitude as fast variable
    meanLon = xeoe(6)

    !Convert the state vector to COE
    call eoe2coe(a=aUsed, n=nUsed, k=k, h=h, p=p, q=q, fr=fr, &
                 meanLon=meanLon, ac=aUsed, Ecc=e, Incl=i,    &
                 Omega=omega, Argp=argp, meanAnom=meanAnom)
    if (present(xcoe)) xcoe(6) = meanAnom
CASE(2,3)
    !True longitude as fast variable
    trueLon = xeoe(6)

    !Convert the state vector to COE
    call eoe2coe(a=aUsed, n=nUsed, k=k, h=h, p=p, q=q, fr=fr, &
                 trueLon=trueLon, ac=aUsed, Ecc=e, Incl=i,    &
                 Omega=omega, Argp=argp, trueAnom=trueAnom)
    if (present(xcoe)) xcoe(6) = trueAnom
END SELECT

if (present(xcoe)) then
    xcoe(1) = aUsed
    xcoe(2) = e
    xcoe(3) = i
    xcoe(4) = omega
    xcoe(5) = argp
end if

!Transition to SI units
aUsed = aUsed * kilo

!Auxiliary variables
kh2 = k*k + h*h
pq2 = p*p + q*q

!Fill the Jacobian
!Init with zero and then only fill the nonzero elements
Jacobian = 0.0_dp

!First row: sensitivity with respect to a/n
!If the sizevar is kept, the partial is always 1.0
Jacobian(1,1) = 1.0_dp

!Second row: sensitivity with respect to eccentricity
Jacobian(2,2) = h/sqrt(kh2)
Jacobian(2,3) = k/sqrt(kh2)
```

```
!Third row: sensitivity with respect to inclination
Jacobian(3,4) = (2.0_dp * fr * p)/((1.0_dp + pq2)*sqrt(pq2))
Jacobian(3,5) = (2.0_dp * fr * q)/((1.0_dp + pq2)*sqrt(pq2))

!Fourth row: sensitivity with respect to RAAN
Jacobian(4,4) =  q/pq2
Jacobian(4,5) = -p/pq2

!Fifth row: sensitivity with respect to argument of perigee
Jacobian(5,2) =  k/kh2
Jacobian(5,3) = -h/kh2
Jacobian(5,4) = -fr*q/pq2
Jacobian(5,5) =  fr*p/pq2

!Sixth row: sensitivity with respect to fast variable
Jacobian(6,2) = -k/kh2
Jacobian(6,3) =  h/kh2
Jacobian(6,6) = 1.0_dp  !No change in type of fast variable

!Assign optional output
if (present(Jac)) Jac = Jacobian

!Compute the transformed matrix via the similarity transformation
Pcoe = Matmul(Jacobian, Matmul(Peoe, transpose(Jacobian)))

!Symmetrize the result
Pcoe = (Pcoe + transpose(Pcoe))/2.0_dp

End Subroutine covEoe2Coe
!=================================================================!
```

Listing B.3: covCoe2RV

```
!=================================================================!
Subroutine covCoe2RV(Pcoe, xcoe, elemType, xcart, Pcart, Jac)
!-----------------------------------------------------------------!
!! Author: Fabian Schiemenz
!! Date: 09/2019
!-----------------------------------------------------------------!
!! Covariance conversion: COE to GCRF
!!
!! Attention: The transition is considerably nonlinear, hence a
!! Gaussian in COE does not result in a pure Gaussian in GCRF.
!! A precise uncertainty characterization in GCRF might need
!! Gaussian sums or the like, depending on the size of the
!! uncertainty.
!!
!! References:
!! Vallado and Alfano 2015: Updated Analytical Partials for
!! Covariance Transformations and Optimization (Rev 2)
!-----------------------------------------------------------------!
!> Keplarian covariance
real(dp), dimension(6,6), intent(in)            :: Pcoe
```

```
!> Input element set
real(dp), dimension(6), intent(in)              :: xcoe
!> Element set:
! 0 = Semi-major axis, mean anomaly
! 1 = Semi-major axis, true anomaly
! 2 = Mean motion, mean anomaly
! 3 = Mean motion, true anomaly
integer(i4), intent(in)                         :: elemType
!> Cartesian covariance
real(dp), dimension(6,6), intent(out)           :: Pcart
!> Converted state vector
real(dp), dimension(6), intent(out), optional   :: xcart
!> Optional output: Jacobian of the transformation
real(dp), dimension(6,6), intent(out), optional  :: Jac
!-----------------------------------------------------------------!
!Locals:
real(dp), dimension(3)      :: r, v, dontCare
real(dp)                    :: Pl, aUsed, nUsed, e, i, omega
real(dp)                    :: argp, trueAnom, eccAnom, meanAnom
real(dp)                    :: sinInc, cosInc, sinOmega, cosOmega
real(dp)                    :: sinArgp, cosArgp, sinNu, cosNu
real(dp)                    :: mu, muSi, e2, dMdv, dMdE
real(dp)                    :: c1, c2, c3, c4, c5, c6, c7
real(dp), dimension(6,6)    :: Jacobian
real(dp), dimension(3,3)    :: p
!-----------------------------------------------------------------!

!Set local gravitational constant mu
mu   = muEarth
muSi = mu * 1.0E9_dp

!Get the mean/true anomalies
if (elemType == 0 .OR. elemType == 2) then
    !Compute the true anomaly from the mean anomaly
    call NewtonM(xcoe(2), xcoe(6), eccAnom, trueAnom)
else
    !True anomaly
    trueAnom = xcoe(6)
end if

!Split the vector
aUsed = xcoe(1)
nUsed = sqrt(mu/aUsed**3.0_dp)
e     = xcoe(2)
i     = xcoe(3)
omega = xcoe(4)
argp  = xcoe(5)

!Semi-parameter
e2 = e*e
Pl = aUsed*(1.0_dp-e2)
```

```
!Convert to GCRF
call coe2rv(P=Pl, Ecc=e, Incl=i, Omega=omega, Argp=argp,     &
            Nu=trueAnom, ArgLat=dontCare(1),                 &
            TrueLon=dontCare(2), LonPer=dontCare(3), R=r, V=v)

if (present(xcart)) then
    xcart(1:3) = r
    xcart(4:6) = v
end if

!Transition to SI units (m and m/s)
r       = r*kilo
v       = v*kilo
aUsed   = aUsed*kilo

!Store trigonometric values
sinInc   = sin(i)
cosInc   = cos(i)
sinOmega = sin(omega)
cosOmega = cos(omega)
sinArgp  = sin(argp)
cosArgp  = cos(argp)
sinNu    = sin(trueAnom)
cosNu    = cos(trueAnom)

!Store the elements of the PQW to ECI transition matrix
!(eq. 14 of Vallado's paper)
p(1,1) =  cosOmega*cosArgp - sinOmega*sinArgp*cosInc
p(1,2) = -cosOmega*sinArgp - sinOmega*cosArgp*cosInc
p(1,3) =  sinOmega*sinInc
p(2,1) =  sinOmega*cosArgp + cosOmega*sinArgp*cosInc
p(2,2) = -sinOmega*sinArgp + cosOmega*cosArgp*cosInc
p(2,3) = -cosOmega*sinInc
p(3,1) =  sinArgp*sinInc
p(3,2) =  cosArgp*sinInc
p(3,3) =  cosInc

!Define constants
c1 = (1.0_dp - e2)/(1.0_dp + e*cosNu)
c2 = sqrt(muSi/(aUsed*(1.0_dp - e2)))
c3 = (2.0_dp * aUsed * e + aUsed*cosNu + aUsed*cosNu*e2)/ &
     ((1.0_dp+e*cosNu)**2.0_dp)
c4 = e/(1.0_dp - e2)
c5 = aUsed*c1
c6 = c5/(1.0_dp + e*cosNu)

!Fill the Jacobian
Jacobian = 0.0_dp

!First column = sensitivity with respect to a
!The following is equal to: c1*(cosNu*p(1:3,1) + sinNu*p(1:3,2))
Jacobian(1:3,1) =  r/aUsed
```

```
!The following is equal to:
!1.0_dp/(2.0_dp * aUsed)*c2*(sinNu*p(1:3,1) - (e+cosNu)*p(1:3,2))
Jacobian(4:6,1) = -v/(2.0_dp * aUsed)

if (elemType == 2 .OR. elemType == 3) then
    !Update partials to mean motion
    Jacobian(:,1) = Jacobian(:,1)*(-2.0_dp/3.0_dp)*aUsed/nUsed
end if

!Second column = sensitivity with respect to eccentricity
Jacobian(1:3,2) = -c3*   (cosNu*p(1:3,1) + sinNu*p(1:3,2))
Jacobian(4:6,2) = -c4*c2*(sinNu*p(1:3,1) - (e+cosNu)*p(1:3,2)) &
                  +p(1:3,2)*c2

!Third column = sensitivity with respect to inclination
Jacobian(1:3,3) = c5*p(1:3,3)*(cosNu*sinArgp + sinNu*cosArgp)
Jacobian(4,3)   =-c2*sinOmega*(sinNu*p(3,1) - (e+cosNu)*p(3,2))
Jacobian(5,3)   = c2*cosOmega*(sinNu*p(3,1) - (e+cosNu)*p(3,2))
Jacobian(6,3)   = c2*p(3,3)* (-sinNu*sinArgp + (e+cosNu)*cosArgp)

!Fourth column = sensitivity with respect to RAAN
Jacobian(1,4) = -c5*(cosNu*p(2,1)+sinNu*p(2,2))
Jacobian(2,4) =  c5*(cosNu*p(1,1)+sinNu*p(1,2))
Jacobian(3,4) =  0.0_dp
Jacobian(4,4) =  c2*(sinNu*p(2,1)-(e+cosNu)*p(2,2))
Jacobian(5,4) = -c2*(sinNu*p(1,1)-(e+cosNu)*p(1,2))
Jacobian(6,4) =  0.0_dp

!Fifth column = sensitivity with respect to argument of perigee
Jacobian(1:3,5) = c5*(cosNu*p(1:3,2) - sinNu*p(1:3,1))
Jacobian(4:6,5) = -c2*(sinNu*p(1:3,2) + (e+cosNu)*p(1:3,1))

!Sixth column = sensitivity with respect to fast variable
!Note that in the paper the true and mean anomaly
!sensitivities are switched

!True anomaly in COE covariance
Jacobian(1:3,6) = c6*(-sinNu*p(1:3,1) + (e+cosNu)*p(1:3,2))
Jacobian(4:6,6) = c2*(-cosNu*p(1:3,1) - sinNu*p(1:3,2))

if (elemType == 0 .OR. elemType == 2) then
    !Update the true anomaly sensitivities to the
    !mean anomaly sensitivities
    !Note: the equations in Vallado have been derived using
    !MathCad. Hence many parts can possibly be simplified.
    !Indeed, this is also the case with the following equations.
    !The long sine/cosine expressions are nothing else
    !than a different form of the true anomaly sensitivities.
    !There is no need to recompute the mean anomaly sensitivities
    !from scratch, since the true anomaly sensitivities can be
    !simply updated.
```

```
    !For the sake of completeness the block below contains
    !the alternate form of the true anomaly sensitivities
    !c7 = aUsed * (e**2.0_dp - 1.0_dp)/(e*cosNu + 1.0_dp)**2.0_dp
    !Jacobian(1,6) = c7 * (e*cosOmega*sinArgp            + &
    !                      cosOmega*cosArgp*sinNu        + &
    !                      cosOmega*sinArgp*cosNu        + &
    !                      e*cosInc*sinOmega*cosArgp     + &
    !                      cosInc*sinOmega*cosArgp*cosNu - &
    !                      cosInc*sinOmega*sinArgp*sinNu)
    !Jacobian(2,6) = c7 * (e*sinOmega*sinArgp            + &
    !                      sinOmega*cosArgp*sinNu        + &
    !                      sinOmega*sinArgp*cosNu        - &
    !                      e*cosInc*cosOmega*cosArgp     - &
    !                      cosInc*cosOmega*cosArgp*cosNu + &
    !                      cosInc*cosOmega*sinArgp*sinNu)
    !Jacobian(3,6) = -c7 * sinInc*(cos(argp+trueAnom)      &
    !                    + e*cosArgp)
    !Jacobian(4,6) =  c2 * (cosOmega*sinArgp*sinNu         &
    !                    - cosOmega*cosArgp*cosNu          &
    !                    + cosInc*sinOmega*cosArgp*sinNu
    !                    + cosInc*sinOmega*sinArgp*cosNu)
    !Jacobian(5,6) = -c2 * (sinOmega*cosArgp*cosNu         &
    !                      -sinOmega*sinArgp*sinNu         &
    !                      +cosInc*cosOmega*cosArgp*sinNu  &
    !                      +cosInc*cosOmega*sinArgp*cosNu)
    !Jacobian(6,6) = -c2 * (-sin(argp+trueAnom)*sinInc)

    dMdv  = (1.0_dp - e2)**1.5_dp / ((1.0_dp + e*cosNu)**2.0_dp)
    dMdE  = -sinNu*((1.0_dp + e*cosNu)*sign(1.0_dp, e + cosNu) &
            + 1.0_dp - 2.0_dp*e2 - e*e2*cosNu) /              &
            ((1.0_dp + e*cosNu)**2.0_dp * sqrt(1.0_dp - e2))

    !Update the true anomaly sensitivities to mean anomaly in
    !the reverse direction
    Jacobian(1:6,6) = Jacobian(1:6,6)/dMdv
    Jacobian(1:6,2) = Jacobian(1:6,2) - dMdE*Jacobian(1:6,6)
end if

!Assign optional output
if (present(Jac)) Jac = Jacobian

!Compute the transformed matrix via the similarity transformation
Pcart = Matmul(Jacobian, Matmul(Pcoe, transpose(Jacobian)))

!Symmetrize the result
Pcart = (Pcart + transpose(Pcart))/2.0_dp

End Subroutine covCoe2RV
!=================================================================!
```

Listing B.4: covRV2Coe

```
!=============================================================!
Subroutine covRV2Coe(Pcart, xcart, xcoe, elemType, Pcoe, Jac)
!-------------------------------------------------------------!
!! Author: Fabian Schiemenz
!! Date: 09/2019
!-------------------------------------------------------------!
!! Covariance conversion: GCRF to COE
!!
!! References:
!! Danielsson, 1995: Semianalytical Satellite Theory
!! Vallado and Alfano 2015: Updated Analytical Partials for
!! Covariance Transformations and Optimization (Rev 2)
!-------------------------------------------------------------!
!> Cartesian covariance
real(dp), dimension(6,6), intent(in)            :: Pcart
!> Cartesian state
real(dp), dimension(6), intent(in)              :: xcart
!> Element type
! 0 = Semi-major axis, mean anomaly
! 1 = Semi-major axis, true anomaly
! 2 = Mean motion, mean anomaly
! 3 = Mean motion, true anomaly
integer(i4), intent(in)                         :: elemType
!> Equinoctial covariance
real(dp), dimension(6,6), intent(out)           :: Pcoe
!> Converted state vector in equinoctial elements
real(dp), dimension(6), intent(out), optional   :: xcoe
!> Optional output: Jacobian of the transformation
real(dp), dimension(6,6), intent(out), optional :: Jac
!-------------------------------------------------------------!
!Locals:
real(dp), dimension(3)   :: r, v, hvec, nvec, Kvec, eccVec
real(dp), dimension(3)   :: v_cross_h, h_cross_r
real(dp)                 :: aUsed, nUsed, e, i, omega, argp
real(dp)                 :: trueAnom, meanAnom
real(dp)                 :: mu, muSi, rMag, rMag2, rMag3, e2
real(dp)                 :: vMag, vMag2, nMag, nMag3, hMag, hMag2
real(dp)                 :: c1, c2, c3, c4, c5, r_dot_v, n_dot_e
real(dp), dimension(4)   :: dontCare
real(dp)                 :: wScale, sgn, dMdv, dMdE, cosNu
real(dp)                 :: cosArgp, nMage, muenMag
real(dp), dimension(6,6) :: Jacobian
!-------------------------------------------------------------!

!Set local gravitational constant mu
mu   = muEarth
muSi = mu * 1.0E9_dp

!Define k
kVec = (/0.0_dp, 0.0_dp, 1.0_dp/)
```

```
!Split vector and convert to SI units
r = xcart(1:3)*kilo
v = xcart(4:6)*kilo

!Compute specific angular momentum vector
call cross(r, v, hvec)

!Compute the node vector
call cross(Kvec, hvec, nVec)

!Compute magnitudes and powers
rMag  = MAG(r)
rMag2 = rMag * rMag
rMag3 = rMag2 * rMag
vMag  = MAG(v)
vMag2 = vMag * vMag
nMag  = MAG(nVec)
nMag3 = nMag * nMag * nMag
hMag  = MAG(hVec)
hMag2 = hMag * hMag

!Compute the eccentricity vector
eccVec = ((vMag2 - muSi/rMag) * r - DOT(r, v) * v)/muSi

!First, convert the state vector to obtain the
!COE elements (kilometer-units)
call rv2coe(R=xcart(1:3), V=xcart(4:6), mu=mu, P=dontCare(1), &
            A=aUsed, Ecc=e, Incl=i, Omega=omega, Argp=argp,   &
            Nu=trueAnom, M=meanAnom, ArgLat=dontCare(2),      &
            TrueLon=dontCare(3), LonPer=dontCare(4))
nUsed = sqrt(mu/aUsed**3.0_dp)

if (present(xcoe)) then
    !Return the output vector
    xcoe(1) = aUsed
    xcoe(2) = e
    xcoe(3) = i
    xcoe(4) = omega
    xcoe(5) = argp
    if (elemType == 0) then
        xcoe(6) = meanAnom
    else
        xcoe(6) = trueAnom
    end if
end if

!Convert to m
aUsed = aUsed*kilo

!From here on everything in SI (m and m/s)
```

```
!Auxiliary scalars
c1 = -1.0_dp/(muSi*e)
c2 = -1.0_dp/nMag
c3 =  1.0_dp/(nMag*nMag)

!Initialize the Jacobian
Jacobian = 0.0_dp

!First row: sensitivity with respect to semi-major axis
Jacobian(1,1:3) = ((2.0_dp * aUsed**2.0_dp) / rMag3) * r
!Vallado's paper is wrong in the next line
Jacobian(1,4:6) = (2.0_dp/(nUsed**2.0 * aUsed)) * v

if (elemType == 2 .OR. elemType == 3) then
    !Adapt to mean motion
    Jacobian(1,:) = Jacobian(1,:)*(-3.0_dp/2.0_dp)*(nUsed/aUsed)
end if

!Second row: sensitivity with respect to eccentricity
Jacobian(2,1)=c1*((v(1)*v(2)-(muSi*r(1)*r(2))/rMag3)*eccVec(2) &
               + (v(1)*v(3)-(muSi*r(1)*r(3))/rMag3)*eccVec(3) &
               - (v(2)**2.0_dp+v(3)**2.0_dp - muSi/rMag       &
               + (muSi*r(1)**2.0_dp)/rMag3)*eccVec(1))
Jacobian(2,2)=c1*((v(1)*v(2)-(muSi*r(1)*r(2))/rMag3)*eccVec(1) &
               + (v(2)*v(3)-(muSi*r(2)*r(3))/rMag3)*eccVec(3) &
               - (v(1)**2.0_dp + v(3)**2.0_dp - muSi/rMag     &
               + (muSi*r(2)**2.0_dp)/rMag3)*eccVec(2))
Jacobian(2,3)=c1*((v(1)*v(3)-(muSi*r(1)*r(3))/rMag3)*eccVec(1) &
               + (v(2)*v(3)-(muSi*r(2)*r(3))/rMag3)*eccVec(2) &
               - (v(2)**2.0_dp + v(1)**2.0_dp - muSi/rMag     &
               + (muSi*r(3)**2.0_dp)/rMag3)*eccVec(3))
Jacobian(2,4)=c1*((r(1)*v(2) - 2.0_dp*r(2)*v(1))*eccVec(2)     &
               + (r(2)*v(2) + r(3)*v(3))*eccVec(1)            &
               + (r(1)*v(3) - 2.0_dp*r(3)*v(1))*eccVec(3))
               !Vallado's paper has a minus in the middle term!
Jacobian(2,5)=c1*((r(2)*v(1) - 2.0_dp*r(1)*v(2))*eccVec(1)     &
               + (r(1)*v(1) + r(3)*v(3))*eccVec(2)            &
               + (r(2)*v(3) - 2.0_dp*r(3)*v(2))*eccVec(3))
               !Vallado's paper has a minus in the middle term!
Jacobian(2,6)=c1*((r(1)*v(1) + r(2)*v(2))*eccVec(3)            &
               + (r(3)*v(1) - 2.0_dp*r(1)*v(3))*eccVec(1)     &
               + (r(3)*v(2) - 2.0_dp*r(2)*v(3))*eccVec(2))

!Third row: sensitivity with respect to  inclination
Jacobian(3,1) =  c2*(v(2)-(hVec(3)*(v(2)*hVec(3)-v(3)*hVec(2))/ &
                    hMag2))
Jacobian(3,2) = -c2*(v(1)-(hVec(3)*(v(1)*hVec(3)-v(3)*hVec(1))/ &
                    hMag2))
Jacobian(3,3) =  c2*(     (hVec(3)*(v(2)*hVec(1)-v(1)*hVec(2))/ &
                    hMag2))
Jacobian(3,4) = -c2*(r(2)-(hVec(3)*(r(2)*hVec(3)-r(3)*hVec(2))/ &
                    hMag2))
```

```
Jacobian(3,5) =  c2*(r(1)-(hVec(3)*(r(1)*hVec(3)-r(3)*hVec(1))/ &
                    hMag2))
Jacobian(3,6) = -c2*(     (hVec(3)*(r(2)*hVec(1)-r(1)*hVec(2))/ &
                    hMag2)) !Vallado's paper is wrong here!

!Fourth row: sensitivity with respect to RAAN
Jacobian(4,1) = -v(3)*nVec(2)*c3
Jacobian(4,2) =  v(3)*nVec(1)*c3
Jacobian(4,3) =  (v(1)*nVec(2) - v(2)*nVec(1))*c3
Jacobian(4,4) =  r(3)*nVec(2)*c3
Jacobian(4,5) = -r(3)*nVec(1)*c3
Jacobian(4,6) =  (r(2)*nVec(1) - r(1)*nVec(2))*c3

!Fifth row: sensitivity with respect to argument of perigee
!These sensitivities are incorrect in Vallado's paper!
n_dot_e       = DOT(nVec, eccVec)
nMage         = nMag*e
cosArgp       = n_dot_e/nMage
sgn           = sign(1.0_dp, eccVec(3)*muSi)
wScale        = -sgn/abs(sin(argp))
muenMag       = muSi*nMage

Jacobian(5,1) = wScale * ( (-hVec(2)*(v(2)*v(2)                &
              + v(3)*v(3) - muSi/rMag + muSi*r(1)*r(1)/rMag3)  &
              - hVec(1)*(v(1)*v(2) - muSi*r(1)*r(2)/rMag3)     &
              + v(3)*muSi*eccVec(1))/muenMag + n_dot_e/e *     &
              ( v(3)*hVec(2)/nMag3 - Jacobian(2,1)/nMage))
Jacobian(5,2) = wScale * ( ( hVec(1)*(v(1)*v(1)                &
              + v(3)*v(3) - muSi/rMag + muSi*r(2)*r(2)/rMag3)  &
              + hVec(2)*(v(1)*v(2) - muSi*r(1)*r(2)/rMag3)     &
              + v(3)*muSi*eccVec(2))/muenMag + n_dot_e/e *     &
              (-v(3)*hVec(1)/nMag3 - Jacobian(2,2)/nMage))
Jacobian(5,3) = wScale * (-(-hVec(2)*(v(1)*v(3)                &
              - muSi*r(1)*r(3)/rMag3)+ hVec(1)*(v(2)*v(3)      &
              - muSi*r(2)*r(3)/rMag3) + v(1)*muSi*eccVec(1)    &
              + v(2)*muSi*eccVec(2))/muenMag + n_dot_e/e *     &
              ((v(2)*hVec(1)-v(1)*hVec(2))/nMag3               &
              - Jacobian(2,3)/nMage))
Jacobian(5,4) = wScale*(-( hVec(1)*(r(1)*v(2)-2.0_dp*r(2)*v(1)) &
              - hVec(2)*(r(2)*v(2) + r(3)*v(3))                &
              + r(3)*muSi*eccVec(1))/muenMag + n_dot_e/e *     &
              (-r(3)*hVec(2)/nMag3 - Jacobian(2,4)/nMage))
Jacobian(5,5) = wScale*(-(-hVec(2)*(r(2)*v(1)-2.0_dp*r(1)*v(2)) &
              + hVec(1)*(r(1)*v(1) + r(3)*v(3))                &
              + r(3)*muSi*eccVec(2))/muenMag + n_dot_e/e *     &
              ( r(3)*hVec(1)/nMag3 - Jacobian(2,5)/nMage))
Jacobian(5,6) = wScale*(-(-hVec(2)*(r(3)*v(1)-2.0_dp*r(1)*v(3)) &
              + hVec(1)*(r(3)*v(2) - 2.0_dp*r(2)*v(3)) -       &
              r(1)*muSi*eccVec(1)-r(2)*muSi*eccVec(2))/muenMag &
              + n_dot_e/e * ((r(1)*hVec(2)-r(2)*hVec(1))/nMag3 &
              - Jacobian(2,6)/nMage))
```

```
!Sixth row: sensitivity with respect to fast variable
!Formulate first for true anomaly and update later to
!mean anomaly, if necessary
r_dot_v         = DOT(r, v)
c4              = rMag2 * vMag2 - muSi*rMag - r_dot_v*r_dot_v
c5              = 1.0_dp / (c4*c4 + r_dot_v*r_dot_v * hMag2)
call CROSS(v, hVec, v_cross_h)
call CROSS(hVec, r, h_cross_r)
Jacobian(6,1:3) = c5 * (c4*(hMag*v + (r_dot_v/hMag)*v_cross_h)
                - r_dot_v*hMag*(2.0_dp * vMag2 * r
                - (muSi*r)/rMag - 2.0_dp*r_dot_v*v))
Jacobian(6,4:6) = c5 * (c4*(hMag*r + (r_dot_v/hMag)*h_cross_r)
                - r_dot_v*hMag*(2.0_dp * rMag2 * v
                - 2.0_dp*r_dot_v*r))

if (elemType == 0 .OR. elemType == 2) then
    !Adapt fast variable partials to mean anomaly
    cosNu = cos(trueAnom)
    e2    = e*e

    !Vallado misses the power of two in the denominator
    dMdv  = (1.0_dp - e2)**1.5_dp / ((1.0_dp + e*cosNu)**2.0_dp)

    !Error in paper for dMdE
    dMdE  = -sin(trueAnom)*((1.0_dp + e*cosNu)*sign(1.0_dp,  &
            e+cosNu) + 1.0_dp - 2.0_dp*e2 - e*e2*cosNu)      &
            /((1.0_dp + e*cosNu)**2.0_dp * sqrt(1.0_dp - e2))

    !Multiply the sensitivities to the respective terms
    Jacobian(6,:) = dMdv*Jacobian(6,:) + dMdE*Jacobian(2,:)
end if

!Assign optional output
if (present(Jac)) Jac = Jacobian

!Compute the transformed matrix via the similarity transformation
Pcoe = Matmul(Jacobian, Matmul(Pcart, transpose(Jacobian)))

!Symmetrize the result
Pcoe = (Pcoe + transpose(Pcoe))/2.0_dp

End Subroutine covRV2Coe
!=================================================================!
```

Listing B.5: covEoe2RV

```
!=================================================================!
Subroutine covEoe2RV(fr,xeoe,elemSwitch,xcart,Peoe,Pcart,Jac)
!-----------------------------------------------------------------!
!! Author: Fabian Schiemenz
!! Date: 09/2019
!-----------------------------------------------------------------!
!! Covariance conversion: EOE to GCRF
```

```
!! Attention: The transition is considerably nonlinear, hence a
!! Gaussian in EOE does not result in a pure Gaussian in GCRF.
!! A precise uncertainty characterization in GCRF might need
!! Gaussian sums or the like, depending on the size of the
!! uncertainty.
!!
!! References:
!! Danielsson, 1995: Semianalytical Satellite Theory
!!                   (contains a typo in dq/dr, page 12)
!! Vallado and Alfano 2015: Updated Analytical Partials for
!! Covariance Transformations and Optimization (Rev 2)
!-----------------------------------------------------------------!
! Inputs
!> Type of element set of components (1=direct, -1=retrograde)
integer(i4), intent(inout)                      :: fr
!> Equinoctial element set
real(dp), dimension(6), intent(in)              :: xeoe
!> Equinoctial covariance matrix
real(dp), dimension(6,6), intent(in)            :: Peoe
!> ElementType-switch
integer(i4), intent(in), optional               :: elemSwitch
! Outputs
!> Converted Cartesian covariance
real(dp), dimension(6,6), intent(out)           :: Pcart
!> Optional output: converted element set
real(dp), dimension(6), intent(out), optional   :: xcart
!> Optional output: Jacobian of the transformation
real(dp), dimension(6,6), intent(out), optional :: Jac
!-----------------------------------------------------------------!
!Locals:
real(dp), dimension(6)      :: z
real(dp), dimension(3)      :: fVec, gVec, wVec, r, v, dhkVec
real(dp)                    :: X, Y, XDot, YDot, rMag, mu
real(dp)                    :: L, ch, ck, cosL, sinL
real(dp)                    :: k, h, p, q, A, bcap, b, C
real(dp)                    :: nUsed, aUsed, sensh, sensk
real(dp)                    :: dXdh, dYdh, dXdk, dYdk
real(dp)                    :: dXdotdh, dYdotdh, dXdotdk, dYdotdk
integer(i4)                 :: eleSwitchUsed
real(dp), dimension(6,6)    :: Jacobian
real(dp)                    :: vMag, e, e2, dMdv, dMde
real(dp)                    :: r_dot_e, muSi, sqrtE2, oneEcosNu
real(dp), dimension(3)      :: eccVec
!-----------------------------------------------------------------!

!Set local gravitational constant mu
mu   = muEarth

if (present(elemSwitch)) then
    eleSwitchUsed = elemSwitch
else
    !Default: direct set with mean motion and mean longitude
```

```
    eleSwitchUsed = 0
end if

!Split the vector
h        = xeoe(2)
k        = xeoe(3)
p        = xeoe(4)
q        = xeoe(5)

!Step 1: Convert the EOE vector into the corresponding Cartesian
!        vector using the full nonlinear relationship and get
!        auxiliary variables which are also needed to convert the
!        covariance (X, Y, XDot, YDot, ...)
SELECT CASE(eleSwitchUsed)
CASE(0)
    !Mean motion and mean longitude
    call eoe2rv(fr = fr, n = xeoe(1), h = h, k = k,             &
                p = p, q = q, meanLon = xeoe(6),                &
                r = z(1:3), v = z(4:6),                         &
                X = X, Y = Y, XDot = XDot, YDot = YDot,         &
                fv = fVec, gv = gVec, wv = wVec, radDist=rMag)
    nUsed = xeoe(1)
    aUsed = mu**(1.0_dp/3.0_dp)/nUsed**(2.0_dp/3.0_dp)
CASE(1)
    !Semi-major axis and mean longitude
    call eoe2rv(fr = fr, a = xeoe(1), h = h, k = k,             &
                p = p, q = q, meanLon = xeoe(6),                &
                r = z(1:3), v = z(4:6),                         &
                X = X, Y = Y, XDot = XDot, YDot = YDot,         &
                fv = fVec, gv = gVec, wv = wVec, radDist=rMag)
    aUsed = xeoe(1)
    nUsed = sqrt(mu/aUsed**3.0_dp)
CASE(2)
    !Mean motion and true longitude
    call eoe2rv(fr = fr, n = xeoe(1), h = h, k = k,             &
                p = p, q = q, trueLon = xeoe(6),                &
                r = z(1:3), v = z(4:6),                         &
                X = X, Y = Y, XDot = XDot, YDot = YDot,         &
                fv = fVec, gv = gVec, wv = wVec, radDist=rMag)

    nUsed = xeoe(1)
    aUsed = mu**(1.0_dp/3.0_dp)/nUsed**(2.0_dp/3.0_dp)
CASE(3)
    !Semi-major axis and true longitude
    call eoe2rv(fr = fr, a = xeoe(1), h = h, k = k,             &
                p = p, q = q, trueLon = xeoe(6),                &
                r = z(1:3), v = z(4:6),                         &
                X = X, Y = Y, XDot = XDot, YDot = YDot,         &
                fv = fVec, gv = gVec, wv = wVec, radDist=rMag)
    aUsed = xeoe(1)
    nUsed = sqrt(mu/aUsed**3.0_dp)
END SELECT
```

```
!Assign optional output
if (present(xcart)) xcart = z

!Transition to SI units
aUsed = aUsed  * kilo
rMag  = rMag   * kilo
r     = z(1:3) * kilo
v     = z(4:6) * kilo
X     = X      * kilo
Y     = Y      * kilo
XDot  = XDot   * kilo
YDot  = YDot   * kilo

!Define auxiliary parameters for the Jacobian
A        = nUsed * aUsed**2.0_dp
bcap     = sqrt(1.0_dp - h**2.0_dp - k**2.0_dp)
b        = 1.0_dp/(1.0_dp + bcap)
C        = 1.0_dp + p**2.0_dp + q**2.0_dp

dXdh     = -(k*Xdot)/(nUsed*(1.0_dp+bcap))       &
           + (aUsed * Y * YDot)/(A * bcap)
dXdk     =  (h*Xdot)/(nUsed*(1.0_dp+bcap))       &
           + (aUsed * Y * XDot)/(A * bcap) - aUsed
dYdh     = -(k*Ydot)/(nUsed*(1.0_dp+bcap))       &
           - (aUsed * X * YDot)/(A * bcap) - aUsed
dYdk     =  (h*Ydot)/(nUsed*(1.0_dp+bcap))       &
                   - (aUsed * X * XDot)/(A * bcap)

dXDotdh  =  (aUsed*YDot*YDot)/(A*bcap) + (A/rMag**3.0_dp) &
            * ((aUsed*k*X)/(1.0_dp + bcap) - (Y*Y)/bcap)
dYDotdh  = -(aUsed*XDot*YDot)/(A*bcap) + (A/rMag**3.0_dp) &
            * ((aUsed*k*Y)/(1.0_dp + bcap) + (X*Y)/bcap)
dXDotdk  =  (aUsed*XDot*YDot)/(A*bcap) - (A/rMag**3.0_dp) &
            * ((aUsed*h*X)/(1.0_dp + bcap) + (X*Y)/bcap)
dYDotdk  = -(aUsed*XDot*XDot)/(A*bcap) - (A/rMag**3.0_dp) &
            * ((aUsed*h*Y)/(1.0_dp + bcap) - (X*X)/bcap)

if (eleSwitchUsed == 2 .OR. eleSwitchUsed == 3) then
    cosL = cos(xeoe(6))
    sinL = sin(xeoe(6))
    ch = aUsed*(2.0_dp*k*h*cosL+2.0_dp*h+(h*h-k*k+1.0_dp)*sinL) &
         /(C*(1.0_dp + k*cosL + h*sinL)**2.0_dp)
    ck = aUsed*(2.0_dp*k*h*sinL+2.0_dp*k+(k*k-h*h+1.0_dp)*cosL) &
         /(C*(1.0_dp + k*cosL + h*sinL)**2.0_dp)

    dhkVec(1) = (p*p-q*q-1.0_dp)*cosL    - 2.0_dp*p*q*fr*sinL
    !The following are not needed, but complete dhkVec
    !dhkVec(2) = (p*p-q*q+1.0_dp)*fr*sinL + 2.0_dp*p*q*cosL
    !dhkVec(3) =  2.0_dp*(p*fr*cosL       - q*sinL)
end if
```

```
!Compute the 6x6 Jacobi matrix
Jacobian = 0.0_dp

!First column: mean motion sensitivity or semi-major axis
!sensitivity
SELECT CASE(eleSwitchUsed)
CASE(0,2)
    !Mean motion
    Jacobian(1:3,1) = -(2.0_dp/(3.0_dp*nUsed))*r
    Jacobian(4:6,1) =  v/(3.0_dp * nUsed)
CASE(1,3)
    !Semi-major axis
    Jacobian(1:3,1) =  r/aUsed
    Jacobian(4:6,1) = -v/(2.0_dp * aUsed)
END SELECT

!Compute the sixth column prior to the second and third, as it
!might be needed for their adaption in case that
!eleSwitchUsed == 2 .OR. eleSwitchUsed == 3.
!Sixth column: sensitivity with respect to fast variable
!Mean longitude
Jacobian(1:3,6) = v/nUsed
Jacobian(4:6,6) = -((nUsed*aUsed**3.0_dp)/(rMag**3.0_dp))*r

if (eleSwitchUsed == 2 .OR. eleSwitchUsed == 3) then
    vMag      = MAG(v)
    muSi      = mu * 1.0E09_dp
    eccVec    = ((vMag*vMag - muSi/rMag)*r - DOT(r, v)*v)/muSi
    e         = MAG(eccVec)
    oneEcosNu = 1.0_dp + DOT(r, eccVec)/rMag
    dMdv      = (1.0_dp - e*e)**1.5_dp / (oneEcosNu*oneEcosNu)

    !Note that all sensitivities where the eccentricity or
    !fast variable are included need to be updated.
    !These are columns 2,3 (h and k), which are updated below
    !and column 6 (mean longitude -> true longitude)
    !dM/dv is equal to dLambda/dL -> direct update of dependency
    Jacobian(:,6) = dMdv*Jacobian(:,6)
end if

!Second column: h-sensitivity
Jacobian(1:3,2) = dXdh * fVec    + dYdh * gVec
Jacobian(4:6,2) = dXdotdh * fVec + dYdotdh * gVec

if (eleSwitchUsed == 2 .OR. eleSwitchUsed == 3) then
    !To update the h-sensitivity, i.e. column 2, the true
    !longitude sensitivity needs to be added with a new
    !sensitivity multiplied. This sensitivity is applicable
    !for the entire column. Hence it can be found from an
    !arbitrary component of the vector. Easiest for this
    !operation is the position part, as its math is simpler.
```

```
    !Based on the position part of the state vector,
    !the desired sensitivity can be computed via
    !sensh = (ch*dhkVec(X) - Jacobian(X,2))/Jacobian(X,6)
    !where X can be selected as 1,2 or 3.
    !This sensitivity can then be multiplied to the entire column
    !to update also the velocity part at the same time.
    !The same procedure also holds for the k component.

    !Compute sensitivity based on first component
    sensh = (ch*dhkVec(1) - Jacobian(1,2))/Jacobian(1,6)

    !Update the column
    Jacobian(:,2) = Jacobian(:,2) + Jacobian(:,6)*sensh
end if

!Third column: k-sensitivity
Jacobian(1:3,3) = dXdk * fVec    + dYdk * gVec
Jacobian(4:6,3) = dXdotdk * fVec + dYdotdk * gVec

if(eleSwitchUsed == 2 .OR. eleSwitchUsed == 3) then
    !Compute sensitivity based on first component
    sensk = (ck * dhkVec(1) - Jacobian(1,3))/Jacobian(1,6)

    !Update the column
    Jacobian(:,3) = Jacobian(:,3) + Jacobian(:,6)*sensk
end if

!Fourth column: p-sensitivity
Jacobian(1:3,4) = 2.0_dp*(fr*q*(Y*fVec-X*gVec)-X*wVec)/C
Jacobian(4:6,4) = 2.0_dp*(fr*q*(YDot*fVec-XDot*gVec)-XDot*wVec)/C

!Fifth column: q-sensitivity
!(Vallado has a typo here: fr must be outside the parenthesis)
Jacobian(1:3,5) = 2.0_dp*fr*(p*(X*gVec-Y*fVec)+Y*wVec)/C
Jacobian(4:6,5) = 2.0_dp*fr*(p*(XDot*gVec-YDot*fVec)+YDot*wVec)/C

!Assign optional output
if (present(Jac)) Jac = Jacobian

!Compute the transformed matrix via the similarity transformation
Pcart = Matmul(Jacobian, Matmul(Peoe, transpose(Jacobian)))

!Symmetrize the result
Pcart = (Pcart + transpose(Pcart))/2.0_dp

End Subroutine covEoe2RV
!=================================================================!
```

Listing B.6: covRV2Eoe

```
!=============================================================!
Subroutine covRV2Eoe(Pcart,xcart,xeoe,elemSwitch,fr,Peoe,Jac)
!-------------------------------------------------------------!
!! Author: Fabian Schiemenz
!! Date: 09/2019
!-------------------------------------------------------------!
!! Covariance conversion: GCRF to EOE
!!
!! References:
!! Danielsson, 1995: Semianalytical Satellite Theory
!! Vallado and Alfano 2015: Updated Analytical Partials for
!! Covariance Transformations and Optimization (Rev 2)
!-------------------------------------------------------------!
!> Cartesian covariance
real(dp), dimension(6,6), intent(in)            :: Pcart
!> Converted element set
real(dp), dimension(6), intent(in)              :: xcart
!> Optional: element switch
integer(i4), intent(in), optional               :: elemSwitch
!> Type of equinoctial element set (1=direct, -1=retrograde)
integer(i4), intent(inout)                      :: fr
!> Equinoctial covariance
real(dp), dimension(6,6), intent(out)           :: Peoe
!> Converted state vector to equinoctial elements
real(dp), dimension(6), intent(out), optional   :: xeoe
!> Optional output: Jacobian of the transformation
real(dp), dimension(6,6), intent(out), optional :: Jac
!-------------------------------------------------------------!
!Locals:
real(dp), dimension(3)      :: fVec, gVec, wVec, dhkVec, eccVec
real(dp)                    :: X, Y, XDot, YDot, rMag, ch, ck
real(dp)                    :: sensh, sensk, cosL, sinL
real(dp)                    :: k, h, p, q, A, bcap, b, C
real(dp)                    :: nUsed, aUsed, L, lambda
real(dp)                    :: dXdh, dYdh, dXdk, dYdk
real(dp)                    :: dXdotdh, dYdotdh, dXdotdk, dYdotdk
real(dp), dimension(3)      :: dvdh, dvdk, r, v
integer(i4)                 :: elemSwitchUsed
real(dp), dimension(6,6)    :: Jacobian
real(dp)                    :: mu, muSi, e, oneEcosNu, e2
real(dp)                    :: vMag, dMdv, fVec1, gVec1
!-------------------------------------------------------------!

!Set local gravitational constant mu
mu   = muEarth
muSi = mu * 1.0E9_dp

!Step 1: Convert the Cartesian vector to equinoctial elements
!        and also fetch the auxiliary variables
call rv2eoe(r=xcart(1:3), v=xcart(4:6), n=nUsed, a=aUsed,       &
            k=k, h=h, p=p, q=q, meanLon=lambda, trueLon=L,      &
```

```
            fr=fr, X=X, Y=Y, XDot=XDot, YDot=YDot, radDist=rMag,&
            fv=fVec, gv=gVec, wv=wVec)

if (present(elemSwitch)) then
    elemSwitchUsed = elemSwitch
else
    !Default: mean motion as sizevar and
    !mean longitude as fastvar
    elemSwitchUsed = 0
end if

if (present(xeoe)) then
    SELECT CASE(elemSwitchUsed)
    CASE(0)
        !Mean motion, mean longitude
        xeoe(1) = nUsed
        xeoe(6) = lambda
    CASE(1)
        !Semi-major axis, mean longitude
        xeoe(1) = aUsed
        xeoe(6) = lambda
    CASE(2)
        !Mean motion, true longitude
        xeoe(1) = nUsed
        xeoe(6) = L
    CASE(3)
        !Semi-major axis, true longitude
        xeoe(1) = aUsed
        xeoe(6) = L
    END SELECT

    xeoe(2) = h
    xeoe(3) = k
    xeoe(4) = p
    xeoe(5) = q
end if

!Transition to SI units for the Jacobian
aUsed = aUsed      * kilo
rMag  = rMag       * kilo
r     = xcart(1:3) * kilo
v     = xcart(4:6) * kilo
X     = X          * kilo
Y     = Y          * kilo
XDot  = XDot       * kilo
YDot  = YDot       * kilo

!Define auxiliary variables
A        = nUsed * aUsed**2.0_dp
bcap     = sqrt(1.0_dp - h**2.0_dp - k**2.0_dp)
b        = 1.0_dp/(1.0_dp + bcap)
C        = 1.0_dp + p**2.0_dp + q**2.0_dp
```

```
dXdh     = -(k*Xdot)/(nUsed*(1.0_dp+bcap)) &
           + (aUsed * Y * YDot)/(A * bcap)
dXdk     =  (h*Xdot)/(nUsed*(1.0_dp+bcap)) &
           + (aUsed * Y * XDot)/(A * bcap) - aUsed
dYdh     = -(k*Ydot)/(nUsed*(1.0_dp+bcap)) &
           - (aUsed * X * YDot)/(A * bcap) - aUsed
dYdk     =  (h*Ydot)/(nUsed*(1.0_dp+bcap)) &
           - (aUsed * X * XDot)/(A * bcap)

dXDotdh  =  (aUsed*YDot*YDot)/(A*bcap) + (A/rMag**3.0_dp) &
            * ((aUsed*k*X)/(1.0_dp + bcap) - (Y*Y)/bcap)
dYDotdh  = -(aUsed*XDot*YDot)/(A*bcap) + (A/rMag**3.0_dp) &
            * ((aUsed*k*Y)/(1.0_dp + bcap) + (X*Y)/bcap)
dXDotdk  =  (aUsed*XDot*YDot)/(A*bcap) - (A/rMag**3.0_dp) &
            * ((aUsed*h*X)/(1.0_dp + bcap) + (X*Y)/bcap)
dYDotdk  = -(aUsed*XDot*XDot)/(A*bcap) - (A/rMag**3.0_dp) &
            * ((aUsed*h*Y)/(1.0_dp + bcap) - (X*X)/bcap)

dvdh     = dXDotdh * fVec + dYDotdh * gVec
dvdk     = dXDotdk * fVec + dYDotdk * gVec

!Compute the 6x6 Jacobi matrix
Jacobian = 0.0_dp

!First row: mean motion or semi-major axis sensitivity
SELECT CASE(elemSwitchUsed)
CASE(0,2)
    !Mean motion
    Jacobian(1,1:3)=((-3.0_dp*nUsed*aUsed) / rMag**3.0_dp) * r
    Jacobian(1,4:6)=-3.0_dp/(nUsed * aUsed**2.0_dp) * v
CASE(1,3)
    !Semi-major axis
    Jacobian(1,1:3)=((2.0_dp * aUsed**2.0_dp) / rMag**3.0_dp) * r
    Jacobian(1,4:6)=2.0_dp/(nUsed**2.0_dp * aUsed) * v
END SELECT

!Second row: sensitivity with respect to h
Jacobian(2,1:3)=-((aUsed*b*h*bcap)/(rMag**3.0_dp)) * r         &
                + (k*(p*XDot - fr*q*YDot)/(A*bcap)) * wVec     &
                - (bcap/A) * dvdk
Jacobian(2,4:6)=((2.0_dp*XDot*Y-X*YDot)*fVec-X*XDot*gVec)/muSi &
                + k*(fr*q*Y - p*X)/(A*bcap) * wVec

!Third row: sensitivity with respect to k
Jacobian(3,1:3)=-((aUsed*b*k*bcap)/(rMag**3.0_dp))  * r        &
                - (h*(p*XDot - fr*q*YDot)/(A*bcap)) * wVec     &
                + (bcap/A) * dvdh
Jacobian(3,4:6)=((2.0_dp*X*YDot-XDot*Y)*gVec-Y*YDot*fVec)/muSi &
                - h*(fr*q*Y - p*X)/(A*bcap) * wVec

!Fourth row: sensitivity with respect to p
Jacobian(4,1:3) = -((C*YDot)/(2*A*bcap)) * wVec
```

```
Jacobian(4,4:6) =  ((C*Y)/(2*A*bcap)) * wVec

!Fifth row: sensitivity with respect to q
!Danielsson misses the fr in Jacobian(5,1:3), but Vallado has it
Jacobian(5,1:3) = -(fr*(C*XDot)/(2*A*bcap)) * wVec
Jacobian(5,4:6) =  (fr*(C*X)/(2*A*bcap)) * wVec

!Sixth row: sensitivity with respect to the fast variable
!Mean longitude
Jacobian(6,1:3) = -v/A + ((p*Xdot - fr*q*YDot)/(A*bcap))*wVec &
                  - ((b*bcap)/A)*(h*dvdh + k*dvdk)
Jacobian(6,4:6) = -(2.0_dp/A)*r + (k*Jacobian(2,4:6)          &
                  - h*Jacobian(3,4:6))/(1.0_dp + bcap)        &
                  + ((fr*q*Y - p*X)/A) * wVec

if (elemSwitchUsed == 2 .OR. elemSwitchUsed == 3) then
    !Adapt mean to true longitude sensitivity
    vMag      = MAG(v)
    muSi      = mu * 1.0E09_dp
    eccVec    = ((vMag*vMag - muSi/rMag)*r - DOT(r, v)*v)/muSi
    e         = MAG(eccVec)
    !oneEcosNu equals 1.0 + h*sin(L) + k*cos(L)
    oneEcosNu = 1.0_dp + DOT(r, eccVec)/rMag
    dMdv      = (1.0_dp - e*e)**1.5_dp / (oneEcosNu*oneEcosNu)

    cosL      = cos(L)
    sinL      = sin(L)
    ch        = aUsed*(2.0_dp*k*h*cosL+2.0_dp*h +          &
                       (h*h-k*k+1.0_dp)*sinL)/             &
                       (C*(1.0_dp + k*cosL + h*sinL)**2.0_dp)
    ck        = aUsed*(2.0_dp*k*h*sinL+2.0_dp*k+           &
                       (k*k-h*h+1.0_dp)*cosL)/             &
                       (C*(1.0_dp + k*cosL + h*sinL)**2.0_dp)

    dhkVec(1) = (p*p-q*q-1.0_dp)*cosL    - 2.0_dp*p*q*fr*sinL
    !dhkVec(2) = (p*p-q*q+1.0_dp)*fr*sinL + 2.0_dp*p*q*cosL
    !dhkVec(3) =  2.0_dp*(p*fr*cosL       - q*sinL)

    !Update the sixth row with dMdv, dMde and the info of
    !rows 2 and 3, which are based on eccentricity
    fVec1 = (1.0_dp - p*p + q*q)/C
    gVec1 = (2.0_dp*fr*p*q)/C
    sensh = (ch*dhkVec(1) - (dXdh*fVec1 + dYdh*gVec1))/    &
            (dMdv*v(1)/nUsed)
    sensk = (ck*dhkVec(1) - (dXdk*fVec1 + dYdk*gVec1))/    &
            (dMdv*v(1)/nUsed)
    Jacobian(6,:) = Jacobian(6,:)/dMdv                     &
                    - (sensh*Jacobian(2,:)+sensk*Jacobian(3,:))
end if

!Assign optional output
if (present(Jac)) Jac = Jacobian
```

```
!Compute the transformed matrix via the similarity transformation
Peoe = Matmul(Jacobian, Matmul(Pcart, transpose(Jacobian)))

!Symmetrize the result
Peoe = (Peoe + transpose(Peoe))/2.0_dp

End Subroutine covRV2Eoe
!=================================================================!
```

B.3 SHORT-TERM ENCOUNTER IDENTIFICATION

Listing B.7 is a complete implementation of Chan's geometric rectilinear motion test. The boundary conditions are taken from the examples presented in Chan (2008, section 3.4).

Listing B.7: isShortTermEncounter

```
!=================================================================!
Subroutine isShortTermEncounter(conjunction, isLeoT, isLeoC, &
                                highConfidence, resultCode)
!-----------------------------------------------------------------!
!! Author: Fabian Schiemenz
!! Date:   03/2020
!-----------------------------------------------------------------!
!! Test if a conjunction is a short-term encounter, which
!! qualifies for the 2D approximation of the 3D Pc integral.
!
!! Exit codes for resultCode:
!! 0: Fulfills both STE-criteria -> safely STE
!! 1: Only one criteria fulfilled, the other in boundary range
!!    -> sufficiently safe STE
!! 2: One criteria fulfilled, the other violated
!!    -> not safely STE
!! 3: Both in boundary range
!!    -> not safely STE
!! 4: Everything else (one in boundary range and the other
!!    violated or both violated) -> not STE
!!
!! The user can decide which exit code to consider as safely STE.
!! The default recommendation is resultCode <= 1 for considering
!! the conjunction to be a short-term encounter.
!!
!! Remarks:
!! - dp is short for REAL64 and i4 for INT32 from iso_fortran_env
!! - MAG computes the magnitude of a vector and
!! - DOT_PRODUCT evaluates the dot product of two vectors
!! - rad2deg equals 180.0_dp/pi
!! - muEarth is Earth's gravitational constant in km^3/s^2
!-----------------------------------------------------------------!
!> Conjunction object to be tested
!> (contains the target and chaser states)
type(tConjunction), intent(inout)   :: conjunction
```

```
!> LEO flag for the target
logical, intent(in)                 :: isLEOT
!> LEO flag for the chaser
logical, intent(in)                 :: isLEOC
!> Optional flag for 17sigma certainty of the result
!> (default is 6 sigma)
logical, intent(in), optional       :: highConfidence
!> The result code which categorizes the conjunction
integer(i4), intent(out)            :: resultCode
!-----------------------------------------------------------------!
!Constants:
real(dp), parameter                 :: alphaGood  = 3.0_dp  !deg
real(dp), parameter                 :: alphaBound = 3.5_dp  !deg
real(dp), parameter                 :: TGood      = 0.02_dp !2%
real(dp), parameter                 :: TBound     = 0.03_dp !3%
!-----------------------------------------------------------------!
!Locals:
logical                             :: use17sigma
real(dp)                            :: rp, vp, rs, vs
real(dp)                            :: phi
real(dp)                            :: Sp, Ss, Lp, Ls, DT, Ds
real(dp)                            :: alphap, alphas, tep, tes
real(dp)                            :: ap, as, tRatp, tRats
!-----------------------------------------------------------------!

!Check optional arguments
if (present(highConfidence)) then
    use17sigma = highConfidence
else
    use17sigma = .FALSE.
end if

!Get the target and the chaser radial distances and speeds at TCA
rp = MAG(conjunction%x_prim(1:3))
vp = MAG(conjunction%x_prim(4:6))
rs = MAG(conjunction%x_sec(1:3))
vs = MAG(conjunction%x_sec(4:6))

!First, compute the approach angle phi
phi = acos(DOT_PRODUCT(conjunction%x_prim(4:6), &
                       conjunction%x_sec(4:6))/(vp*vs))

!Initialize the required rectilinear motion length S, which is
!30km for 6sigma in LEO,  85km for 17sigma in LEO and
!90km for 6sigma in GEO, 255km for 17sigma in GEO.
!By default this routine uses the 6 sigma boundaries.
if (use17sigma) then
    if (isLEOT) then
        Sp =  85.0_dp
    else
        Sp = 255.0_dp
    end if
```

```
    if (isLEOC) then
        Ss =  85.0_dp
    else
        Ss = 255.0_dp
    end if
else
    if (isLEOT) then
        Sp = 30.0_dp
    else
        Sp = 90.0_dp
    end if
    if (isLEOC) then
        Ss = 30.0_dp
    else
        Ss = 90.0_dp
    end if
end if

!Step three is to compute the encounter region length
Lp = Sp/(2.0*sin(phi/2.0_dp))
Ls = Ss/(2.0*sin(phi/2.0_dp))

!Next, compute the radial error D by
!inverting the relationship L^2 = (D+R)^2 - R^2
DT = sqrt(rp*rp+Lp*Lp) - rp
Ds = sqrt(rs*rs+Ls*Ls) - rs

!Evaluate the deflection angle alpha
alphap = atan(DT/Lp)*rad2deg
alphas = atan(Ds/Ls)*rad2deg

!Finally, compute the time to transverse the encounter region
tep = Lp/vp
tes = Ls/vs

!Compute the orbital periods
ap  = -muEarth / ((vp*vp) - 2.0E0_dp*(muEarth/rp))
conjunction%Tp  = twoPi * sqrt(ap**3.0_dp / muEarth)
as  = -muEarth / ((vs*vs) - 2.0E0_dp*(muEarth/rs))
conjunction%Ts  = twoPi * sqrt(as**3.0_dp / muEarth)

!Calculate the ratio of the encounter shell transit time
!with respect to the orbital period
tRatp = tep/conjunction%Tp
tRats = tes/conjunction%Ts

!Set the return value
!Initialize with worst case, update accordingly
resultCode = 4
if (alphap < alphaGood .AND. alphas < alphaGood) then
    if (tRatp < TGood .AND. tRats < TGood) then
        resultCode = 0
```

```
    else if (tRatp < TBound .AND. tRats < TBound) then
        resultCode = 1
    else
        resultCode = 2
    end if
else if (alphap < alphaBound .AND. alphas < alphaBound) then
    if (tRatp < TGood .AND. tRats < TGood) then
        resultCode = 1
    else if (tRatp < TBound .AND. tRats < TBound) then
        resultCode = 3
    end if
else
    if (tRatp < TGood .AND. tRats < TGood) then
        resultCode = 2
    end if
end if

End Subroutine
!=================================================================!
```

B.4 INERTIAL TO MAHALANOBIS SPACE TRANSFORMATION

Nonlinear collision probability algorithms typically perform a transition from inertial space to Mahalanobis space in order to decouple the three-dimensional collision probability integral (equation 6.3). A complete routine to perform the transition to Mahalanobis space has been worked out as part of this thesis. The Fortran sources are given in the following and contain detailed descriptions of the steps required to transform the combined covariance and spherical hard body volumes.

Listing B.8: GCRF2MahalanobisSpace

```
!=================================================================!
Subroutine GCRFtoMahaEncFrame(xT, xC, Pt, Pc, rA, U,            &
                              deltaStateMahaenc, rho_Maha_enc)
!-----------------------------------------------------------------!
!! Author: Fabian Schiemenz
!! Date:   03/2020
!-----------------------------------------------------------------!
!! This subroutine transforms the objects, relative positions,
!! relative velocities and the combined covariance into the fully
!! normalized Mahalanobis encounter frame.
!! After the transformation the covariance equals the identity
!! matrix (it resembles a sphere) and the combined object becomes
!! an ellipsoid.
!! Once the probability density function has equal properties in
!! all directions, the relative velocity direction can easily be
!! decoupled from the other directions.
!!
!! The transformation is a four-step process:
!! 1) Obtain the matrix to perform the covariance diagonalization
```

```
!! 2) Formulate the scaling matrix S, which scales the
!!    diagonalized covariance to the identity matrix
!! 3) Obtain the transformation matrix W from the GCRF frame to
!!    the encounter frame (y-axis in relative velocity direction)
!! 4) Formulate an initial version of the overall transformation
!!    and update
!!
!! Remarks:
!! - dp is short for REAL64 from iso_fortran_env
!! - get_eigenvalues computes the eigenvalues and eigenvectors
!! - transformVector performs the Matrix vector multiplication
!! - deltaCov3Encounter performs the transformation into the
!!   encounter frame (see equations 6.4-6.10)
!! - eye creates an identity matrix
!-----------------------------------------------------------------!
!> Target state in GCRF
real(dp), dimension(6), intent(in)        :: xT
!> Chaser state in GCRF
real(dp), dimension(6), intent(in)        :: xC
!> Target covariance in GCRF
real(dp), dimension(6,6), intent(in)      :: Pt
!> Chaser covariance in GCRF
real(dp), dimension(6,6), intent(in)      :: Pc
!> Circumscribing radius of both objects
real(dp), intent(in)                      :: rA
!> Relative distance in Mahalanobis space
real(dp), dimension(3), intent(out)       :: deltaStateMahaenc
!> Converted hard body principal axes lengths
real(dp), dimension(3), intent(out)       :: rho_Maha_enc
!> Overall transformation matrix from GCRF to the
!> encounter Mahalanobis frame
real(dp), dimension(3,3), intent(out)     :: U
!-----------------------------------------------------------------!
!Locals:
real(dp), dimension(6)                    :: deltaState
real(dp), dimension(6,6)                  :: Pcomb
real(dp), dimension(:), allocatable       :: eigValReal
real(dp), dimension(:,:), allocatable     :: EigVec
real(dp), dimension(3,3)                  :: Penc, HBRmat
real(dp), dimension(3,3)                  :: Q, S, SQ, I, W
real(dp), dimension(3)                    :: sigmaAxes
real(dp), dimension(6)                    :: deltaStateSQ
real(dp), dimension(6)                    :: deltaStateMahaencIni
real(dp), dimension(2,2)                  :: HBellipse
real(dp), dimension(2,3)                  :: U2
real(dp), dimension(3,3)                  :: P_SQ, ROT2
real(dp)                                  :: phi
!-----------------------------------------------------------------!

!Define the delta state in GCRF
deltaState = xT - xC
Pcomb      = PT + PC
```

```
!Get the eigenvectors and eigenvalues of the covariance ellipsoid
allocate(eigvalReal(3), eigValImag(3), EigVec(3,3))
call get_eigenvalues(matrix           = Pcomb(1:3,1:3), &
                     eigenvalues_real = eigvalReal,     &
                     eigenvectors     = EigVec)

!Define the transformation matrix from GCRF to the diagonal frame
Q                 = transpose(EigVec)

!Define the scaling transformation from the inverse of the
!critical values, which equals the square root of the inverse of
!the eigenvalues
sigmaAxes         = sqrt(1.0_dp/eigvalReal(1:3))
S                 = 0.0_dp
S(1,1)            = sigmaAxes(1)
S(2,2)            = sigmaAxes(2)
S(3,3)            = sigmaAxes(3)

!Define the matrix combining both operations
SQ                = Matmul(S, Q)

!Next, transform the state vector of the primary to the
!scaled frame via SQ
call transformVector(deltaState(1:3),  deltaStateSQ(1:3), SQ)
call transformVector(deltaState(4:6),  deltaStateSQ(4:6), SQ)

!The covariance has to become the identity matrix in the
!scaled frame (after this P_SQ must numerically match eye(3))
P_SQ = Matmul(SQ, Matmul(Pcomb(1:3,1:3), transpose(SQ)))

!Next, transform deltaStateSQ and the combined covariance
!P_SQ from the diagonal frame into the encounter frame.
!The covariance remains a unit sphere, as this is only a rotation
call deltaCov3Encounter(deltaStateSQ, P_SQ,         &
                        deltaStateMahaencIni, Penc, W)

!Next, define the matrix U, which describes the combined sequence
!GCRF -> Mahalanobis encounter
U = Matmul(W, SQ)

!Transformation of the Hard body volume
!The original hard body sphere undergoes the same sequence of
!transformations as the covariance ellipsoid.
!A sphere can also be modeled via a matrix in the same way as
!a covariance (one simply pretends that the hard-body sphere
!is also a covariance).
!The transformation sequence is described by U. The last
!operation of U is the transformation to the encounter frame.
!To include the projection onto the xz-encounter plane, it is
!necessary to cut out the row belonging to the relative motion
!direction (second row for y).
```

```
!Hence, it is possible to define a matrix U2 to transform the
!sphere via the appropriate rotations, stretching and the
!projection on the encounter plane.

!Define the hard body in the same matrix-way as covariances
!(mind to square rA, as (co-)variances are squared quantities)
call eye(3, I)
HBRmat = rA * rA * I

!Define the matrix U2, which includes the projection on the
!xz-plane of the encounter frame
U2(1,:)   = U(1,:)
U2(2,:)   = U(3,:)

!Perform the transformation
HBellipse = Matmul(U2, Matmul(HBRmat, transpose(U2)))

!To compute the principal axes lengths of the ellipse in the
!encounter plane, the eigenvalues of HBellipse are required
deallocate(eigvalReal, eigValImag, EigVec)
allocate(eigvalReal(2), eigValImag(2), EigVec(2,2))
call get_eigenvalues(matrix           = HBellipse,  &
                     eigenvalues_real = eigvalReal, &
                     eigenvectors     = EigVec)

!The lengths of the main axes are given by the roots of the
!eigenvalues. The second component of the output vector,
!rho_Maha_enc(2), is a dummy component and set to zero to
!maintain the xz nature of the projection.
rho_Maha_enc(1) = sqrt(eigvalReal(1))
rho_Maha_enc(2) = 0.0_dp
rho_Maha_enc(3) = sqrt(eigvalReal(2))

!The axes lengths obtained from the critical values are with
!respect to the orientation given in the eigenvectors. To ensure
!that this orientation is consistent with the encounter frame,
!the angular difference of the major eigenvector and the
!encounter-plane x-axis needs to be computed.
!This angle is then used to rotate the relative state, thereby
!making sure that the relative state and the orientation of the
!encounter plane projection of the hard-body ellipsoid are in the
!same frame.
phi       = atan(EigVec(2,1)/EigVec(1,1))

!Formulate ROT2 (rotation about y) using phi
ROT2(1,:) = [cos(phi), 0.0_dp, -sin(phi)]
ROT2(2,:) = [0.0_dp,   1.0_dp,    0.0_dp]
ROT2(3,:) = [sin(phi), 0.0_dp,  cos(phi)]

!Transform the encounter frame vector to the proper alignment
call transformVector(deltaStateMahaencIni(1:3), &
                     deltaStateMahaenc, ROT2)
```

```
!Also account for the final rotation in the overall
!transformation matrix U.
!The matrix U is required to express the propagated GCRF state
!difference in an earlier Mahalanobis frame (e.g. GCRF tube end
!position difference in tube start Mahalanobis frame in case of
!the adjoining tube method)
U = Matmul(ROT2, U)

End Subroutine
!=============================================================!
```

B.5 GENERAL PERTURBATIONS MONTE-CARLO COLLISION PROBABILITY

The Monte-Carlo technique is a popular method for validating collision probability calculations. It is easy to implement and essentially requires only three steps to obtain accurate collision probabilities:

1. Sampling
2. Propagation
3. Collision detection

The sampling step requires the six-dimensional epoch covariances to perturb the target and chaser epoch states according to their pdfs. In the second step the perturbed epoch state vectors are propagated in a timeframe corresponding to the expected extent of the encounter region. The final step then checks if at any time the target and chaser objects come closer to each other than their combined hard body radius. In this case a collision is logged. The final collision probability is the ratio of collisions over the absolute number of trials.

The realization of these steps needs to be carefully implemented. First of all, statistically sound sampling bounds are required. The Monte-Carlo method relies on the law of large numbers. Given sufficient trials, the collision probability is guaranteed to converge against the true collision probability. Some early (and unfortunately also recent) works did not consider the minimum required number of iterations for statistically meaningful results and simply used e.g. 10^3 (Chao and Park, 2002) or 10^6 (Gondelach and Linares, 2020) trials, whereas the actually required number depends on the true collision probability. The larger it is, the fewer samples are needed. For most practically relevant conjunctions the required trials range between 10^8 and $> 10^{20}$. For this reason, the MC method is only used to validate other collision probability algorithms, given specially designed HIE conjunctions that can be verified using 10^6-10^8 trials. An excellent discussion on the subject may be found in Alfano (2009).

The final collision detection step requires finding all minimum distances between the target and chaser in order to check if any of the minima is smaller than the combined HBR. There are many algorithms

for this purpose. A prominent approach is the Alfano Negron Close Approach Software (ANCAS) method (Alfano and Negron, 1993; Alfano, 1994), which is based on localized cubic polynomial root finding, however also the fine conjunction detection search of the smart sieve algorithm can be used. Recently, also promising results have been reported using Chebyshev proxy polynomial root finding (Denenberg, 2020).

So far, Monte-Carlo P_c calculations were limited to the SP perturbation theory. The underlying issue is the incompatibility of the sampling step with the Kozai/Brouwer TLE mean orbital elements definition. There is simply no statistically consistent and formally correct technique to sample the TLE orbital elements according to GCRF epoch covariances (see also the discussion in section 4.4). With the help of the numerical GP covariance propagation introduced and validated in section 4.4, it is however possible to formulate a workaround that bridges the gap between Monte-Carlo collision probabilities and the GP theory, as is demonstrated in the following.

Prerequisites (t_f = propagation end time, t_0 = target/chaser epoch times):

- Target and chaser TLE epoch sets: $\mathbf{x}_{t_0,T,m}$ and $\mathbf{x}_{t_0,C,m}$
- GP propagator (equation 4.62)
- Algorithm 4.1: $\mathbf{x}_{t_0,m}, t_f \rightarrow \mathbf{\Phi}(t_f, t_0)$
- Target and chaser GCRF covariance matrices (e.g. derived in a manner similar to Flohrer et al., 2008)

At each time-step it is possible to express the perturbed state (denoted by a superscript asterisk) as the nominal state and a GCRF perturbation vector:

$$\mathbf{x}^*_{t_i,T/C,GCRF} = \mathbf{x}_{t_i,T/C,GCRF} + \delta\mathbf{x}_{t_i,T/C,GCRF} \tag{B.1}$$

The GCRF epoch state vectors can be obtained from the TLE sets using $t_i = t_0$ in equation 4.62:

$$\mathbf{x}_{t_0,T/C,GCRF} = GP(t_0, t_0, \mathbf{x}_{t_0,T/C,m}) \tag{B.2}$$

The epoch state GCRF perturbations are derived from the epoch covariance matrices (or pdfs), as described in Chao and Park (2002), which allows to evaluate equation B.1 independently for the target and chaser at their respective epoch times:

$$\begin{aligned} \mathbf{x}^*_{t_0,T/C,GCRF} &= \mathbf{x}_{t_0,T/C,GCRF} + \delta\mathbf{x}_{t_0,T/C,GCRF} \\ \Leftrightarrow \delta\mathbf{x}_{t_0,T/C,GCRF} &= \mathbf{x}^*_{t_0,T/C,GCRF} - \mathbf{x}_{t_0,T/C,GCRF} \end{aligned} \tag{B.3}$$

It is not possible to rephrase $\mathbf{x}^*_{0,T,GCRF}$ as a perturbed TLE, however the STM obtained from algorithm 4.1 allows to linearly propagate the GCRF epoch state perturbation independently of the nominal state:

$$\delta\mathbf{x}_{t_f,T/C,GCRF} = \mathbf{\Phi}(t_f, T/C, t_0)\ \delta\mathbf{x}_{t_0,T/C,GCRF} \tag{B.4}$$

The perturbed and propagated state vector is then found as:

$$\mathbf{x}^{*}_{t_f,T/C,GCRF} \cong GP(t_0, t_f, \mathbf{x}_{t_0,T/C,m}) + \mathbf{\Phi}(t_f, T/C, t_0)\ \delta\mathbf{x}_{t_0,T/C,GCRF} \tag{B.5}$$

Equation B.5 needs to be evaluated for the target and chaser RSOs at all times of interest within the encounter region. It only requires analytical GP propagations for its evaluation and is therefore fully compliant with the GP theory. The only weak spot is the usage of linear error propagation in order to obtain the perturbed state at the final propagation time. The algorithm is therefore subject to the same restrictions as linear covariance propagation, such that its application needs to be limited to scenarios for which the linearity assumption is justified.

As a proof of concept, the MC collision probability computation results were compared against the Alfano and McKinley adjoining tube algorithms (cf. section 6.1.2) using the following encounter:

```
Target: INTELSAT 901 (IS-901)
1 26824U 01024A   20055.87828646 -.00000304 +00000-0 +00000-0 0  9998
2 26824 001.6396 091.3706 0002107 345.7158 167.9664 00.99201678068575

Chaser: MEV-1
1 44625U 19067B   20055.77389073 -.00000305  00000-0  00000+0 0  9998
2 44625   1.6363  91.3103 0002012 351.6056 124.8523  0.99201881  1262

Time of closest approach (JD): 2458905.55735990
Encounter region time limit: TCA ± 0.25 days
Epoch RTN covariance (km² and km²/s²): diag(10⁻⁶,10⁻⁶,10⁻⁶,10⁻¹⁰,4·10⁻¹⁰,10⁻¹²)
```

To artificially increase the collision probability and reduce the number of required Monte-Carlo iterations, the hard body radii of both objects have been increased to 50 m each, resulting in a combined HBR of 100 m. Table B.2 depicts the collision probability results. Dagum bounds with 5% error and 95% confidence were used to compute the number of required trials (3 998 185).

GP MC	Adj. tube (Alfano)	Adj. tube (no CovSym)	Adj. tube (McKinley)
$1.1210 \cdot 10^{-3}$	$1.0592 \cdot 10^{-3}$	$1.0592 \cdot 10^{-3}$	$1.0603 \cdot 10^{-3}$

Table B.2: Evaluation of GP Monte-Carlo collision probability

Further testing of the method is recommended, but the possibility of GP collision probability validation by means of MC calculations is a very attractive new feature. The analytic nature of the GP theory makes its evaluation significantly faster than typical for SP evaluations.

B.6 RELATIVISTIC ABERRATION CORRECTION

Aberration is an effect that belongs to the class of the apparent displacements of celestial objects. The apparent direction in the sky at which a celestial object appears is not the actual direction from which the light was emitted. Several corrections are required to determine the true origin from the observations.

The correction for aberration is one of several steps that must be applied to optical measurements before they can be used for orbit determination. As Airbus had just installed a robotic telescope in Extremadura, Spain, when this work was started, an auxiliary development of this thesis was the derivation of a universal relativistic aberration correction algorithm.

Classical textbook references like Seidelmann (2006, section 3.25) only consider the forward direction of the problem, i.e. the computation of the apparent direction of a light source, given its true direction, however for orbit determination the inverse problem, i.e. finding the true direction when the observations are given, is of concern. The relativistic treatment of both problems (addition and removal of aberration) is considered in the following. Further information on the topic, such as the different types of aberration (stellar, annual, orbital, diurnal) can be found in textbooks like Barbieri (2006, chapter 7) or Seidelmann (2006, section 3.25).

The derivation starts from the generalized vectorial form of the Lorentz transformation, which is derived in Fock (1964, section 1.§10) and represents a set of equations for transforming the space and time coordinates of one inertial frame into those of another that moves with constant velocity relative to the first (moving frame quantities are indicated by a superscript prime):

$$\mathbf{r}' = \mathbf{r} - \gamma \boldsymbol{\nu} t + (\gamma - 1)\boldsymbol{\nu}(\boldsymbol{\nu} \cdot \mathbf{r})/\nu^2 \tag{B.6}$$

$$t' = \gamma\left[t - (\boldsymbol{\nu} \cdot \mathbf{r})/c^2\right] \tag{B.7}$$

where $\gamma = \frac{1}{\sqrt{1-(\nu/c)^2}} = \beta^{-1}$ and c denotes the speed of light. The inverse relationship describes the transformation of the space and time coordinates of the inertial frame with respect to the moving frame (Fock, 1964, section 1.§10):

$$\mathbf{r} = \mathbf{r}' + \gamma \boldsymbol{\nu} t' + (\gamma - 1)\boldsymbol{\nu}(\boldsymbol{\nu} \cdot \mathbf{r}')/\nu^2 \tag{B.8}$$

$$t = \gamma\left[t' + (\boldsymbol{\nu} \cdot \mathbf{r}')/c^2\right] \tag{B.9}$$

Equations B.6 and B.8, as well as B.7 and B.9 only differ by the sign of $\boldsymbol{\nu}$. This set of equations describes the generalized vectorial Lorentz transformation for space and time. In the present form, equations B.6-B.9 cannot be used to correct for the effect of aberration due to

their inherent time dependence. To this end, the derivative of $\mathbf{r}'$ with respect to t' needs to be computed from equations B.6 and B.7:

$$\frac{d\mathbf{r}'}{dt'} = \frac{d\mathbf{r}}{dt}\frac{dt}{dt'} - \gamma\mathbf{v}\frac{dt}{dt'} + (\gamma-1)\mathbf{v}\left(\mathbf{v}\cdot\frac{d\mathbf{r}}{dt}\frac{dt}{dt'}\right)/v^2 \tag{B.10}$$

To further simplify equation B.10, a description of dt/dt' is required. Differentiating equation B.7 with respect to dt yields:

$$\frac{dt'}{dt} = \gamma\left[\frac{dt}{dt} - \left(\mathbf{v}\cdot\frac{d\mathbf{r}}{dt}\right)/c^2\right] = \gamma\left[1 - \left(\mathbf{v}\cdot\frac{d\mathbf{r}}{dt}\right)/c^2\right] \tag{B.11}$$

Letting $\mathbf{u} = d\mathbf{r}/dt$, $\mathbf{u}' = d\mathbf{r}'/dt'$ and inserting equation B.11 into eq. B.10 yields:

$$\begin{aligned}\mathbf{u}' &= \mathbf{u}\frac{1}{\gamma[1-(\mathbf{v}\cdot\mathbf{u})/c^2]} - \gamma\mathbf{v}\frac{1}{\gamma[1-(\mathbf{v}\cdot\mathbf{u})/c^2]} \\ &\quad + (\gamma-1)\mathbf{v}\left(\mathbf{v}\cdot\mathbf{u}\frac{1}{\gamma[1-(\mathbf{v}\cdot\mathbf{u})/c^2]}\right)/v^2 \\ \Leftrightarrow \mathbf{u}' &= \frac{\mathbf{u} - \gamma\mathbf{v} + (\gamma-1)\mathbf{v}(\mathbf{v}\cdot\mathbf{u})/v^2}{\gamma[1-(\mathbf{v}\cdot\mathbf{u})/c^2]}\end{aligned} \tag{B.12}$$

The inverse transformation starts by computing the derivative of equation B.8 with respect to dt:

$$\frac{d\mathbf{r}}{dt} = \frac{d\mathbf{r}'}{dt'}\frac{dt'}{dt} + \gamma\mathbf{v}\frac{dt'}{dt} + (\gamma-1)\mathbf{v}\left(\mathbf{v}\cdot\frac{d\mathbf{r}'}{dt'}\frac{dt'}{dt}\right)/v^2 \tag{B.13}$$

The derivative of eq. B.9 with respect to t' results in:

$$\frac{dt}{dt'} = \gamma\left[\frac{dt'}{dt'} + \left(\mathbf{v}\cdot\frac{d\mathbf{r}'}{dt'}\right)/c^2\right] = \gamma\left[1 + (\mathbf{v}\cdot\mathbf{u}')/c^2\right] \tag{B.14}$$

which upon insertion into equation B.13 leads to:

$$\begin{aligned}\mathbf{u} &= \mathbf{u}'\frac{1}{\gamma[1+(\mathbf{v}\cdot\mathbf{u}')/c^2]} + \gamma\mathbf{v}\frac{1}{\gamma[1+(\mathbf{v}\cdot\mathbf{u}')/c^2]} \\ &\quad + (\gamma-1)\mathbf{v}\left(\mathbf{v}\cdot\mathbf{u}'\frac{1}{\gamma[1+(\mathbf{v}\cdot\mathbf{u}')/c^2]}\right)/v^2 \\ \Leftrightarrow \mathbf{u} &= \frac{\mathbf{u}' + \gamma\mathbf{v} + (\gamma-1)\mathbf{v}(\mathbf{v}\cdot\mathbf{u}')/v^2}{\gamma[1+(\mathbf{v}\cdot\mathbf{u}')/c^2]}\end{aligned} \tag{B.15}$$

Equations B.12 and B.15 are independent of time and only differ by the sign of the relative velocity $\mathbf{v}$. Formula B.12 expresses the velocity in the dashed (moving) system, given the relative observer velocity vector $\mathbf{v}$ and the light velocity vector $\mathbf{u}$. This case corresponds to the addition of aberration, whereas equation B.15 represents the inverse relationship and is required to remove the effect of aberration from an apparent target position.

Denoting the true direction vector from the observer to the light source as $\mathbf{p}$ and the direction vector from the observer to the apparent

position as $\mathbf{p}'$, the velocity vectors of the *incoming* light correspond to $\mathbf{u} = -c\mathbf{p}$, as well as $\mathbf{u}' = -c\mathbf{p}'$. The vector $\boldsymbol{v}$ corresponds to the inertial observer velocity. Using these relations, the addition of aberration is obtained from equation B.12 as:

$$\begin{aligned} -c\mathbf{p}' &= \frac{-c\mathbf{p} - \gamma\boldsymbol{v} - (\gamma-1)\boldsymbol{v}(\boldsymbol{v}\cdot c\mathbf{p})/v^2}{\gamma[1 + (\boldsymbol{v}\cdot c\mathbf{p})/c^2]} \\ \Leftrightarrow c\mathbf{p}' &= \frac{c\mathbf{p} + \gamma\boldsymbol{v} + (\gamma-1)\boldsymbol{v}(\boldsymbol{v}\cdot c\mathbf{p})/v^2}{\gamma[1 + (\boldsymbol{v}\cdot c\mathbf{p})/c^2]} \end{aligned} \tag{B.16}$$

Removing aberration requires obtaining $\mathbf{u}$ from $\mathbf{u}'$. Using equation B.15 with $\mathbf{u}' = -c\mathbf{p}'$ yields:

$$\begin{aligned} -c\mathbf{p} &= \frac{-c\mathbf{p}' + \gamma\boldsymbol{v} - (\gamma-1)\boldsymbol{v}(\boldsymbol{v}\cdot c\mathbf{p}')/v^2}{\gamma[1 - (\boldsymbol{v}\cdot c\mathbf{p}')/c^2]} \\ c\mathbf{p} &= \frac{c\mathbf{p}' - \gamma\boldsymbol{v} + (\gamma-1)\boldsymbol{v}(\boldsymbol{v}\cdot c\mathbf{p}')/v^2}{\gamma[1 - (\boldsymbol{v}\cdot c\mathbf{p}')/c^2]} \end{aligned} \tag{B.17}$$

Both cases may be combined into a single equation. To this end, the notation is changed slightly by indicating the direction vector prior correction as $\mathbf{p}$ and the corrected (with the addition or removal of aberration) direction as $\mathbf{p}_c$:

$$c\mathbf{p}_c = \frac{c\mathbf{p} \pm \gamma\boldsymbol{v} + (\gamma-1)\boldsymbol{v}(\boldsymbol{v}\cdot c\mathbf{p})/v^2}{\gamma[1 \pm (\boldsymbol{v}\cdot\mathbf{p})/c]} \tag{B.18}$$

Dividing by c results in:

$$\mathbf{p}_c = \frac{\frac{1}{\gamma}\mathbf{p} \pm \boldsymbol{v}/c + \left(\frac{\gamma-1}{\gamma}\right)\boldsymbol{v}(\boldsymbol{v}\cdot\mathbf{p})/v^2}{1 \pm (\boldsymbol{v}\cdot\mathbf{p})/c} \tag{B.19}$$

Since $\gamma = \beta^{-1}$ and thus $(\gamma-1)/\gamma = 1-\beta$, equation B.19 can be written as:

$$\mathbf{p}_c = \frac{\beta\mathbf{p} \pm \boldsymbol{v}/c + (1-\beta)\frac{c^2}{v^2}(\boldsymbol{v}/c)(\boldsymbol{v}/c)\cdot\mathbf{p}}{1 \pm (\boldsymbol{v}\cdot\mathbf{p})/c} \tag{B.20}$$

Since $\beta = \sqrt{1-(v/c)^2} \Leftrightarrow 1-\beta^2 = (v/c)^2 \Leftrightarrow (v/c)^2 = (1+\beta)(1-\beta)$ $\Leftrightarrow (c/v)^2 = 1/((1+\beta)(1-\beta))$, it follows that:

$$\mathbf{p}_c = \frac{\beta\mathbf{p} \pm \boldsymbol{v}/c + (\mathbf{p}\cdot\boldsymbol{v}/c)(\boldsymbol{v}/c)/(1+\beta)}{1 \pm \mathbf{p}\cdot\boldsymbol{v}/c} \tag{B.21}$$

Equation B.21 is quoted in many textbooks, e.g. Barbieri (2006, eq. 7.5) or Seidelmann (2006, eq. 3.252-3), however always featuring only the plus sign. The general derivation starting from the Lorentz transformation demonstrates that the inverse problem, i.e. the removal of aberration from optical measurements, is achieved by using the minus sign in two places of the equation. A Fortran implementation of the correction, which is able to add or remove aberration, is given in algorithm B.9.

Listing B.9: relativisticAberrationCorrection

```
!=============================================================!
Subroutine aberration(vrel, p, add)
!! Perform aberration correction
!
!! To add aberration set add=.TRUE., else it is removed.
!-------------------------------------------------------------!
!! Author: Fabian Schiemenz
!! Date:   02/2018
!-------------------------------------------------------------!
! In/Out:
!> Position unit vector in GCRF inertial frame
!> The corrected vector will be returned
real(dp), dimension(3), Intent(inout)   :: p
! Input only:
!> To add aberration, set add=.TRUE.
!> In all other cases it will be removed
logical, Intent(in), optional           :: add
!> Velocity of the observer in the inertial frame
real(dp), dimension(3), Intent(in)      :: vrel
!-------------------------------------------------------------!
!Locals:
real(dp)                                :: beta, pv
real(dp), dimension(3)                  :: pc
logical                                 :: addMode
!-------------------------------------------------------------!

!Fortran does not support short-circuit boolean evaluation, hence
!check if add is present AND true
addMode = present(add)
if (addMode) addMode = add

! Define beta
beta  = sqrt(1-(MAG(vrel)/c)**2)
pv    = DOT(p, vrel)

if (addMode) then
    !Add aberration
    pc = (beta*p + vrel/c+((pv/c)*vrel/c)/(1+beta))/(1 + pv/c)
else
    !Remove aberration
    pc = (beta*p - vrel/c+((pv/c)*vrel/c)/(1+beta))/(1 - pv/c)
end if

!Update the input direction vector
p = pc

End Subroutine
!=============================================================!
```

BIBLIOGRAPHY

Åkesson, Fredrik and Lehoczky, John P. (1998). *Discrete Eigenfunction Expansion of Multi-Dimensional Brownian Motion and the Ornstein-Uhlenbeck Process*. Tech. rep. Carnegie-Mellon University.

Alarcon Rodriguez, J. R., Martinez Fadrique, F., and Klinkrad, H. (2002). "Collision risk assessment with a "smart sieve" method." In: *Joint ESA-NASA Space-Flight Safety Conference*. ESA SP-486. Noordwijk, the Netherlands: ESA publications devision, pp. 159–164.

Alfano, Salvatore (Apr. 1994). "Determining Satellite Close Approaches, Part II." In: *The Journal of the Astronautical Sciences* 42, pp. 143–152.

Alfano, Salvatore (Oct. 2002). "Aerospace Support to Space Situation Awareness." In: *MIT Lincoln Laboratory Satellite Operations and Safety Workshop*. Haystack Observatory, Chelmsford, MA.

Alfano, Salvatore (2003). "Relating position uncertainty to maximum conjunction probability." In: *Advances in the Astronautical Sciences: proceedings of the 2003 AIAA/AAS Astrodynamics Specialists Conference*. Vol. 116. 1. Paper AAS 03-548, pp. 757–766.

Alfano, Salvatore (2005). "A numerical implementation of spherical object collision probability." In: *Journal of Astronautical Sciences* 53.1, pp. 103–109.

Alfano, Salvatore (2006). "Addressing nonlinear relative motion for spacecraft collision probability." In: *AIAA/AAS astrodynamics specialist conference*. Paper AIAA 2006-6760. Keystone, CO.

Alfano, Salvatore (2007a). "Beta Conjunction Analysis Tool." In: *Advances in the Astronautical Sciences: proceedings of the 2007 AAS/AIAA Astrodynamics Specialist Conference*. Vol. 129. Paper AAS 07-393. San Diego, CA: Univelt, Inc., pp. 2245–2256.

Alfano, Salvatore (2007b). "Review of conjunction probability methods for short-term encounters." In: *Advances in the Astronautical Sciences: proceedings of the 17th AAS/AIAA Space Flight Mechanics Meetings*. Vol. 127. Paper AAS 07-148. San Diego, CA: Univelt, Inc.

Alfano, Salvatore (2009). "Satellite conjunction monte carlo analysis." In: *Advances in the Astronautical Sciences: proceedings of the 19th AAS/AIAA Spaceflight Mechanics Meeting*. Vol. 134. Paper AAS 09-233. San Diego, CA: Univelt, Inc.

Alfano, Salvatore (2012a). "Eliminating assumptions regarding satellite conjunction analysis." In: *The Journal of the Astronautical Sciences* 59.4, pp. 676–705.

Alfano, Salvatore (2012b). "Toroidal path filter for orbital conjunction screening." In: *Celestial Mechanics and Dynamical Astronomy* 113.3, pp. 321–334.

Alfano, Salvatore (2013). "Determining a probability-based distance threshold for conjunction screening." In: *Journal of Spacecraft and Rockets* 50.3, pp. 686–690.

Alfano, Salvatore and Negron, David Jr. (1993). "Determining Satellite Close Approaches." In: *The Journal of the Astronautical Sciences* 41.2, pp. 217–225.

Alfano, Salvatore and Finkleman, David (2014). "On selecting satellite conjunction filter parameters." In: *Acta Astronautica* 99, pp. 193–200.

Alfano, Salvatore and Oltrogge, Daniel (2018). "Probability of Collision: Valuation, variability, visualization, and validity." In: *Acta Astronautica* 148, pp. 301–316.

Alfriend, Kyle T., Akella, Maruthi R., Frisbee, Joseph, Foster, James L., Lee, Deok-Jin, and Wilkins, Matthew (1999). "Probability of collision error analysis." In: *Space Debris* 1.1, pp. 21–35.

Alfriend, Kyle T. and Park, Inkwan (2016). "When does the uncertainty become non-Gaussian?" In: *17th Advanced Maui Optical and Space Surveillance Technologies Conference*. Maui, HI.

Alspach, Daniel L. and Sorenson, Harold W. (1972). "Nonlinear Bayesian estimation using Gaussian sum approximations." In: *IEEE transactions on automatic control* 17.4, pp. 439–448.

Anderson, Rodney L., Born, George H., and Forbes, Jeffrey M. (2009). "Sensitivity of orbit predictions to density variability." In: *Journal of Spacecraft and Rockets* 46.6, pp. 1214–1230.

Anderson, Rodney L., Guignet, Christian P., Born, George H., and Forbes, Jeffrey M. (2013). "Effect of Density Model Time-Delay Errors on Orbit Prediction." In: *Journal of Spacecraft and Rockets* 50.5, pp. 1096–1105.

Anderson, Theodore W. and Darling, Donald A. (1954). "A test of goodness of fit." In: *Journal of the American statistical association* 49.268, pp. 765–769.

Anz-Meador, Phillip D., Opiela, John N., Shoots, Debra, and Liou, Jer Chyi (2018). *History of on-orbit satellite fragmentations 15th Edition*. Tech. rep. Lyndon B. Johnson Space Center, Houston, Texas: NASA Orbital Debris Program Office.

Apruzese, John P., Strobel, Darrell F., and Schoeberl, Mark R. (1984). "Parameterization of IR cooling in a Middle Atmosphere Dynamics Model: 2. Non-LTE radiative transfer and the globally averaged temperature of the mesosphere and lower thermosphere." In: *Journal of Geophysical Research: Atmospheres* 89.D3, pp. 4917–4926.

Aristoff, Jeffrey M., Horwood, Joshua T., Singh, Navraj, and Poore, Aubrey B. (2014). "Nonlinear uncertainty propagation in orbital elements and transformation to cartesian space without loss of realism." In: *2014 AIAA/AAS Astrodynamics Specialist Conference*. Paper AIAA 2014-4167. San Diego, CA.

Balch, Michael Scott, Martin, Ryan, and Ferson, Scott (2019). "Satellite conjunction analysis and the false confidence theorem." In: *Proceedings of the Royal Society A* 475.2227:20180565.

Barbieri, Cesare (2006). *Fundamentals of astronomy*. Taylor & Francis. ISBN: 978-0-750-30886-1.

Barnett, Vic (1999). *Comparative statistical inference*. Vol. 3. John Wiley & Sons. ISBN: 978-0-471-97643-1.

Bartels, J. (1949). *The standardized index, Ks, and the planetary index, Kp*. IATME Bull. 12b. IUGG Publ Office. Paris, pp. 97–112.

Bartels, J., Heck, N. H., and Johnston, H. F. (1939). "The three-hour-range index measuring geomagnetic activity." In: *Terrestrial Magnetism and Atmospheric Electricity* 44.4, pp. 411–454.

Bartels, J. and Veldkamp, J. (1953). "International data on magnetic disturbances, second quarter, 1953." In: *Journal of Geophysical Research* 58.4, pp. 543–545.

Bates, David Robert (1959). "Some problems concerning the terrestrial atmosphere above about the 100 km level." In: *Proceedings of the Royal Society A* 253.1275, pp. 451–462.

Bowman, Bruce, Tobiska, W. Kent, Marcos, Frank A., Huang, Cheryl, Lin, Chin, and Burke, William (2008). "A new empirical thermospheric density model JB2008 using new solar and geomagnetic indices." In: *2008 AIAA/AAS Astrodynamics Specialists Conference*. Paper AIAA 2008-6438. Honolulu, HI.

Brouwer, Dirk (1959). "Solution of the Problem of Artificial Satellite Theory Without Drag." In: *Astronomical Journal* 64.11, pp. 378–397.

Bruinsma, Sean L. (2013). "The Semi-Empirical Thermosphere Model DTM2012." In: *6th European Conference on Space Debris*. Darmstadt, Germany.

Bruinsma, Sean L. (2015). "The DTM-2013 thermosphere model." In: *Journal of Space Weather and Space Climate* 5, A1.

Bruinsma, Sean L. and Forbes, Jeffrey M. (2008). "Medium-to large-scale density variability as observed by CHAMP." In: *Space Weather* 6.8.

Bussy-Virat, Charles D., Ridley, Aaron J., and Getchius, Joel W. (2018). "Effects of uncertainties in the atmospheric density on the probability of collision between space objects." In: *Space Weather* 16.5, pp. 519–537.

Cai, Han, Hussein, Islam, and Jah, Moriba K. (2020). "Possibilistic admissible region using outer probability measure theory." In: *Acta Astronautica* 177, pp. 246–257.

Casali, S., Hall, D., Snow, D., Hejduk, M., Johnson, L., Skrehart, B., and Baars, L. (2019). "Effect of Cross-Correlation of Orbital Error on Probability of Collision Determination." In: *Advances in the Astronautical Sciences: proceedings of the 2018 AAS/AIAA Astrodynamics Specialist Conference*. Vol. 167. Paper AAS 18-272. San Diego, CA: Univelt, Inc., pp. 2083–2096.

Cerven, William Todd (2011). "Covariance error assessment, correction, and impact on probability of collision." In: *21st AAS/AIAA Space Flight Mechanics Meeting*. Paper AAS 11-154. New Orleans, LA.

Cerven, William Todd (2013). "Improved empirical covariance estimation." In: *2013 AIAA/AAS Astrodynamics Specialists Conference*. Paper AAS 13-768. Hilton Head Island, SC.

Chamberlain, Thomas P. and Hunten, Donald M. (1990). *Theory of planetary atmospheres: an introduction to their physics and chemistry*. Vol. 36. Academic Press. ISBN: 978-0-080-96313-6.

Chan, Kenneth F. (1997). "Collision Probability Analyses for Earth Orbiting Satellites." In: *Advances in the Astronautical Sciences: proceedings of the 7th AAS/JRS/CSA Symposium, International Space Conference of Pacific-Basin Societies*. Vol. 96. Paper AAS 97-491. San Diego, CA: Univelt, Inc., pp. 1033–1048.

Chan, Kenneth F. (2003). "Improved Analytical Expressions for Computing Spacecraft Collision Probabilities." In: *Advances in the Astronautical Sciences: proceedings of the 13th AAS/AIAA Spaceflight Mechanics Meeting*. Vol. 114. Paper AAS 03-184. San Diego, CA: Univelt, Inc., pp. 1197–1216.

Chan, Kenneth F. (2008). *Spacecraft collision probability*. El Segundo, CA: Aerospace Press. ISBN: 978-1-884-98918-6.

Chao, Chia-Chun and Park, Timothy (2002). "A Benchmark Monte Carlo Numerical Verification of Computed Probability of Collision." In: *2002 AIAA/AAS Astrodynamics Specialist Conference*. Paper AIAA 2002-4633. Monterey, CA.

Cook, Dayne G. (2001). "Solar Radiation Pressure Modelling Issues For High Altitude Satellites." Master's thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology.

Coppola, Vincent T. (2012). "Including velocity uncertainty in the probability of collision between space objects." In: *22nd AAS/AIAA Spaceflight Mechanics Meeting*. Paper AAS 12-247. Charleston, SC.

Coppola, Vincent T., Woodburn, James, and Hujsak, Richard (2004). "Effects of cross correlated covariance on spacecraft collision probability." In: *14th AAS/AIAA Space Flight Mechanics Meeting*. Paper AAS 04-181. Maui, HI.

Darling, Donald A. (1957). "The Kolmogorov-Smirnov, Cramér-von Mises tests." In: *The Annals of Mathematical Statistics* 28.4, pp. 823–838.

Delande, Emmanuel, Houssineau, Jérémie, and Jah, Moriba K. (2018a). "A New Representation of Uncertainty for Data Fusion in SSA Detection and Tracking Problems." In: *21st International Conference on Information Fusion (FUSION)*. IEEE, pp. 1309–1316.

Delande, Emmanuel, Houssineau, Jérémie, and Jah, Moriba K. (2018b). "Physics and human-based information fusion for improved resident space object tracking." In: *Advances in Space Research* 62.7, pp. 1800–1812.

Delande, Emmanuel, Jah, Moriba K., and Jones, Brandon (2019). "A New Representation of Uncertainty for Collision Assessment." In: *Advances in the Astronautical Sciences: proceedings of the 29th AAS/AIAA Spaceflight Mechanics Meeting*. Vol. 168. Paper AAS 19-452. San Diego, CA: Univelt, Inc., pp. 3021–3040.

Delaye, Eric (1986). "Analysis of the effect of gravity model errors on the state error covariance." Master's thesis. University of Texas at Austin.

DeMars, Kyle J., Bishop, Robert H., and Jah, Moriba K. (2013). "Entropy-based approach for uncertainty propagation of nonlinear dynamical systems." In: *Journal of Guidance, Control, and Dynamics* 36.4, pp. 1047–1057.

DeMars, Kyle J. and Jah, Moriba K. (2013). "Probabilistic initial orbit determination using gaussian mixture models." In: *Journal of Guidance, Control, and Dynamics* 36.5, pp. 1324–1335.

Denenberg, Elad (2020). "Satellite closest approach calculation through Chebyshev Proxy Polynomials." In: *Acta Astronautica* 170, pp. 55–65.

Doornbos, Eelco, Klinkrad, Heiner, and Visser, Pieter (2008). "Use of two-line element data for thermosphere neutral density model calibration." In: *Advances in Space Research* 41.7, pp. 1115–1122.

Drob, Douglas P., Emmert, John T., Meriwether, John W., Makela, Jonathan J., Doornbos, Eelco, Conde, Mark, Hernandez, Gonzalo, Noto, John, Zawdie, Katherine A., McDonald, Sarah E., et al. (2015). "An update to the Horizontal Wind Model (HWM): The quiet time thermosphere." In: *Earth and Space Science* 2.7, pp. 301–319.

Dudok de Wit, Thierry and Bruinsma, Sean L. (2011). "Determination of the most pertinent EUV proxy for use in thermosphere modeling." In: *Geophysical Research Letters* 38.19.

Dudok de Wit, Thierry and Bruinsma, Sean L. (2017). "The 30 cm radio flux as a solar proxy for thermosphere density modelling." In: *Journal of Space Weather and Space Climate* 7, A9.

Emmert, John, Byers, Jeff, Warren, Harry, and Segerman, Alan (2014). "Propagation of Forecast Errors from the Sun to LEO Trajectories: How Does Drag Uncertainty Affect Conjunction Frequency?" In: *15th Advanced Maui Optical and Space Surveillance Technologies Conference*. Maui, HI.

Emmert, John T. (2008). "A physicist's tour of the upper atmosphere." In: *Physics today* 61.12, p. 70.

Emmert, John T. (2015). "Thermospheric mass density: A review." In: *Advances in Space Research* 56.5, pp. 773–824.

Emmert, John T., Warren, H. P., Segerman, A. M., Byers, J. M., and Picone, J. M. (2017). "Propagation of atmospheric density errors to satellite orbits." In: *Advances in Space Research* 59.1, pp. 147–165.

Escobar, Diego, Águeda, Alberto, Martín, Luis, and Martínez, Francisco M. (2011). "Efficient ALL vs. ALL collision risk analyses." In: *12th Advanced Maui Optical and Space Surveillance Technologies Conference*. Maui, HI, p. 32.

Esteva, Maria, Xu, Weijia, Simone, Nevan, Gupta, Amit, and Jah, Moriba K. (2020). "Modeling Data Curation to Scientific Inquiry: A Case Study for Multimodal Data Integration." In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*. Virtual Event, China, pp. 235–242.

Farrell, Damien (2016). "DataExplore: An application for general data analysis in research and education." In: *Journal of Open Research Software* 4.1, pp. 1–8.

Faulds, Anthony L. and Spencer, David B. (2003). "Satellite close-approach filtering using genetic algorithms." In: *Journal of spacecraft and rockets* 40.2, pp. 248–252.

Flohrer, T., Krag, H., and Klinkrad, H. (2008). "Assessment and Categorization of TLE Orbit Errors for the US SSN Catalogue." In: *9th Advanced Maui Optical and Space Surveillance Technologies Conference*. Maui, HI.

Flohrer, Tim, Jilete, B., Krag, H., Funke, Q., Braun, V., and Mancas, A. (2016). "ESA activities on satellite laser ranging to non-cooperative objects." In: *Proceedings of the 20th International Workshop on Laser Ranging*. Postdam, Germany.

Fock, Vladimir (1964). *The theory of space, time and gravitation*. 2nd ed. Pergamon Press. ISBN: 978-0-080-10061-6.

Forbes, Jeffrey M. (1972). *Low-Altitude Satellite Ephemeris Prediction*. Tech. rep. 72-0428. Bedford, MA: US Air Force Cambridge Research Laboratories.

Foster, James Lee and Estes, Herbert S. (Aug. 1992). *A parametric analysis of orbital debris collision probability and maneuver rate for space vehicles*. Tech. rep. NASA/JSC-25898.

Genz, Alan and Keister, Bradley D. (1996). "Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight." In: *Journal of Computational and Applied Mathematics* 71.2, pp. 299–309.

George, Eric R. and Harvey, Seth (2011). *A comparison of satellite conjunction analysis screening tools*. Tech. rep. Kirtland Air Force Space: Air Force Research Lab, Space Vehicles Directorate.

Gersten, Robert H., Gore, Rogers C., and Hall, Norman S. (1967). "Statistical properties of orbit perturbations induced by the Earth's anomalous gravity." In: *Journal of Spacecraft and Rockets* 4.9, pp. 1145–1150.

Ghrist, Richard W. and Plakalovic, Dragan (2012). "Impact of non-Gaussian error volumes on conjunction assessment risk analysis." In: *2012 AIAA/AAS Astrodynamics specialist conference*. Minneapolis, MN.

Golub, Gene H. and Van Loan, Charles F. (1996). *Matrix computations*. Vol. 3. Johns Hopkins University Press.

Gondelach, David J. and Linares, Richard (2020). "Atmospheric Density Uncertainty Quantification for Satellite Conjunction Assessment." In: *AIAA Scitech Forum*. Paper AIAA 2020-0232. Orlando, FL.

Greve, Gabriel H., Hopkinson, Kenneth M., and Lamont, Gary B. (2018). "Evolutionary sensor allocation for the Space Surveillance Network." In: *The Journal of Defense Modeling and Simulation* 15.3, pp. 303–322.

Hamming, Richard Wesley (1971). *Introduction to applied numerical analysis*. McGraw-Hill. ISBN: 978-0-070-25889-1.

Healy, Liam M. (1995). "Close conjunction detection on parallel computer." In: *Journal of Guidance, Control, and Dynamics* 18.4, pp. 824–829.

Hedin, Alan E., Niemann, H. B., Kasprzak, W. T., and Seiff, A. (1983). "Global empirical model of the Venus thermosphere." In: *Journal of Geophysical Research: Space Physics* 88.A1, pp. 73–83.

Hedin, Alan E., Fleming, E. L., Manson, A. H., Schmidlin, F. J., Avery, S. K., Clark, R. R., Franke, Steven J., Fraser, G. J., Tsuda, T., Vial, F., et al. (1996). "Empirical wind model for the upper, middle and lower atmosphere." In: *Journal of atmospheric and terrestrial physics* 58.13, pp. 1421–1447.

Heiskanen, Weikko A. and Moritz, Helmut (1967). *Physical geodesy*. Series of books in geology. W. H. Freeman and Company. ISBN: 978-0-716-70233-7.

Hejduk, Matthew D. (2017). "Collision Avoidance Short Course Part I: Theory." In: *18th Advanced Maui Optical and Space Surveillance Technologies Conference*. NASA Robotic CARA. Maui, HI.

Hejduk, Matthew D., Plakalovic, D., Newman, Lauri K., Ollivierre, J. C., Hametz, M. E., Beaver, B. A., and Thompson, R. C. (2013). "Trajectory error and covariance realism for launch COLA operations." In: *23rd AAS/AIAA Space Flight Mechanics Meeting*.

Hejduk, Matthew D. and Snow, D. E. (2018). "The effect of neutral density estimation errors on satellite conjunction serious event rates." In: *Space Weather* 16.7, pp. 849–869.

Hejduk, Matthew D., Snow, D. E., and Newman Lauri, K. (2019). "Satellite conjunction assessment risk analysis for "dilution region" events: issues and operational approaches." In: *Space Traffic Management Conference*. Austin, TX.

Higham, Nicholas J. (1988). "Computing a nearest symmetric positive semidefinite matrix." In: *Linear algebra and its applications* 103, pp. 103–118.

Hinks, Joanna and Psiaki, Mark (2010). "Simultaneous orbit and atmospheric density estimation for a satellite constellation." In: *2010 AIAA/AAS Astrodynamics Specialist Conference*. Paper AIAA 2010-8258. Toronto, Canada.

Hoffman, Joe D. and Frankel, Steven (2001). *Numerical methods for engineers and scientists*. Vol. 2. CRC press. ISBN: 978-0-824-70443-8.

Holzinger, Marcus J., Scheeres, Daniel J., and Alfriend, Kyle T. (2012). "Object Correlation, Maneuver Detection, and Characterization Using Control Distance Metrics." In: *Journal of Guidance, Control, and Dynamics* 35.4, pp. 1312–1325.

Hoots, Felix R. and Roehrich, Ronald L. (1980). *Models for propagation of NORAD element sets*. Tech. rep. Peterson Air Force Base, Colorado Springs, CO: U.S. Air Force Aerospace Defense Command.

Hoots, Felix R., Crawford, Linda L., and Roehrich, Ronald L. (1984). "An analytic method to determine future close approaches between satellites." In: *Celestial mechanics* 33.2, pp. 143–158.

Hoots, Felix R., Schumacher, Paul W. Jr., and Glover, Robert A. (2004). "History of analytical orbit modeling in the US space surveillance system." In: *Journal of Guidance, Control, and Dynamics* 27.2, pp. 174–185.

Horwood, Joshua T. and Poore, Aubrey B. (2011). "Adaptive Gaussian sum filters for space surveillance." In: *IEEE Transactions on Automatic Control* 56.8, pp. 1777–1790.

Horwood, Joshua T., Aragon, Nathan D., and Poore, Aubrey B. (2011). "Gaussian sum filters for space surveillance: theory and simulations." In: *Journal of Guidance, Control, and Dynamics* 34.6, pp. 1839–1851.

Horwood, Joshua T., Aragon, Nathan D., and Poore, Aubrey B. (2012). "Adaptive Gaussian sum filters for space surveillance tracking." In: *The Journal of the Astronautical Sciences* 59.1-2, pp. 308–326.

Houssineau, Jeremie and Bishop, Adrian N. (2018). "Smoothing and filtering with a class of outer measures." In: *SIAM/ASA Journal on Uncertainty Quantification* 6.2, pp. 845–866.

Huber, Marco F., Bailey, Tim, Durrant-Whyte, Hugh, and Hanebeck, Uwe D. (2008). "On entropy approximation for Gaussian mixture random vectors." In: *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. IEEE, pp. 181–188.

Jekeli, Christopher (2000). *Inertial navigation systems with geodetic applications*. Walter de Gruyter. ISBN: 978-3-11-015903-5.

Johnson, Nicholas (1996). "First natural collision of cataloged earth satellites." In: *Orbital Debris Quarterly News*. Vol. 1. 2. NASA Orbital Debris Program Office.

Johnson, Thomas M. (2015). "SSA Sensor Calibration Best Practices." In: *16th Advanced Maui Optical and Space Surveillance Technologies Conference*. Maui, HI.

Jones, Brandon, Delande, Emmanuel, Zucchelli, Enrico, and Jah, Moriba K. (2019). "Multi-Fidelity Orbit Uncertainty Propagation with Systematic Errors." In: *20th Advanced Maui Optical and Space Surveillance Technologies Conference*. Maui, HI.

Julier, Simon J. and Uhlmann, Jeffrey K. (1997). "A New Extension of the Kalman filter to Nonlinear Systems." In: *AeroSense '97, Signal processing, sensor fusion, and target recognition VI*. Vol. 3068. International Society for Optics and Photonics. Orlando, FL, pp. 182–193.

Julier, Simon J. and Uhlmann, Jeffrey K. (2004). "Unscented filtering and nonlinear estimation." In: *Proceedings of the IEEE* 92.3, pp. 401–422.

Junkins, John L., Akella, Maruthi R., and Alfriend, Kyle T. (1996). "Non-Gaussian error propagation in orbital mechanics." In: *Advances in the Astronautical Sciences: proceedings of the annual Rocky Mountain Guidance and Control Conference*. Vol. 92. San Diego, CA: Univelt Inc., pp. 283–298.

Kaula, William M. (1959). "Statistical and harmonic analysis of gravity." In: *Journal of Geophysical Research* 64.12, pp. 2401–2421.

Kaya, Denise, Ericson, Nancy, and Davis, Don (2004). "AFSPC Astrodynamic Standard Software (AAS 04-124)." In: *14th AAS/AIAA Space Flight Mechanics Meeting*. Maui, HI.

Kelso, T. S. and Alfano, Salvatore (2005). "Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space (SOCRATES)." In: *Advances in the Astronautical Sciences: proceedings of the 15th AAS/AIAA Space Flight Mechanics Meetings*. Vol. 120, pp. 317–326.

Kelso, T. S., Parkhomenko, N., Vasiliev, V. P., Shargorodskiy, V. D., Nazarenko, A., Yurasov, V., Tanygin, S., and Hiles, R. (2013). "What Happened to BLITS? An Analysis of the 2013 Jan 22 Event." In: *14th Advanced Maui Optical and Space Surveillance Technologies Conference*. Maui, HI.

Kessler, Donald J. (1991). "Collisional cascading: The limits of population growth in low earth orbit." In: *Advances in Space research* 11.12, pp. 63–66.

Kessler, Donald J. and Cour-Palais, Burton G. (1978). "Collision frequency of artificial satellites: The creation of a debris belt." In: *Journal of Geophysical Research: Space Physics* 83.A6, pp. 2637–2646.

King-Hele, Desmond George (1987). *Satellite orbits in an atmosphere: theory and application*. Springer Science & Business Media. ISBN: 978-0-216-92252-5.

Klinkrad, Heiner (2006). *Space Debris Models and Risk Analysis*. Springer. ISBN: 978-3-540-25448-5.

Kozai, Yoshihide (1959). "The motion of a close earth satellite." In: *Astronomical Journal* 64.11, pp. 367–377.

Krier, Georges (2017). "Satellite collision probability for long-term encounters and arbitrary primary satellite shape." In: *7th European conference on space debris*. Darmstadt, Germany.

Lane, M. H. and Cranford, K. H. (1969). "An improved analytical drag theory for the artificial satellite problem." In: *Astrodynamics Conference*. Paper AIAA 69-925. Princeton, NJ.

Laporte, François (2014). "JAC software, solving conjunction assessment issues." In: *15th Advanced Maui Optical and Space Surveillance Technologies Conference*. Maui, HI.

Leonard, J. M., Forbes, Jeffrey M., and Born, George H. (2012). "Impact of tidal density variability on orbital and reentry predictions." In: *Space Weather* 10.12.

Li, Xiao-Rong and Jilkov, Vesselin P. (2001a). "Survey of maneuvering target tracking: II. Ballistic target models." In: *Signal and Data Processing of Small Targets 2001*. Vol. 4473. International Society for Optics and Photonics, pp. 559–581.

Li, Xiao-Rong and Jilkov, Vesselin P. (2001b). "Survey of maneuvering target tracking: III. Measurement models." In: *Signal and Data Processing of Small Targets 2001*. Vol. 4473. International Society for Optics and Photonics, pp. 423–447.

Lidtke, Aleksander A. (2016). "High collision probability conjunctions and space debris remediation." PhD thesis. University of Southampton.

Liou, Jer Chyi (2005). "Accidental collisions of cataloged satellites identified." In: *Orbital Debris Quarterly News*. Vol. 9. 2. NASA Orbital Debris Program Office.

Liou, Jer Chyi (2020). "Risks from Orbital Debris and Space Situational Awareness." In: *2nd IAA Conference on Space Situational Awareness (ICSSA)*. NASA Johnson Space Center. Washington, DC.

Liou, Jer Chyi, Kieffer, M., Dreq, A., and Sweet, A. (2020). "The 2019 U.S. Government Orbital Debris Mitigation Standard Practices." In: *Orbital Debris Quarterly News*. Vol. 24. 1. NASA Orbital Debris Program Office.

Liu, Huixin, Doornbos, Eelco, and Nakashima, Junichiro (2016). "Thermospheric wind observed by GOCE: Wind jets and seasonal variations." In: *Journal of Geophysical Research: Space Physics* 121.7, pp. 6901–6913.

Luo, Yazhong and Yang, Zhen (2017). "A review of uncertainty propagation in orbital mechanics." In: *Progress in Aerospace Sciences* 89, pp. 23–39.

Mandea, Mioara and Korte, Monika (2010). *Geomagnetic observations and models*. Vol. 5. Springer. ISBN: 978-90-481-9857-3.

Marcos, Frank A. (1998). "Precision low Earth orbit determination using atmospheric density calibration." In: *Advances in the Astronautical Sciences: proceedings of the 1997 AAS/AIAA Astrodynamics Specialist Conference*. Vol. 97, pp. 501–513.

Marcos, Frank A. (2006). "New satellite drag modeling capabilities." In: *44th AIAA Aerospace Sciences Meeting and Exhibit*, p. 470.

McKinley, David P. (2006). "Development of a nonlinear probability of collision tool for the Earth observing system." In: *2006 AIAA/AAS Astrodynamics Specialist Conference*. Paper AIAA 2006-6295. Keystone, CO.

Menvielle, Michel (1998). "The geomagnetic indices: derivation, meaning, and availability." In: *ESA Workshop on Space Weather*. Noordwijk, The Netherlands: ESA Space Environments and Effects Analysis Section.

Menvielle, Michel, Iyemori, Toshihiko, Marchaudon, Aurélie, and Nosé, Masahito (2011). "Geomagnetic indices." In: *Geomagnetic observations and models*. Springer, pp. 183–228.

Michael, John R. (1983). "The stabilized probability plot." In: *Biometrika* 70.1, pp. 11–17.

Miller, James G. (2004). "A New Sensor Resource Allocation Algorithm for the Space Surveillance Network in Support of the Special Perturbations Satellite Catalog." In: *Advances in the Astronautical Sciences: proceedings of the 2003 AAS/AIAA Astrodynamics Specialist Conference*. Vol. 116. Paper AAS 03-669. San Diego, CA: Univelt, Inc.

Miller, James G. (2007). "A new sensor allocation algorithm for the space surveillance network." In: *Military Operations Research*, pp. 57–70.

Moe, Kenneth (1962). "Stochastic Models of the Errors in Orbital Predictions for Artificial Earth Satellites." In: *ARS Journal* 32, pp. 1726–1728.

Montenbruck, Oliver and Gill, Eberhard (2000). *Satellite orbits: models, methods and applications*. Springer-Verlag Berlin Herdelberg. ISBN: 978-3-54067-280-7.

Moritz, Helmut (1980). *Advanced Physical Geodesy*. Sammlung Wichmann. Neue Folge: Buchreihe Bd. 13. Herbert Wichmann. ISBN: 978-3-879-07106-7.

Nader, R. and Kelso, T. S. (2014). "The Pegasus incident: The loss of the first Ecuadorian satellite and its recovery." In: *65th International Astronautical Congress*.

National Research Council (2012). *Continuing Kepler's Quest: Assessing Air Force Space Command's Astrodynamics Standards*. National Academies Press. ISBN: 978-0-309-26142-5.

Nazarenko, Andrey I., Cefola, Paul J., and Yurasov, Vasiliy (1998). "Estimating atmosphere density variations to improve LEO orbit prediction accuracy." In: *8th AAS/AIAA Space Flight Mechanics Meeting*. Paper AAS 98-190. Breckenridge, CO, pp. 1235–1264.

Newman, Lauri K., Frigm, R., and McKinley, D. (2009). "It's Not a Big Sky After All: Justification for a Close Approach Prediction and Risk Assessment Process." In: *Advances in the Astronautical Sciences: proceedings of the 2009 AAS/AIAA Astrodynamics Specialist Conference*. Vol. 135. San Diego, CA: Univelt, Inc., pp. 1113–1132.

Newman, Lauri K., Frigm, Ryan C., Duncan, Matthew G., and Hejduk, Matthew D. (2014). "Evolution and implementation of the NASA robotic conjunction assessment risk analysis concept of operations." In: *15th Advanced Maui Optical and Space Surveillance Technologies Conference*. Maui, HI.

Newman, Lauri K., Hejduk, Matthew D., and Johnson, Lauren C. (2016). "Operational implementation of a Pc uncertainty construct for conjunction assessment risk analysis." In: *17th Advanced Maui Optical and Space Surveillance Technologies Conference*. Maui, HI.

Novak, Daniel M., Biamonti, Davide, Gross, Jeremy, and Milnes, Martin (2013). "A Powerful, Cost Effective, Web Based Engineering Solution Supporting Conjunction Detection and Visual Analysis." In: *6th European Conference on Space Debris*. ESA SP-723. Darmstadt, Germany: ESA publications devision.

Patera, Russell P. (2001). "General method for calculating satellite collision probability." In: *Journal of Guidance, Control, and Dynamics* 24.4, pp. 716–722.

Patera, Russell P. (2003). "Satellite collision probability for nonlinear relative motion." In: *Journal of Guidance, Control, and Dynamics* 26.5, pp. 728–733.

Patera, Russell P. (2005). "Calculating collision probability for arbitrary space vehicle shapes via numerical quadrature." In: *Journal of guidance, control, and dynamics* 28.6, pp. 1326–1328.

Pechenick, Kay R. (1988). *A Derivation of the Gravity Error Covariance Formulas Including the Off-Diagonal Components*. Tech. rep. Prepared for Goddard Spaceflight Center, Doc.-Id. TR-NAS5-29417-1A. Applied Technology Associates, Inc.

Petit, Gérard and Brian Luzum, eds. (2010). *IERS Conventions (2010)*. IERS Technical Note 36. Frankfurt am Main: Verlag des Bundesamts für Kartographie und Geodäsie. ISBN: 978-3-89888-989-6.

Picone, J. M., Hedin, A. E., Drob, D. P., and Aikin, A. C. (2002). "NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues." In: *Journal of Geophysical Research: Space Physics* 107.A12.

Picone, J. M., Meier, R. R., and Emmert, John T. (2013). "Theoretical tools for studies of low-frequency thermospheric variability." In: *Journal of Geophysical Research: Space Physics* 118.9, pp. 5853–5873.

Pitz, Wolfgang and Miller, David (2010). "The TerraSAR-X satellite." In: *IEEE Transactions on Geoscience and Remote Sensing* 48.2, pp. 615–622.

Poore, Audrey B., Aristoff, Jeffrey M., Horwood, Joshua T., Armellin, Roberto, Cerven, William T., Cheng, Yang, Cox, Christopher M., Erwin, Richard S., Frisbee, Joseph H., Hejduk, Matthew D., et al. (2016). *Covariance and uncertainty realism in space surveillance and tracking*. Tech. rep. Fort Collins, CO: Numerica Corporation.

Rauch, H. E. (1965). "Optimum estimation of satellite trajectories including random fluctuations in drag." In: *AIAA Journal* 3.4, pp. 717–722.

Roble, Raymond G. (1995). "Energetics of the mesosphere and thermosphere." In: *The Upper Mesosphere and Lower Thermosphere: A Review of Experiment and Theory, Geophysical Monograph Series* 87, pp. 1–21.

Rodriguez Fernandez, Oscar, Utzmann, Jens, and Hugentobler, Urs (2019a). "Correlation of optical observations to catalogued objects using multiple hypothesis filters." In: *1st NEO and Debris Detection Conference*. Darmstadt, Germany: ESA Space Safety Programme Office.

Rodriguez Fernandez, Oscar, Utzmann, Jens, and Hugentobler, Urs (2019b). "SPOOK - A comprehensive Space Surveillance and Tracking analysis tool." In: *Acta Astronautica* 158, pp. 178–184.

Royston, Patrick (1993). "Graphical Detection of Non-Normality by Using Michael's Statistic." In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 42.1, pp. 153–158.

Rump, Siegfried M. (2009). "Inversion of extremely ill-conditioned matrices in floating-point." In: *Japan Journal of Industrial and Applied Mathematics* 26.2-3, pp. 249–277.

Runnalls, Andrew R. (2007). "Kullback-Leibler approach to Gaussian mixture reduction." In: *IEEE Transactions on Aerospace and Electronic Systems* 43.3, pp. 989–999.

Sabol, Chris, Sukut, Thomas, Hill, Keric, Alfriend, Kyle T., Wright, Brendan, Li, You, and Schumacher, Paul (2010). "Linearized orbit covariance generation and propagation analysis via simple Monte Carlo simulations." In: *20th AAS/AIAA Space Flight Mechanics Meeting*, pp. 14–17.

Sagnières, Luc and Sharf, Inna (2017). "Uncertainty characterization of atmospheric density models for orbit prediction of space debris." In: *7th European Conference on Space Debris*. Darmstadt, Germany: ESA Space Debris Office.

Salmond, Deborah J., Atherton, D. P., and Bather, J. A. (1988). "Mixture reduction algorithms for uncertain tracking." In: *IFAC Proceedings Volumes* 21.9, pp. 775–780.

Schutz, Bob, Tapley, Byron, and Born, George H. (2004). *Statistical orbit determination*. Elsevier. ISBN: 978-0-126-83630-1.

Seidelmann, P. Kenneth (2006). *Explanatory supplement to the astronomical almanac*. University Science Books. ISBN: 978-1-891-38945-0.

Shelton, Christopher T. and Junkins, John L. (2019). "Probability of collision between space objects including model uncertainty." In: *Acta Astronautica* 155, pp. 462–471.

Singh, Navraj, Horwood, Joshua T., and Poore, Aubrey B. (2012). "Space object maneuver detection via a joint optimal control and multiple hypothesis tracking approach." In: *22nd AAS/AIAA Space Flight Mechanics Meeting*. Vol. 143. Paper AAS 12-159, pp. 843–862.

Sorenson, Harold W. and Alspach, Daniel L. (1971). "Recursive Bayesian estimation using Gaussian sums." In: *Automatica* 7.4, pp. 465–479.

Storz, Mark F., Bowman, Bruce R., Branson, James I., Casali, Stephen J., and Tobiska, W. Kent (2005). "High accuracy satellite drag model (HASDM)." In: *Advances in Space Research* 36.12, pp. 2497–2505.

Tapping, Ken (2013). "The 10.7 cm solar radio flux (F10.7)." In: *Space Weather* 11.7, pp. 394–406.

Terejanu, Gabriel, Singla, Puneet, Singh, Tarunraj, and Scott, Peter D. (2008a). "A novel Gaussian Sum Filter Method for accurate solution to the nonlinear filtering problem." In: *2008 11th International Conference on Information Fusion*. IEEE, pp. 1–8.

Terejanu, Gabriel, Singla, Puneet, Singh, Tarunraj, and Scott, Peter D. (2008b). "Uncertainty propagation for nonlinear dynamic systems using Gaussian mixture models." In: *Journal of Guidance, Control, and Dynamics* 31.6, pp. 1623–1633.

Terejanu, Gabriel, Singla, Puneet, Singh, Tarunraj, and Scott, Peter D. (2011). "Adaptive Gaussian sum filter for nonlinear Bayesian estimation." In: *IEEE Transactions on Automatic Control* 56.9, pp. 2151–2156.

The Space Foundation (2020). *The Space Economy Scorecard 2020*. http://web.archive.org/web/20200502154510/www.thespacereport.org/scorecard. Accessed May 2nd, 2020.

USSF (2020). *Initial operational capability and operational acceptance of Space Fence*. http://web.archive.org/web/20200504202328/https://www.spaceforce.mil/News/Article/2129325/ussf-announces-initial-operational-capability-and-operational-acceptance-of-spa. Accessed May 4th, 2020.

Utzmann, Jens, Vesselinova, Maria, and Rodriguez Fernandez, Oscar (2019). "Airbus Robotic Telescope." In: *1st NEO and Debris Detection Conference*. Darmstadt, Germany: ESA Space Safety Programme Office.

Vallado, David (2004). "Covariance transformations for satellite flight dynamics operations." In: *2003 AIAA/AAS Astrodynamics Specialists Conference*. Paper AAS-03-526. Big Sky, MT.

Vallado, David (2013). *Fundamentals of Astrodynamics and Applications*. 4th ed. Microcosm Press. ISBN: 978-1-88188-318-0.

Vallado, David, Crawford, Paul, Hujsak, Richard, and Kelso, T. S. (2006). "Revisiting spacetrack report# 3." In: *2006 AIAA/AAS Astrodynamics Specialists Conference*. Paper AIAA 2006-6753-Rev2. Keystone, CO.

Vallado, David and Finkleman, David (2014). "A critical assessment of satellite drag and atmospheric density modeling." In: *Acta Astronautica* 95, pp. 141–165.

Vallado, David and Alfano, Salvatore (2015). "Updated Analytical Partials for Covariance Transformations and Optimization." In: *2015 AIAA/AAS Astrodynamics Specialists Conference*. Paper AAS 15-537. Vail, CO.

Vallverdú, Jordi (2015). *Bayesians versus frequentists: a philosophical debate on statistical reasoning*. Springer. ISBN: 978-3-662-48636-8.

VanderPlas, Jake (2014). "Frequentism and Bayesianism: A Python-driven Primer." In: *Proceedings of the 13th Python in Science Conference (SciPY 2014)*. Austin, Texas.

Vishwajeet, Kumar, Singla, Puneet, and Jah, Moriba K. (2014). "Nonlinear uncertainty propagation for perturbed two-body orbits." In: *Journal of Guidance, Control, and Dynamics* 37.5, pp. 1415–1425.

Vittaldev, Vivek (2015). "Uncertainty propagation and conjunction assessment for resident space objects." PhD thesis. The University of Texas at Austin.

Vittaldev, Vivek and Russell, R. P. (2016). "Multidirectional Gaussian Mixture Models for Nonlinear Uncertainty Propagation." In: *CMES-Computer Modeling In Engineering & Sciences* 111.1, pp. 83–117.

von Mises, Richard and Pollaczek-Geiringer, Hilda (1929). "Praktische Verfahren der Gleichungsauflösung." In: *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik* 9.2, pp. 152–164.

Wertz, James R. (2012). *Spacecraft Attitude Determination and Control*. Springer Science & Business Media. ISBN: 978-90-277-1204-2.

Wetterer, Charles J., Linares, Richard, Crassidis, John L., Kelecy, Thomas M., Ziebart, Marek K., Jah, Moriba K., and Cefola, Paul J. (2014). "Refining space object radiation pressure modeling with bidirectional reflectance distribution functions." In: *Journal of Guidance, Control, and Dynamics* 37.1, pp. 185–196.

Wilkins, Matthew P. and Alfriend, Kyle T. (2000). "Characterizing orbit uncertainty due to atmospheric uncertainty." In: *2000 AIAA/AAS Astrodynamics Specialist Conference*. Paper AIAA-2000-3931. Denver, CO, pp. 19–29.

Williams, Jason L. (2003). "Gaussian mixture reduction for tracking multiple maneuvering targets in clutter." Master's thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology.

Williams, Jason L. and Maybeck, Peter S. (2003). "Cost-function-based Gaussian mixture reduction for target tracking." In: *Proceedings of the sixth international conference on information fusion*. Vol. 2. IEEE. Cairns, Queensland, Australia, pp. 1047–1054.

Williams, Patrick S. (2012). "Coupling between nonlinear estimation and dynamic sensor tasking applied to satellite tracking." PhD thesis. The Pennsylvania State University.

Woodburn, James (Aug. 2019). Personal communication.

Woodburn, James, Coppola, Vincent T., and Stoner, Frank (2009). "A description of filters for minimizing the time required for orbital conjunction computations." In: *Advances in the Astronautical Sciences: proceedings of the 2009 AAS/AIAA Astrodynamics Specialist Conference*. Vol. 135. 2. San Diego, CA: Univelt, Inc., pp. 1157–1173.

Woodburn, James and Coppola, Vincent T. (2013). "Effect of Coordinate Selection on Orbit Determination." In: *23rd AIAA/AAS Spaceflight Mechanics Meeting*. Paper AAS13-825. Kauai, HI.

Wright, David (2009). *Colliding satellites: consequences and implications*. Tech. rep. Union of Concerned Scientists.

Wright, James R. (1981). "Sequential Orbit Determination with Auto-Correlated Gravity Modeling Errors." In: *Journal of Guidance and Control* 4.3, pp. 304–309.

Wright, James R. (1994). "Orbit determination solution to non-Markov gravity error problem." In: Paper AAS 94-176. Cocoa Beach, FL, pp. 1153–1172.

Wright, James R. (2011). "QQ-Plot for Sequential Orbit Determination." In: *Advances in the Astronautical Sciences: proceedings of the 2011 AAS/AIAA Astrodynamics Specialist Conference*. Vol. 142. Paper AAS 11-607. San Diego, CA: Univelt, Inc., pp. 3189–3205.

Wright, James R., Woodburn, James, Truong, Son, and Chuba, William (2008a). "Orbit gravity error covariance." In: *18th AAS/AIAA Space Flight Mechanics Meeting*. Paper AAS 08-157. Galveston, TX.

Wright, James R., Woodburn, James, Truong, Son, and Chuba, William (2008b). "Sample Orbit Covariance Function and Filter Smoother Consistency Tests." In: *18th AAS/AIAA Space Flight Mechanics Meeting*. Paper AAS 08-159. Galveston, TX.

Wright, James R., Woodburn, James, Truong, Son, and Chuba, William (2008c). "Orbit Covariance Inner Integrals with Polynomials." In: *18th AAS/AIAA Space Flight Mechanics Meeting*. Paper AAS 08-161. Galveston, TX.

Xu, Wenyao (2008). "Uncertainty in magnetic activity indices." In: *Science in China Series E: Technological Sciences* 51.10, pp. 1659–1664.

Yang, Minxian (2008). "Normal log-normal mixture, leptokurtosis and skewness." In: *Applied Economics Letters* 15.9, pp. 737–742.

Zhao, Feng, Shin, Jaewon, and Reich, James (2002). "Information-driven dynamic sensor collaboration." In: *IEEE Signal processing magazine* 19.2, pp. 61–72.

COLOPHON

This thesis was typeset in LaTeX[1] using the typographical look-and-feel `classicthesis`[2] developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". The bibliography was processed using BibLaTeX[3].

[1] *https://latex-project.org*

[2] *https://bitbucket.org/amiede/classicthesis*

[3] *https://ctan.org/pkg/biblatex*

Covariance and uncertainty realism for low Earth orbiting satellites via quantification of dominant force model uncertainties