Probabilistic Uncertainty Quantification in Space Object Tracking and Orbital Safety Analysis

by

Yannick Sztamfater Garcia

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

Aerospace Engineering

Universidad Carlos III de Madrid

Advisor(s):

Joaquin Miguez Arenas
Manuel Sanjurjo-Rivo

September 2025

To my family, my girlfriend, and my closest friends in Madrid, Barcelona, and London.

ACKNOWLEDGEMENTS

They call this section Acknowledgements, which feels too formal and understated —like I'm just politely nodding at people who helped me survive this thesis. In truth, what follows is less an acknowledgment and more a heartfelt expression of gratitude.

I am beyond grateful and feel extremely lucky to have had Joaquin Miguez and Manuel Sanjurjo Rivo as supervisors during this journey. To Joaquin: thank you for your invaluable guidance and insight throughout my PhD. You are a brilliant researcher and an even better person — always kind and committed to my progress and well-being. I look up to you not only professionally, but also personally. To Manuel: thank you for always being available to me (even after your departure from UC3M), and for consistently offering creative and useful perspectives on the research questions that arose. I'm inspired by your ability to remain cheerful, witty, and unflustered, even amid stress and change.

Thank you to my UCSD research stay hosts, Aaron J. Rosengren and Boris Kramer, who were both extremely helpful — not just during my stay, but also afterward, in the writing and development of further research. To Aaron: thank you for helping me settle into life in San Diego, for introducing me to your lab, for letting me use your gym, for showing me the delights of The Waterfront, and for your tattoo advice. To Boris: thank you for your valuable insights into my research problem and for proposing suitable methodologies. I'm also grateful for the opportunity to join your lab and work with your students — it made my time there significantly more enjoyable.

Thank you to my parents — for, well, creating me, and for always reminding me how proud they are. Thank you to my siblings, Axel, Alan, and Melody, for being my timeless best friends and for always (incorrectly) telling people that I either work at NASA or am an astronaut.

A very special thank you to my amazing girlfriend, Tiphène, for moving to Madrid for me and with me, working full-time in a third language she wasn't yet comfortable with, and for helping me draw (ever-improving) boundaries between life and work. Thank you for understanding my commitments — whether it's the PhD, karate, or band practice — and for always offering your full support.

An enormous shoutout to my fellow Chiefs: Luca, Ricardo, Lucas, Donato, Raul, Bun Kim, Irene and Thomas for truly becoming some of my best friends and making university life (and life in general) 100 times more fun and meaningful. From coffee breaks and three-hour lunches to house parties at Ducato's, tacos by the thousands, and late-night techno shenanigans — you made it unforgettable.

A special thank you to Ricardo, Giuseppe (GuicyP), and Eddie HD for always being down to rock vocal and guitar strings together with Los Cohetes. Black!

Thank you to my friend Rashid, for being there from the start of my PhD journey - through some pretty obscure mathematics classes — and always having time for a coffee or a dinner, and to Guillermo Escribano, for helping me land on my feet early on and for sharing knowledge and tools without which the learning curve would have been much steeper.

Thank you, Karate family at Dojo Akeru for providing me with therapeutic bone-breaking, abdominal bruising and face-kicking doses: indeed a crucial outlet during my PhD days.

This work has been supported by Comunidad de Madrid (project ref. IND2020/TIC-17539), *Agencia Estatal de Investigación* (ref. PID2021-125159NB-I00 TYCHE) funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe", and the Office of Naval Research (award N00014-22-1-2647).

PUBLISHED AND SUBMITTED CONTENT

- Yannick Sztamfater Garcia, Manuel Sanjurjo Rivo, Guillermo Escribano, Harold Molina-Bulla and Joaquin Miguez, "An approximate model for the computation of in-orbit collision probabilities using importance sampling", *Advances in Space Research*, vol. 75(4), pages 3791-3805. 15 February 2025. doi: https://doi.org/10.1016/j.asr.2024.12.074. The material in this published journal paper is wholly included in this thesis and is not singled out with typographic means and references. It represents the bulk of the methodology and results of Chapter 3, and especially in Sections 3.2.3, 3.2.4, 3.3.2, and 3.4.

- Yannick Sztamfater Garcia, Joaquin Miguez, and Manuel Sanjurjo Rivo, "Sequential filtering techniques for simultaneous tracking and parameter estimation". Manuscript submitted to *Journal of Astronautical Science* on 15 March 2025. The material in this submitted journal paper is wholly included in this thesis and is not singled out with typographic means and references. It represents the bulk of the methodology and results in part two of Chapter 4, and especially in Sections 4.3 and 4.7. Some of this material is also included in Chapter 2, Sections 2.4 and 2.6.

- Yannick Sztamfater Garcia, Manuel Sanjurjo Rivo, Guillermo Escribano and Joaquin Miguez, "Rare event sampling schemes for the efficient computation of in-orbit collision probabilities", in *2nd NEO and Space Debris Conference (ESA)*, January 2023. The material in this conference paper is partly included in this thesis and is not singled out with typographic means and references. It is an early version of the methodology used in Chapter 3, Sections 3.2.3 and 3.2.4.

- Yannick Sztamfater Garcia, Boris Kramer, Aaron Rosengren, Manuel Sanjurjo Rivo, and Joaquin Miguez, "Multifidelity Monte Carlo for the estimation of re-entry windows", at the *11th European Conference for Aerospace Sciences, June 2025. The material in this conference paper is wholly included in this thesis and is not singled out with typographic means and references. It is an early version of the methodology used in Chapter 5, Sections 5.1 and 5.4.*

- *Yannick Sztamfater Garcia, Manuel Sanjurjo Rivo, Guillermo Escribano and Joaquin Miguez, "Novel method for the Computation of In-Orbit Collision Probability by Multilevel Splitting and Surrogate Modelling", in 34th AIAA-SciTech Forum, January 2024. The material in this conference paper is partly included in this thesis and is not singled out with typographic means and references. It is an early version of the methodology used in Chapter 3, but represents an obsolete version of the methodology now present in the thesis.*

ABSTRACT

The near-Earth space environment is currently experiencing a rapid surge in the number of resident space objects (RSOs). This surge is attributed to advancements in launch capabilities, cost reductions in space missions, the expansion of internet infrastructure and the rise of nano-satellite technologies, enabling the simultaneous deployment of multiple, cost-effective satellites. While space surveillance and tracking along with collision avoidance procedures presently meet the safety needs of the current RSO population, there are concerns that they might soon prove inadequate. The likelihood of collisions is poised to increase significantly given the current growth rates of RSOs. Orbital collisions, although currently rare, pose inherent challenges as they can disperse thousands of new fragments into Earth's orbit, making accurate tracking nearly impossible due to their small size. Predicting the likelihood of such events by obtaining suitable risk metrics, and accurately tracking as many objects as possible are tasks which lie at the heart of space safety for the current world.

Standard methods which compute the probability of collision (PoC) often make restrictive assumptions which produce numerically efficient solutions, the quality of which may decrease over long time-spans. The crude Monte Carlo (CMC) method makes few assumptions and can produce high quality estimates provided enough samples are used; however, its computational cost can be prohibitively high. In the first area of study of this thesis, we introduce a new method based on importance sampling (IS), which is shown to attain the accuracy of a CMC simulation at a fraction of the computational cost. To accomplish this, we propose a surrogate dynamical model which exploits Keplerian conjunction geometry and applies a linear correction to approximate the full-dynamics of the conjunction scenario, reducing the computational load of evaluating the dynamical model. This linear approximation is satisfactorily validated by means of high-fidelity model evaluations. The PoC results, when the method is applied on LEO and GEO scenarios, show agreement with an independent CMC benchmark simulation for the calculation of the PoC at a fraction of the computational cost. However, PoC computations are inherently reliant on the uncertainty around the initial state estimates of the spacecraft being studied. A large uncertainty may lead to a probability dilution, which is an undesirable result in terms of safety.

It is therefore important to be able to track RSOs accurately and at an affordable computational cost, in addition to correctly characterising the uncertainty present, which is the main objective of the second area of study in the thesis. Orbital dynamics are highly nonlinear, and current operational methods assume Gaussian representations of the objects' states and employ linearizations which cease to hold true in observation-free propagation. Monte Carlo-based filters can provide a means to approximate the a posteriori probability distribution of the states more accurately by providing support in the portion of the state space which overlaps the most with the processed observations. Moreover, dynamical models are not able to capture the full extent of realistic forces experienced in the near-Earth space environment, and hence fully deterministic propagation methods may fail to achieve the desired accuracy. By representing orbital dynamics through stochastic differential equations, we are able to simultaneously estimate the scale of the process noise incurred by the assumed uncertainty in the system, and robustly track the state of the spacecraft. We propose three algorithms which are capable of tracking a space object and estimating the magnitude of the system's uncertainty. The proposed filters are successfully applied to a LEO scenario, demonstrating the ability to accurately track a spacecraft state and estimate the scale of the uncertainty online, in various simulation setups.

An application where uncertainty plays a significant role is uncontrolled re-entry, which represents the fate of many orbiting objects. Commonly used re-entry prediction methods use orbital propagation models that assume Gaussian uncertainties, which may fail to provide accurate representations of the state's probability density function during propagation, making it difficult to obtain reliable decay-time metrics. CMC simulations, though expensive, typically enable accurate representations of the states at the time of decay, provided that sufficiently many samples are used. In the third area of study of this thesis, we propose an approach based on multifidelity Monte Carlo to provide decay time window estimates by optimally distributing work between lower-fidelity dynamical models and accurate high-fidelity models. The method is applied in a decay scenario involving real re-entered objects for which decay time windows are available. This allows for comparison with a ground truth time value as well as with an independent high-fidelity CMC simulation.

In summary, this thesis provides alternative methods in an attempt to bypass common assumptions in the dynamical models and state distribution representations as well as appropriately account for the uncertainty present in various space traffic management prediction and estimation processes, in an efficient manner.

RESUMEN

El entorno espacial cercano a la Tierra está actualmente experimentando un rápido aumento en el número de objetos espaciales residentes (RSOs, por sus siglas en inglés). Este incremento se debe a los avances en las capacidades de lanzamiento, la reducción de costos en las misiones espaciales, la expansión de la infraestructura de internet y el auge de las tecnologías de nanosatélites, que permiten el despliegue simultáneo de múltiples satélites de bajo coste. Si bien la vigilancia y el seguimiento espaciales, junto con los procedimientos de evasión de colisiones, satisfacen actualmente las necesidades de seguridad de la población de RSOs, existen preocupaciones de que pronto puedan resultar insuficientes. La probabilidad de colisión está destinada a aumentar significativamente dado el ritmo de crecimiento actual de los RSOs. Las colisiones orbitales, aunque poco frecuentes por ahora, representan un desafío inherente, ya que pueden dispersar miles de fragmentos en la órbita terrestre, cuya detección resulta casi imposible debido a su reducido tamaño. Predecir la probabilidad de tales eventos mediante métricas de riesgo adecuadas y realizar un seguimiento preciso del mayor número posible de objetos son tareas fundamentales para la seguridad espacial en el mundo actual.

Los métodos estándar para calcular la probabilidad de colisión (PoC, por sus siglas en inglés) suelen basarse en suposiciones restrictivas que permiten soluciones numéricamente eficientes, pero cuya precisión puede degradarse con el tiempo. El método “crude Monte Carlo" (CMC) requiere pocas suposiciones y puede ofrecer estimaciones de alta calidad si se emplea un número suficiente de muestras; sin embargo, su costo computacional puede resultar prohibitivo. En la primera línea de investigación de esta tesis, se introduce un nuevo método basado en “importance sampling" (IS), que logra una precisión comparable a la del CMC pero con una fracción del costo computacional. Para ello, se propone un modelo dinámico simplificado que aprovecha la geometría kepleriana del encuentro y aplica una corrección lineal para aproximar la dinámica completa del escenario de conjunción, reduciendo el coste computacional. Esta aproximación lineal se valida satisfactoriamente mediante evaluaciones de un modelo de alta fidelidad. Los resultados de PoC, al aplicar el método en escenarios LEO y GEO, muestran concordancia con una simulación independiente de CMC, a un coste computacional significativamente menor. Sin embargo, los cálculos de PoC dependen inherentemente de la incertidumbre en torno a las estimaciones iniciales del estado de las naves espaciales. Una incertidumbre

elevada puede generar un fenómeno conocido como dilución de probabilidad, lo cual es indeseable desde el punto de vista de la seguridad orbital.

Por ello, resulta fundamental poder realizar un seguimiento adecuado de los RSOs con un coste computacional asequible, además de caracterizar adecuadamente la incertidumbre presente, lo cual constituye el objetivo principal de la segunda línea de estudio de esta tesis. La dinámica orbital es altamente no lineal, y los métodos operacionales actuales asumen representaciones gaussianas de los estados de los objetos y emplean linearizaciones que dejan de ser válidas en propagaciones largas. Los filtros basados en Monte Carlo permiten aproximar de manera más precisa la distribución a posteriori de los estados, al enfocarse en las regiones del espacio de estados que se superponen con las observaciones disponibles. Además, los modelos dinámicos no capturan completamente las fuerzas realistas en el entorno espacial cercano a la Tierra, por lo que los métodos deterministas pueden fallar en lograr la precisión deseada. Al representar la dinámica orbital mediante ecuaciones diferenciales estocásticas, es posible estimar simultáneamente la escala del ruido de proceso asociado a la incertidumbre del sistema, y realizar un seguimiento robusto del estado del objeto. Se proponen tres algoritmos capaces de realizar este seguimiento y estimar la magnitud de la incertidumbre del sistema. Los filtros propuestos se aplican con éxito a un escenario LEO, demostrando su capacidad para seguir con precisión el estado de una nave espacial y estimar en línea la escala de la incertidumbre bajo distintos esquemas de simulación.

Una aplicación donde la incertidumbre juega un papel crítico es la reentrada no controlada, que representa el destino de muchos objetos en órbita. Los métodos habituales de predicción de reentrada emplean modelos de propagación orbital que asumen incertidumbres gaussianas, lo que puede resultar en representaciones poco precisas de la función de densidad de probabilidad del estado durante la propagación, dificultando la estimación fiable del tiempo de caída. Las simulaciones CMC, aunque costosas, permiten representar correctamente el estado al momento de la reentrada, siempre que se utilice un número suficiente de muestras. En la tercera línea de investigación de esta tesis, se propone un enfoque basado en el “multifidelity Monte Carlo" para estimar ventanas de tiempo de reentrada, distribuyendo de manera óptima el coste entre modelos dinámicos de baja y alta fidelidad. El método se aplica a dos casos reales de reentrada de objetos con tiempos de caída conocidos, lo que permite comparaciones tanto con valores reales como con simulaciones independientes de alta fidelidad mediante CMC.

En resumen, esta tesis ofrece métodos alternativos para evitar suposiciones restrictivas en modelos dinámicos y representaciones del estado, y para tener en cuenta adecuadamente la incertidumbre presente en distintos procesos de predicción y estimación en la gestión del tráfico espacial, de manera eficiente.

CONTENTS

List of figures xx

List of tables xxii

Acronyms and abbreviations xxiii

List of symbols xxv

1 Introduction 1

1.1. Motivation . . . 1

1.2. Challenges . . . 6

1.3. Objectives . . . 8

1.4. Contributions . . . 9

1.5. Organisation of the thesis . . . 10

2 Tools and theory 11

2.1. Notation . . . 11

2.2. Orbital dynamical models . . . 12

2.2.1. Keplerian model . . . 12

2.2.2. Cowell formulation . . . 14

2.2.3. Observations . . . 16

2.3. Probability theory . . . 16

2.4. Numerical integration . . . 18

2.4.1. Deterministic integration . . . 18

2.4.2. Stochastic integration . . . 20

2.5. State space model 21
2.6. Estimation theory 21

3 Orbital collision risk assessment **25**
3.1. State of the art 26
3.2. Methodology 28
3.2.1. Initial conditions 28
3.2.2. Collision events 29
3.2.3. Reduction to 2D space 29
3.2.4. Surrogate model 34
3.3. PoC computation 37
3.3.1. Crude Monte Carlo analysis 37
3.3.2. PoC by importance sampling 39
3.3.3. Proposed Methodology 40
3.4. Results and discussion 43
3.4.1. Test cases 43
3.4.2. Importance sampling: LEO 45
3.4.3. Importance sampling: GEO 50
3.5. Concluding remarks 54

4 Spacecraft tracking **57**
4.1. State of the art 58
4.1.1. State-only tracking 58
4.1.2. State tracking and unknown parameter estimation 61
4.2. Methodology: state-only tracking 63
4.2.1. Gaussian sequential filters 63
4.2.2. Particle filters 66
4.2.3. Hybrid filters 67
4.3. Methodology: state tracking and parameter estimation 73
4.3.1. Hybrid filters 73
4.4. Test cases: state-only tracking 77
4.4.1. LEO 77

4.4.2. GEO 79
4.5. Results: state-only tracking 80
4.5.1. Simulation setup 80
4.5.2. LEO 81
4.5.3. GEO 85
4.6. Test case: state-tracking and parameter estimation 90
4.6.1. LEO 90
4.7. Results: state tracking and parameter estimation 91
4.7.1. Simulation configuration 91
4.7.2. Estimating the nominal parameter 92
4.7.3. Estimating the parameter with simplified dynamics. 97
4.7.4. Estimating a 3-dimensional parameter 99
4.8. Concluding remarks 101

5 Spacecraft re-entry prediction 103

5.1. State of the art 104
5.1.1. Background 104
5.1.2. Dynamical modelling standards 105
5.1.3. Re-entry time window predictions 106
5.1.4. Monte Carlo-based methods 106
5.2. Test cases 107
5.3. Methodology 109
5.3.1. State-space definition 109
5.3.2. Re-entry event definition 110
5.3.3. High-fidelity crude Monte Carlo 110
5.3.4. Stochastic re-entry analysis 111
5.3.5. Multifidelity Monte Carlo 114
5.4. Results and discussion 116
5.4.1. Propagation models 118
5.4.2. Roseycubesat-1 118
5.4.3. GOCE 122

5.5. Concluding remarks 125

6 Conclusions and future work 127

6.1. Summary and conclusions. 127

6.2. Future work. 130

6.2.1. Orbital collision risk assessment 130

6.2.2. Spacecraft re-entry prediction. 133

Bibliography 134

LIST OF FIGURES

1.1 The change in RSO number in the period 1957-2025 published in ESA's 2025 space environment report 3

1.2 The historical trend of numbers of fragments for the fragmentation events in the period 1957-2025, published in ESA's 2025 space environment report. 4

1.3 The evolution of the number of re-entering objects sorted by object type in the period 1957-2025 published in ESA's 2025 space environment report. 6

2.1 The Keplerian orbital elements, which describe elliptical orbits of objects in space. 13

3.1 Illustration of the two variables, Γ_c and Δ_c comprising the collision metric used to define the dimensionality reduction mapping. 32

3.2 The euclidean distance between two spacecraft is plotted as a function of time for a period of four days, with different integration step-sizes. 33

3.3 The evolution of the two variables Γ and Δ, plotted for an object with a simulation time-span of four days. (left). The figure on the right shows the magnitude of $||\boldsymbol{\xi}||$ as it evolves through time. 34

3.4 Correlation plot of $\mathbb{R}^{12}$ to $\mathbb{R}^2$ for 215 samples drawn from $\pi_0(\boldsymbol{x}_0)$. 36

3.5 Conjunction mapping showing the defined 2D metric for the proposed model samples in LEO. 47

3.6 a) The HF mapping of the samples predicted by the proposed method to collide. b) The HF mapping of the samples predicted by the proposed method to not collide. 48

3.7 Confusion matrix outlining the number of true positives, false positives, true negatives and false negatives obtained with Algorithm 1. 48

3.8 States of object A at t_0' drawn from the proposal distribution (blue), and the subset of these which end up in collision at TCA (red). 49

3.9 States of object B at t'_0 drawn from the proposal distribution (blue), and the subset of these which end up in collision at TCA (red). 50

3.10 Conjunction mapping showing the 2D metric for the proposed model samples in GEO. 51

3.11 a) The HF mapping of the samples predicted by the method to collide. b) The HF mapping of the samples predicted by the method to not collide. . 52

3.12 Confusion matrix outlining the ratio of true positives, false positives, true negatives and false negatives between the predicted and the HF values. . . 53

3.13 States of object C at t'_0 drawn from the proposal distribution (blue), and the subset of these which end up in collision at TCA (red). 53

3.14 States of object D at t'_0 drawn from the proposal distribution (blue), and the subset of these which end up in collision at TCA (red). 54

4.1 Hierarchy of the main types of filters used in OD and RSO tracking. The main groups are batch and sequential methods, each spanning their subtypes (gray boxes). 62

4.2 The CKF of degree 3, in which $d_{ckf} = 2d$ cubature points are drawn around a mean value, $\hat{x}_k$, spanning the covariance C_k. 64

4.3 Position RMSE (km) against the number of samples for the LEO scenario, for the EnKF, EnPF, EnCKF and the BPF. 81

4.4 Position RMSE (km) against minimum observation separation of the simulated observation set in LEO. 82

4.5 Position RMSE (km) and standard deviation of samples (shaded region) for the CKF. Simulation of five days. 83

4.6 Position RMSE (km) and standard deviation of samples (shaded region) for the EnKF. Simulation of five days. 84

4.7 Position RMSE (km) and standard deviation of samples (shaded region) for the EnCKF. Simulation of five days. 84

4.8 Position RMSE (km) and standard deviation of samples (shaded region) for the BPF. Simulation of five days. 84

4.9 Position RMSE (km) and standard deviation of samples (shaded region) for the EnPF. Simulation of five days. 85

4.10 Position RMSE (km) against the number of samples for the GEO scenario, for the EnKF, EnPF, EnCKF and the BPF. 86

4.11 Position RMSE (km) against minimum observation time separation of the simulated observation set in GEO. 87

4.12 Position RMSE (km) and standard deviation of samples (shaded region) for the CKF. Simulation of five days. 88

4.13 Position RMSE (km) and standard deviation of samples (shaded region) for the EnKF. Simulation of five days. 88

4.14 Position RMSE (km) and standard deviation of samples (shaded region) for the EnCKF. Simulation of five days. 89

4.15 Position RMSE (km) and standard deviation of samples (shaded region) for the BPF. Simulation of five days. 89

4.16 Position RMSE (km) and standard deviation of samples (shaded region) for the EnPF. Simulation of five days. 89

4.17 Parameter estimates shown in solid lines for the EnKFup (blue), the EnPFup (red) and the NHF (green), and nominal values (dotted line). 93

4.18 Estimated posterior pdf of σ_W for the EnKFup (top), the EnPFup (middle) and the NHF (bottom) at two different time-steps. 94

4.19 RMSEs in position for the three proposed algorithms which estimate the given σ_W. 94

4.20 RMSEs in velocity for the three proposed algorithms which estimate the given σ_W. 95

4.21 RMSEs in position for the three proposed algorithms which adjust the process noise magnitude, σ_W to account for the difference between the reference orbit HF model and the LF model used by the filters. 98

4.22 RMSEs in velocity for the three proposed algorithms which adjust the process noise magnitude, σ_W, to account for the difference between the reference orbit HF model and the LF model used by the filters. 98

4.23 The RTN components of the $\bar{\sigma}_W$ estimated by the EnKFup (in blue), the EnPFup (in red) and the NHF (in green) algorithms, in the presence of a simplified dynamical model. 100

5.1 a) The re-entry of (GOCE) over the Falkland islands on 11 November 2013. b) Space debris that landed in the Kenyan village of Mukuku, on 12 January 2025. 104

5.2 Deterministic histogram of re-entry times for the GOCE scenario (in blue), and the real re-entry time (shown as a thick black line). 112

5.3 A set of histograms showing the different stochastic diffusion coefficient scales and their corresponding window sizes. 113

5.4 The variation of window sizes (in days) with respect to σ_W. The starting point, the ending point, and the point at which window size begins to change significantly are singled out. 114

5.5 The mean altitude of the collection of high-fidelity samples used in the MFMC algorithm is shown in this figure. 119

5.6 MFMC and CMC estimators for the mean re-entry time for Roseycubesat-1, for different computational budgets, along with their 3σ values. 120

5.7 The standard deviations of the MFMC estimator and CMC estimator for different computational budgets. The MFMC's $B_c = 1,000$ sec is chosen as reference for the variance equivalent CMC budget. 121

5.8 Multifidelity Monte Carlo (MFMC) re-entry time window shown in cyan for Roseycubesat-1, bounded by the estimators $\hat{s}^{lb}$ and $\hat{s}^{ub}$. The MFMC's computational budget was $B_c = 1,000$ sec. The figure also shows the average estimate $\hat{s}^{avg}$ (red dotted line), and a CMC simulation of $B_c = 2,500$ sec (green histogram), and the ±20% RSL recommended in the literature. 122

5.9 MFMC and CMC estimators for the mean target time (last available measurement) for GOCE, for different computational budgets. 123

5.10 The standard deviations of the MFMC estimator and CMC estimator for different computational budgets. 124

5.11 MFMC re-entry time window for GOCE, bounded by $\hat{s}^{lb}$ and $\hat{s}^{ub}$. Also shown is the average estimate $\hat{s}^{avg}$ and an independent CMC simulation. . 125

6.1 Sequential construction of IS collision proposal distribution by way of resampling, using stochastic propagation. 133

6.2 Comparison of the 6 orbital elements retrieved from TLE data (in blue) and from a HF propagation of the initial conditions (in red). 135

6.3 Histogram of re-entry times obtained by means of a HF deterministic CMC simulation of 1,000 samples, and the real re-entry time. 135

LIST OF TABLES

3.1 Physical properties of the spacecraft in the LEO conjunction case. 44

3.2 GEO orbital parameters 45

3.3 Performance comparison for LEO conjunction: PoC estimates, run-time and standard deviation of the PoC estimator (σ). 46

3.4 Performance comparison for the GEO conjunction: PoC estimates, run-time and standard deviation of the PoC estimator (σ). 51

4.1 Methods for noise characterisation. 61

4.2 Physical properties of object C 79

4.3 Physical properties of object D 79

4.4 Estimation RMSE for the five algorithms applied to the LEO scenario. . . 83

4.5 Estimation RMSE for the five algorithms applied to the GEO scenario. . . 87

4.6 Physical properties of object E 91

4.7 Performance comparison for the three algorithms (RMSE and computational cost) in the case of nominal parameters. 96

4.8 Prediction errors for the three algorithms (RMSE and computational cost) in the case of nominal parameters. 97

4.9 Performance comparison for the three algorithms in the case where simplified dynamics are used. 99

4.10 Prediction errors for the three algorithms (RMSE and computational cost) in the case where simplified dynamics are used. 99

4.11 Performance comparison for the three algorithms (RMSE and computational cost) in the case where three σ_W components are estimated. 100

5.1 Physical properties of the decaying Roseycubesat-1 spacecraft 108

5.2 Physical properties of the decaying GOCE spacecraft. 109

5.3 Performance comparison of MFMC and CMC on the Roseycubesat-1 test case, in terms of computational cost (in seconds) and standard deviation when estimating reentry time (in days since t_0). 120

5.4 Performance comparison of MFMC and CMC on the GOCE test case. . . 124

6.1 Performance comparison for LEO conjunction using stochastic framework: PoC estimates and run-time. 133

ACRONYMS AND ABBREVIATIONS

Acronym	Definition
STM	space traffic management
SSA	space situational awareness
SST	space surveillance and tracking
ECI	Earth-centred inertial coordinate frame
RTN	radial-transverse-normal coordinate frame
UTC	universal time coordinated
CCSDS	consultative committee for space data systems
GPS	global positioning system
RSO	resident space object
ISS	international space station
MSE	mean squared error
RMSE	root mean squared error
LEO	low-Earth orbit
MEO	medium-Earth Orbit
GEO	geostationary Earth orbit
SRP	solar radiation pressure
TLE	two-line element
RAAN	right ascension of the ascending node
RA	right ascension
DEC	declination
pmf	probability mass function
pdf	probability density function
r.v.	random variable
d.o.f.	degrees of freedom
CDF	cumulative density function
QoI	quantity of interest
CMC	crude Monte Carlo
IS	importance sampling
BFIS	bi-fidelity importance sampling

MFMC	multifidelity Monte Carlo
LLN	law of large numbers
i.i.d	independent identically distributed
PoC	probability of collision
ODE	ordinary differential equation
SDE	stochastic differential equation
PDE	partial differential equation
RK4	Runge-Kutta of order 4
SRK	stochastic Runge-Kutta
TCA	time of closest approach
CAM	collision avoidance maneuver
ASAT	anti-satellite missile test
OD	orbit determination
IOD	initial orbit determination
CMM	covariance matching method
MPE	measurement prediction error
LSQ	least-squares filter
NBLS	nonlinear batch least-squares filter
KF	Kalman filter
EKF	extended Kalman filter
UKF	unscented Kalman filter
CKF	cubature Kalman filter
SCKF	square-root cubature Kalman filter
EnKF	ensemble Kalman filter
EnCKF	ensemble cubature Kalman filter
GMM	Gaussian mixture model
BSF	batch-sequential filter
PF	particle filter
BPF	bootstrap particle filter
EnPF	ensemble particle filter
NPF	nested particle filter
NHF	nested hybrid filtr
SVD	singular value decomposition
HBR	hard-body radius
HF	high-fidelity
LF	low-fidelity
RSL	remaining spacecraft lifetime

LIST OF SYMBOLS

Symbol	**Description**
	General
d	dimension
$\mathbb{M}$	mass
$\mathcal{R}$	3D region of space
$\mathbb{R}^d$	d-dimensional real-valued variable space
ϱ	pmf
$p(\cdot)$	pdf
$\pi_0(\cdot)$	initial pdf
$\mu(\cdot)$	proposal pdf
$P(\varepsilon)$	probability of event ε
$\boldsymbol{x}$	state random variable
$\boldsymbol{y}$	predicted observation random variable
$\boldsymbol{z}$	observation
t	continuous time index
k	observation discrete time index
l	integration discrete time index
$\boldsymbol{T}$	vector of observation time-stamps
$\boldsymbol{T}'$	vector of integration time-stamps
σ_W	stochastic diffusion coefficient
σ	standard deviation
Σ_0	initial covariance matrix
$\boldsymbol{W}$	Wiener process
$\boldsymbol{r}$	position vector
$\mathbf{v}$	velocity vector
$\boldsymbol{\theta}$	parameter vector
M	number of observations
L	number of numerical integration time-steps
N	number of samples
i / j	sample index

$\boldsymbol{\upsilon}$	process noise vector
$\boldsymbol{Q}$	process noise covariance matrix
$\boldsymbol{s}$	measurement noise vector
Ω	measurement noise covariance matrix
L_k^i	Likelihood of sample i at time k
w	sample weight
h	numerical integration step-size
F_l	RK4 intermediate integration step dynamical model evaluation
	Functions and thresholds
$\mathbb{E}[\cdot]$	expectation function
$g(\cdot)$	general integrable test function
$f(\cdot)$	drift function, dynamical model evaluation
$\mathcal{M}(\cdot)$	measurement function
$\mathbb{I}$	indicator function
$\Phi(\cdot)$	propagation function
$\Xi(\cdot)$	$\mathbb{R}^{12} \to \mathbb{R}^2$ function
$\tau(\cdot, t)$	re-entry time computation function
φ	resampling threshold
γ	collision HBR threshold in $\mathbb{R}^{12}$
γ'	collision HBR threshold in $\mathbb{R}^2$
β	altitude threshold
	Orbital dynamics
$\check{\mu}$	gravitational parameter
α	semi-major axis
e	eccentricity
i	inclination
ω	argument of perigee
Ω	longitude of the ascending node
v	true anomaly
$\mathcal{T}$	orbital period
$\boldsymbol{F}$	gravitational force
$G_\oplus$	Earth gravitational constant
R_E	Earth radius
$\boldsymbol{a}$	acceleration vector
Bc	ballistic coefficient
$\check{\rho}$	atmospheric density
D	Drag force
C_D	drag coefficient
C_R	solar radiation pressure coefficient

$\mathcal{H}$	angular momentum
ζ	angle between eccentricity and angular momentum vectors
n_q	instantaneous angular velocity
U_n	line of nodes
	Tracking
N_1	number of samples, first layer of NHF
N_2	number of samples, second layer of NHF
$\boldsymbol{C}_k$	Gaussian covariance matrix at time index k
K_G	Kalman gain
$\boldsymbol{S}_k$	factorised Gaussian covariance matrix
$\boldsymbol{C}_{xy}$	cross covariance matrix
d_{ckf}	number of cubature samples
λ	secondary layer sample weight in NHF
κ	jittering perturbation in NHF
χ	extended state vector
	PoC calculation
x_c	importance sampling proposal mean
x_{ref}	reference state for 2D linearisation
$\boldsymbol{C}_c$	importance sampling proposal covariance matrix
$\mathcal{C}_\gamma$	set of time instants of collision ($\mathbb{R}^{12}$)
$\bar{\mathcal{C}}'_\gamma$	set of time instants of collision ($\mathbb{R}^{2}$)
$\boldsymbol{\xi}$	2D conjunction metric state vector
ϵ	approximation error
$\boldsymbol{J}^{HF}$	high-fidelity Jacobian matrix
d_L	minimnum $\mathbb{R}^{12}$ distance
$\mathring{d}_c$	minimnum $\mathbb{R}^{2}$ distance
	Re-entry
$\mathcal{A}$	altitude of state
K	number of propagation models
$\boldsymbol{c}$	propagation model computational costs vector
$\boldsymbol{m}$	vector of sample numbers for MFMC
B_c	computational budget
$\mathcal{B}$	variance to be optimised
g_n	auxiliary optimisation vector in MFMC
$\boldsymbol{\rho}$	correlation vector
$\boldsymbol{Q}$	MFMC weight coefficients
$\hat{s}$	QoI to be estimated
n	propagation model index

1

INTRODUCTION

Into the night
They fall through the sky
No one should fly
Where eagles dare

Iron Maiden
Where Eagles Dare
(1983)

1.1 Motivation

On 27 June 2024, astronauts at the International Space Station (ISS) were forced to cease all activity and shelter in the Starliner spacecraft, which was docked. A Russian satellite had broken up in the proximity of the station, and a cloud of debris swarmed past the ISS, putting the lives of the astronauts at risk. At an altitude of around 440 km, even small orbital fragments can prove lethal to operational spacecraft, and certainly any crew on-board. Approximately one hour later, the astronauts were allowed to re-enter the space station [1]. It is worth noting that this was a last-resort scenario, as no collision avoidance manoeuver (CAM) was performed due to the short notice, meaning that the fates of the station, and of the humans on-board were in a way, left to chance.

The launch in October of 1957 of the first artificial satellite, the Sputnik I, marked the beginning of the space age, roughly 67 years ago at the time of writing. Homo Sapiens are thought to have been around for over 300,000 years [2]. The fraction of our timelime spent as a spacefaring civilisation is merely 0.02%. In such a short amount of time, our understanding about space has grown in unimaginable ways. The space race, which occurred during the cold war in the latter half of the 20th century, represented a show of force between two of the then-greatest superpowers, who had just unlocked the ability to use space at their will. This dynamic highlighted a rather obscure realisation: those who control space may control the world.

Fast-forward to 2025, and our species is responsible for five different interstellar spacecraft, two of which have already exited our solar system and are now in full-blown interstellar space, and over 250 interplanetary missions to the Moon, Mars, asteroids and beyond. But space, for better or worse, is not just for exploration, but for exploitation. In addition to the countless scientific, Earth-observation, and astronomical missions launched, space's most valued uses are those which satisfy our civilisation's needs for communication, security, and surveillance. There are more than 38,000 objects orbiting the Earth, as of March 2025 [3], out of which only around a third are operational spacecraft. The remainder represents long-defunct satellites, residual mission parts and rocket bodies, as well as pieces left behind by fragmentation events.

The scenario at the beginning of this chapter represents an extreme example of the impact of space debris, since it is usually possible to detect the future intrusion of a piece of debris into a spacecraft's 'keep out zone' and perform a well-planned debris avoidance manoeuver. However, while the problem is still manageable with the current space population, the tide is rapidly shifting. The solar cycle is responsible for high-orbit defunct spacecraft being dragged down to lower, more populated orbits, increasing congestion, but other phenomena have humans to be accountable for. Events such as the anti-satellite missile (ASAT) tests carried out by China and Russia in 2007 and 2022, respectively, together released over 4,000 fragments into low-Earth orbit (LEO) [4][5]. These events mark deliberate and inconsiderate acts of space misuse. Figure 1.1, taken from the European Space Agency's (ESA) 2025 space environment report [3] shows the increase in the number of RSOs since the beginning of space exploration, including a breakdown of the type of object and its respective proportion.

The deployment of internet infrastructure satellites by providers such as Starlink and OneWeb has brought about a sharp increase in spacecraft numbers in the last 10 years. Starlink is responsible for a fleet of around 7,000 operational spacecraft in LEO, which have been forced to perform over 75,000 CAMs between 2023 and 2024 [6][7], with conjunction data messages probably exceeding several hundreds of thousands. The advent of nanosatellite technology, bringing thousands of cubesats into LEO, together with the reduction in launch costs and the on-going endeavour to attain reusable thruster mechanisms can help paint a realistic picture of what the next 10 years may look like.

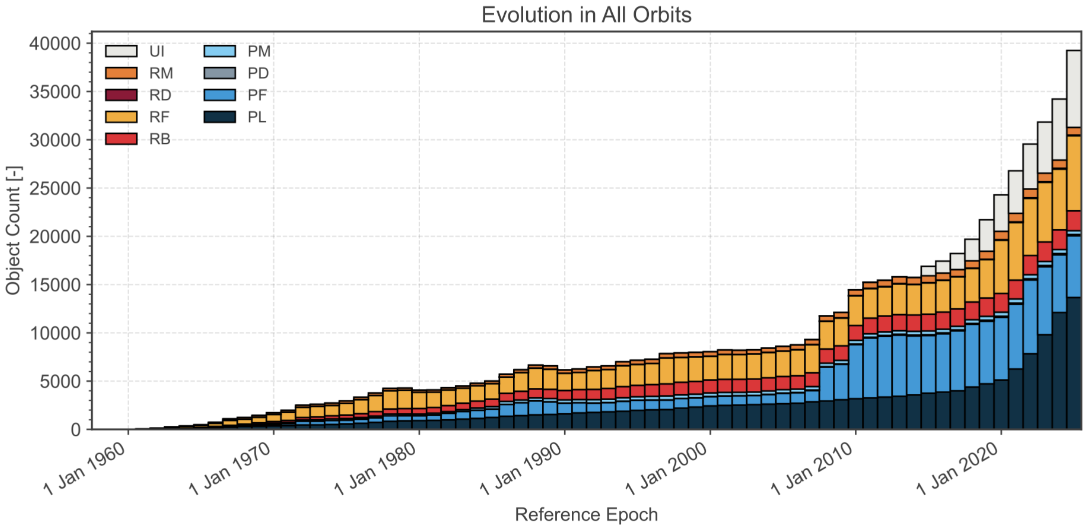


Figure 1.1: The change in the number of RSOs in the period 1957-2025 published in ESA's 2025 space environment report. UI = Unidentified, RM = Rocket mission related object, RD = Rocket debris, RF = Rocket fragmentation debris, RB = Rocket body, PM = Payload mission related object, PD = Payload debris, PF = Payload fragmentation debris, PL = Payload [3].

Space situational awareness (SSA) is the branch of space engineering which ensures that services being provided by different space players output the desired performance and promote safety in operations. These include space surveillance and tracking (SST) services, weather monitoring, and risk assessment of activities. The sub-branch of SSA which deals with the policies, rules and technology of ensuring space law and order is space traffic management (STM), which lays out the guidelines for organisations and agencies regarding navigation and safety thresholds for the preservation of the space population.

One in-orbit collision could be catastrophic in a myriad of ways. First and foremost, it would represent the complete loss of the two objects involved. Also pressing is the effect it would have on the surrounding space. Hours after collision, the debris cloud would spread throughout the entire orbital plane, with deviations in all directions, posing an imminent threat to other spacecraft in said orbital plane. In addition, due to solar cycle variability and its effect on the atmosphere, the orbits of these fragments may be lowered by varying drag levels, inducing further scattering within the cloud. The phenomenon by which collisions lead to a cascading effect of destruction is termed 'the Kessler syndrome', named after the scientist who theorised it [8]. It postulates that an increase in collision-derived fragments may give rise to further collisions and fragmentations, leading to an exponential rise in orbital threats to the space environment, and a point of no return may be reached.

Figure 1.2 shows the number of fragments associated with fragmentation events since the beginning of the space era. It is clear that the segment of the 2005-2010 bin is due to deliberate events such as those mentioned in the previous paragraphs. While the study of fragmentation events and the evolution of debris clouds is an important aspect of ensuring orbital safety post-collision, accurate methods to compute the risk of occurrence of collision-induced fragmentations are critical for the provision of collision warning and avoidance services.

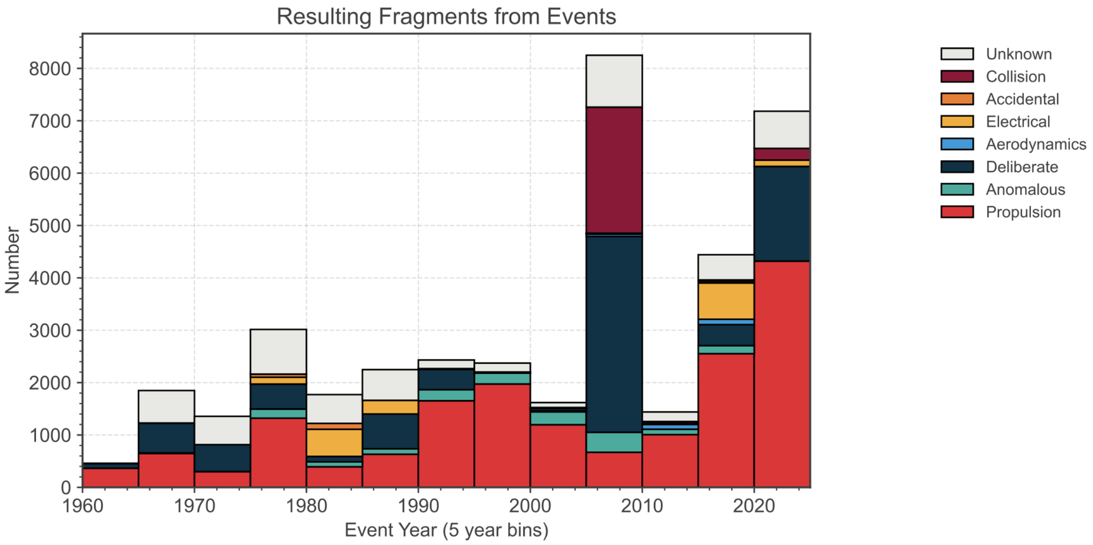


Figure 1.2: The historical trend of numbers of fragments for the fragmentation events in the period 1957-2025, published in ESA's 2025 space environment report [3].

In addition, it is desirable to avoid maneuvering unless the risk is significant, as a way to prevent satellite owners from wasting fuel on an otherwise preventable manoeuver. The ISS, for instance, has the luxury of being conservative in its CAM threshold, as there are humans on-board, and the increased costs associated with larger fuel consumption can be assumed by a cohort of agencies from multiple nations. Other spacecraft may want to aim to reduce the number of manoeuvers at all costs. This demands not only accurate, but efficient methodologies that can be run online or with a buffer of a few hours.

However, collision risk prediction methods inherently rely on the uncertainty of the system. Spacecraft follow trajectories which, in some cases, where sufficient available data is available, can be very well tracked. In these cases, the computation of the probability of collision may be a reliable metric to assess the need for a CAM. Other spacecraft may have less data or the data quality may be lower. In such cases, the probability computation metrics are subject to this uncertainty, and may become less reliable. Therefore, in order to prevent collisions, it is also necessary to improve tracking capabilities, by incorporating robust methods which can characterise large uncertainties and perform sufficiently well in practical scenarios. In addition, reliable tracking capabilities are necessary for the detection and vigilance of suspicious or malicious spacecraft whose intentions may be to harm critical infrastructure or, in the likely case of future space weaponisation, human populations. Another fundamental aspect of being able to track spacecraft is data association, the process by which a catalog ID number is assigned to objects and to which newly acquired observations are matched. This enables tracking to be carried out in the first place: by processing subsequent observations related to a specific object, but also by detecting manoeuvers. This last application can be crucial, as a maneuvered spacecraft will naturally appear with different observed coordinates, and in the absence of any publicly available manoeuver data, instead of classifying said observations as belonging to a newly spotted object, the challenge lies in correctly assigning it to the correct object.

The third main aspect of STM is spacecraft re-entry. Spacecraft nowadays have an associated useful lifetime, after which they are directed to decay into the atmosphere and undergo re-entry. Re-entry occurs when the altitude of a spacecraft reaches levels at which the drag forces are too large to overcome and bring the spacecraft deeper into the atmosphere, causing an exponential increase in drag force experienced and the subsequent burn-up of the object, as it plummets down to the surface. Depending on the object's composition and shape, usually, if the object is small enough, the burn-up phase culminates in complete disintegration, which sometimes gifts us with a shining spectacle in the night-sky: shooting stars. Other times, the object is too large and it inevitably makes its way to the surface. In reality, this is rarely an issue, since 70% of the planet is composed of oceans, and with vast empty lands around the globe, the probability of it affecting a given population is slim. Notwithstanding, incidents such as the re-entry of the purposefully de-orbited space station MIR in 2001, whose decay appeared in the skies over Fiji, in the Pacific ocean, prompted authorities to begin paying attention due to possible impact on local populations, and to create impact mitigation plans. These incidents are capable of causing considerable disruption in terrestrial life, and their extent of impact should be mitigated, or at least predicted to a sufficient degree. Figure 1.3 shows the number of re-entering objects in terms of the object type, since the beginning of the space era. A substantial increase can be clearly seen after the 2020 mark. Oftentimes, observations of a decaying spacecraft drop off as it enters deeper layers of the atmosphere. At that point, it is critical to be able to propagate the state reliably up to its terminus. Tied in with the tracking aspect of SST, propagation methods must be able to predict, with a realistic measure of certainty, the time and location of a possible re-entry.

The three focus points of this thesis are based on the three broad problems described. These may lie on different sides of the spectrum of SSA, but are inherently reliant on each other, and all share a common aspect: the enhancement of uncertainty treatment in astrodynamics.

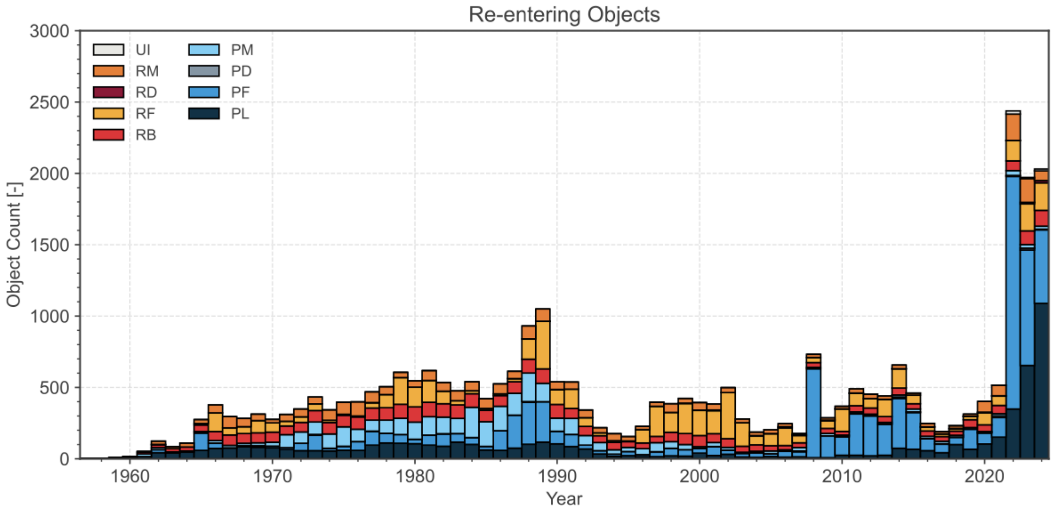


Figure 1.3: The evolution of the number of re-entering objects sorted by object type (without human spaceflight) in the period 1957-2025, published in ESA's 2025 space environment report. UI = Unidentified, RM = Rocket mission related object, RD = Rocket debris, RF = Rocket fragmentation debris, RB = Rocket body, PM = Payload mission related object, PD = Payload debris, PF = Payload fragmentation debris, PL = Payload [3].

1.2 Challenges

For applications such as tracking and re-entry predictions, one of the principal methodological components is the propagation of spacecraft states from the current time to the next. The challenge here, as well as in methods for the computation of collision probability, lies in the representation of uncertainty. To successfully perform tracking of an orbiting object or simply obtain accurate predictions of the state at later times, one must have an estimate of the uncertainty of the system, i.e., how much we can trust our predictions. The limitations in standard methodology to accomplish these tasks arise from the following problems:

Problem 1.1) Traditionally, orbital dynamics have been modelled as a deterministic system. This entails that given a set of initial conditions, and a final time of integration, the end result remains unchanged, i.e., there is no room for stochasticity in the dynamical process. The truth is that orbital dynamics are quasi-deterministic: largely deterministic, but with elements of randomness which are oftentimes unaccounted for [9]. Not computing the effect of imperfect dynamical modelling or unpredictable randomness in the intrinsic nature of the system is a hindrance to the correct estimation of spacecraft trajectories.

Problem 1.2) In relation to the previous point, the inclusion of process noise has only recently started gaining traction, and the methodologies to do so remain largely ad-hoc or arbitrary. There is therefore a general lack of proper noise modelling in RSO filtering applications (among others).

Problem 1.3) Tied to the absence of process noise in the dynamical model, for re-entries this translates to simplistic algorithmic policies when predicting the possible time ranges of re-entry events. This usually results in symmetrical and constant re-entry windows around estimates, which, although they are assumed to capture said events with a certain percentage of confidence, they actually represent blanket policies which produce borderline uninformative estimates of time windows.

Problem 2) Observations, if processed within the framework of a filtering algorithm, enable the refinement of predictions. The assumption of Gaussian state uncertainties simplifies the treatment of uncertainty during propagation. In reality, during a long-enough propagation without observations, depending on the orbital regime, the predicted uncertainty distribution deforms with time and begins to exhibit higher moments, which simply cannot be captured by a Gaussian distribution. This results in a gradual degradation of estimation quality.

The computation of the probability of collision (PoC) between RSOs is also subject to several challenges and problems, which are summarised below.

Problem 3) Collisions are extremely rare events for now, which means that there is little data to show the conditions for a collision. In addition, objects orbit the Earth at staggering speeds exceeding the 7 km/s mark. Two objects orbiting in opposite directions can exceed a relative speed of over 14 km/s. The aim is to follow an object with a characteristic orbital period of hours, but collision events occur in the millisecond range. This places an enormous demand on numerical integration resolution, since most collisions may be missed in between the time-steps of regular integration methods. The rarity of these events also makes the application of data-driven methodologies unfruitful and means that the methods must almost always be simulation-based and simulation-validated.

Problem 4) The vast majority of methods employ major simplifications of the dynamical process, which render these computations efficient. However, at times, this efficiency comes with the sacrifice of accuracy. In addition, state uncertainties at the time of close approach (TCA) between two spacecraft are often treated as Gaussian distributions, which once more, simplifies the problem, but may not reflect an accurate representation of the truth. Unfortunately, methods which bypass said assumptions are oftentimes associated to prohibitive computational costs.

1.3 Objectives

To overcome the problems and challenges outlined in the previous subsection, this thesis sets out to provide solutions which offer alternative ways of carrying out the relevant tasks in an efficient, yet accuracy-preserving way. To be specific, the methods introduced in this work are aimed at the following goals.

Objective 1) Propagation methods which treat uncertainty in a mathematically rigorous way are required to fully capture the stochastic behaviour of orbital dynamical states during propagation. To accomplish this, numerical integration methods designed to solve stochastic differential equations (SDEs) should be used instead of those for ordinary differential equations (ODEs). Such stochastic methods must necessarily include reasonable values for the magnitude of the dynamical model noise present, avoiding the use of ad-hoc solutions based purely on trial and error, such as arbitrarily inflated covariance matrices.

Objective 2) There is a need for methodologies which handle the highly nonlinear nature of orbital dynamics, and the presence of non-Gaussian uncertainties for long observation-free propagation. In essence, sample-based methodologies are chosen over pure Gaussian methods for the tracking of space objects, the refinement of which is an objective of this thesis.

Objective 3) Tied to the previous objective is the development of methodologies to provide reliable metrics for re-entry time window predictions. In order to bypass the static, symmetric windows for spacecraft re-entry, an objective is to implement in their place, window bounds obtained in an informed way through the inclusion of process noise in the propagation process, and efficient variations of sample-based methodologies, which are known for their ability to attain sufficient accuracy given a sufficient computational budget.

Objective 4) Methods which are able to bypass the need for assumptions and simplifications of the dynamical model or the computation of quantities of interest (QoI) are, as mentioned, sample-based. This typically entails simulations of large numbers of samples to effectively cover the target state space. However, these methods are computationally costly when studying rare events, and the implementation of efficient modifications to the traditional sample-based methods is one of the aims of this thesis, in particular, for the computation of the conditions associated with collisions.

Objective 5) Another objective is to study collisions in an alternative space free of the aforementioned curse of numerical integration resolution. This should enable the focus of computational power in the relevant parts of the integration (close to collision) and provide a means to evaluate the previously suggested methods in an efficient, reliable and intuitive manner.

1.4 Contributions

The contributions of this thesis aim to satisfy the objectives of Subsection 1.3 to the maximum extent. The addressal of ***objective*** 1 involves the representation of orbital dynamics as a stochastic system, given by an SDE. The drift function is the output of a function which evaluates the accelerations due to orbital dynamic forces, and a process noise term is included to account for random error and unmodelled accelerations. To dote the noise with an appropriate intensity, the magnitude of the lowest-order modelled acceleration is recorded and used as the scale of the stochastic diffusion term. This is the approach used for observation-free propagation, such as in the case of re-entry prediction (to satisfy ***objective*** 3), where the states are propagated towards the end of the decaying object's life.

However, for the implementation of tracking algorithms, one of the main contributions of the thesis is the active simultaneous estimation of the process noise scale of the orbital dynamics SDE, as well as state tracking. To this end, filtering algorithms that extend the state vector to include additional parameters (in this case the diffusion coefficient) are adopted, enabling us to track the state and characterise noise simultaneously and online. Moreover, the algorithms described constitute sample-based filtering methods which are capable of approximating the posterior state distribution without the need for Gaussian uncertainties at every time-step, enabling the computation of higher moments of the probability distribution. These are a mixture of particle filtering methods, which produce unevenly weighed samples to concentrate probability mass in certain regions of the state space depending on observations, and Gaussian methods, which provide a reliable and efficient mechanism to update samples, provided these already cover the state-space sufficiently. These contributions cover ***objective*** 2.

In order to provide a solution for ***objective*** 4 and ***objective*** 5, in the study of PoC calculations, an alternative metric to Euclidean distance is proposed, which reduces the dimensionality of the collision problem by computing just two variables. These two variables reflect the separation of the spacecraft at critical times (when at the line of nodes intersecting the two orbits). The evolution in this space enables the capture of collision or close-approach events more smoothly. The entire mapping is implemented with different variants of the crude Monte Carlo (CMC) methodology, aimed at achieving a similar order of accuracy (given by their bypassing of limiting assumptions), but with a fraction of the computational cost and an increase in reliability, given by their reduction in estimator variance.

In summary, the contributions of this thesis are the provision of alternative methodologies to appropriately and accurately tackle the modelling of uncertainty in tracking and re-entry prediction services, as well as estimating quantities of interest such as collision probability in an efficient, yet reliable manner. All of these aims have the overarching goal of by-passing limiting assumptions of Gaussianity, and reducing the computational cost of commonly employed tactics to do so, such as CMC.

1.5 Organisation of the thesis

This thesis is composed of three main areas of study; probability of collision computation, spacecraft tracking, and re-entry window predictions. Chapter 2 delves into the tools and concepts used throughout the thesis which are common to all three of the topics studied. The computation of the probability of collision is tackled in Chapter 3, by describing the proposed conjunction metric, in addition to an importance sampling-based method to improve computational efficiency compared to CMC. The development of filters for spacecraft tracking is studied in Chapter 4, which is itself divided into two sections: Section 4.2, which deals with the study of hybrid filters for the tracking of spacecraft states and Section 4.3, devoted to the study of filters which simultaneously track the state and unknown parameters. The final chapter, Chapter 5 is dedicated to the assessment of a multifidelity Monte Carlo approach to predict re-entry time windows of decaying spacecraft, and is followed by concluding remarks about all topics in Chapter 6.

2

TOOLS AND THEORY

[*Instrumental*]

Iron Maiden
Losfer Words
(1984)

This chapter is dedicated to the introduction of the tools and principles that are employed throughout the thesis and are common to the three main chapters. A description of the orbital dynamics models used throughout the document is provided, as well as suitable numerical integration schemes. In addition, brief primers on relevant concepts in estimation and probability theory are included to provide a self-contained academic report.

2.1 Notation

Throughout the report, we adhere to the following notation to ease readability. Regular-face letters are used to represent scalars (e.g., $x \in \mathbb{R}$), bold-face lower-case letters for column vectors (e.g., $\boldsymbol{x} = [\boldsymbol{r}^\top, \boldsymbol{v}^\top]^\top \in \mathbb{R}^d$), where d is the state dimension, and bold-face upper-case letters for matrices, for example

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \in \mathbb{R}^{2\times 2}.$$

For a vector $\boldsymbol{x} = [x_1, \ldots, x_d]^\top$, we denote its Euclidean norm as $x = ||\boldsymbol{x}|| = \sqrt{\sum_{i=1}^{d} x_i^2}$.

2.2 Orbital dynamical models

2.2.1. Keplerian model

A Keplerian orbit is the unperturbed locus of an object under the gravitational influence of another object. Let object A represent a large object, and object B, a significantly smaller one. Object B's orbit traces an elliptical path, with object A at a focus of the ellipse; this is the first of Kepler's laws. The second law states that as object B moves along the orbit, the line connecting the centre of object A to that of object B sweeps out equal areas in equal times, independent of the location of the object along the orbit. The third law states that the orbital period, $\mathcal{T}$, of object B is related to the length of the semi-major axis α according to $\mathcal{T}^2 \propto \alpha^3$.
These physical laws are a direct consequence of Newton's laws of gravitation. To complete our description, in every elliptical orbit around the Earth, two characteristic points can be identified: its perigee, where the object is closest to the Earth, and its apogee, where the object lies the furthest from the Earth [10].

In this thesis, the Earth-centred Cartesian inertial coordinate system (ECI) is primarily used, where the origin is the centre of the Earth, and the position of a spacecraft is represented in 3-dimensional space through r_x, r_y and r_z components. A spacecraft state may also be exactly described using the six Keplerian orbital elements, illustrated in Figure 2.1 for completeness. Note that the ascending node is the point at which a satellite ascending from the southern hemisphere to the northern hemisphere crosses a reference (equatorial) plane [11][12].

The six Keplerian orbital elements are given by the following:

- α is the semi-major axis of the ellipse, or longest diameter (not seen in Figure 2.1).

- e is the eccentricity of the ellipse. It is the deviation from circularity (also not seen in the figure).

- i is the inclination, or the angular distance of the orbital plane with respect to the equatorial plane.

- $\check{\Omega}$ is the the longitude of the ascending node, which is the clockwise angle from the principal direction to the ascending node, in the reference plane.

- ω is the the argument of perigee, which is the counterclockwise angle measured from the ascending node to the perigee along the orbit plane.

- ν is the true anomaly: the angular distance from perigee to the current position of the satellite, measured at the focus of the orbit (usually Earth), in the direction of orbital motion.

Sometimes, mean anomaly MA can be used instead of true anomaly ν. It is an angular measure of time (0 to 2π over one period), assuming constant angular motion as if the body were moving in a circular orbit with the same period.

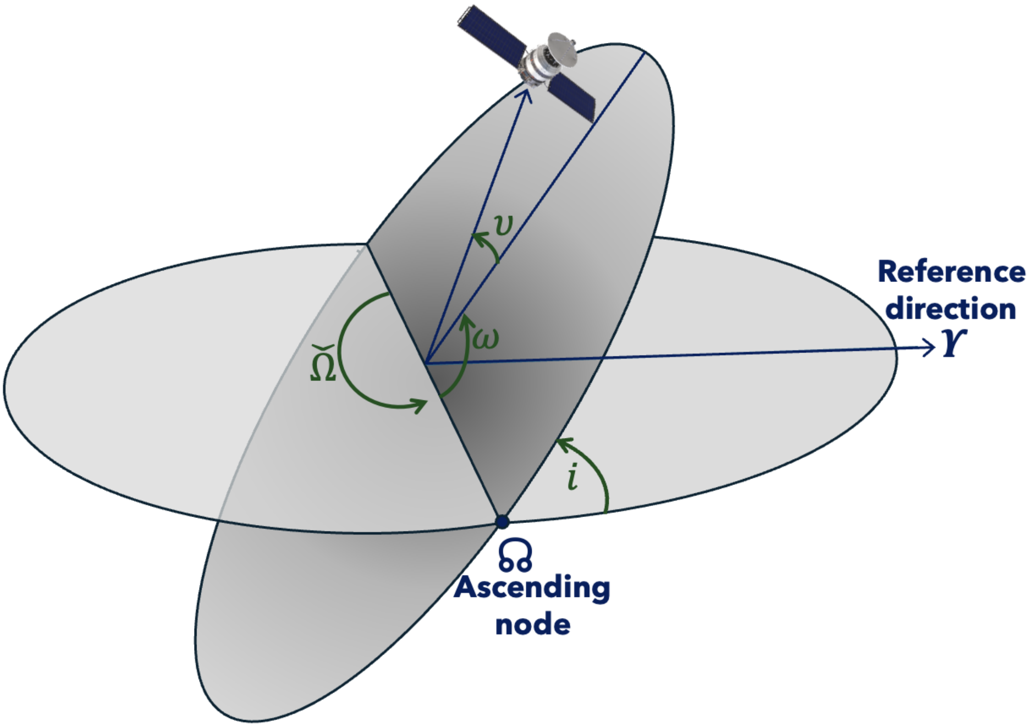


Figure 2.1: The Keplerian orbital elements, which, along with the eccentricity and semi-major axis, fully describe elliptical orbits of objects in space.

The primary force involved in the Keplerian framework, in ECI coordinates, is given by Newton's universal law of gravitation

$$\boldsymbol{F}_{A,B} = \frac{G_{\oplus}\mathbb{M}_A\mathbb{M}_B}{r^2}\frac{\boldsymbol{r}}{r}, \tag{2.1}$$

where $\boldsymbol{F}_{A,B}$ is the gravitational force between the two objects, $G_{\oplus}$ is the universal gravitational constant, $\mathbb{M}_A$ is the mass of the larger body (in this case, the Earth), $\mathbb{M}_B$ is the mass of the smaller body (in this case, the satellite), $\boldsymbol{r}$ is the position vector of object B, and r is the magnitude of the distance between the two bodies' centres of mass, (or the position vector). In this inertial reference frame, the relative acceleration between body A

and body B is given by

$$\begin{aligned}\boldsymbol{a} &= \boldsymbol{a}_B - \boldsymbol{a}_A \\ &= \frac{-G_{\oplus}(\mathbb{M}_A + \mathbb{M}_B)\,\boldsymbol{r}}{r^3} \\ &= -\frac{\breve{\mu}\,\boldsymbol{r}}{r^3},\end{aligned} \tag{2.2}$$

where $\boldsymbol{a}_A$ and $\boldsymbol{a}_B$ are the acceleration vectors of the Earth and the object, respectively, and $\breve{\mu} = G_{\oplus}(\mathbb{M}_A + \mathbb{M}_B)$. In the problems of interest in this work, $\mathbb{M}_B$ is much smaller than $\mathbb{M}_A$, hence $\breve{\mu}$ can be approximated as $\breve{\mu} \approx G_{\oplus}\mathbb{M}_A$.

2.2.2. Cowell formulation

Cowell's formulation is a way of representing each of the perturbing forces acting on a satellite at a given time [13]. Eq. (2.2) describes the gravitational acceleration in the Keplerian scenario. To describe orbital dynamics in a realistic scenario, additional forces must be modelled. Perturbations constitute forces (with their associated accelerations) that affect $\boldsymbol{a}$. Let $\boldsymbol{a}_{\text{pert.}}$ represent a combination of all the accelerations caused by the relevant perturbations. Then, Eq (2.2) becomes

$$\boldsymbol{a} = -\frac{\breve{\mu}\,\boldsymbol{r}}{r^3} + \boldsymbol{a}_{\text{pert}}. \tag{2.3}$$

Cowell's method provides a direct way to recast the second-order differential Eq. (2.3) as one of first-order, by defining

$$\mathrm{v} = \frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} \quad \text{and} \quad \frac{\mathrm{d}^2\boldsymbol{r}}{\mathrm{d}t^2} \equiv f(\boldsymbol{x}, t),$$

where $f(\boldsymbol{x}, t)$ is a function $f : \mathbb{R}^d \times \mathbb{R} \longrightarrow \mathbb{R}^d$. Then the following first-order system can be defined

$$f(\boldsymbol{x}, t) \equiv \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\boldsymbol{r}\\ \boldsymbol{v}\end{bmatrix} = \begin{bmatrix}\boldsymbol{v}\\ -\frac{\breve{\mu}r}{r^3} + \boldsymbol{a}_{\text{pert}}\end{bmatrix}. \tag{2.4}$$

RSOs may exist in LEO, MEO and GEO, among others. In LEO the main source of uncertainty in the modelling arises due to atmospheric drag, and this is explained by the difficulty in accurately modelling atmospheric dynamics and density as a function of time and space [14]. In GEO, on the other hand, solar radiation pressure (SRP) represents the main perturbing effect. The main constitutive perturbing components of the acceleration term are described in the following.

Gravitational perturbation Earth's gravitational potential $\check{U}$ (included merely for completeness) can be expressed as

$$\begin{aligned}\check{U}(r,\check{\phi}) = \frac{\check{\mu}}{r}\Bigg\{1 - &\sum_{l=2}^{\infty}\left(\frac{R_{\oplus}}{r}\right)^{l} J_l P_l(\sin\check{\phi}) \\ &+ \sum_{l=2}^{\infty}\sum_{m=1}^{l}\left(\frac{R_{\oplus}}{r}\right)^{l} P_{l,m}(\sin\check{\phi}) \\ &\times\Big[C_{l,m}\cos{(m\check{\lambda})} + S_{l,m}\sin{(m\check{\lambda})}\Big]\Bigg\} \qquad (2.5)\end{aligned}$$

where $\check{\phi}$ is the latitude, $\check{\lambda}$ is the longitude, P_l and P_{lm} are Legendre functions, C_l and S_{l_m} are related to tesseral and sectorial harmonics and J_l are zonal harmonics coefficients [15]. Zonal harmonics are the main contributors to the gravitational potential. The perturbed acceleration due to Earth's oblateness (J_2) is given by

$$a_{J_2} = \frac{3}{2}\frac{\check{\mu}\;J_2 R_{\oplus}^2}{r^4}\left[\frac{x}{r}\left(5\frac{z^2}{r^2} - 1\right)i \;+\; \frac{y}{r}\left(5\frac{z^2}{r^2} - 1\right)j \;+\; \frac{z}{r}\left(5\frac{z^2}{r^2} - 3\right)k\right]. \qquad (2.6)$$

Atmospheric drag Atmospheric drag has a significant effect at low altitudes and is essentially due to air friction, which opposes the object's motion. It is given by

$$D \;=\; \frac{1}{2}\check{\rho}\;\|\boldsymbol{v}_{rel}\|^2 C_D \mathbb{A}, \qquad (2.7)$$

where D is the force due to drag, $\check{\rho}$ is the atmospheric density, $\boldsymbol{v}_{rel}$ is the relative velocity vector between the satellite and the atmosphere, C_D is the drag coefficient, and $\mathbb{A}$ is the cross-sectional area in the direction of velocity. The acceleration due to this force is

$$a_{\text{drag}} \;=\; \frac{1}{2}\check{\rho}\;\boldsymbol{v}_{\text{rel}}{}^2\left(\frac{C_D\mathbb{A}}{\mathbb{M}}\right) = \;\frac{1}{2}\check{\rho}\;B_c\boldsymbol{v}_{\text{rel}}{}^2, \qquad (2.8)$$

where $B_c = \frac{C_D\mathbb{A}}{\mathbb{M}}$ is the ballistic coefficient.

Solar radiation pressure At high altitudes, due to lower levels of protection from solar rays, photons may strike an object's surface and transfer significant impulse, which may affect its momentum. This leads to small changes in the orbit's eccentricity and longitude of perigee. The overall extent of the effect depends on the satellite's mass and surface area. This acceleration is given by

$$a_{\text{SRP}} \;=\; -\check{\rho}_{SRP}\left(\frac{C_R\mathbb{A}_{\text{sun}}}{\mathbb{M}}\right)\frac{\boldsymbol{r}_{ss}}{\|\boldsymbol{r}_{ss}\|}, \qquad (2.9)$$

where $\check{\rho}_{SRP}$ is the incoming solar pressure, derived from the solar flux, 1370 W/m^2, C_R is the reflectivity coefficient, which depends on the reflective and absorptive properties of the material, $\mathbb{A}_{\text{sun}}$ is the cross-sectional area with respect to the Sun (usually not the same as the one for drag) and $\boldsymbol{r}_{ss}$ is the position vector of the spacecraft with respect to the Sun.

2.2.3. Observations

In the context of spacecraft tracking, available data may come in the form of radar, optical, or any other sensor measurements depending on the orbital regime. In LEO, observations may be acquired by ground-based radar sensors and are typically comprised of range, range rate, azimuth and elevation datasets. In GEO, observations typically come from ground-based telescope networks and are comprised of right ascension (RA) and declination (DEC) readings.

High accuracy in orbital state estimation requires precise measurement models that account for geometric and dynamic factors and corrections to incorporate environmental effects. For the time being, we assume a general (abstract) model in which an observation z at time t is represented as

$$z(t) = \mathcal{M}(\boldsymbol{x}, t) + \boldsymbol{s}(t), \tag{2.10}$$

where $\mathcal{M}(\cdot, t)$ is the transfer function of the sensor(s) available at time t and $\boldsymbol{s}(t)$ is the observational noise vector, which we model as Gaussian, with zero mean and covariance matrix $\Omega(t)$, i.e., $\boldsymbol{s}(t) \sim \mathcal{N}(0, \Omega(t))$.

2.3 Probability theory

In science and engineering, a distinction exists between theoretical and experimental results. Precise measurements and accurate dynamical models can yield close approximations to theoretical predictions. However, inherent randomness in the measured quantities and system parameters introduces deviations from the expected outcomes [16]. It remains crucial to recognise that uncertainty is an ever-present factor in engineering applications, requiring the development of robust strategies to manage it.

A random variable (r.v.) is a measurable map from a probability space to the space of real numbers. Let $(\mathcal{G}, \mathcal{F}, \mathbb{P})$ be a probability space, where $\mathcal{G}$ is the sample space, $\mathcal{F}$ is a sigma-algebra of subsets of $\mathcal{G}$, and $P : \mathcal{F} \to [0, 1]$ is a probability measure [17]. Then, an r.v. is a function $X : \mathcal{G} \to \mathbb{R}$, such that X is measurable. Let $\mathcal{E}$ denote some condition on X (e.g., $X > 0$ or $-1 < X < 1$). For conciseness, we write $P(\mathcal{E})$ to denote the probability of the event $\{\omega \in \mathcal{G} : \mathcal{E}(\omega) \text{ is true}\} \in \mathcal{F}$.

A probability distribution is a general term that describes how probabilities are assigned to different possible values of an r.v. If the r.v. X is discrete, the probability distribution is given by a probability mass function (pmf) $\varrho_X(x)$, such that

$$P(X = x_i) = \varrho_i, \ \sum_i \varrho_i = 1.$$

If the r.v. X is continuous, it is given as a cumulative distribution function (CDF), which provides the probability that X takes a value less than or equal to x.

$$\mathrm{CDF}(x) = P(X \leq x) = \int_{-\infty}^{x} p_X(x)dt,$$

where $p_X(x)$ is the probability density function (pdf) of X. Continuous r.v.s have an associated pdf which describes how the probability is distributed over a range of values and is given by

$$p_X(x) = \frac{\mathrm{d}}{\mathrm{d}x}\mathrm{CDF}_X(x).$$

Note that from now on, we denote $p_X(x)$ by $p(x)$ and $\varrho_X(x)$ by $\varrho(x)$ to simplify notation. This notation is argument-wise, so for r.v.s X and Y, with their corresponding (and possibly different) pdfs $p(x)$ and $p(y)$, their joint pdf is denoted $p(x, y)$ and the conditional pdf of X, given $Y = y$ is given by $p(x \mid y)$.

Pdfs can be fully described by their moments, which characterise and scale the shape of the pdf. The four most commonly used moments are the mean, variance, skewness and kurtosis [9]. The mean, or expectation, is a measure of the central location of a probability distribution (e.g., the average value), while the variance is the weighted average of the squared deviations from the expected value. An important characteristic of Gaussian pdfs, for example, is that they can be completely described in terms of their first and second moments. Other types of distributions include mixtures, uniform and exponential distributions. These may require additional or alternative ways to represent their shapes. The third statistical moment is the skewness, and is a measure of asymmetry in the distribution, while the fourth moment, kurtosis, is a measure of the weight of the tails in the distribution [18].

If X is discrete, its expectation is

$$\mathbb{E}[X] = \sum_{x} x\varrho(x),$$

where $\varrho(x)$ is its pmf. If X is continuous, its expectation is

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} xp(x)dx,$$

where $p(x)$ is its pdf.

A common assumption in dynamical systems is that uncertainties in r.v.s can be represented as Gaussian distributions. The Gaussian (or normal) distribution with mean $\hat{x}$ and covariance matrix Σ is denoted $\mathcal{N}(\hat{x}, \Sigma)$. The Gaussian pdf for a d-dimensional r.v. is given by

$$p\,(x) \;=\; \frac{1}{\left(\sqrt{(2\pi|\Sigma|)^{d}}\right)} \exp\left[-\frac{1}{2}(x - \hat{x})^{\top}\Sigma^{-1}(x - \hat{x})\right], \tag{2.11}$$

where $\hat{x}$ is the mean, $|\Sigma|$ is the determinant of the covariance matrix Σ and d is the state dimension.

As an example, for multi-dimensional r.v.s, the mean is given by

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \boldsymbol{x}\ p(x_1, x_2, \dots, x_d) \mathrm{d}x_1 \mathrm{d}x_2 \dots \mathrm{d}x_d,$$

while the covariance matrix is given by

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{x_1}^2 & \mathrm{Cov}(x_1, x_2) & \mathrm{Cov}(x_1, x_3) & \dots & \mathrm{Cov}(x_1, x_d) \\ \mathrm{Cov}(x_2, x_1) & \sigma_{x_2}^2 & \mathrm{Cov}(x_2, x_3) & \dots & \mathrm{Cov}(x_2, x_d) \\ \mathrm{Cov}(x_3, x_1) & \mathrm{Cov}(x_3, x_2) & \sigma_{x_3}^2 & \dots & \mathrm{Cov}(x_3, x_d) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(x_d, x_1) & \mathrm{Cov}(x_d, x_2) & \mathrm{Cov}(x_d, x_3) & \dots & \sigma_{x_d}^2 \end{bmatrix},$$

where $\mathrm{Cov}(x_i, x_j) = \mathbb{E}[(x_i - \hat{x}_i)(x_j - \hat{x}_j)]$.

Covariance matrices are symmetric and positive-semidefinite, and their diagonal elements are the variances of each element, i.e $\boldsymbol{\sigma} = \sqrt{\mathrm{diag}(\boldsymbol{\Sigma})}$, where $\boldsymbol{\sigma}$ is the standard deviation vector. Gaussian r.v.s are said to be statistically independent if their covariance matrix contains only diagonal entries, with the off-diagonals being equal to zero (i.e., they are uncorrelated) [19].

2.4 Numerical integration

2.4.1. Deterministic integration

Traditionally, orbital dynamics have been modelled as a deterministic system of equations of motion. The dynamical model is given by the ordinary differential equation (ODE) in Eq. (2.4), where $f(\boldsymbol{x}, t)$ is the dynamical model evaluation of $\boldsymbol{x}$ at t, $-\frac{\check{\mu}\boldsymbol{r}}{r^3}$ represents Keplerian motion, and $\boldsymbol{a}_{\mathrm{pert}}$ represents perturbative accelerations such as J_2 effects, drag, or SRP. In this thesis, $f(\boldsymbol{x}, t)$ is evaluated by simply computing Cowell's formulation of the desired forces numerically, or by implementing the high precision orbit propagator (HPOP) library in MATLAB [20], which takes into account the following accelerations

- atmospheric drag, following either the Jacchia-Bowman 2008 drag model, the USSA76 model or the NRLMSISE-00 model,
- solar radiation pressure,
- the GGM03C model for Earth gravitational potential up to degree and order (70,70),
- Luni-Solar perturbations,
- other planets' gravitational influence,

- solid Earth tides,
- relativity perturbations.

HPOP is a variable-fidelity dynamical model, as any or all of the forces mentioned may be switched on or off, depending on the required level of fidelity. In order to compute spacecraft trajectories, forward propagation must be carried out. Assume a time grid $\boldsymbol{T}' = \{t'_l\}_{l=0}^{L}$, where t'_0 is the initial time and $t'_1, \ldots, t'_L$ are time instants at which the value of the state $\boldsymbol{x}(t'_l)$ must be approximated. Numerical schemes enable us to (approximately) sample the state at any given time instants, i.e., they determine the conditional probability density function (pdf) of $\tilde{\boldsymbol{x}}_l \approx \boldsymbol{x}(t'_l)$ given $\tilde{\boldsymbol{x}}_{l-1} \approx \boldsymbol{x}(t'_{l-1})$, which is denoted $p(\tilde{\boldsymbol{x}}_l|\tilde{\boldsymbol{x}}_{l-1})$. Moreover, for any $m > 0$ the conditional pdf $p(\tilde{\boldsymbol{x}}_{l+m}|\tilde{\boldsymbol{x}}_l)$ can be expressed as

$$p(\tilde{\boldsymbol{x}}_{l+m}|\tilde{\boldsymbol{x}}_l) = \int \ldots \int \prod_{i=1}^{m} p(\tilde{\boldsymbol{x}}_{l+i}|\tilde{\boldsymbol{x}}_{l+i-1}) \mathrm{d}\boldsymbol{x}_{l+i} \ldots \mathrm{d}\boldsymbol{x}_{l+m-1},$$

and sampling $\tilde{\boldsymbol{x}}_{l+m} \sim p(\tilde{\boldsymbol{x}}_{l+m}|\tilde{\boldsymbol{x}}_l)$ amounts to running m steps of the numerical scheme starting at $\tilde{\boldsymbol{x}}_l$.

To aid in the understanding of the following chapters, we define a propagation function $\Phi(\cdot)$ as

$$\begin{aligned} \Phi_l &: \mathbb{R}^6 \mapsto \mathbb{R}^6, \\ \tilde{\boldsymbol{x}}_{l-1} &\rightsquigarrow \tilde{\boldsymbol{x}}_l = \Phi_l(\tilde{\boldsymbol{x}}_{l-1}), \end{aligned} \tag{2.12}$$

which incorporates the numerical integration of the equations of motion associated with a particular dynamical model.

In this work, the deterministic integrators used are

- 4th order Runge-Kutta [21]: a standard explicit numerical integrator of order four with fixed step-size for ODEs.
- MATLAB's ode113 [22]: a variable-order, variable step-size numerical integrator which uses an Adams-Bashford-Moulton predictor-corrector method. It is a multistep method, and adapts step-sizes automatically.
- SGP4: The simplified general perturbations method (SGP4) was first devised by the authors of [23] and later expanded in Ref. [24]. It is a semi-analytical propagation method, which uses a modification of the Lagrange planetary equations and is able to separate secular and short-period effects, approximating the long-term evolution of the dynamics analytically. It takes into account gravitational field terms J_2 to J_4, luni-solar gravitational perturbations, and drag. The latter is modelled using empirical drag and ballistic coefficients by computing the change in semi-major axis over some finite time interval.

However, the deterministic approach does not account for inaccuracies in the dynamical model or random errors that accumulate during numerical propagation. We adopt, hence, a stochastic model, to represent uncertainty more rigorously.

2.4.2. Stochastic integration

In order to take into account the inherent uncertainty in the system, as well as other sources of error, it is possible to model orbital dynamics as a stochastic system. In this vein, Eq. (2.4) is recast as a stochastic differential equation (SDE) in the Itô sense. Such a system incorporates a diffusion term to represent propagation uncertainty [9], and is given, in its general form, by

$$\mathrm{d}\boldsymbol{x} = f(\boldsymbol{x}, t')\mathrm{d}t' + \sigma_W(\boldsymbol{x}, t')\mathrm{d}\boldsymbol{W}, \tag{2.13}$$

where $f(\boldsymbol{x}, t')$ represents the deterministic drift of the stochastic state $\boldsymbol{x}(t')$ and a diffusion (noise) term driven by a d-dimensional Wiener process $\boldsymbol{W}(t')$ is incorporated. $\boldsymbol{\sigma}_W(\boldsymbol{x}, t')$ is a 6×6 diffusion coefficient representing the scale of the assumed uncertainty. This stochastic parametrisation of unknown model accelerations captures the uncertainty in the dynamical model due to unmodelled dynamics or errors [25] and provides a potentially more realistic characterisation of spacecraft motion under uncertainty.

The Itô SDE in Eq. (2.13) can be discretised using a stochastic numerical scheme with a step-size $h_l = t'_l - t'_{l-1} > 0$. Due to its simplicity, we begin with the Euler-Maruyama discretisation scheme, which is described as

$$\tilde{\boldsymbol{x}}_l = \tilde{\boldsymbol{x}}_{l-1} + f(\tilde{\boldsymbol{x}}_{l-1}, t'_{l-1})h_l + \boldsymbol{\sigma}_W(\tilde{\boldsymbol{x}}_{l-1}, t'_{l-1})\boldsymbol{W}_l, \tag{2.14}$$

where $l \in \mathbb{N}$ is discrete time, $\tilde{\boldsymbol{x}}_l$ is the approximate state at time t'_l, $\tilde{\boldsymbol{x}}_l \approx \boldsymbol{x}(t'_l)$ and $\boldsymbol{W}_l = \boldsymbol{W}(t'_l) - \boldsymbol{W}(t'_{l-1})$ is a Gaussian r.v. with zero mean and diagonal covariance matrix $h_l\mathbb{I}$. We assume that the process noise is only present in the velocity components, hence this matrix is given by

$$\boldsymbol{\sigma}_W(\boldsymbol{x}) = \mathrm{diag}\Big(0,\ 0,\ 0,\ \sigma_W(v_x),\ \sigma_W(v_y),\ \sigma_W(v_z)\Big) \in \mathbb{R}^{6\times 6}. \tag{2.15}$$

For the rest of this work, the stochastic diffusion coefficient is assumed to be independent of $\boldsymbol{x}$, i.e., $\boldsymbol{\sigma}_W(\boldsymbol{x}_l) = \boldsymbol{\sigma}_W$.

Denoting the conditional pdf as $p(\tilde{\boldsymbol{x}}_l|\tilde{\boldsymbol{x}}_{l-1})$. it is easy to see that

$$p(\tilde{\boldsymbol{x}}_l|\tilde{\boldsymbol{x}}_{l-1}) = \mathcal{N}(\tilde{\boldsymbol{x}}_l; \tilde{\boldsymbol{x}}_{l-1} + h_l f(\tilde{\boldsymbol{x}}_{l-1}, t'_{l-1}), \sigma_W \sigma_W^\top).$$

If other schemes are used, the expression for $p(\tilde{\boldsymbol{x}}_l|\tilde{\boldsymbol{x}}_{l-1})$ may become more involved, but sampling $\tilde{\boldsymbol{x}}_l \sim p(\tilde{\boldsymbol{x}}_l|\tilde{\boldsymbol{x}}_{l-1})$ is always possible because it amounts to running one step of the numerical scheme at hand. In principle, any stochastic discretisation scheme can be used. However, for sufficient precision, Euler schemes in orbital dynamics demand very low step-sizes, which increase the computational cost. In this work, the stochastic Runge-Kutta (SRK) integration scheme of [26] is used, which is a well-known stochastic adaptation of the classical Runge-Kutta of order four (RK4), though with strong order-one convergence [27]. The SRK discretisation scheme is, according to the Butcher tableau values, given by

$$\tilde{\boldsymbol{x}}_l = \tilde{\boldsymbol{x}}_{l-1} + \frac{1}{6}[F_0 + 2F_1 + 2F_2 + F_3]h + \frac{1}{6}[G_0 + 2G_1 + 2G_2 + G_3]\Delta\boldsymbol{W}_l, \tag{2.16}$$

$$\begin{aligned}
\text{where}\quad F_0 &= f\left(\tilde{\boldsymbol{x}}_{l-1}, t'_{l-1}\right),\\
F_1 &= f\left(\tilde{\boldsymbol{x}}_{l-1} + \frac{1}{2}hF_0 + \frac{1}{2}G\Delta\boldsymbol{W}_l\right),\\
F_2 &= f\left(\tilde{\boldsymbol{x}}_{l-1} + \frac{1}{2}hF_1 + \frac{1}{2}G\Delta\boldsymbol{W}_l\right),\\
F_3 &= f\left(\tilde{\boldsymbol{x}}_{l-1} + \frac{1}{2}hF_2 + \frac{1}{2}G\Delta\boldsymbol{W}_l\right).
\end{aligned}$$

Note that since we assume a stochastic diffusion coefficient which is independent of $\boldsymbol{x}$, $G_0 = G_1 = G2 = G3$.

2.5 State space model

The state-space model provides a mathematical framework to describe the system evolution over time, capturing the relationship between the state, its inputs, and observable outputs. Assume that observations are collected at times $\boldsymbol{T} = \{t_k\}_{k=1}^M$ and denoted $\boldsymbol{z}_k = \boldsymbol{z}(t_k)$ for conciseness. Also, assume that the numerical scheme of Subsection 2.4.2 is aligned with the observation times in the sense that there are integer indices $l_1, \ldots, l_M$ such that $t_k = t'_{l_k}$. Let us denote $\boldsymbol{x}_k = \tilde{\boldsymbol{x}}_{l_k}$, hence, $\boldsymbol{x}_k \approx \boldsymbol{x}(t_k)$. In this work, considering discrete time corresponding to discrete observation times, the state-space model consists of

- $\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0)$: the prior pdf of the state,
- $\boldsymbol{x}_k \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$: the transition density of the state, obtained with Eq. (2.4) and
- $\boldsymbol{z}_k \sim p(\boldsymbol{z}_k|\boldsymbol{x}_k)$: the conditional pdf of the observation given the state, obtained with Eq. (2.10). $\boldsymbol{z}_k$ is the observation vector at time t_k, assumed to be conditionally independent of all other observations given the state.

2.6 Estimation theory

Estimation theory is the framework which allows us to execute inference of unknown states or parameters based on observed or simulated data [28], i.e., it allows us to make estimations of a QoI. Estimation theory plays a key role in the processes of filtering, prediction and decision-making in a diverse range of fields and represents the underlying framework behind

- state space filtering, or improving estimations of r.v.s as observations are sequentially collected,
- dynamical parameter estimation, when inferring fixed or varying parameters in a given system,
- estimation of parameters associated with target events, or the computation of expected states in a system which lead to the occurrence of specific events, such as re-entries or collisions.

Estimators

The simplest type of estimator is the point estimator, which provides a single value as an estimate of some unknown QoI. An example of this is the computation of the sample mean $\bar{x}$ from a set of data-points $\{x^i\}_{i=1}^N$ drawn from r.v. X. $\bar{x}$ is the point estimator of the true mean $\mathbb{E}[X]$ [18], and is given by

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x^i.$$

Crude Monte Carlo (CMC) involves the generation of N samples to compute empirical averages, i.e., to estimate the expectation of a function $g(X)$ under a pdf $p(x)$, i.e.,

$$\mathbb{E}[g(X)] = \int g(x)p(x)\mathrm{d}t \approx \frac{1}{N}\sum_{i=1}^{N} g(x^i), \tag{2.17}$$

where $x^i \sim p(x)$ are i.i.d random samples from $p(x)$.

Error metrics

The bias of an estimator $\bar{x}$ measures how far its expectation is from the true value x_{true}, i.e.,

$$\text{Bias}(\bar{x}) = \mathbb{E}[\bar{x}] - x_{\text{true}}.$$

An estimator is said to be unbiased if $\text{Bias}(\bar{x}) = 0$. Consistency is a property that ensures that as the sample size N reaches infinity, the estimate converges to the true value, that is, $\lim_{N\to\infty} \bar{x} = x_{\text{true}}$. Other error metrics which are often studied are the mean squared error (MSE), given by

$$\text{MSE}(\bar{x}) = \mathbb{E}[(\bar{x} - x_{\text{true}})^2],$$

and the root mean squared error $\text{RMSE} = \sqrt{\text{MSE}}$.

CMC provides a consistent empirical estimator of expectations with respect to the probability law of the r.v. X.

The estimation of a QoI by means of simulation-based methods in the absence of observations relies on probabilistic models and simulations. It is usually carried out with CMC simulations and its variations, to provide unbiased estimates of the QoI, where the integrable test function $g(\cdot)$ can be a dynamical model evaluation, or, in the case of rare event studies, an indicator function. Variations of CMC are used in Chapter 3 for the estimation of the probability of collision (PoC) as the main QoI, and in Chapter 5 for the estimation a re-entry time window.

Bayesian filtering

In Bayesian statistics, knowledge about the variables in a model is updated with new information in an observation dataset. The newly updated pdf is termed the *posterior* pdf. In the case of state-space filtering with nonlinear settings, the posterior distribution cannot generally be computed exactly, so methods that approximate it are used instead.

The likelihood function, $L(\cdot)$, quantifies the compatibility of a dataset with the predicted state. It is proportional to the conditional probability density of the observation given the state. The form of $L(\cdot)$ depends on the assumed probability distribution of the data. Given the observation model of Eq. (2.10), the conditional pdf of the observation z_k given the state $\boldsymbol{x}_k$ (the likelihood) is

$$L(\mathcal{M}_k(\boldsymbol{x}_k), \Omega_k | z_k) = p(z_k|\boldsymbol{x}_k) = \frac{1}{(2\pi)^{d/2}|\Omega_k|^{1/2}} \exp\left(-\frac{1}{2}(z_k - \mathcal{M}_k(\boldsymbol{x}_k))^\top \Omega_k^{-1} (z_k - \mathcal{M}_k(\boldsymbol{x}_k))\right), \tag{2.18}$$

i.e., it is assumed Gaussian, with covariance matrix Ω_k. It is further assumed that observations are conditionally independent given the state $\boldsymbol{x}_k$ and can be evaluated for any value of $\boldsymbol{x}_k$. Taking the log-likelihood for numerical stability, we obtain

$$\log L(\mathcal{M}_k(\boldsymbol{x}_k), \Omega_k | z_k) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\Omega_k| - \frac{1}{2}(z_k - \mathcal{M}_k(\boldsymbol{x}_k))^\top \Omega_k^{-1} (z_k - \mathcal{M}_k(\boldsymbol{x}_k)). \tag{2.19}$$

For continuous data with pdf $p(z_k|\boldsymbol{x}_k)$ at time t_k, the likelihood function is

$$L(\boldsymbol{x}_k|z_k) = \prod_{j=1}^{\mathcal{J}} p(z_k^j|\boldsymbol{x}_k), \text{ for } j = 1, \ldots, \mathcal{J},$$

where $\mathcal{J}$ is the number of observations at t_k.

Note, now, that the conditional pdf of $\boldsymbol{x}_k$ given $\boldsymbol{x}_{k-1}$ (i.e., the prior) is

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = p(\tilde{\boldsymbol{x}}_{l_k}|\tilde{\boldsymbol{x}}_{l_{k-1}}) = \int \ldots \int \prod_{i=l_{k-1}}^{l_k} p(\tilde{\boldsymbol{x}}_i|\tilde{\boldsymbol{x}}_{i-1}) \mathrm{d}\tilde{\boldsymbol{x}}_{l_{k-1}+1} \ldots \mathrm{d}\tilde{\boldsymbol{x}}_{l_k-1},$$

hence, we can sample $\boldsymbol{x}_k$ from $\boldsymbol{x}_{k-1}$ simply taking $l_k - l_{k-1}$ steps of the numerical scheme of choice.

From a Bayesian perspective, the statistical characterisation of the state $\boldsymbol{x}_k$ at time t_k given the data available up to that time is contained in the posterior pdf $p(\boldsymbol{x}_k|z_{1:k})$. A Bayesian filter is a recursive algorithm that computes (or at least approximates) the sequence of pdfs $p(\boldsymbol{x}_k|z_{1:k})$, $k = 1, 2, \ldots, M$, as observations are collected sequentially. To be specific, given the filtering pdf at time t_{k-1}, $p(\boldsymbol{x}_{k-1}|z_{1:k-1})$, one can use a Chapman-Kolmogorov equation to obtain the one-step-ahead prediction pdf

$$p(\boldsymbol{x}_k|z_{1:k-1}) = \int p(\boldsymbol{x}_{k-1}|\boldsymbol{x}_{k-1})p(\boldsymbol{x}_{k-1}|z_{1:k-1})\mathrm{d}\boldsymbol{x}_{k-1}, \tag{2.20}$$

and when z_k becomes available, update the prediction to obtain the new filtering pdf

$$p(\boldsymbol{x}_k|z_{1:k}) \propto p(z_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|z_{1:k-1}), \tag{2.21}$$

where $p(z_k|\boldsymbol{x}_k)$ is the conditional pdf of the observation z_k given the state $\boldsymbol{x}_k$ (i.e., the likelihood of $\boldsymbol{x}_k$).

If the state space model is parameterised by some random unknown parameter vector $\boldsymbol{\theta}$, it is possible to seek the joint posterior pdf $p(\boldsymbol{x}_k, \boldsymbol{\theta}|z_{1:k})$, which admits a similar decomposition

$$p(\boldsymbol{x}_k, \boldsymbol{\theta}|z_{1:k}) \propto p(z_k|\boldsymbol{x}_k, \boldsymbol{\theta})p(\boldsymbol{x}_k, \boldsymbol{\theta}|z_{1:k-1}). \tag{2.22}$$

Filters provide a recursive estimation of the posterior pdf as observations are processed, which enables the calculation of any QoI associated with said pdf. The posterior pdf in Eq. (2.21) can only be calculated exactly when the state space is linear and Gaussian. In astrodynamics, however, since the dynamics are nonlinear and uncertainties are possibly non-Gaussian, there is no exact solution, so it must be numerically approximated. This is achieved by the implementation of Bayesian filters, which are able to compute the sequence of posterior pdfs of the state. Chapter 4 introduces various kinds of filters for the task described in the context of tracking orbital spacecraft and unknown parameters.

3

ORBITAL COLLISION RISK ASSESSMENT

For the hundredth time I check the declination
Now the fear starts to grow
Even my computer shows
There are no errors in the calculations

Iron Maiden
When Two Worlds Collide (1998)

This chapter deals with one of the central aspects of STM: collision risk assessment. Having methods to accurately evaluate the risk of in-orbit collisions in an efficient yet reliable manner is a crucial component of orbital safety. In the following sections, a discussion on the available literature regarding probability of collision computation is laid out, highlighting the limitations of commonly used methods, and setting the stage for the development of the proposed method which is in principle able to mitigate said issues, whilst ensuring a reliable assessment of the probability of collision. The methodology section first describes a reduced dimension mapping with which to model conjunction events in an efficient manner, followed by a description of the importance sampling (IS) approach followed, which is able to obtain the accuracy and reliability of an independent CMC simulation at a reduced cost. The proposed method is successfully validated on a LEO and a GEO scenario by comparison with a large independent CMC simulation.

3.1 State of the art

The methodology used to compute the PoC usually depends on the type of encounter. In LEO scenarios, the most frequent are short-term encounters due to the large relative velocities involved, while in GEO cases, long-term encounters are the most frequent due to objects having similar orbital characteristics.

For short encounters, various numerical and analytical methods have been proposed [29], which, in general, offer either accuracy at the cost of a large computational burden, or high efficiency but with degradation in estimation quality. To ease computations, several assumptions (outlined below) are usually employed:

1. Spacecraft states (and their associated uncertainties) are represented as Gaussian distributions at the time of conjunction.

2. At critical times, rectilinear motion is assumed due to the high relative velocities involved.

3. Objects are endowed with a relative velocity that is large enough that the velocity covariance can be deemed static.

4. Spacecraft are modelled as spherical objects.

The aforementioned assumptions, although helpful for the efficient estimation of in-orbit collision probabilities, limit the applicability of such methods on a wider range of cases such as those with low velocity, or when the probability density functions (pdfs) involved are non-Gaussian. Some of the more relevant methodologies which have been proposed in this domain include works which pose the intrinsically 3-dimensional problem of computing the probability of two objects entering the same spatial volume as a 2D spatial integral which is to be integrated numerically [30], whilst the authors of [31] further show that the 3D uncertainty integration and the area associated with the 2D simplification of the integral on the conjunction plane are equivalent, leading to a new class of efficient PoC calculation methods. On the other hand, semi-analytical approaches have been proposed in which one dimension of the volume integral is solved analytically, resulting in an encounter plane, over which cylindrical coordinates are used for integration [32]. However, this procedure may be inefficient if the shape is non-spherical. Analytical power series descriptions to solve for the case with spherical covariance in the spacecraft states, or error series expansions of the discretised integral have been proposed in an attempt to improve computational speed, whilst maintaining accuracy in some cases [33][29][34].

Works such as [35] numerically explore the use of polynomial chaos expansion techniques with CMC simulations for which no Gaussian assumptions are made, though accuracy seems limited by the polynomial surrogate models of the target state that is used. Other authors explore a general description of uncertainty using Gaussian mixture models (see [36]), which provide better accuracy than plain Gaussian representation at an affordable cost, particularly after long propagation arcs. However, works such as [37] stress the fact that it is a challenging task to achieve results comparable to CMC when non-Gaussian behaviour is strong, particularly in such a multi-dimensional problem.

For long-term encounters, an integral metric including time is required, consisting of the covariance position ellipsoid swept by spacecraft during the encounter time. Studying long-term encounters must necessarily include proper uncertainty propagation, since space objects may be in close proximity to each other for long periods of time, making relative dynamics a critical aspect. Works which deal with long-term encounters include [38], where a geometrical approach using cylinders to compute the swept volume is introduced. Here, the main source of inaccuracy arises from the Gaussian assumption. In the cited work [39], an upper bound is defined in the PoC using an analytical formulation in formation-flying-type encounters. In recent works such as [40], a collision risk assessment method is proposed which tackles non-Gaussian uncertainty using local Gaussian approximations of propagated pdfs, in a fashion similar to Gaussian mixture models (GMMs). GMMs are more appropriate when nonlinear effects are significant. The authors of [41] propose a GMM representation coupled with differential algebra (DA) for propagation which automatically determines the number of components required to capture the behaviour of the propagation uncertainty, while the works in [42] and [43] apply GMMs and state transition tensors to several scenarios with varying dynamical complexity, and achieve a satisfactory level of efficiency and accuracy. A comparison of some of the above methods in terms of accuracy and run-time compared to a CMC simulation is included in Ref. number [44], and a more recently updated in-depth review of some of these PoC calculation methods can also be found in [45].

A widely used alternative for PoC computation is the CMC simulation, which provides a consistent estimate of the probability, making as few prior assumptions as possible [45][29][46]. It is therefore a suitable benchmark method for the assessment of different techniques. CMC entails simulating N i.i.d. samples from an initial distribution and applying a specific transformation (e.g., time propagation). The collision event is declared based on exceeding a user-defined threshold [47]. However, as mentioned in [48], if collision events are rare, obtaining a single collision in a CMC simulation may necessitate the generation of an extraordinarily large number of samples: for an expected PoC of 10^{-m}, at least 10^{m+2} samples are required, rendering this method impractical for studying orbital collisions due to prohibitively long computational run-times. Alternative methods exist and can be devised to improve the efficiency of MC-style estimation of rare quantities of interest without compromising the accuracy and unbiasedness of CMC simulations, and that is the goal of this chapter.

Works which apply rare-event sampling to safety and collision avoidance problems in other fields include [49] and [50], where extreme value theory is applied to the estimation of head-on collisions in road traffic, and in roundabouts, respectively. The authors of [51] use a sequential Monte Carlo (SMC) method to estimate the collision risk of aircraft in a given air traffic area. Other authors make use of the cross-entropy method to generate better proposals in IS, and apply it to the estimation of risk of air traffic collisions [52] and maritime collisions [53].

In the orbital conjunction domain, the adaptive splitting technique has been used to analyse the Iridium-Cosmos collision [54], whilst works such as [55] and [56] employ a sequential Monte Carlo square (SMC^2) algorithm (originally derived in [57]), which consists of two nested importance samplers. IS and line sampling have been used in combination with DA to tackle the PoC computation in Keplerian test cases [58].

3.2 Methodology

3.2.1. Initial conditions

The state of each RSO is represented by a 6-dimensional r.v. composed of the position and velocity vectors of its centre of mass (CoM) in the ECI reference frame coordinates. Let $\boldsymbol{x}_0 = \boldsymbol{x}(t_0)$ for simplicity. In particular, let $\boldsymbol{x}_{q,0} = [\boldsymbol{r}_{q,0}^\top \ \boldsymbol{v}_{q,0}^\top]^\top$ be the state vector of object $q \in \{A, B\}$ at time $t = 0$, where $\boldsymbol{r}_{q,0}$ is position and $\boldsymbol{v}_{q,0}$ is velocity. Since $\boldsymbol{x}_{q,0}$ is random, it can be characterised by pdf $\pi_{q,0}(\boldsymbol{x}_{q,0})$. The two 6-dimensional vectors $\boldsymbol{x}_{A,0}$ and $\boldsymbol{x}_{B,0}$ can be concatenated into a single 12-dimensional state $\boldsymbol{x}_0 = [\boldsymbol{x}_{A,0}^\top \ \boldsymbol{x}_{B,0}^\top]^\top \in \mathbb{R}^{12}$ that yields the joint initial state of the system, with pdf $\pi_0(\boldsymbol{x}_0)$. Usually, the marginal pdfs are modelled as independent Gaussian distributions, i.e.,

$$\boldsymbol{x}_{A,0} \sim \pi_{A,0} \equiv \mathcal{N}\left(\hat{\boldsymbol{x}}_{A,0}, \boldsymbol{\Sigma}_{A,0}\right), \quad \boldsymbol{x}_{B,0} \sim \pi_{B,0} \equiv \mathcal{N}\left(\hat{\boldsymbol{x}}_{B,0}, \boldsymbol{\Sigma}_{B,0}\right)$$

and they yield a Gaussian joint pdf of the form

$$\boldsymbol{x}_0 \sim \pi_0 \equiv \mathcal{N}(\hat{\boldsymbol{x}}_0, \boldsymbol{\Sigma}_0),$$

where $\hat{\boldsymbol{x}}_0 = [\hat{\boldsymbol{x}}_{A,0}^\top, \hat{\boldsymbol{x}}_{B,0}^\top]^\top$ is the 12-dimensional mean and

$$\boldsymbol{\Sigma}_0 = \begin{bmatrix} \boldsymbol{\Sigma}_{A,0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_{B,0} \end{bmatrix}$$

is the 12 × 12 covariance matrix. Let us note that the form of matrix $\boldsymbol{\Sigma}_0$ above implies that there is no correlation between the states of objects A and B. This is a reasonable assumption when the two initial conditions are obtained through independent OD procedures. However, it is not a necessary assumption for the PoC computation methodology introduced in the rest of this section. The algorithm described in Section 3.3.2 can work with an arbitrary choice of $\boldsymbol{\Sigma}_0$.

3.2.2. Collision events

A collision occurs when the volume occupied by each of the space objects has a non-empty intersection. The hard body radius (HBR) of each of the two objects involved in a conjunction can be defined as the largest length spanned between any point in said object and its centre of mass. In particular, for object $q \in \{A, B\}$, the HBR is

$$\gamma_q = \sup_{\boldsymbol{r} \in \mathcal{R}_q(t)} \|\boldsymbol{r} - \boldsymbol{r}_q(t)\|,$$

where $\mathcal{R}_q(t) \subset \mathbb{R}^3$ is the 3D region occupied by object q, and $\boldsymbol{r}_q(t)$ is the position vector of the centre of mass of object q (but note that the HBRs γ_A and γ_B are independent of time t).

Following [59], the combined HBR, $\gamma = \gamma_A + \gamma_B$, is used to simplify the computation of the PoC. Let us define the set $C_\gamma := \{t \in [0, \infty) : \|\boldsymbol{r}_A(t) - \boldsymbol{r}_B(t)\| \leq \gamma\}$ of time instants where the two objects are closer than the combined HBR γ. A collision occurs at time

$$T_\gamma = \begin{cases} \infty, & \text{if } C_\gamma = \emptyset \\ \inf C_\gamma, & \text{otherwise} \end{cases} .$$

Since the position vectors $\boldsymbol{r}_q(t)$, $q \in \{A, B\}$, are random, it follows that T_γ is a random variable as well. If we are interested in conjunctions that occur within a certain time window, say $t \in [0, T]$, then the probability of a collision is

$$P_{T,\gamma} = \mathbb{P}\left(T_\gamma \leq T\right),$$

where $\mathbb{P}(\mathcal{E})$ denotes the probability of the event characterised by the condition $\mathcal{E}$.

3.2.3. Reduction to 2D space

Traditional PoC computation methods rely on Euclidean distance as the reference metric to define thresholds for the estimation of collision risk. This is the natural approach for problems where Euclidean distance behaves smoothly as a function of time at critical time intervals. In orbital dynamics, however, this is not the case. Spacecraft may come into close proximity or even collide within the time-steps of typical numerical integration schemes, as velocities of objects in standard orbital regimes can exceed the 7 km/s mark. As a consequence, a collision event can be "missed" due to a lack of sufficient time resolution.

To overcome this limitation, an alternative 2D geometrical space is herein proposed, less sensitive to time resolution hindrances. The 2D metric developed in this chapter aims to capture collision events in a smoother manner. Since $\gamma \ll \min\{\|\boldsymbol{r}_B(t)\|, \|\boldsymbol{r}_A(t)\|\}$, a collision can only occur for two non-coplanar objects, at the relative line of nodes of the two osculating orbital planes. Firstly, using osculating orbital elements, one can write the position vectors in a reference frame defined by the direction of the relative line of nodes, $\boldsymbol{u}_n$, and another direction in the orbital plane, perpendicular to $\boldsymbol{u}_n$, given by $\boldsymbol{u}_{p,q}(t) = \boldsymbol{u}_{\mathcal{H},q}(t) \times \boldsymbol{u}_n$; $q \in \{A, B\}$, where $\boldsymbol{u}_{\mathcal{H},q}(t) = \boldsymbol{\mathcal{H}}_q(t)/\mathcal{H}_q(t)$ and $\mathcal{H}_q(t)$ is the angular momentum of object q at time t. To be specific, we have

$$\boldsymbol{r}_q(t) = \frac{\mathcal{H}_q(t)^2/\breve{\mu}}{1 + e_q(t)\cos\nu_q(t)}\left[\cos\left(\nu_q(t) - \zeta_q(t)\right)\boldsymbol{u}_n + \sin\left(\nu_q(t) - \zeta_q(t)\right)\boldsymbol{u}_{p,q}(t)\right] \tag{3.1}$$

for $q \in \{A, B\}$, where $\zeta_q(t)$ is the angle between the eccentricity vector and the relative line of nodes, $\breve{\mu}$ is the Earth's gravitational parameter, $e_q(t)$ is the osculating eccentricity at time t of object q and $\nu_q(t)$ is the osculating true anomaly at time t of object q. Time dependencies have been explicitly included in Eq. (3.1) for clarity, but we drop them hereafter to simplify notation.

From Eq. (3.1), the relative position between the two objects becomes

$$\begin{aligned}\boldsymbol{r}_B - \boldsymbol{r}_A = &\left[\frac{\mathcal{H}_B^2/\breve{\mu}}{1 + e_B\cos\nu_B}\cos\left(\nu_B - \zeta_B\right) - \frac{\mathcal{H}_A^2/\breve{\mu}}{1 + e_A\cos\nu_A}\cos\left(\nu_A - \zeta_A\right)\right]\boldsymbol{u}_n\\ &+ \frac{\mathcal{H}_B^2/\breve{\mu}}{1 + e_B\cos\nu_B}\sin\left(\nu_B - \zeta_B\right)\boldsymbol{u}_{p,B}\\ &- \frac{\mathcal{H}_A^2/\breve{\mu}}{1 + e_A\cos\nu_A}\sin\left(\nu_A - \zeta_A\right)\boldsymbol{u}_{p,A}\end{aligned}$$

For clarity, node indices run from $k = 1$ to $k = c$. As long as $\boldsymbol{u}_{p,B}$ and $\boldsymbol{u}_{p,A}$ are not colinear, the terms of the relative distance in those directions are not necessarily small but may be close to the line of nodes, i.e., $\nu_A \approx \zeta_A + \delta\nu_A$ and $\nu_B \approx \zeta_B + \delta\nu_B$, where $\delta\nu_q$ indicates a small change in ν_q. Using this approximation, the relative position close to the relative line of nodes can be written as

$$\boldsymbol{r}_B - \boldsymbol{r}_A = r_{A,c}\left[(1 - \Gamma_c)\delta\nu_B\boldsymbol{u}_{p,B} - \delta\nu_A\boldsymbol{u}_{p,A} - \Gamma_c\boldsymbol{u}_n\right], \tag{3.2}$$

where

$$r_{q,c} = (\mathcal{H}_q^2/\breve{\mu})/(1 + e_q\cos\nu_q) \tag{3.3}$$

and $\Gamma_c = 1 - r_{B,c}/r_{A,c}$. Hence, the distance between objects A and B is

$$\|\boldsymbol{r}_B(t) - \boldsymbol{r}_A(t)\| = r_{A,c}\sqrt{\Gamma_c^2 + \delta\nu_A^2 + (1 - \Gamma_c)^2\delta\nu_B^2 - 2(1 - \Gamma_c)\delta\nu_B\delta\nu_A\boldsymbol{u}_{p,A}\cdot\boldsymbol{u}_{p,B}}, \tag{3.4}$$

Eq. (3.4) does not explicitly show the time dependence of the fast variables $\delta\nu_q$, $q \in \{A, B\}$, and the fact that they are *not* independent. In order to provide a description which takes these facts into account, let us describe the time dependence of the true anomalies as

$$\delta\nu_A = n_A\Delta t \quad \text{and} \quad \delta\nu_B = \Delta_c + n_B\Delta t,$$

where

- Δt is the time elapsed since t_n (the time at which $\nu_A = \zeta_A$, i.e., the passage of object A through the relative node) up to the current time t,
- n_A and n_B are the instantaneous angular velocities at time t_n,
- and $\Delta_c = \nu_B(t_n) - \zeta_B(t_n)$.

The evolution of the distance between the two objects can now be expressed as an explicit function of time, namely

$$\|\boldsymbol{r}_B(t) - \boldsymbol{r}_A(t)\| = r_{A,c}\sqrt{D_1 + D_2\Delta t + D_3\Delta t^2}, \tag{3.5}$$

where

$$\begin{aligned} D_1 &= \Gamma_c^2 + (1-\Gamma_c)^2\Delta_c^2, \\ D_2 &= 2(1-\Gamma_c)^2 n_B\Delta_c^2 - 2(1-\Gamma_c)n_A\Delta_c\boldsymbol{u}_{p,A}\cdot\boldsymbol{u}_{p,B} \quad \text{and} \\ D_3 &= n_A^2 + (1-\Gamma_c)^2 n_B^2 - 2(1-\Gamma_c)n_A n_B\boldsymbol{u}_{p,A}\cdot\boldsymbol{u}_{p,B}. \end{aligned}$$

From Eq. (3.5), the relative distance is minimised when $\Delta t = -D_2/(2D_3)$, which yields

$$\|\boldsymbol{r}_B(t) - \boldsymbol{r}_A(t)\|_{\min} = r_{A,c}\sqrt{D_1 - \frac{D_2^2}{4D_3}}. \tag{3.6}$$

If we additionally assume that the instantaneous angular velocities are similar, $n_A \approx n_B$, meaning that the semi-major axes of the two orbits are similar[1] and $(1-\Gamma_c) \approx 1$, which is a necessary condition for collision, then

$$\|\boldsymbol{r}_B(t) - \boldsymbol{r}_A(t)\|_{\min} = r_{A,c}\sqrt{\Gamma_c^2 + \frac{\Delta_c^2}{2}\left[1 + \boldsymbol{u}_{p,A}\cdot\boldsymbol{u}_{p,B}\right]} \leq r_{A,c}\sqrt{\Gamma_c^2 + \Delta_c^2}, \tag{3.7}$$

where the bound on the right-hand side depends only on the variables Γ_c and Δ_c. We can now define a new collision threshold γ' as

$$\gamma' = \frac{\|\boldsymbol{r}_B(t) - \boldsymbol{r}_A(t)\|_{\min}}{r_{A,c}}. \tag{3.8}$$

Consequently, the collision event condition can be defined in an alternative way as a function of:

- Γ_c, the non-dimensional distance of the osculating orbits of the two objects along the relative line of nodes, and

[1] For the purposes of this analysis, it is assumed that collisions occur between objects in orbits which are similar in size. This represents the most common scenario, though a counterexample is, of course, possible: a collision may occur between objects in orbits with different semi-major axes and eccentricities. If the assumption $n_A \approx n_B$ is removed, then the methodology remains applicable, but Eq. (3.6) requires further analysis in order to obtain a suitable upper bound for $\|\boldsymbol{r}_B(t) - \boldsymbol{r}_A(t)\|_{\min}$ in terms of Γ_c, Δ_c and, possibly, the ratio n_A/n_B.

- Δ_c, the angular distance to the relative line of nodes of the second object when the first object is at the relative line of nodes.

Figure 3.1 illustrates the two variables of interest, Γ_c and Δ_c, at the intersection of the two orbits. The line of nodes $\boldsymbol{U}_n$ of RSOs A and B (orbits A and B, respectively), as well as vectors $\boldsymbol{r}_{B,k}$ and $\boldsymbol{r}_{A,k}$ are shown for clarity in the two-diagram illustration.

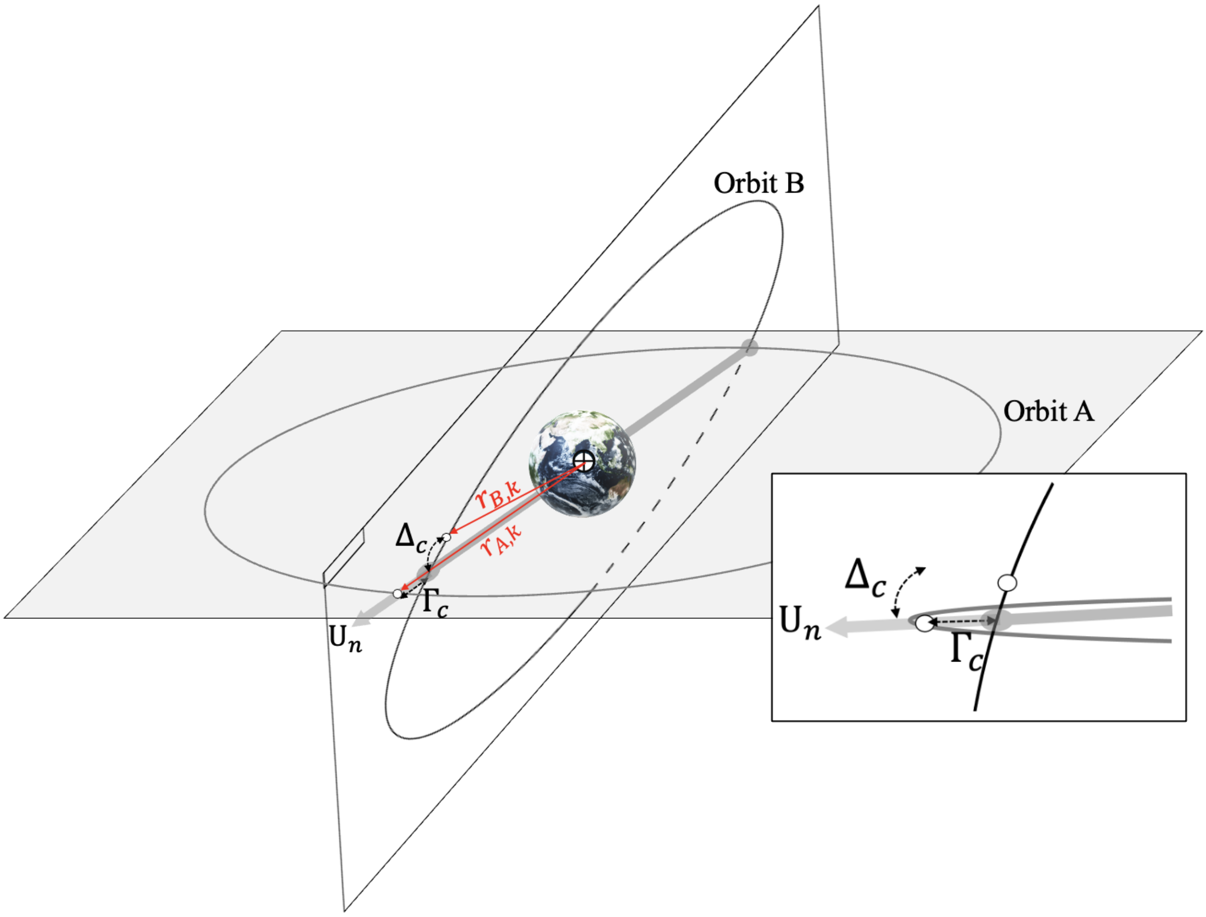


Figure 3.1: Illustration of the two variables, Γ_c and Δ_c comprising the collision metric used to define the dimensionality reduction mapping. The line of nodes $\boldsymbol{U}_n$ (solid gray line) of RSOs A and B (orbits A and B, respectively), as well as vectors $\boldsymbol{r}_{B,k}$ and $\boldsymbol{r}_{A,k}$ are shown for clarity. It is straightforward to see that if Γ_c and Δ_c are both zero, the objects occupy the same point in space. In the square on the right, a close-up of the two variables can be seen (both with respect to the line of nodes).

In order to illustrate the motivation for the proposed 2D metric, in Figures 3.2 and 3.3, we show two alternatives when assessing collisions. In Figure 3.2, the Euclidean distance between spacecraft in LEO is plotted as a function of time for a period of four days with different integration time-steps. Aside from the fact that large oscillations in relative distance can be observed in critical time intervals, unless the integration time grid is sufficiently fine, collision events are likely to be missed, as the collision event may occur in between the time-steps defined by the numerical integration scheme. Figure 3.2 (top) shows a fine time-grid integration, whilst the middle figure shows a coarser integration. The bottom figure illustrates how the fine integration reaches collision (red dot), whilst the coarser one misses it, showcasing the enormous difficulty in getting the time-steps or the tolerances just right to observe a collision.

In Figure 3.3, the proposed conjunction variables are plotted for the same four days. On the left, the figure shows the evolution of the two variables Γ and Δ throughout the nodes, showing a smoother evolution of the 2D state, compared to the significant oscillations observed in Figure 3.2. On the right, we show the magnitude of $\boldsymbol{\xi}_k := [\Gamma_k \; \Delta_k]^\top$ as it evolves in time (represented by the node number), showing a smoother curve (almost straight). This has several implications. It provides a potentially more suitable space for the definition of thresholds and interpolation, as well as for the development of surrogate models. Furthermore, analysts and operators may benefit from a more intuitive geometric picture of conjunction risk evolution.

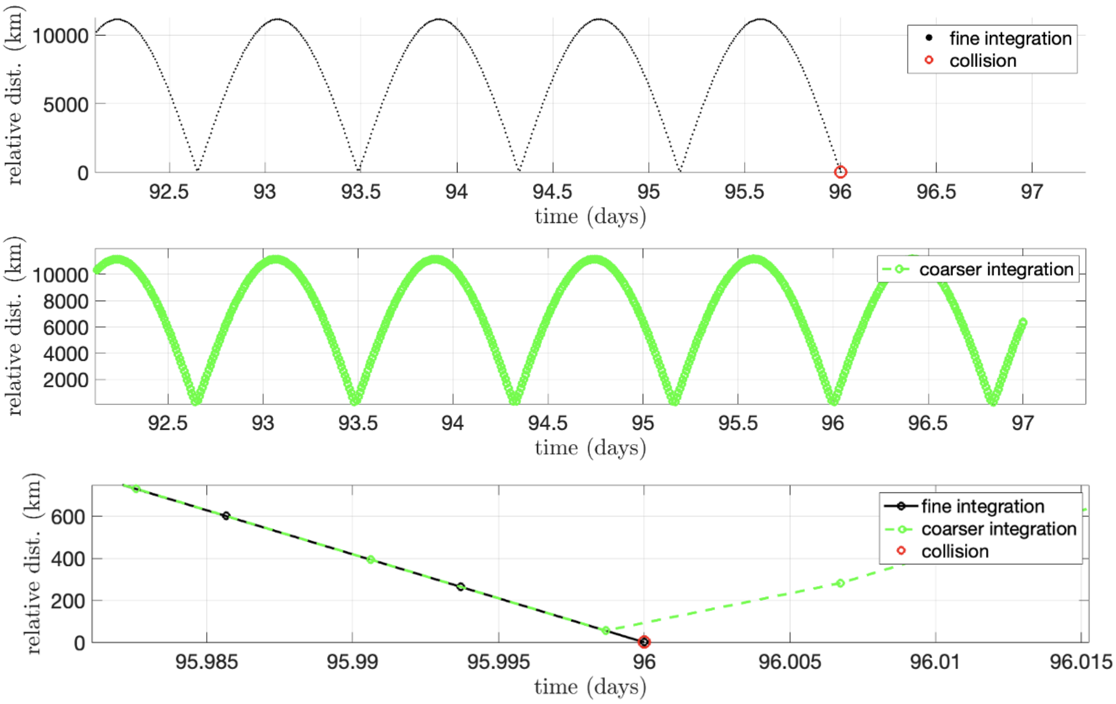


Figure 3.2: The euclidean distance between two spacecraft is plotted as a function of time for a period of four days, with emphasis in the final hours of propagation. The figure on the top shows a fine (very small step-size) integration scheme, the middle plot shows a coarser (larger step-size) propagation, and the bottom figure shows the difficulty in reaching collision with a slightly coarser integration time-grid.

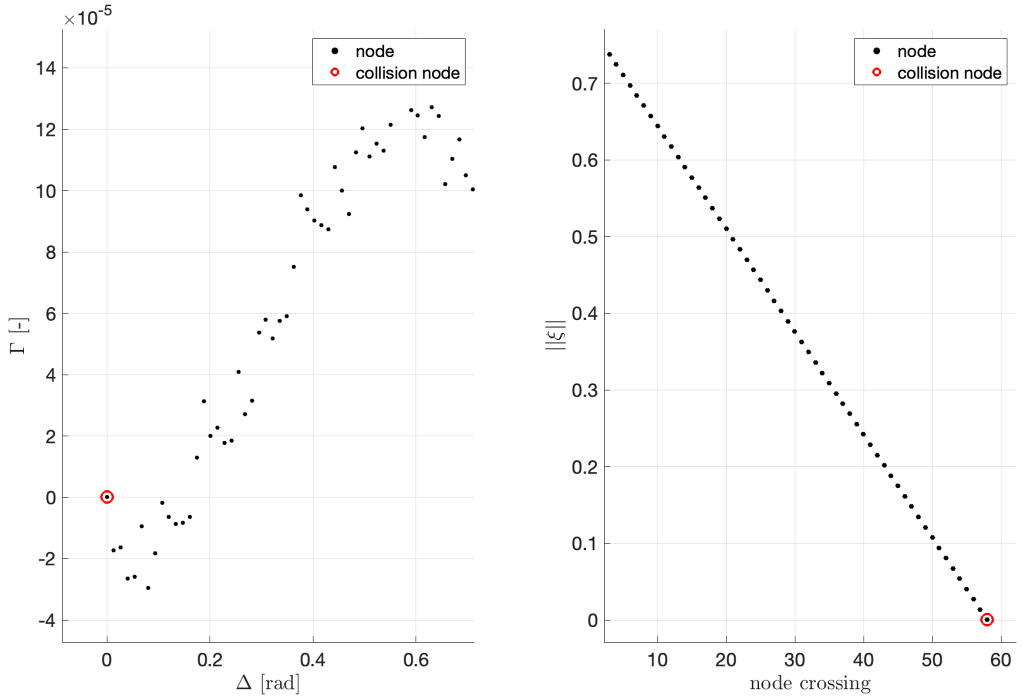


Figure 3.3: The 2D metric is plotted for the same object throughout a simulation time-span of the same four days. The picture shows the evolution throughout the nodes of the two variables Γ and Δ (left) which make up $\boldsymbol{\xi}$. The magnitude of $||\boldsymbol{\xi}||$ is shown as it evolves through time, represented by the node number (right). This represents a stark comparison to the oscillations seen in Figure 3.2.

Accordingly, if we define the set

$$C_{\gamma'} := \left\{ t \in [0, \infty) : \sqrt{\Gamma_c^2(t) + \Delta_c^2(t)} \leq \gamma' \right\}, \tag{3.9}$$

then the time of collision can be approximated as

$$T_{\gamma'} := \begin{cases} \inf C_{\gamma'}, & \text{if } C_{\gamma'} \neq \emptyset, \\ \infty, & \text{otherwise} \end{cases}$$

and the PoC for $t \in [0, T]$ is estimated as

$$P_{T,\gamma} = \mathbb{P}(T_\gamma \leq T) \approx \mathbb{P}(T_{\gamma'} \leq T) =: P_{T,\gamma'}. \tag{3.10}$$

3.2.4. Surrogate model

The propagation of the initial state, $\boldsymbol{x}_0$ is needed to assess whether condition $T_{\gamma'} \leq T$ is fulfilled. The accuracy of the dynamical propagation is imposed by the conditions inherent to a collision, i.e., an event that typically occurs in space volumes of the order of hundreds of cubic meters and in time intervals smaller than one second (depending on the specific geometry of the collision). Keplerian propagation does not provide the required accuracy and, consequently, high-fidelity (HF) propagation is required.

The computational cost of HF propagators is high, making the CMC approach prohibitive, even with the reduced 2D mapping described in Section 3.2.3. In what follows, a surrogate model is introduced to reduce the computational cost of orbital propagation while keeping the accuracy level required for the computation of the PoC in Eq. (3.10). Let us denote a deterministic propagation method that maps the state's initial condition $\boldsymbol{x}_0$ to some time $t \in (0, T]$ (like Eq. (2.12)) as

$$\begin{aligned} \Phi_t : \ & \mathbb{R}^{12} && \mapsto && \mathbb{R}^{12} \\ & \boldsymbol{x}_0 && \rightsquigarrow && \boldsymbol{x}_t = \Phi_t(\boldsymbol{x}_0). \end{aligned} \tag{3.11}$$

Additionally, we denote the projection of the 12D state $\boldsymbol{x}_t$ onto the reduced 2D space of the variables $\{\Gamma(t), \Delta(t)\}$ as

$$\begin{aligned} \Xi_t : \ & \mathbb{R}^{12} && \mapsto && \mathbb{R}^{2} \\ & \boldsymbol{x}_t && \rightsquigarrow && \boldsymbol{\xi}(t) = \Xi_t(\boldsymbol{x}_t) = [\Gamma(t), \Delta(t)]^\top. \end{aligned} \tag{3.12}$$

To test the validity of the proposed projection in capturing collision events, we can interpret the HF mapping of a sample $\boldsymbol{x}_0^i \sim \pi_0$ using the model in Eq. (3.12) and the detection (or not) of a collision in $\mathbb{R}^{12}$ as a test. Specifically, given the initial condition $\boldsymbol{x}_0^i$ we first compute $\boldsymbol{\xi}_k^i$ using Eq. (3.12) and then evaluate the associated $\mathbb{R}^{12}$ distance metric $\left\|\boldsymbol{r}_B^i(t) - \boldsymbol{r}_A^i(t)\right\|_{\min}$ in Eq. (3.7) by propagating through Eq. (3.11). For samples for which $\|\boldsymbol{\xi}_k^i\| < \gamma'$, if $\left\|\boldsymbol{r}_B^i(t) - \boldsymbol{r}_A^i(t)\right\|_{\min} < \gamma$ the result of the test is positive and we declare a collision for the ith sample $\boldsymbol{x}_0^i$, i.e., a true positive. Otherwise, for samples for which $\|\boldsymbol{\xi}_k^i\| > \gamma'$, if $\left\|\boldsymbol{r}_B^i(t) - \boldsymbol{r}_A^i(t)\right\|_{\min} > \gamma$, the result is negative (no collision), and we declare a true negative. However, there may be two possible outcomes that would signify model failure: a false positive/negative may be declared if a sample which reaches collision in $\mathbb{R}^2$, does not in $\mathbb{R}^{12}$, or a sample which does not reach collision in $\mathbb{R}^2$, reaches collision in $\mathbb{R}^{12}$. A correlation plot for 215 samples drawn from the initial distribution $\pi_0(\boldsymbol{x}_0)$ is shown in Figure 3.4, where samples which are true positives and true negatives must appear in the first and third quadrants of the plot. Good agreement can be observed, with a very small number of false positive/negatives appearing in the second and fourth quadrants.

Let us now denote the times of the relative node passage (ascending and descending) as t_k, $k = 1, \ldots, c$, with $t_c \leq T$ (and let us remark that the sequence $\{t_k\}_{k=1}^c$ depends on the propagation method Φ_t).

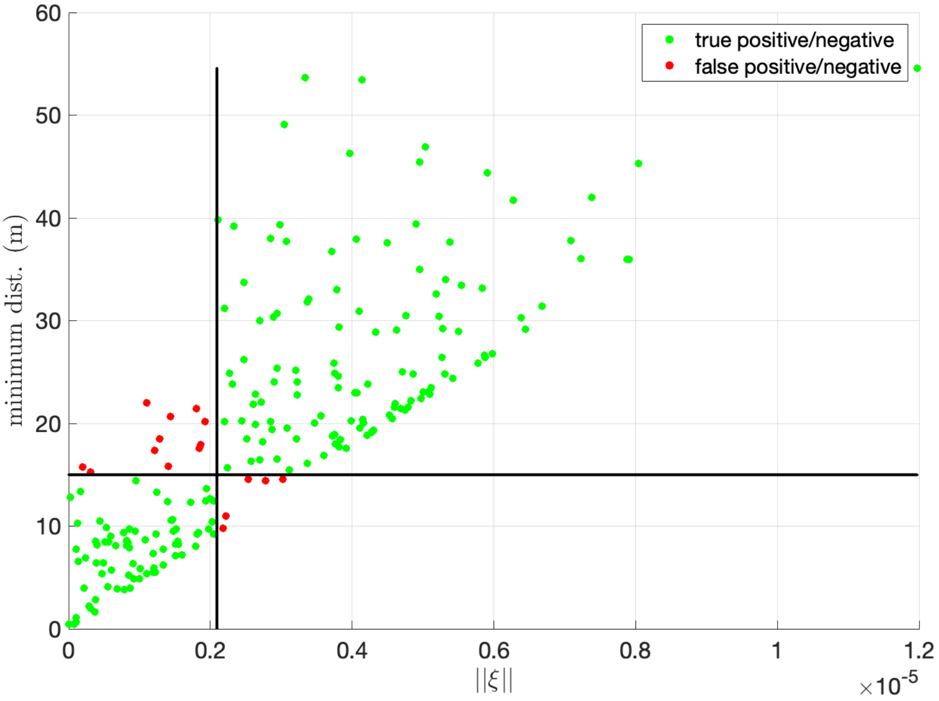


Figure 3.4: Correlation plot for 215 samples drawn from $\pi_0(\boldsymbol{x}_0)$. The true positives and negatives of the test appear in green in the first and third quadrants of the plot, whilst false positives and negatives appear in red on the second and fourth quadrants.

At each t_k, the coordinates of vector $\boldsymbol{\xi}_k := \boldsymbol{\xi}(t_k)$ are computed as

$$\Gamma(t_k) := \Gamma_k = 1 - \frac{r_{B,k}}{r_{A,k}} \quad \text{and} \quad \Delta(t_k) := \Delta_k = \nu_B(t_k) - \zeta_B(t_k) \tag{3.13}$$

where

$$r_{A,k} = \boldsymbol{r}_A(t_k), \qquad r_{B,k} = \frac{\mathcal{H}^2(t_k)/\check{\mu}}{1 + e(t_k)\cos\nu_B(t_k)}$$

and we have dropped the time dependence of r_q in the notation. From Eq. (3.13) and a given propagation method Φ_t, we can explicitly construct a sequence of 2D projections $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_c$.

Keplerian dynamics are not accurate enough to compute the PoC, however we can still use them to reduce the cost of the computations. To see this, let Φ_t^{HF} and Φ_t^{Kep} denote a HF propagation scheme and a Keplerian propagation scheme, respectively. They yield different 2D projections, namely, $\boldsymbol{\xi}_k^{HF} = \Xi_{t_k}\left(\Phi_{t_k}^{HF}(\boldsymbol{x}_0)\right)$ and $\boldsymbol{\xi}_k^{Kep} = \Xi_{t_k}\left(\Phi_{t_k}^{Kep}(\boldsymbol{x}_0)\right)$. If we think of $\boldsymbol{\xi}_k^{Kep}$ as an approximation of $\boldsymbol{\xi}_k^{HF}$ then we can define an approximation error $\boldsymbol{\epsilon}_k = \boldsymbol{\epsilon}_k(\boldsymbol{x}_0)$ and obtain

$$\boldsymbol{\xi}_k^{HF} = \boldsymbol{\xi}_k^{Kep} + \boldsymbol{\epsilon}_k. \tag{3.14}$$

The approximation error $\boldsymbol{\epsilon}_k(\boldsymbol{x}_0)$ is unknown, but we can estimate it by means of a Taylor expansion around a reference initial condition, denoted $\boldsymbol{x}_{\text{ref},0}$. In particular, a first order expansion yields

$$\boldsymbol{\epsilon}_k \approx \tilde{\boldsymbol{\epsilon}}_k = \boldsymbol{\epsilon}_{\mathrm{ref},k} + \boldsymbol{J}_k^{HF}\left(\boldsymbol{x}_0 - \boldsymbol{x}_{\mathrm{ref},0}\right), \tag{3.15}$$

where the error in 2D space for the reference initial condition is given by

$$\boldsymbol{\epsilon}_{\mathrm{ref},k} = \Xi_{t_k}\left(\Phi_{t_k}^{HF}(\boldsymbol{x}_{\mathrm{ref},0})\right) - \Xi_{t_k}\left(\Phi_{t_k}^{Kep}(\boldsymbol{x}_{\mathrm{ref},0})\right) \tag{3.16}$$

and $\boldsymbol{J}_k^{HF}$ is the Jacobian of the composition $\Xi_{t_k} \circ \Phi_{t_k}^{HF}$ evaluated at $\boldsymbol{x}_{\mathrm{ref},0}$. Matrix $\boldsymbol{J}_k^{HF}$ can be numerically approximated using finite differences.

A second order approximation of $\boldsymbol{\epsilon}_k$ is also feasible, albeit at a higher computational cost. For simplicity, we assume the first-order approximation in Eq. (3.15) through the rest of this section.

Combining Eq. (3.14) and Eq. (3.15) yields an estimate of the HF vector

$$\begin{aligned}\boldsymbol{\xi}_k^{HF} \approx \tilde{\boldsymbol{\xi}}_k &= \boldsymbol{\xi}_k^{Kep} + \tilde{\boldsymbol{\epsilon}}_k \\ &= \boldsymbol{\xi}_k^{Kep} + \boldsymbol{\epsilon}_{\mathrm{ref},k} + \boldsymbol{J}_k^{HF}\left(\boldsymbol{x}_0 - \boldsymbol{x}_{\mathrm{ref},0}\right).\end{aligned} \tag{3.17}$$

We note that the HF propagation of a set of initial states around $\boldsymbol{x}_{\mathrm{ref},0}$ (say $\Phi_{t_k}^{HF}(\boldsymbol{x}_0^{ii})$, for $ii = 1, \ldots, I$) is still needed to approximate the Jacobian $\boldsymbol{J}_k^{HF}$ using finite differences and, hence, the error $\boldsymbol{\epsilon}_{\mathrm{ref},k}$.

The Taylor expansion can also be used to identify the initial state vectors that lead to close encounters of objects A and B. Assuming the model in Eq. (3.17) and minimising the quadratic norm of $\tilde{\boldsymbol{\xi}}_k$ yields

$$\boldsymbol{x}_{c,0} = \boldsymbol{x}_{\mathrm{ref},0} - \left((\boldsymbol{J}_k^{HF})^\top \boldsymbol{J}_k^{HF}\right)^{-1} (\boldsymbol{J}_k^{HF})^\top \left(\Xi_{t_k}\left(\Phi_{t_k}^{Kep}(\boldsymbol{x}_{\mathrm{ref},0})\right) + \boldsymbol{\epsilon}_{\mathrm{ref},k}\right), \tag{3.18}$$

where $\boldsymbol{x}_{c,0}$ is the initial condition that yields the collision solution and, hence, a collision at time t_k. Note that Eq. (3.18) does not provide a closed-form expression for $\boldsymbol{x}_{c,0}$, but it is straightforward to approximate it numerically using, e.g., a gradient search. It is worth noting that, depending on the orbital regime, the accuracy of the first order approximation validity may degrade with very long propagation times, and large distances from $\boldsymbol{x}_{\mathrm{ref},0}$.

3.3 PoC computation

3.3.1. Crude Monte Carlo analysis

CMC has been used in previous works to establish a benchmark for PoC computation [60], as it yields uncertainty propagation without any geometrical simplification. In particular, a detailed procedure to compute the PoC using CMC is described in [46].

For completeness, the details of the approach implemented in the present work are gathered below. Following the notation of Section 2.4, time indices associated to numerical integration time-steps are denoted t'. The procedure runs through three steps as follows:

1. Generate N independent samples from the joint prior distribution π_0, denoted $\boldsymbol{x}^i(0) = \begin{bmatrix} \boldsymbol{x}^i_{A,0} \\ \boldsymbol{x}^i_{B,0} \end{bmatrix}$, $i = 1, \dots, N$.

2. Select a sufficiently fine time grid $t'_0 = 0, t'_1, \dots, t'_L = T$ and use a HF numerical scheme to generate N sequences $\boldsymbol{x}^i_{t'_l} = \Phi^{HF}_{t'_l}\left(\boldsymbol{x}^i_{t'_{l-1}}\right)$, $l = 0, 1, \dots, L$, then denote $\boldsymbol{x}^i_{q,t'_l} = \begin{bmatrix} \boldsymbol{r}^i_{q,t'_l} \\ \boldsymbol{v}^i_{q,t'_l} \end{bmatrix}$, for $q \in \{A, B\}$, and compute

$$d^i_L := \min_{0 \le l \le L} \|\boldsymbol{r}^i_{B,t'_l} - \boldsymbol{r}^i_{A,t'_l}\|, \tag{3.19}$$

   the minimum Euclidean distance between the orbiting objects A and B on the grid $t'_0 = 0, \dots, t'_L = T'$ in the ith independent simulation.

3. Let $I_\gamma(d)$ be the indicator function

$$I_\gamma(d) = \begin{cases} 1, & \text{if } d < \gamma \\ 0, & \text{otherwise} \end{cases}, \tag{3.20}$$

   where γ is the combined HBR. The PoC can now be estimated as

$$\bar{P}^{N,L}_{T',\gamma} = \frac{1}{N} \sum_{i=1}^{N} I_\gamma(d^i_L).$$

The PoC can also be estimated by mapping the simulated states $\boldsymbol{x}^i_{t'}$ onto the 2D space defined by the variables Γ^i_k and Δ^i_k. To be specific, let $t'_1, \dots, t'_c$ be the times of the node passages generated by the numerical scheme $\Phi^{HF}_{t'}$ and compute

$$\boldsymbol{\xi}^{i,HF}_k = \Xi_{t'_k}(\boldsymbol{x}^i_{t'_k}) = \begin{bmatrix} \Gamma^i_k(t'_k) \\ \Delta^i_k(t'_k) \end{bmatrix}, \quad i = 1, \dots, N,$$

for each t'_k. The minimum distance metric over all node passages is denoted

$$\mathring{d}^i_c = \min_{1 \le k \le c} \|\boldsymbol{\xi}^{i,HF}_k\|, \tag{3.21}$$

where $r^i_{A,c}(t'_k)$ is obtained as in Eq. (3.3) and the PoC can then be estimated as

$$\bar{P}^{N,c}_{T',\gamma'} = \frac{1}{N} \sum_{i=1}^{N} I_\gamma(\mathring{d}^i_c).$$

Let us note that both $\bar{P}^{N,L}_{T',\gamma}$ and $\bar{P}^{N,c}_{T',\gamma'}$ are CMC estimators of the PoC based on HF propagation with the scheme $\Phi^{HF}_{t'}$ (i.e., *not* the 2D surrogate model of Eq. (3.17)).

3.3.2. PoC by importance sampling

Collisions are rare events, meaning that $P_{T',\gamma} << 1$. As a consequence, the PoC computation via CMC is very costly. In particular, it requires

(a) the numerical integration of the dynamical model on a high-resolution time grid $t'_0 < \cdots < t'_L$ (where L is large and the steps $t'_l - t'_{l-1}$ can be very small),

(b) a very large number of samples N, because only a very small subset of the domain of the pdf π_0 corresponds to trajectories leading to a collision.

Problem (a) can be mitigated using time-stepping techniques where the elements t'_l, $l = 1, \ldots, L$, in the time-grid are detected adaptively. As for (b), the support of the initial condition can be explored efficiently using importance sampling (IS). In the remainder of this section we describe the plain IS methodology (for the estimation of $P_{T',\gamma}$) and then discuss its formulation as applied to our problem.

Let $\mathcal{S}_{\gamma,0} \subset \mathbb{R}^{12}$ denote the subset of feasible initial conditions that lead to a collision in the interval $[0, T']$ assuming a HBR of γ, i.e., the collision probability in Eq. (3.10) can be rewritten as $P_{T',\gamma} = \int_{\mathcal{S}_{\gamma,0}} \pi_0(\boldsymbol{x})\mathsf{d}\boldsymbol{x}$, where π_0 is the initial (usually Gaussian) distribution, centred around the initial state $\boldsymbol{x}_0$. If $P_{T',\gamma} \approx 10^{-m}$ then one needs $O(10^{m+2})$ CMC samples to obtain a low-variance estimate of the PoC [48]. A more efficient strategy to approximate the latter integral is to choose a proposal pdf (often called *importance function*) μ_0, which may be a Gaussian distribution centred on a state $\boldsymbol{x}_{c,0}$ which leads to collision. This pdf should be chosen such that $\frac{\pi_0(\boldsymbol{x})}{\mu_0(\boldsymbol{x})} < \infty$ for all $\boldsymbol{x}$ in the support of π_0 and

$$\int_{\mathcal{S}_{\gamma,0}} \mu_0(\boldsymbol{x})\mathsf{d}\boldsymbol{x} >> \int_{\mathcal{S}_{\gamma,0}} \pi_0(\boldsymbol{x})\mathsf{d}\boldsymbol{x}.$$

Assume that we generate N i.i.d. samples $\boldsymbol{x}_0^1, \ldots, \boldsymbol{x}_0^N$ from the proposal pdf, μ_0, and assign them importance weights of the form $w(\boldsymbol{x}_0^i) \propto \frac{\pi_0(\boldsymbol{x}_0^i)}{\mu_0(\boldsymbol{x}_0^i)}$, $i = 1, \ldots, N$, then it can be easily shown that

$$\lim_{N\to\infty} \sum_{i=1}^{N} w_0^i g(\boldsymbol{x}_0^i) = \int g(\boldsymbol{x})\pi_0(\boldsymbol{x})\mathsf{d}\boldsymbol{x} \quad \text{almost surely (a.s.),}$$

where $w_0^i = \frac{w(\boldsymbol{x}_0^i)}{\sum_{j=1}^{N} w(\boldsymbol{x}_0^j)}$ is the normalised importance weight of the sample $\boldsymbol{x}_0^i$ and $g : \mathbb{R}^{12} \mapsto \mathbb{R}$ is any bounded real test function [61][62].

In particular, if the test function is the indicator for the set $\mathcal{S}_{\gamma,0}$, namely,

$$I_{\mathcal{S}_{\gamma,0}}(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \boldsymbol{x} \in \mathcal{S}_{\gamma,0} \\ 0, & \text{otherwise} \end{cases}$$

then

$$\bar{P}^{\text{IS},N}_{T',\gamma} = \sum_{i=1}^{N} w_0^i I_{\mathcal{S}_{\gamma,0}}(\boldsymbol{x}_0^i) \tag{3.22}$$

is a consistent estimator [61] of the PoC, i.e., $\lim_{N\to\infty} \bar{P}^{\text{IS},N}_{T',\gamma} = P_{T',\gamma}$ a.s.. Note that in order to evaluate the indicator $I_{\mathcal{S}_{\gamma,0}}(\boldsymbol{x}_0^i)$ one needs to propagate the initial condition $\boldsymbol{x}_0^i$ using an HF propagator $\Phi_t^{HF}(\cdot)$, evaluate the distance d_L^i as in Eq. (3.19) and then obtain $I_{\mathcal{S}_{\gamma,0}}(\boldsymbol{x}_0^i) = I_\gamma(d_L^i)$ using the definition of Eq. (3.20). The IS estimator can be more accurate (i.e., present less variance) than the CMC estimator $P^{N,L}_{T',\gamma}$ for the same value of N. If the proposal pdf μ_0 is suitably chosen, one can, in a similar fashion to CMC, calculate the IS PoC estimator in the 2D space, by using the indicator $I_\gamma(\mathring{d}_c^i)$ in Eq. (3.22). The choice of proposal density μ_0 is problem-specific, and this pdf may have virtually any form, including Gaussian mixtures, heavy-tailed pdfs (e.g., Student's t-distribution), or other non-Gaussian distributions. However, these alternative proposals may also increase computational complexity and may require more sophisticated tuning. A Gaussian proposal is chosen here because it is easy to implement and initial conditions (ICs) are usually given in terms of a mean state and a covariance matrix. It is important to note that the proposal refers to states at t_0' which end up in collision after propagation. Consequently, the proposal samples mapped onto the reduced space are not Gaussian.

3.3.3. Proposed Methodology

The proposed method for PoC computation combines the approximate 2D model of Section 3.2.3, specific details of which are provided in Algorithm 1. It consists of two stages:

1. We follow an iterative procedure to estimate an initial condition, denoted $\boldsymbol{x}_{c,0}$, which yields a collision within the interval $t' \in [0, T']$.
2. We construct a Gaussian importance function $\mu_0(\boldsymbol{x})$ centred around the state $\boldsymbol{x}_{c,0}$. Then, we use μ_0 to generate samples and importance weights that, in turn, yield an approximation of the PoC as in Eq. (3.22).

The 'while loop' between lines 2 and 7 of Algorithm 1 is iterated until a reference initial condition, $\boldsymbol{x}^j_{\text{ref},0}$, that leads to a collision at some time t_k' $(1 \le k \le c)$ is found. At each iteration j of the loop, the computation of $\boldsymbol{\epsilon}_{\text{ref},k}$ requires both the HF and Keplerian propagations of the current reference vector $\boldsymbol{x}^j_{\text{ref},0}$ (see Eq. (3.16)). Additionally, the Jacobian matrix $\boldsymbol{J}_k^{HF}$ is approximated by central finite differences, due to its ease of computation, which also requires the HF propagation of a small collection of points in $\mathbb{R}^{12}$ located close to $\boldsymbol{x}^j_{\text{ref},0}$ [2].

[2] The step-size of the finite-differences method may be tuned empirically offline by studying the equivalence between a small number of approximate samples and their HF counterpart.

Algorithm 1 IS-based PoC computation in 2D space

```
Inputs:
  - Numbers of samples $N_1$ and $N_2$.
  - The HBR $\gamma$ or 2D collision threshold $\gamma'$.
  - The initial distribution $\pi_0$ with mean $\hat{\boldsymbol{x}}_0$ and covariance matrix $\boldsymbol{\Sigma}_0$.
  - An initial reference state $\boldsymbol{x}^0_{\mathrm{ref},0} = \hat{\boldsymbol{x}}_0$.
Outputs:
  - The IS estimator of the PoC, denoted $\bar{P}^{\mathrm{IS},N}_{T',\gamma'}$.
Procedure:

   1. Set $\mathring{d}_{\min} = \infty$ and $j = 0$

      while $\mathring{d}_{\min} > \gamma'$

   2. Use $\boldsymbol{x}^j_{\mathrm{ref},0}$ to compute $\boldsymbol{\epsilon}_{\mathrm{ref},k}$ in Eq. (3.16), and Jacobian $\boldsymbol{J}^{\mathrm{HF}}_k$ for node passages $t'_k$, $k = 1, \ldots, c$.

   3. Draw $N_2$ samples $\boldsymbol{x}^1_0, \ldots, \boldsymbol{x}^{N_2}_0$ from the distribution $\mathcal{N}(\boldsymbol{x}^j_{\mathrm{ref},0}, \boldsymbol{\Sigma}_0)$ and map them using the surrogate model of Eq. (3.17) to obtain $\tilde{\boldsymbol{\xi}}^1_k, \ldots, \tilde{\boldsymbol{\xi}}^{N_2}_k$.

   4. Compute $\mathring{d}^i_c = \min_{1\leq k\leq c} ||\bar{\boldsymbol{\xi}}^i_k||$ and $i_* = \operatorname{argmin}_{1\leq k\leq m} \mathring{d}^i_c$. then set $\mathring{d}_{\min} = \mathring{d}^{i_*}_c$.

   5. Let $\boldsymbol{x}^{j+1}_{\mathrm{ref},0} = \boldsymbol{x}^{i_*}_0$ and set $j \leftarrow j + 1$

      end while

   6. Plug $\boldsymbol{x}^j_{\mathrm{ref},0}$ into Eq. (3.18) to compute 'refined' condition $\boldsymbol{x}_{c,0}$. Construct the proposal function $\mu_0 \equiv \mathcal{N}(\boldsymbol{x}_{c,0}, \boldsymbol{C}_{c,0})$ where $\boldsymbol{C}_{c,0}$ is a user-defined covariance matrix.

   7. Set $\boldsymbol{x}_{\mathrm{ref},0} = \boldsymbol{x}_{c,0}$ and use $\Phi^{HF}_t$ to compute $\boldsymbol{\epsilon}_{\mathrm{ref},k}$ and $\boldsymbol{J}^{HF}_k$ associated to this state (where the subscript $k$, $1 \leq k \leq c$, indicates the relevant crossing with the line of nodes).

   8. Draw $N_1$ i.i.d samples $\boldsymbol{x}^i_0 \sim \mu_0(\boldsymbol{x}_0), i = 1, \ldots, N_1$ and map them using the approximate model of Eq. (3.17), yielding $\tilde{\boldsymbol{\xi}}^i_k$, $i = 1, \ldots, N_1$.

   9. Compute normalised importance weights $w^i = \frac{w^{*,i}}{\sum_{l=1}^{N_1} w^{*,l}}$, $i = 1, \ldots, N_1$, where $w^{*,i} = \frac{\pi_0(\boldsymbol{x}^i_0)}{\mu_0(\boldsymbol{x}^i_0)}$ are the non-normalised weights.

  10. Compute $\mathring{d}^i_c = \min_{1\leq k\leq m} ||\tilde{\boldsymbol{\xi}}^i_k||$ for $i = 1, \ldots, N_1$, then approximate the PoC as $\bar{P}^{\mathrm{IS},N_1}_{T',\gamma'} = \sum_{i=1}^{N_1} w^i I_{\gamma'}(\mathring{d}^i_c)$.
```

Once $\boldsymbol{J}_k^{HF}$ and $\boldsymbol{\epsilon}_{\text{ref},k}$ have been computed, one can generate samples around the reference state (line 3 of Algorithm 1) and map them (at low computational cost) using the approximate model of Eq. (3.17). The sample that comes closest to a collision (hence attaining the smallest $\mathring{d}^i$ in line 4) becomes the new reference state.

The final reference state, which itself leads to a collision, is refined in line 8 of Algorithm 1 to obtain an initial condition that yields a (nearly) zero distance between objects A and B at the time of collision. This is done by minimising the quadratic norm of $\tilde{\boldsymbol{\xi}}_k$ w.r.t. the initial condition $\boldsymbol{x}_0$. Since the model is a linear approximation (see Eq. (3.17)), it is ensured that there is always a solution. This is given by Eq. (3.18), and constitutes the state $\boldsymbol{x}_{c,0}$.

The importance function is selected as the Gaussian distribution $\mu_0 \equiv \mathcal{N}(\boldsymbol{x}_{c,0}, \boldsymbol{C}_{c,0})$, where the covariance matrix $\boldsymbol{C}_{c,0}$ can be selected in various ways, but should remain relatively narrow to ensure that a significant proportion of the samples drawn from μ_0 belong to the set $\mathcal{S}_{\gamma,0}$ that leads to a collision. To be specific, the covariance matrix should guarantee that the proposal generates samples $\boldsymbol{x}_0^i$ that, after mapping to the time of closest approach (TCA), remain at a distance which is of the same order of magnitude as the HBR. As a quantitative example, the diagonal blocks of the $\boldsymbol{C}_{c,0}$ matrix obtained in the LEO example of Section 3.4.2 are given by

$$\boldsymbol{C}_{c,0} = \begin{bmatrix} \boldsymbol{C}_{c,0}^{\boldsymbol{r}_A} & 0 & 0 & 0 \\ 0 & \boldsymbol{C}_{c,0}^{\boldsymbol{v}_A} & 0 & 0 \\ 0 & 0 & \boldsymbol{C}_{c,0}^{\boldsymbol{r}_B} & 0 \\ 0 & 0 & 0 & \boldsymbol{C}_{c,0}^{\boldsymbol{v}_B} \end{bmatrix} \tag{3.23}$$

and the traces of the covariance matrices for the positions $\boldsymbol{r}_q$ and velocities $\boldsymbol{v}_q$, where $q \in \{A, B\}$, are given by

$$\begin{aligned} \text{Tr}(\boldsymbol{C}_{c,0}^{\boldsymbol{r}_A}) &= 6.7351\,\text{m}^2, \\ \text{Tr}(\boldsymbol{C}_{c,0}^{\boldsymbol{v}_A}) &= 1.4810 \times 10^{-10}\,\text{m}^2/\text{s}^2, \\ \text{Tr}(\boldsymbol{C}_{c,0}^{\boldsymbol{r}_B}) &= 6.3426\,\text{m}^2, \\ \text{Tr}(\boldsymbol{C}_{c,0}^{\boldsymbol{v}_B}) &= 2.9635 \times 10^{-10}\,\text{m}^2/\text{s}^2. \end{aligned} \tag{3.24}$$

In order to check whether the proposal covariance is adequate or not, we can compare the standard deviation of the samples in the 2D plane with the collision threshold, given by $\gamma' = 1 \times 10^{-6}$. The standard deviations of the samples in $\mathbb{R}^2$ are $\sigma_{\Gamma_c} = 0.8867 \times 10^{-6}$ and $\sigma_{\Delta_c} = 0.9008 \times 10^{-6}$, and 99% of the samples generated by the proposal are contained in the rectangle

$$(-2\sigma_\Delta, 2\sigma_\Delta)\times(-2\sigma_\Gamma, 2\sigma_\Gamma) = (-1.8016\times10^{-6}, 1.8016\times10^{-6})\times(-1.7733\times10^{-6}, 1.7733\times10^{-6}).$$

All that remains is to draw N_1 importance samples from μ_0 and map them to the node of collision at t_k' using the approximate model of Eq. (3.17) (line 9 of Algorithm 1). Finally, we compute the importance weights in line 11 (this is straightforward using the prior pdf π_0 and the importance pdf μ_0), the 2D values $\mathring{d}_c^i$ for the N_1 samples and approximate the PoC. Note that, by construction of the surrogate 2D model, we expect $I_\gamma(\mathring{d}_c^i) = I_{\mathcal{S}_{\gamma,0}}(\boldsymbol{x}_0^i)$ for *most* initial conditions $\boldsymbol{x}_0^i$, i.e., $I_\gamma(\mathring{d}_c^i) = 1$ for states $\boldsymbol{x}_0^i$ leading to collision and $I_\gamma(\mathring{d}_c^i) = 0$ otherwise. This feature is numerically validated in Sections 3.4.2 and 3.4.3.

3.4 Results and discussion

3.4.1. Test cases

In this section, two test cases, a LEO encounter and a GEO encounter, are investigated to validate the proposed methodology. HF propagation is carried out using the HPOP library in MATLAB (see Section 2.4), which applies the Jacchia-Bowman 2008 drag model, as well as the remaining forces mentioned in said section. The numerical scheme used is MATLAB's ode113 suite. All simulations are run on a Linux machine with an Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz processor and 20 cores.

LEO

This case study is the well-known collision which occurred on 10 February 2009 between defunct USSR satellite Cosmos-2251 (object A) and operational communications satellite Iridium-33 (object B). The time of collision of the two satellites is $t_c \equiv$ 10 February 2009, 16:55:59 UTC, obtained from the observed conjunction geometry [63]. The states of both spacecraft at TCA containing the position and velocity expressed in ECI coordinates are

$$\boldsymbol{x}_{\mathrm{A,c}} = [\text{-1457.77km} \quad \text{1589.34km} \quad \text{6814.11km} \quad \text{-7.00km/s} \quad \text{-2.44km/s} \quad \text{-0.93km/s}]^\top$$

and

$$\boldsymbol{x}_{\mathrm{B,c}} = [\text{-1457.77km} \quad \text{1589.36km} \quad \text{6814.19km} \quad \text{3.58km/s} \quad \text{-6.17km/s} \quad \text{2.20km/s}]^\top.$$

The position and velocity uncertainties for the covariance matrix of the initial state are extracted from the consultative committee for space data systems historical archives. The covariance matrix is assumed equal for both objects and expressed in the cross-track, along-track, and out-of-plane (RTN) coordinate system, namely

$$\mathbf{\Sigma}_0^{RTN} = \begin{bmatrix} 41.44 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2533.00 & 0 & 0 & 0 & 0 \\ 0 & 0 & 70.98 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.74 \times 10^{-3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.05 \times 10^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.53 \times 10^{-5} \end{bmatrix} \begin{matrix} \mathrm{m}^2 \\ \mathrm{m}^2 \\ \mathrm{m}^2 \\ \mathrm{m}^2/\mathrm{s}^2 \\ \mathrm{m}^2/\mathrm{s}^2 \\ \mathrm{m}^2/\mathrm{s}^2 \end{matrix}$$

This matrix can be easily converted to ECI coordinates in order to match the coordinate system of both states. The assumed physical properties of the two satellites are detailed in Table 3.1.

Table 3.1: Physical properties of the spacecraft in the LEO conjunction case.

	Object A	Object B
Mass (kg)	689	800
$\mathbb{A}$ (m^2)	6.12	14.25
C_D (-)	2	2.5

To obtain suitable initial states for the proposed method, these conjunction conditions are propagated in HF and backwards in time for four days, then perturbed at t_0. The elements of the covariance matrix are then scaled by a factor of 0.1.

In order to detect collisions using the surrogate model of Section 3.2.4 it is convenient to define a collision threshold in 2D space. In LEO, for the given spacecraft altitudes and characteristic lengths (given by $\sqrt{\mathbb{A}}$), we set a HBR $\gamma = 10$ m which, using Eq. (3.8), yields a transformed threshold $\gamma' = \gamma / r_{A,c}(t_k) = 1 \times 10^{-6}$. For a given 2D state vector $\boldsymbol{\xi}_k$, a collision is detected when $\|\boldsymbol{\xi}_k\| < \gamma'$.

GEO

This case study is a simulated low-velocity collision between two GEO satellites (objects C and D). It is included to test the limits of the 2D metric proposed in this paper, since co-planar orbits can be problematic in the definition of the line of nodes. The TCA of the two satellites is $t_c \equiv$ 17 November 2016, 08:25:43 UTC, obtained from the simulated conjunction geometry. The states of both spacecraft at TCA expressed in ECI coordinates are

$$\boldsymbol{x}_{\mathrm{C,c}} = [42094.92\mathrm{km} \quad \text{-}1113.11\mathrm{km} \quad 2170.41\mathrm{km} \quad 0.08\mathrm{km/s} \quad 3.07\mathrm{km/s} \quad \text{-}0.02\mathrm{km/s}]^\top$$

and

$$\boldsymbol{x}_{\mathrm{D,c}} = [42094.92\mathrm{km} \quad \text{-}1113.10\mathrm{km} \quad 2170.41\mathrm{km} \quad 0.15\mathrm{km/s} \quad 3.04\mathrm{km/s} \quad \text{-}0.39\mathrm{km/s}]^\top$$

The initial covariance matrix in RTN coordinates,

$$\Sigma_0^{RTN} = \begin{bmatrix} 41.44 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2533.00 & 0 & 0 & 0 & 0 \\ 0 & 0 & 70.98 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.74 \times 10^{-3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.05 \times 10^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 5.53 \times 10^{-5} \end{bmatrix} \begin{array}{l} \mathrm{m}^2 \\ \mathrm{m}^2 \\ \mathrm{m}^2 \\ \mathrm{m}^2/\mathrm{s}^2 \\ \mathrm{m}^2/\mathrm{s}^2 \\ \mathrm{m}^2/\mathrm{s}^2 \end{array}$$

is selected to be the same as in the LEO scenario, and also equal for both objects. The orbital elements and physical properties of the spacecraft along with the orbital parameters for the computer simulations are listed in Table 3.2.

Table 3.2: GEO orbital parameters

	Object C	Object D
α (km)	42166.30	42000
e (-)	0.000059	0.0178
i (deg)	0.0517	0.1396
ω (deg)	4.5870	3.4907
Ω (deg)	0.4102	0.9514
ν (deg)	1.2593	1.8113
Mass (kg)	2040	3000
$\mathbb{A}_{\mathrm{SRP}}$ (m^2)	10	10
C_R (-)	1.2058	1.3

To obtain suitable initial states for the proposed method, these conjunction conditions are propagated backwards in time seven days with HF propagation and perturbed at t_0. The collision threshold in 2D space is $\gamma' = \gamma / r_{A,c}(t_k) = 2.38 \times 10^{-7}$.

3.4.2. Importance sampling: LEO

Performance and comparison with CMC

We compare Algorithm 1 with a CMC simulation on the basis of the PoC estimator, its standard deviation and the run-times of the two methods. The computational cost of Algorithm 1 for this scenario is relatively high, as the HF propagation of samples in this case involves calls to atmospheric models for drag computation which are costly. Nevertheless the number of HF samples to be propagated to obtain the proposed estimator is significantly lower than the number needed for the CMC simulation.

The results for this orbital configuration are summarised in Table 3.3, including a parallelised CMC simulation run with $N = 200,000$ samples, a number consistent with the expected PoC and a relative deviation of 10% [48]. The results for the calculation of the PoC with the proposed method using $N_1 = 100,000$ surrogate samples, out of which 66,848 collide, are compared with the PoC results of the CMC simulation. In addition, the computational run-times and the empirical standard deviation of the estimate, σ, are also displayed. To obtain σ for the IS estimator, Algorithm 1 (the IS surrogate) is (independently) run 100 times.

Table 3.3: Performance comparison for LEO conjunction: PoC estimates, run-time and standard deviation of the PoC estimator (σ).

	PoC estimate	Run-time (s)	σ
CMC simulation (HF)	1.3550×10^{-3}	2.0739×10^{6}	8.2254×10^{-5}
IS surrogate	1.2832×10^{-3}	14,926	5.9902×10^{-5}

From Table 3.3 it can be seen that the CMC method and the IS surrogate model of Algorithm 1 yield similar estimates of the PoC with sample standard deviations of the same order of magnitude. The proposed scheme, however, attains these results with a computational cost which is two orders of magnitude smaller.

Figure 3.5 shows the collection of samples generated by Algorithm 1, using the proposal function μ_0, after they are mapped to the 2D space at the crossing with the line of nodes where the collision takes place. The samples actually yielding a collision are depicted in red, while the samples *not* leading to a collision are depicted in green. It is clearly seen that the IS surrogate generates a large proportion of the initial samples inside the event threshold at TCA and hence, a large proportion of the simulated samples correspond to a collision event. Also, there is a significant proportion of samples close to the threshold γ' (both below and above) which enables the accurate estimation of the PoC.

Surrogate model validation

We can interpret the approximate propagation of a sample $\boldsymbol{x}_0^i \sim \mu_0$ using the model in Eq. (3.17) and the detection (or not) of a collision as a test. Specifically, given the initial condition $\boldsymbol{x}_0^i$ we first compute $\tilde{\boldsymbol{\xi}}_k^i$ using Eq. (3.17) and then the associated metric $\mathring{d}_c^i$ using Eq. (3.21). If $\mathring{d}_c^i < \gamma$ the result of the test is positive and we declare a collision for the ith sample $\boldsymbol{x}_0^i$. Otherwise, if $\mathring{d}_c^i > \gamma$, the result is negative (no collision). However, since the propagation model of Eq. (3.17) is only approximate, the test $\mathring{d}_c^i \gtrless \gamma$ may yield an erroneous result. There are two possible errors:

(1) A sample $\boldsymbol{x}_0^i \in \mathcal{S}_{\gamma,0}$ that actually leads to a collision (when propagated with HF), yields $\mathring{d}_c^i > \gamma$ with Algorithm 1 and hence we fail to detect the collision. This error is usually termed a misdetection or a false negative.

(2) A sample $\boldsymbol{x}_0^i \notin \mathcal{S}_{\gamma,0}$ that does not lead to a collision (when propagated with HF) yields $\mathring{d}_c^i < \gamma$ with Algorithm 1, and hence we declare collision. This error is usually termed a false alarm or a false positive.

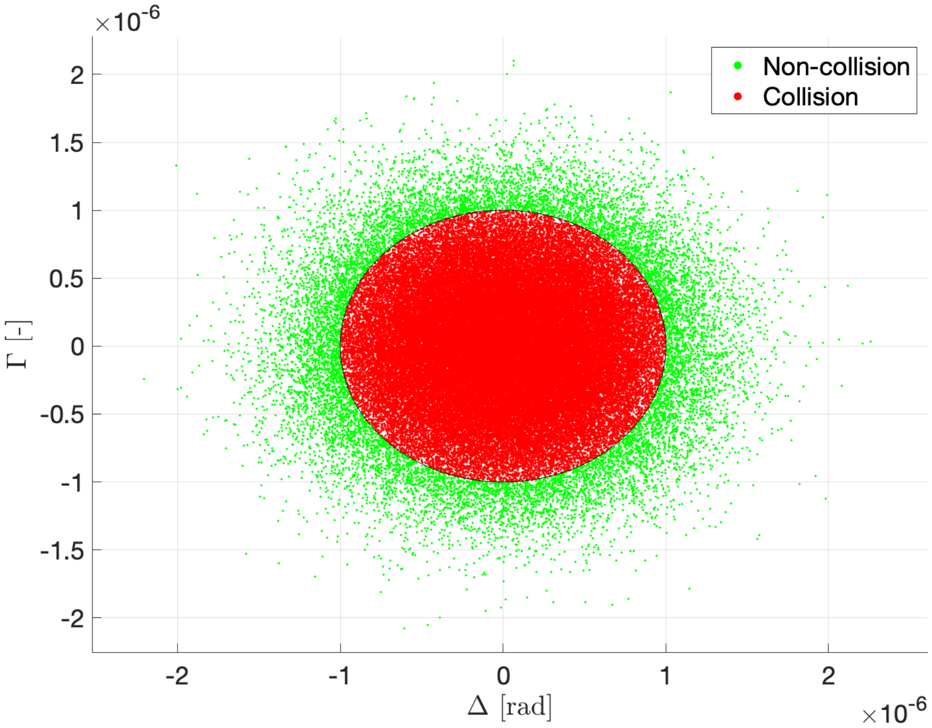


Figure 3.5: Conjunction mapping showing the defined 2D metric for the proposed model samples in LEO. The collision region is the circle centred at $(\Delta, \Gamma) = (0, 0)$ and radius $\gamma' = 10^{-6}$.

It is of interest to assess the performance of the approximate propagation model in Eq. (3.17) in terms of how often the collision test $\mathring{d}_c^i \gtrless \gamma$ results in an error of either type.

Detecting a false alarm or misdetection for the ith sample can only be done by propagating the ith sample $\boldsymbol{x}_0^i$ with HF, which is a costly operation. However, given the good agreement between the HF 2D mapping $\boldsymbol{\xi}_k^{i,HF}$ and the approximate 2D mapping $\tilde{\boldsymbol{\xi}}_k^i$, it is sufficient to only compute the HF vector $\boldsymbol{\xi}_k^{i,HF}$ for the samples which are close to the detection threshold γ' in Figure 3.5.

Figure 3.6a) displays the result of performing a HF propagation of the 450 initial samples closest to the threshold that were declared positive (i.e., leading to a collision). The green dots represent the 2D HF projections which are still within the collision area, while the red crosses indicate the samples for which the 2D metric is bigger than the threshold γ' when computed with HF. We observe that the number of false alarms (red crosses) is small relative to the number of true positives (green dots), which indicates a low false alarm rate.

Similarly, Figure 3.6b) shows the result of performing a HF propagation of the 450 initial samples closest to the threshold that were declared negative (i.e., no collision). The green dots represent the 2D projections which remain outside the collision area after HF propagation, while the red crosses indicate the samples which actually yield a collision. The number of false negatives (red crosses) is again small relative to the number of true negatives (green dots). This indicates a low misdetection rate.

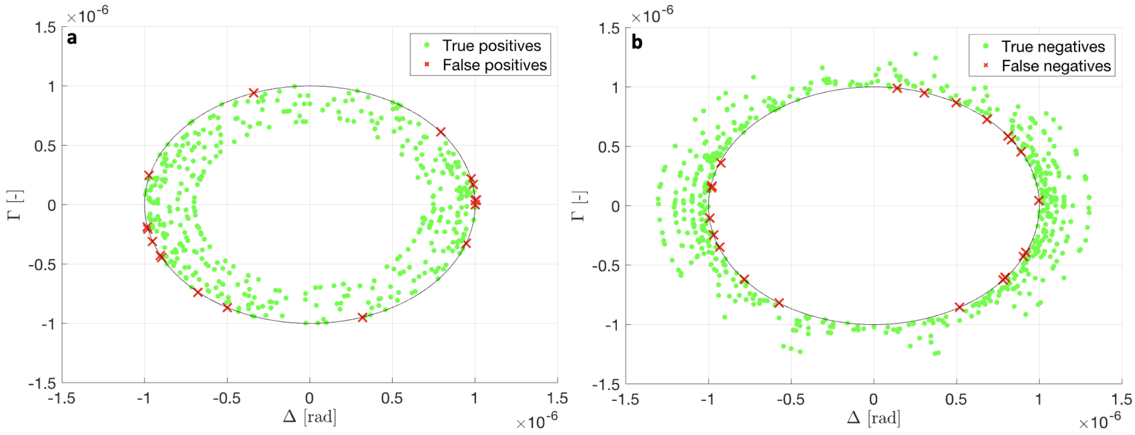
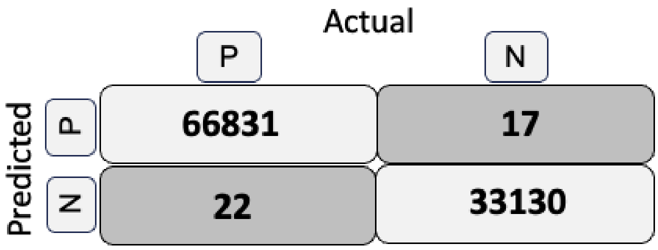


Figure 3.6: a) The HF mapping of the samples predicted by the proposed method to collide. b) The HF mapping of the samples predicted by the proposed method to not collide. In both cases, a false positive/negative (indicated by crosses) division can be observed. The collision region is the circle centred at $(\Delta, \Gamma) = (0, 0)$ and radius $\gamma' = 10^{-6}$.

The confusion matrix in Figure 3.7 displays the estimated error rates. Out of 100,000 samples, there are 66,831+22 actual collisions (HF) and 17+33,130 non-collisions (HF). Out of the total number of collisions, there are 22 misdetections, which yields a misdetection rate of $\frac{22}{66{,}853} = 3.291 \times 10^{-4}$. Similarly, the false alarm rate is $\frac{17}{33{,}147} = 5.129 \times 10^{-4}$. Let us remark that all misclassified samples in Figures 3.6a) and 3.6b) are very close to the threshold. This indicates that, in practice, one can significantly reduce the misdetection rate by artificially inflating the threshold γ', at the expense of an increase in the false alarm rate.



		Actual	
		P	N
Predicted	P	66831	17
	N	22	33130

Figure 3.7: Confusion matrix outlining the number of true positives, false positives, true negatives and false negatives obtained with Algorithm 1. The false alarm rate is $\frac{17}{33{,}147} = 5.129 \times 10^{-4}$ and the misdetection rate is $\frac{22}{66{,}853} = 3.291 \times 10^{-4}$.

Spatial distribution

We can visualize the proposal distribution samples in Cartesian coordinate pairs at t_0' in Figures 3.8 and 3.9 for the LEO case. The states which lead to collision appear in red, while those which do not lead to collision appear in blue. Operationally, one may examine the proximity to collision states in each dimension to assess whether or not a collision is likely within some temporal horizon.

Although these figures may appear to show a lack of continuity among colliding states, showcased by the apparent lack of convexity in certain areas of the collection of colliding samples, this is likely an illusion caused by the dimensionality of the problem, as a single 12-dimensional continuous set is difficult to illustrate in dimension pairs as represented in these figures.

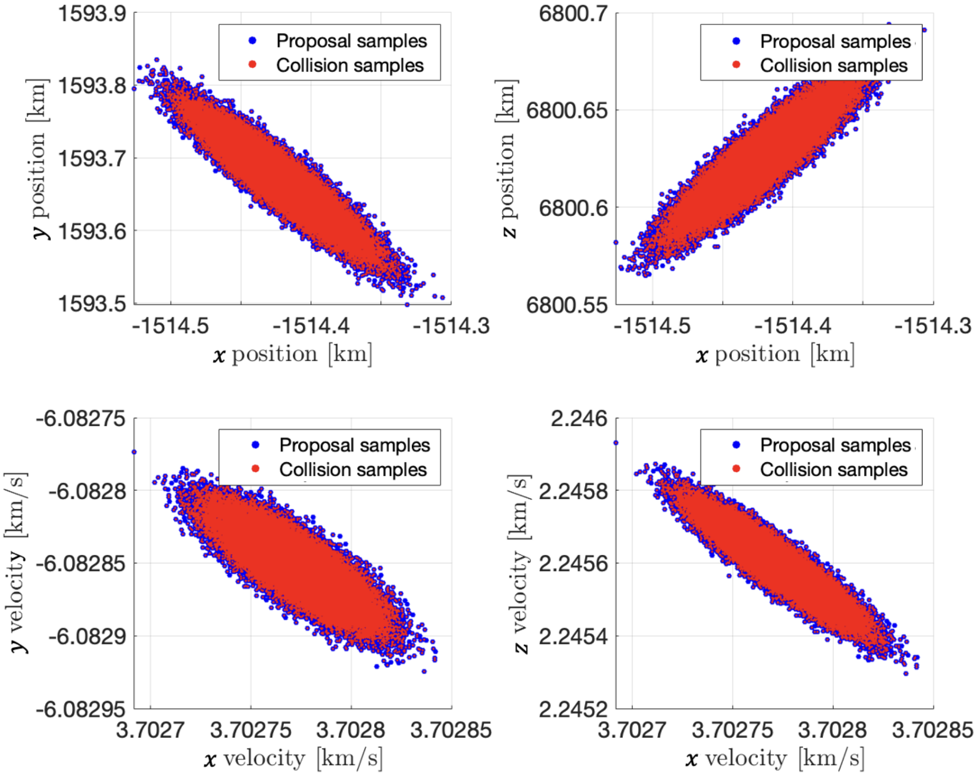


Figure 3.8: States of object A at t_0' drawn from the proposal distribution (blue), and the subset of these which end up in collision at TCA (red).

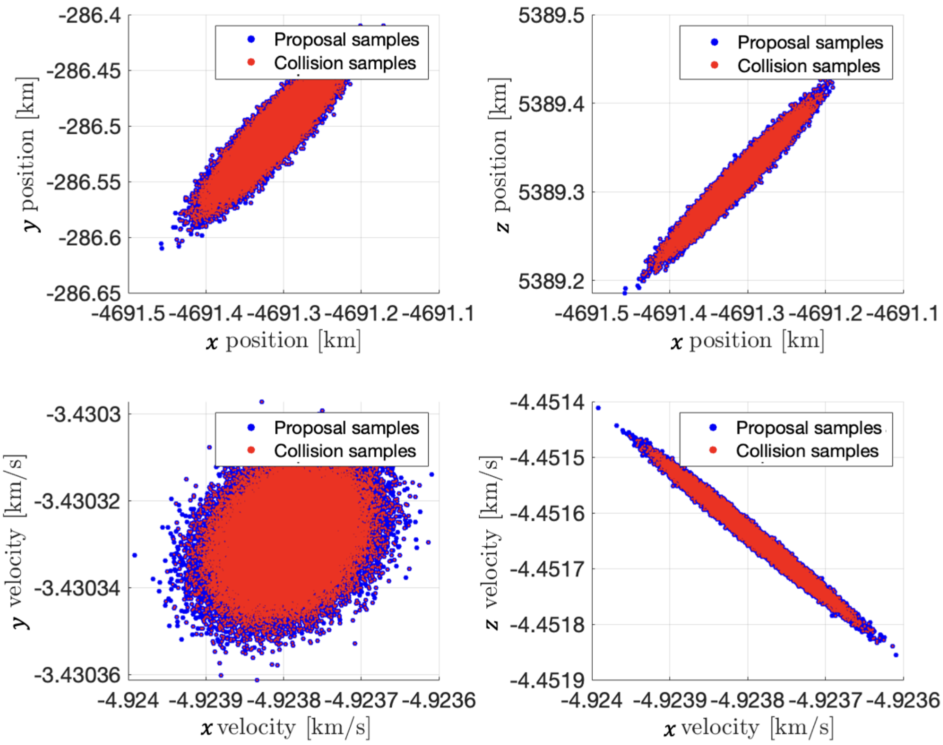


Figure 3.9: States of object B at t_0' drawn from the proposal distribution (blue), and the subset of these which end up in collision at TCA (red).

3.4.3. Importance sampling: GEO

Performance and comparison with CMC

The performance of Algorithm 1 is compared with a CMC simulation on the basis of the PoC estimator and its standard deviation, and the run-times of both methods. The computational cost for the HF propagation of a GEO object is lower than in the LEO case because it does not involve atmospheric drag term computations. In addition, due to a larger semi-major axis, the number of passes is lower, so the calculation of $\boldsymbol{J}^{HF}$ is significantly more efficient.

The results for this orbital configuration are summarised in Table 3.4, including a parallelised CMC simulation with $N = 1,150,000$ samples. This number is consistent with a relative deviation in the PoC estimator of at most 10% [48]. The results for the calculation of the PoC of the proposed method using $N_1 = 100,000$ surrogate samples, out of which 88,123 collide, are compared with the PoC results of the CMC simulation. In addition, the computational run-times and the empirical standard deviation of the estimate, σ, are also displayed.

From Table 3.4, it can be seen that both estimators yield similar PoC estimates with similar standard deviations. However, the computational run-time of the proposed method is two orders of magnitude lower than the run-time of the CMC simulation. Both methods have been run on the same platform.

Table 3.4: Performance comparison for the GEO conjunction: PoC estimates, run-time and standard deviation of the PoC estimator (σ).

	PoC estimate	Run-time (s)	σ
CMC simulation (HF)	1.7913×10^{-4}	51,580	1.3059×10^{-5}
IS surrogate	1.9011×10^{-4}	401	1.5706×10^{-5}

Figure 3.10 shows the collection of samples generated by applying the IS surrogate of Algorithm 1, using the proposal function μ_0, after they are mapped to the 2D space at the crossing with the line of nodes where the collision takes place. The samples actually yielding a collision (with the 2D metric below the collision threshold) are depicted in red, while the samples *not* leading to a collision are depicted in green. It is clearly seen that Algorithm 1 generates a large proportion of the initial samples inside the event threshold and, hence, a large proportion of the simulated samples correspond to a collision event. Also, there is a significant proportion of samples close to the threshold (both below and above) which enables the accurate estimation of the PoC.

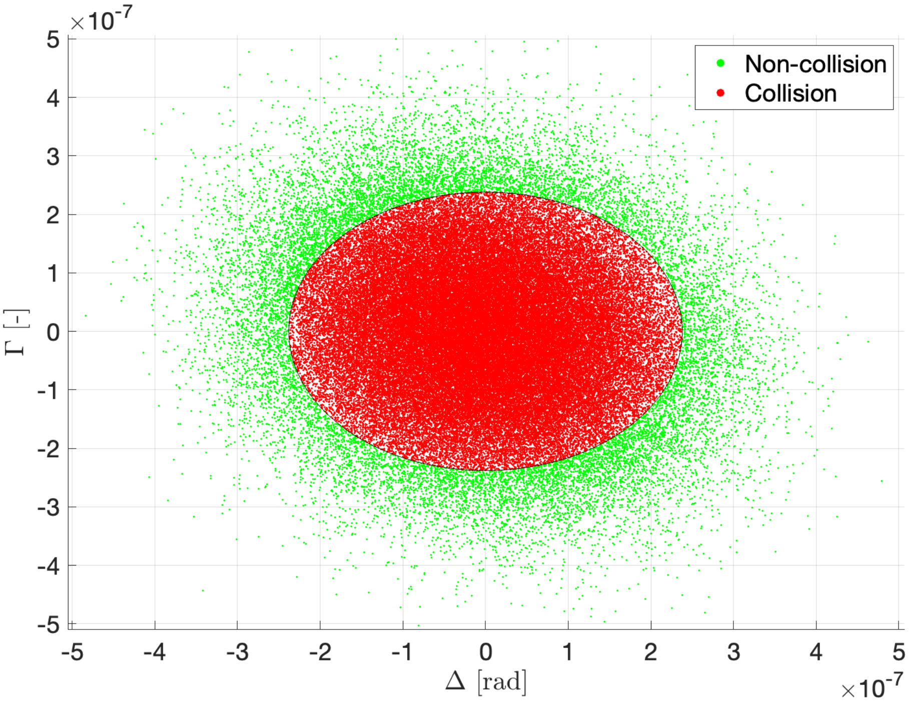


Figure 3.10: Conjunction mapping showing the 2D metric for the proposed model samples in GEO. The collision region is the circle centred at $(\Delta, \Gamma) = (0, 0)$ and radius $\gamma' = 2.38 \times 10^{-7}$.

Surrogate model validation

We study the reliability of the PoC computation method by looking into the false alarm and misdetection rates, in the same way as for the LEO test case.

Figure 3.11a) displays the result of performing a HF propagation of the 450 initial samples closest to the threshold that are declared positive (i.e., collisions). The green dots represent the 2D HF projections which are still within the collision area, while the red crosses indicate the samples for which the 2D metric is bigger than the threshold γ' when computed with HF. We observe that the number of false alarms (red crosses) is small relative to the number of true positives (green dots), which indicates a low false alarm rate (although greater than in the LEO scenario).

Similarly, Figure 3.11b) shows the result of performing a HF propagation of the 450 initial samples closest to the threshold that were declared negative (i.e., no collisions). The green dots represent the 2D projections which remain outside the collision area after HF propagation, while the red crosses indicate the samples which actually yield a collision. The number of false negatives (red crosses) is relatively small compared to the number of true negatives (green dots). It is clear, however, that the proportion of misdetections is greater than the proportion of false alarms. A plausible cause for this difference is that the 2D mapping in Section 3.2.3 is better suited to non-co-planar orbits –while GEO orbits tend to lie on similar planes, and hence the line of nodes may be hard to detect or even undefined.

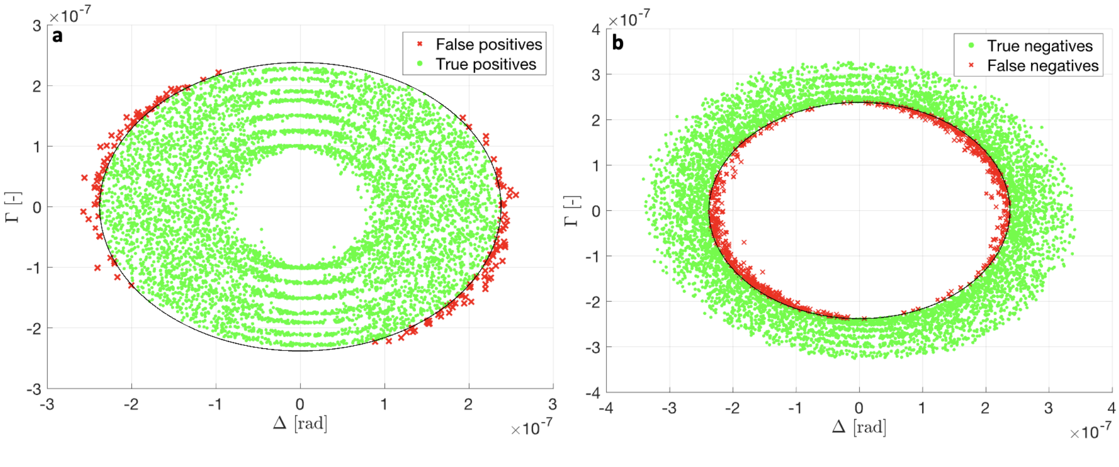
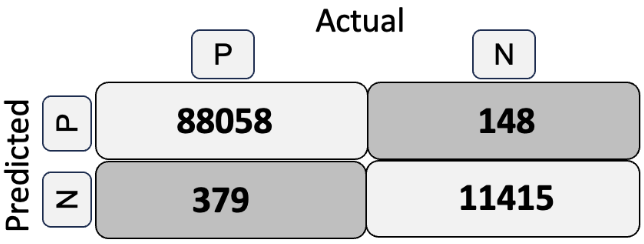


Figure 3.11: a) The HF mapping of the samples predicted by the method to collide. b) The HF mapping of the samples predicted by the method to avoid collision. In both cases, a false positive/negative (indicated by crosses) division can be observed. The collision region is the circle centred at $(\Delta, \Gamma) = (0, 0)$ and radius $\gamma' = 2.38 \times 10^{-7}$.

The confusion matrix in Figure 3.12 displays the estimated error rates. Out of 100,000 samples, there are 88,058+379 actual collisions and 148+11,415 non-collisions. Out of the total number of collisions, there are 379 misdetections, which yields a misdetection rate of $\frac{379}{88{,}437} = 0.004$. Similarly, the false alarm rate is $\frac{148}{11{,}563} = 0.013$.



		Actual	
		P	N
Predicted	P	88058	148
	N	379	11415

Figure 3.12: Confusion matrix outlining the ratio of true positives, false positives, true negatives and false negatives between the predicted and the HF values.

Spatial distribution

We can visualize the proposal distribution samples in Cartesian coordinate pairs at t_0' in Figures 3.13 and 3.14 for the LEO case. The states which lead to collision appear in red, while those which do not lead to collision appear in blue. Operationally, one may examine the proximity to collision states in each dimension to assess whether or not a collision is likely.

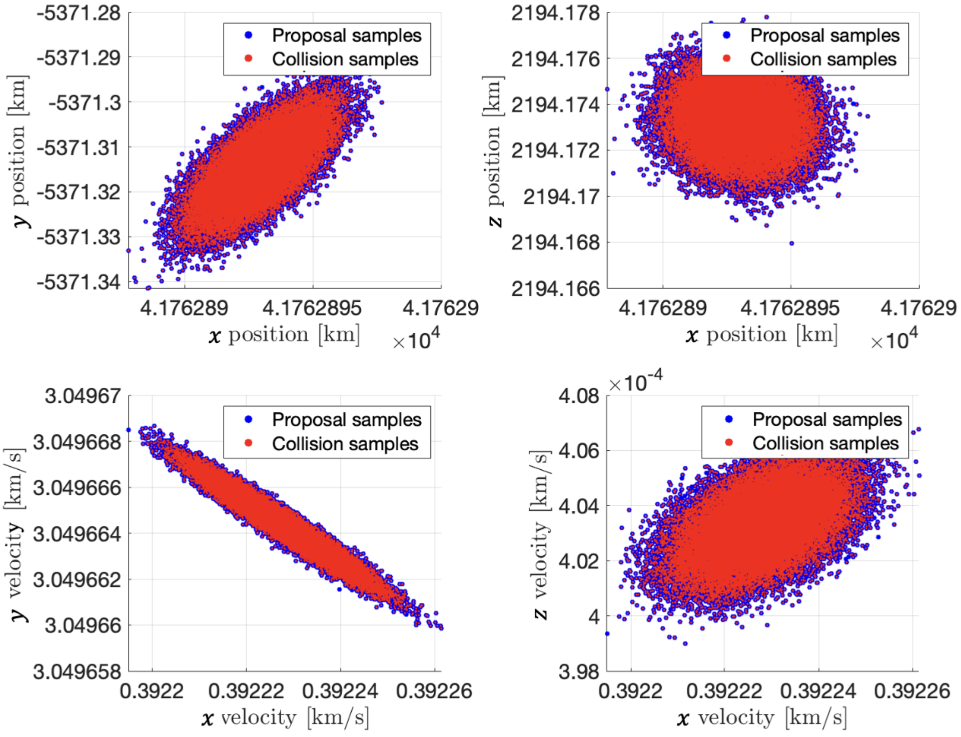


Figure 3.13: States of object C at t_0' drawn from the proposal distribution (blue), and the subset of these which end up in collision at TCA (red).

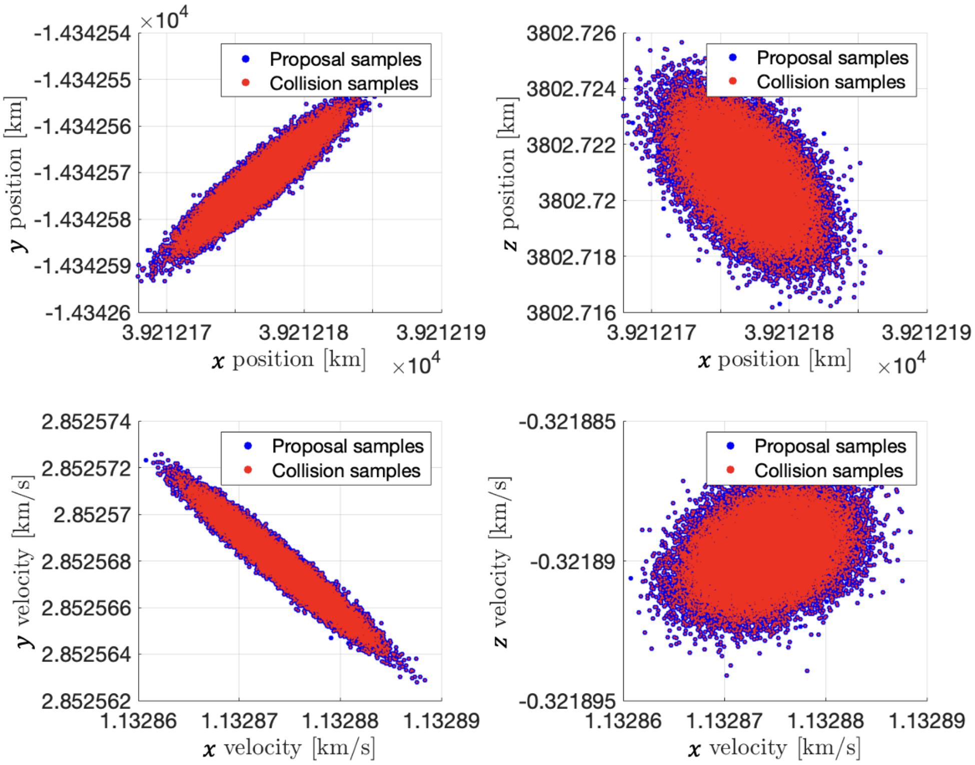


Figure 3.14: States of object D at t_0' drawn from the proposal distribution (blue), and the subset of these which end up in collision at TCA (red).

Although a priori, these figures may seem to show a lack of continuity, showcased by the apparent lack of convexity in the collection of colliding samples, this is likely explained by the difficulty in illustrating a continuous set in 12 dimensions.

3.5 Concluding remarks

The PoC computation method proposed in this chapter builds upon a novel collision risk metric for space objects. The aim is to bypass the need for simplifying assumptions involving their state distributions and the relevant dynamics at collision. This could satisfy the need to provide reliable safety guidelines for satellite operators for the prevention of collisions, as well as unnecessary collision avoidance manoeuvres.

The new technique relies on a reduced-dimension orbital propagation model that involves (i) a convenient metric for the risk of collision that can be evaluated in a 2D space and (ii) a suitable threshold for this metric which enables the detection of collisions. Based on this 2D model, the PoC is approximated using importance sampling. A key ingredient is the construction of an importance function that can be used to generate a significant proportion of state samples which lead to collision, enabling the approximation of potentially small PoCs with a relatively small number of samples. This importance function may have any form, though a Gaussian proposal is proposed herein due to its ease of computation. The proposal is, however, based on states $\mathbf{x}_0$ (at time t'_0) which end up under the collision threshold after being mapped to TCA, but these propagated samples are no longer Gaussian-distributed. Therefore, no Gaussianity assumption is made on the states at TCA.

We show how the new method attains similar or better accuracy than a CMC simulation with a fraction of the computational cost in two different scenarios –including a LEO conjunction and a GEO conjunction. We also illustrate the reliability of the new method by numerically studying the false alarm and misdetection rates. In both test cases, the two error rates are relatively low, although the performance in the LEO test is clearly better. In the GEO test case, the orbits of the two spacecraft are nearly co-planar and this makes the detection of the line of nodes harder (this detection is a key step of the proposed method). Nevertheless, the misdetection rates remain of order 10^{-4} in LEO and 10^{-3} in GEO, and can be further reduced (at the expense of an increase in false alarms).

It is important to note that similar instantaneous angular velocities between the objects are assumed in the derivation of the collision condition for the 2D mapping. This assumption is not required for the general application of the approximate model, though it simplifies the derivation for the most common scenarios. The method can still be applied to handle cases with differing angular velocities, but the evaluation of the collision condition may become more complex. Future work could focus on exploring more general scenarios to enhance and validate the flexibility of the methodology. Another thread for future research could be the incorporation of uncertainties in the parameters of the dynamical model, e.g., the solar radiation pressure. This can be done in a straightforward way by extending the current 12D state vector into a 14D vector (for one unknown parameter) or a 16D vector (for two unknown parameters). The IS surrogate of Algorithm 1 can be applied to construct a proposal function on the extended state space and then estimate the PoC by importance sampling. Another feasible approach is the incorporation of uncertainties into a process noise term applied to the dynamics of the surrogate model. In either case, research would be needed to analyse the effects of the increased uncertainty, e.g., whether it may lead to a dilution of the PoC.

The successful application of any method which computes the PoC is inherently reliant on the initial conditions uncertainty. Probability dilution describes the phenomenon by which poor knowledge of the spacecraft states may lead to a low PoC, due to the dilution effect of an enlarged covariance matrix on the computed probability of collision. Therefore, a crucial factor in performing reliable PoC calculations is the acquisition of accurate orbit determination data in the first place. To do this, tracking methods must sufficiently capture the uncertainties associated with state estimations and, where possible, reduce them. The next chapter deals with the development of algorithms to accomplish this.

4

SPACECRAFT TRACKING

Ship of white light in the sky
Nobody there to reason why
Here I am, I'm not really there
Smiling faces ever so rare

Iron Maiden
Strange World (1980)

Chapter 1 outlines the growing problems faced in STM and the requirements for a safe near-Earth space environment, which specifically include the improvement of tracking capabilities. This chapter is dedicated to the development of algorithms which aim to fulfill the need for efficient, yet accurate and informative RSO tracking.

A review of the state of the art is hereby included, delving into the types of filtering methods that have been proposed over the years, followed by two separate parts. The first one postulates a set of sequential filters which are described and assessed in regards to the task of accurately estimating the state of a spacecraft in the presence of irregularly spaced measurements. Some of these filters are commonly used, both in academic work and in industry, and others are somewhat unknown in the astrodynamics community but are popular in other fields. The second part introduces variations of some of these algorithms, as well as the development of new ones, which aim to sequentially estimate the state, but also perform stochastic parametrisation, which constitutes the characterisation of process noise to improve uncertainty representation.

4.1 State of the art

4.1.1. State-only tracking

Orbit determination

Tracking and cataloguing RSOs are crucial aspects of space surveillance and tracking (SST) services such as reentry prediction or conjunction assessment. The data processing core of their operations is based on OD methods which are able to provide orbital state and uncertainty estimates given observational data. Although the target tracking problem has a long history in many different applications, OD is challenging due to its unique characteristics. Indeed, methods aimed at estimating the state of an RSO from measurements of radar, electro-optical sensors, laser, or other sensors (generally not on-board sensors) have to deal with sparse data and nonlinear dynamical and measurement models.

OD methods can be classified into three broad categories:

- batch least-squares estimators[64], where all observations are processed together, yielding a solution through iterative fitting;
- sequential filters [65], where the orbital model parameters are updated with each new observation;
- recursive least squares, or batch sequential filters (BSF) [66][67], which combine batch least squares and sequential filter properties.

In this thesis, the focus lies on sequential filtering because of its potential to yield efficient online OD. More specifically, within sequential filters, we identify three popular families: Gaussian filters (starting with [68]), particle filters [69] and hybrid particle filters [70][71], depending on the way the prediction (in which the prior estimate is propagated to the next observation) and update steps (in which the prior estimate is corrected with new observational data) are performed.

Filtering algorithms

The Kalman filter (KF), originally proposed by Rudolph Kalman in 1960 [68], is a common tool in various fields such as electrical engineering, signal processing, econometrics and navigation and guidance systems. It provides an optimal solution to linear systems in the presence of additive Gaussian noise, but is not directly applicable to nonlinear systems, for which approximate methods such as the extended Kalman filter (EKF) [72], the unscented Kalman filter (UKF) [73] and the cubature Kalman filter (CKF) [74] are more appropriate. The EKF, which introduces linearisations for the transition or observation functions is compared with a least squares algorithm (LSQ) in [75], demonstrating that although LSQ is more robust, the EKF is more efficient in real-time. The EKF is also regularly studied as a means to compare the performance of more advanced algorithms [76] [77]. Some of the more relevant papers include [78] and [79]. The authors of [78] study the dependence of the performance on the accuracy of the numerical Jacobian and determine it to be a suitable method under limited criteria. The authors of [79] apply the EKF to LEO satellites while making use of precise gravity models. They claim to achieve radial and cross-track position errors of less than 100 m, as it is usually the along-track observability which degrades in LEO orbits.

The UKF (e.g., [73]) and the CKF [74] propagate a set of sigma (or cubature) points, improving probability distribution representation after nonlinear transformations and bypassing the need for linearisations. CKF methods include a third-degree flavour (3CKF), comparable to the UKF, and the fifth-degree flavour (5CKF), applied in [80] on a LEO scenario and shown to provide higher accuracy than the 3CKF. Neither of the cubature methods is common in OD. They have, however been applied in RSO tracking works such as [81], which applies a CKF to tracking in the presence of GPS observations over a period of 24 hours. The authors determine that the algorithm is sensitive to the computation of the covariance matrix, and that the filter diverges when the re-observation times are large enough (observations are sparse). A comparison between the CKF and the UKF can be found in [82]. In all Kalman filters, the Kalman gain matrix maps residuals to corrections of the initial estimate, balancing the weight of observational data and model dynamics.

Unlike Kalman-style methods, particle filters (PFs) can account for probability distributions which are possibly non-Gaussian. PFs are recursive Monte Carlo methods that can be used to numerically approximate the sequence of posterior probability distributions of the state given a set of observations [83][84][85]. The basic theory of particle filtering deals with dynamic variables only; hence all other parameters are assumed to be known. The simplest form of PF is the bootstrap particle filter (BPF) [69][86], where Monte Carlo samples at each time are drawn from the state distribution given by the dynamical model. However, customised proposal distributions for the generation and propagation of samples can be constructed to improve convergence [87]. Particles are then allocated importance weights according to their likelihood. Resampling of the particles using these weights is a fundamental step in particle filtering [88]. However, weight degeneracy (a phenomenon which occurs when one single particle accumulates most of the weight [89]), can easily lead to sample impoverishment after the resampling step. Ref. [90] analyzes this issue in PFs, and proposes strategies like regularisation and resampling thresholds to mitigate it. Resampling is not needed at every time step; one can choose to resample once a threshold is reached. This is often done by evaluating the effective sample size, which is an estimate of the variance of the (non-normalised) weights [85]. The downsides associated with weight degeneracy can often be challenging to mitigate without incurring a large computational effort. Ref. [91] reviews various PF implementations for space object tracking, emphasising adaptability to non-Gaussian uncertainties and multi-modal distributions. The authors demonstrate significant performance advantages over the UKF in different orbital scenarios.

However, PFs have not been extensively used in the orbit determination domain. Some of the reasons for this include the following:

1. PFs require a substantial number of particles to achieve adequate accuracy (around three orders of magnitude larger than the state dimension [92]), which incurs a significant computational cost. The associated computational run-time issue can be alleviated partly by the use of parallel computing.

2. Due to the quasi-deterministic nature of astrodynamics, PFs are prone to suffer from weight degeneracy and particle impoverishment, which require elaborate modifications or clever resampling criteria and techniques [93].

However, prominent studies of PFs in RSO tracking include [94], where “roughening" as well as prior editing are proposed to deal with the issue of sample degeneracy and impoverishment of the BPF in LEO and conclude that adding more particles may bring more accuracy but may also increase the rate of sample impoverishment. In [95], the authors apply a regularised PF (RPF) to process laser ranges as observations. They use simulated data with probabilistic initial orbit determination (IOD), and show the RPF to converge if the initial particle cloud has sufficiently many particles (around 2,000). In addition, works like [76] highlight significant improvements in PF accuracy (with errors around 25% lower than the EKF) in cases with large initial uncertainties, while [96] uses a PF for automatic maneuver detection, where each particle represents a possible spacecraft state under various maneuver hypotheses, allowing the algorithm to probabilistically infer the most likely maneuver scenario.

The ensemble Kalman filter (EnKF) [97] combines Monte Carlo sampling with Kalman updates to construct a randomly distributed ensemble, making it popular for high dimensional problems. It aims at achieving substantial coverage of the state space, whilst propagating equally-weighted particles and, hence, not requiring sample-weighing and resampling steps. It is therefore often more robust than PFs in cases where the latter suffers from weight degeneracy. The authors of [98] apply an EnKF to spacecraft tracking in LEO with simulated observations and a covariance inflation parameter, though they achieve errors in position which are considerably larger than the errors attained by the UKF. Ref. [99] applies Gaussian mixture models (GMM) to IOD, enabling probabilistic representations of uncertainties, while [100] introduces a kernel-based Gaussian mixture filter that combines ensemble sampling with non-parametric density estimation. These approaches significantly improve robustness and accuracy with limited data. Other hybridisations include works like [101], which combine the UKF with a PF to manage non-Gaussian uncertainties, achieving a balance between computational efficiency and estimation accuracy which outperforms the standalone UKF and PF by up to 40% in simulated scenarios. The authors of [102] propose the ensemble cubature Kalman filter (EnCKF), which merges the EnKF and the CKF's individual advantages, i.e., their advantages when dealing with high dimensional integrals and nonlinear settings. They show that when the number of ensemble members is larger than the state dimension, the EnCKF is expected to be the most computationally expensive among the CKF, EnKF, and EnCKF [102]. The EnCKF has not been used in astrodynamics, to date.

4.1.2. State tracking and unknown parameter estimation

For all OD methods, estimation performance depends on the correct characterisation of the process and measurement noise. In fact, the development of optimal estimators in the Bayesian sense (i.e. those that solve the problem of the exact and complete characterisation of the posterior pdf) is based on the knowledge of suitable models for the dynamical and measurement noise. In the case of KFs in linear time invariant systems, the identification of noise covariances is a problem that has existed for 50 years [103], and is not yet fully solved. The problem of dealing with inexact knowledge of the noise present in the model can be formulated as follows: Given a vector time series and a library of models of system dynamics, find a suitable process and measurement noise model and the best system dynamics for the time series. The traditional classification of methods [104] for dealing with this problem is summarised in Table 4.1, where MPE stands for measurement prediction error.

Table 4.1: Methods for noise characterisation.

	State and noise simultaneous estimation	**Noise-only estimation**
Probability-based	Bayesian inference	Maximum likelihood estimation
Statistical analysis of MPE	Covariance matching method (CMM)	Correlation methods

Methods that deal with noise characterisation alone are based on the solution of a sub-optimal state estimator, whilst algorithms that deal with noise and state estimation simultaneously do so by augmenting the state with parameters that allow the noise to be characterised in some way. Note that noise may be characterised, but not estimated. It is possible, however, to estimate the variance of noise or the diffusion coefficient of a stochastic differential equation. It has been shown that estimators in this type of problem are necessarily nonlinear [105], even in the case of linear dynamic and measurement models, given the nonlinear relationship between the elements of the augmented state. It is therefore necessary to use nonlinear filters (such as PFs or hybrid schemes) in this type of problem. Moreover, the inclusion and study of process noise in orbital dynamics has received relatively little attention in the astrodynamics community [106]. This process noise is commonly accounted for by scaling or inflating its covariance matrix throughout the propagation, but the magnitude of the scaling is often rather arbitrary.

To achieve realistic covariance estimates, refined uncertainty quantification techniques are essential. Works such as [107] and [108] carry out innovation-based approaches for updating the noise covariance during filtering over a sliding time window and use KF variations which incorporate empirical accelerations modelled as first-order Gauss-Markov processes. These methods involve estimating and incorporating uncertainties from either observations or dynamical models into the OD process, enhancing the accuracy of the filter. Other works achieve this by employing consider parameter analysis; e.g., [109] and [110] augment the process noise covariance matrix to more appropriately represent uncertainty in the time update phase.

Section 4.2 describes the methods introduced at the beginning of this chapter, which are capable of tracking the state of a spacecraft along its orbit. Results of this part of the methodology can be found in Section 4.5. In Section 4.3, some of the afore-mentioned methods are expanded to provide a framework with which to simultaneously estimate unknown parameters and track RSO states. To deal with the problem of characterising uncertainty online, the parameter to be estimated is chosen to be the stochastic diffusion coefficient σ_W in Eq. (2.15), which represents the scale of the process noise. However, other parameters can be estimated with this modified methodology, such as the ballistic coefficient in LEO, solar radiation pressure in GEO, or bias in the measurement equation. Results of this part of the methodology can be found in Section 4.7.

Most of the methods studied in this chapter can be seen in Figure 4.1, where the gray boxes contain methods which estimate the state of the spacecraft alone, and the blue boxes contain methods which can either estimate the state alone or simultaneously track the state as well as unknown parameters.

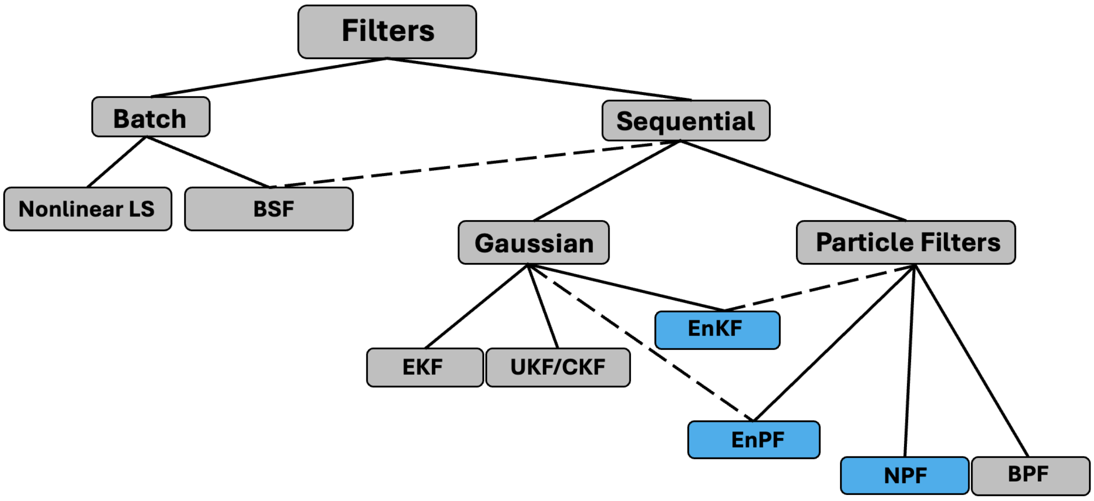


Figure 4.1: Hierarchy of the main types of filters used in OD and RSO tracking. The main groups are batch and sequential methods, each spanning their sub-types. Filters which are able to track the state alone are shown in gray boxes. The algorithms shown in the blue boxes are methods which can also be used to simultaneously track the state and unknown parameters.

4.2 Methodology: state-only tracking

4.2.1. Gaussian sequential filters

Kalman filter

The rationale behind the KF serves as the foundation for its many variants, some of which are discussed in this chapter. This filter works under the assumption that the posterior density at each time-step follows a Gaussian distribution, which can be fully described by its mean and covariance [111]. State estimation is performed by computing a weighted average of the predicted state and the predicted measurements, adjusting the confidence given to each based on the covariance values. This weighing mechanism is contained in the Kalman gain, which is designed to minimise estimator error covariance at the kth observation time [102][112]. Given the probabilistic state-space equations introduced in Section 2.5, and the time-grid $\boldsymbol{T} = \{t_k\}_{k=1}^{M}$ corresponding to observation time-stamps (introduced in Section 2.6), the KF follows a two-step iterative process: prediction and update, both of which are applied sequentially over time.

- **Prediction stage:** The system dynamic equation is used to propagate the state from the previous time-step, t_{k-1}, to generate a predicted state at the current time-step, t_k. Additionally, the process noise vector $\boldsymbol{\upsilon}_{k-1}$, which is assumed to be normally distributed with zero mean and covariance matrix $\boldsymbol{Q}_{k-1}$, is incorporated.

- **Update stage:** The predicted state is used to compute the expected measurement. The discrepancy between this predicted measurement and the actual measurement is then multiplied by the Kalman gain and combined with the prior state to produce the a posteriori state estimate, denoted $\bar{\boldsymbol{x}}_k$.

Due to the inherently nonlinear nature of orbital dynamics, linear approximations tend to degrade in performance over time and cannot maintain accuracy for extended periods. For systems exhibiting only mild nonlinearity, the extended Kalman filter (EKF) [72] offers a near-optimal solution [112]. However, in highly nonlinear scenarios, linearisation becomes both computationally expensive and imprecise.

Cubature Kalman filters

A commonly used class of nonlinear Kalman filters is the CKF [113], which uses spherical radial cubature rules to approximate the integrals that arise when applying Bayes' rule to Gaussian systems [113][114]. The third-degree CKF is a slightly more expensive alternative to the KF, but may provide gains in efficiency compared to the EKF or the UKF. It is described in Algorithm 2, where $\boldsymbol{\upsilon}_k$ is the process noise vector with zero mean and variance $\boldsymbol{Q}_k$, and $\boldsymbol{s}_k$ is the measurement noise vector with zero mean and variance $\boldsymbol{\Omega}_k$. Let

d represent the state dimension, and define the number of cubature points to be $d_{ckf} = 2d$, computed as

$$\boldsymbol{x}^i_{k-1} = \boldsymbol{S}_{k-1}\eta^i + \hat{\boldsymbol{x}}_{k-1}, \tag{4.1}$$

where $\boldsymbol{S}_{k-1}$ denotes the factorised state covariance matrix, obtained through either Cholesky factorisation or the QR algorithm. The values of η^i are defined as

$$\eta^i = \sqrt{d}\mathbb{I}^i, \quad \text{for } i = 1, \ldots, d, \text{ and } \eta^i = -\sqrt{d}\mathbb{I}^i, \quad \text{for } i = d+1, \ldots, d_{ckf}$$

where $\mathbb{I}^i$ is a unit vector with a 1 at the ith position and zeros elsewhere. Each cubature point is assigned an equal weight, given by the constant value $w^i = \frac{1}{d_{ckf}}$ which scales inversely with state dimension [102]. Figure 4.2 provides a visual representation of third-degree cubature point sampling around a mean value.

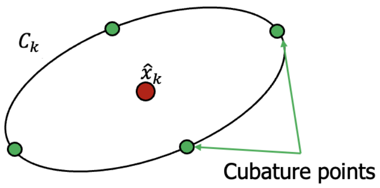


Figure 4.2: The CKF of degree 3, in which $d_{ckf} = 2d$ cubature points are drawn around a mean value, $\hat{x}_k$, spanning the covariance C_k.

Cubature points are propagated to the current time-step t_k, and the mean and covariance are recomputed under the assumption that they sufficiently represent the propagated distribution. Cubature points are then mapped into the observation space for comparison with actual measurements. If the measurement equation remains linear, the standard KF update equations are applied; however, when dealing with nonlinear measurements, as is common in orbital dynamics, a modified set of equations is necessary. The Kalman gain K_G computation is then taken as the quotient between the measurement covariance $\boldsymbol{C}_y$ and the cross-covariance between the predicted observations and the predicted states, $\boldsymbol{C}_{xy}$. As a new observation $\boldsymbol{z}_k$ arrives at time t_k, the state can now be updated as $\bar{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k + K_G(\boldsymbol{z}_k - \hat{\boldsymbol{y}}_k)$, where $\hat{\boldsymbol{y}}_k$ is the mean predicted observation.

In standard CKF implementations, two essential properties of the error covariance matrix, symmetry and positive definiteness, can sometimes be lost due to numerical errors arising from matrix inversion or square root operations. To mitigate these issues, the square root cubature Kalman filter (SCKF) [114] can be used. The SCKF ensures that the covariance matrix remains symmetric and positive (semi)definite, preventing numerical instabilities that could otherwise cause filter divergence. In this thesis, however, the CKF is used as no complications have thus far been experienced during its implementation.

Algorithm 2 Cubature Kalman filter (CKF)

```
Inputs:
- Prior Gaussian pdf $\pi_0(\boldsymbol{x}_0)$ at time $t_0$, given by mean state $\hat{\boldsymbol{x}}_0$ and covariance $\Sigma_0$.
- $M$ observation time-stamps $\{t_k\}_{k=1}^{M}$.
- A set of observations $z_k$ and their noise covariance matrices $\Omega_k,\ k = 1, \dots, M$.
Outputs:
- Estimated state $\bar{\boldsymbol{x}}_k$ and covariance matrix $\boldsymbol{C}_{x,k}$ at each observation time-stamp.
Procedure: for each observation epoch $t_k$

                         Prediction

 1. Factorise the prior covariance to obtain $\boldsymbol{S}_{k-1}$
    $\boldsymbol{C}_{x,k-1} = \boldsymbol{S}_{k-1}\boldsymbol{S}_{k-1}^{\top}$
 2. Compute new cubature points according to equation
    $\boldsymbol{x}_{k-1}^{i} = \boldsymbol{S}_{k-1}\eta^{i} + \hat{\boldsymbol{x}}_{k-1}$
 3. Propagate cubature points through state transition equation:
    $\tilde{\boldsymbol{x}}_{k}^{i} = f(\boldsymbol{x}_{k-1}^{i}, \boldsymbol{\upsilon}_{k-1})$
 4. Estimate predictive state
    $\tilde{\boldsymbol{x}}_k = \sum_{i=1}^{d_{ckf}} w^{i}\tilde{\boldsymbol{x}}_{k}^{i}$
 5. Estimate predictive covariance, in presence of process noise:
    $\boldsymbol{C}_{x,k}^{*} = \boldsymbol{w}\sum_{i=1}^{d_{ckf}} (\tilde{\boldsymbol{x}}_{k}^{i}\tilde{\boldsymbol{x}}_{k}^{i\top}) - (\hat{\boldsymbol{x}}_k\hat{\boldsymbol{x}}_{k}^{\top}) + \boldsymbol{Q}_{k-1}$

                           Update

 6. Factorise predictive covariance:
    $\boldsymbol{C}_{x,k}^{*} = \boldsymbol{S}_k\boldsymbol{S}_{k}^{\top}$
 7. Draw new cubature points after propagation:
    $\boldsymbol{x}_{k}^{i} = \boldsymbol{S}_k\eta^{i} + \tilde{\boldsymbol{x}}_k$
 8. Transform cubature points by the measurement equation:
    $\boldsymbol{y}_{k}^{i} = \mathcal{M}_k(\boldsymbol{x}_{k}^{i})$
 9. Compute expected value of the measurement:
    $\hat{\boldsymbol{y}}_k = \sum_{i=1}^{d_{ckf}} w^{i}\boldsymbol{y}_{k}^{i}$
10. Compute measurement covariance and cross-covariance, respectively given by:
    $\boldsymbol{C}_y = \frac{1}{2d_{ckf}}\sum_{i=1}^{d_{ckf}}\left(\boldsymbol{y}_{k}^{i}\boldsymbol{y}_{k}^{i\,\top}\right) - (\hat{\boldsymbol{y}}_k\,\hat{\boldsymbol{y}}_{k}^{\top}) + \boldsymbol{\Omega}_k$ and
    $\boldsymbol{C}_{xy} = \frac{1}{2d_{ckf}}\sum_{i=1}^{d_{ckf}}\left(\boldsymbol{x}_{k}^{i}\boldsymbol{y}_{k}^{i\,\top}\right) - (\hat{\boldsymbol{x}}_k\hat{\boldsymbol{y}}_{k}^{\top})$
11. Calculate the Kalman Gain $K_G$:
    $K_G = \boldsymbol{C}_{xy}/\boldsymbol{C}_y$
12. Update the state estimate with actual measurement $z_k$:
    $\hat{\boldsymbol{x}}_k = \tilde{\boldsymbol{x}}_k + K_G(z_k - \hat{\boldsymbol{y}}_k)$.
13. Update the covariance estimate:
    $\boldsymbol{C}_{x,k} = \boldsymbol{C}_{x,k}^{*} - K_G\boldsymbol{C}_yK_{G}^{\top}$
```

4.2.2. Particle filters

Gaussian filters are remarkably efficient to use when the state pdfs all remain (at least approximately) Gaussian throughout the simulation. Oftentimes, however, these methods fail to capture the distribution of states during propagation, due to significant nonlinearities in the dynamical model.

The underlying principle behind PFs is Monte Carlo integration. Let X be an r.v. with pdf $p(x)$. If one wishes to estimate $g(X)$ for some test function $g(\cdot)$, a natural way to proceed is to compute the expectation

$$\mathbb{E}[g(X)] = \int g(x)p(x)\mathrm{d}x.$$

This expectation can be approximated by drawing N samples from the pdf $p(x)$, and then computing the sample mean

$$\mathbb{E}^N[g(X)] = \frac{1}{N}\sum_{i=1}^{N} g(x). \tag{4.2}$$

However, oftentimes it may not be possible to draw samples from the target pdf. In that case, a popular alternative is to use IS schemes (see Chapter 3, Section 3.3.2), where the MC samples are drawn from an importance function $q(x)$. To account for the mismatch between $q(x)$ and $p(x)$, importance weights are assigned to these samples, given by

$$w^i \propto \frac{p(x^i)}{q(x^i)},\ i = 1, ..., N. \tag{4.3}$$

These weights are normalised so that $\sum_{i=1}^{N} w^i = 1$, which implies that it is sufficient to compute $p(x^i)$ up to a proportionality constant. Then, the IS estimator of $\mathbb{E}[f(X)]$ is

$$\mathbb{E}^{IS,N}[f(X)] = \sum_{i=1}^{N} f(x^i)w^i. \tag{4.4}$$

Bootstrap particle filter

The standard PF, often referred to as the bootstrap particle filter (BPF) [69][86], is a (rather simple) sequential IS algorithm. Assume that at time t_{k-1}, samples x^i_{k-1} and weights w^i_{k-1} have been computed. From Eq. (2.20), we see that the predictive pdf $p(x_k|z_{1:k-1})$ is an integral w.r.t. the pdf $p(x_{k-1}|z_{1:k-1})$ and, therefore, it can be approximated as

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})\mathrm{d}x_{k-1} \approx p^{IS,N}(x_k|z_{1:k-1}) := \sum_{i=1}^{N} w^i_{k-1}p(x_k|x^i_{k-1}). \tag{4.5}$$

As a consequence, the filtering pdf at time t_k can itself be estimated, namely

$$p^{IS,N}(x_k|z_{1:k}) \propto p(z_k|x_k) \sum_{i=1}^{N} w_{k-1}^i p(x_k|x_{k-1}^i). \tag{4.6}$$

If the approximate predictive pdf is used as the importance function, i.e., we draw $x_k^i \sim p^{IS,N}(x_k|z_{1:k-1}),\ i = 1,\ldots,N$, then the importance weights at time t_k become

$$w_k^i \propto \frac{p^{IS,N}(x_k|z_{1:k})}{p^{IS,N}(x_k|z_{1:k-1})} \propto p(z_k|x_k^i),\ i = 1,\ldots,N. \tag{4.7}$$

Algorithm 3 shows an implementation of the BPF for spacecraft tracking. At time t_0, N particles are drawn from the prior pdf, $\boldsymbol{x}_0^i \sim p(\boldsymbol{x}_0),\ i = 1,\ldots,N$. Then each particle $\boldsymbol{x}_k^i,\ i = 1,\ldots,N$ is propagated forwards from time t_{k-1} to t_k, by sampling $\tilde{\boldsymbol{x}}_k^i \sim p(\boldsymbol{x}_k|\tilde{\boldsymbol{x}}_{k-1}^i),\ i = 1,\ldots,N$. These predictive samples are passed through the observation function $\mathcal{M}(\cdot)$ to obtain predicted measurements $\boldsymbol{y}_k^i$, for $i = 1,...,N$. This enables the weights to be computed more easily as a function of the likelihoods, which are a direct proxy for how well the predicted observations match the actual observations, i.e., $w_k^i \propto p(\boldsymbol{z}_k|\boldsymbol{x}_k^i) \propto e^{-\frac{1}{2}(z_k-y_k^i)^\top \Omega_k^{-1}(z_k-y_k^i)}$. Resampling with replacement by using the weights of the particles follows. Particles with higher weights are randomly replicated, whilst those with lower weights are randomly discarded. Note that since the output of the algorithm at time t_k is the collection of weighted particles $\{\boldsymbol{x}_k^i, w_k^i\}_{i=1}^N$ that enable the approximation of integrals w.r.t the filtering density $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$, it is straightforward to compute any kind of estimators, such as the posterior mean estimator, or the covariance estimator.

The BPF represents the simplest of PFs. It often suffers from a weight degeneracy problem, where the importance weight tends to concentrate on a single particle [89]. This problem can be tackled by using a large number of samples, but this is costly to run, as the computational cost of a PF is $O(N)$. An alternative solution is to devise more sophisticated extensions to the algorithm to bypass these drawbacks [115]–[118]. For the time being, let it serve as motivation for the implementation of the following algorithm, the EnKF, which avoids the computation of weights altogether.

4.2.3. Hybrid filters

Ensemble Kalman filter

The ensemble Kalman filter (EnKF) is a recursive Monte Carlo filter that replaces the computation of weights of the BPF by a particle update using an empirical Kalman gain matrix. The method has proved robust in many applications (e.g., [97], [119]) but it enjoys limited convergence guarantees compared to the PF [120].

Algorithm 4, shows the EnKF, which follows [121]. The update stage, in contrast to the PF, uses an MC estimate of the Kalman gain instead of particle weights and resam-

Algorithm 3 Bootstrap particle filter (BPF)

Inputs:

- N iid samples $\{\boldsymbol{x}_0^i\},\ i = 1, \ldots, N$, from the prior pdf $p(\boldsymbol{x}_0)$ at time t_0.
- M observation time-stamps, $\{t_k\}_{k=1}^M$.
- A set of observations z_k and their noise covariance matrices $\Omega_k,\ k = 1, ...M$.

Outputs:

- Weighted particle sets $\{\tilde{\boldsymbol{x}}_k^i, w_k^i\}_{i=1}^N,\ k = 1, \ldots, M$.

Procedure: for each observation epoch t_k

Prediction

1. Propagate samples stochastically from time t_{k-1} to t_k, using numerical scheme and dynamical model of choice, obtaining
$\tilde{\boldsymbol{x}}_k^i \sim p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^i),\ i = 1, ..., N$.

Update

2. Compute predicted measurements $\boldsymbol{y}_k^i = \mathcal{M}_k(\tilde{\boldsymbol{x}}_k^i)$ and evaluate likelihoods $L_k^i \propto p(z_k | \tilde{\boldsymbol{x}}_k^i),\ i = 1, \ldots, N$.
3. Compute normalised importance weights:
$w_k^i \propto L_k^i,\ i = 1, ..., N$.
4. Resample the weighted set $\{\tilde{\boldsymbol{x}}_k^i, w_k^i\}_{i=1}^N$ N times with replacement to generate new particles $\boldsymbol{x}_k^i,\ i = 1, ..., N$.

pling, meaning that all samples are equally weighted. In order to use the observation to refine the state, the Kalman gain K_G^N weighs the ratio of how much the filter is to "trust" either the kth measurement or the kth propagated state. In this framework, a collection of N samples is termed an 'ensemble', which contains information about the empirical mean and covariance. These ensemble particles are then updated with a variation of the classical Kalman update equations, given by

$$\boldsymbol{x}_k^i = \tilde{\boldsymbol{x}}_k^i + K_G^N(z_k - \boldsymbol{y}_k^i + \boldsymbol{s}_k^i),\ \boldsymbol{s}_k^i \sim \mathcal{N}(0, \Omega_k),\ i = 1, ..., N, \tag{4.8}$$

where $z_k - \boldsymbol{y}_k^i$ is the observation residual and $\boldsymbol{s}_k^i$ is the observation noise vector with covariance matrix Ω_k. Having updated the ensemble, the estimator of the state is simply the mean of all particles, given by

$$\bar{\boldsymbol{x}}_k^N = \frac{1}{N} \sum_{i=1}^N \boldsymbol{x}_k^i, \tag{4.9}$$

and an empirical covariance can be computed as

$$\boldsymbol{C}_{x,k}^N = \frac{1}{N} \sum_{i=1}^N (\boldsymbol{x}_k^i - \bar{\boldsymbol{x}}_k^N)(\boldsymbol{x}_k^i - \bar{\boldsymbol{x}}_k^N)^\top. \tag{4.10}$$

These updated ensembles are then recursively propagated to the next time-step.

Algorithm 4 Ensemble Kalman filter (EnKF)

Inputs:
- N iid samples $\{\boldsymbol{x}_0^i\}, i = 1, \dots, N$, from the prior pdf $p(\boldsymbol{x}_0)$ at time t_0.
- M observation time-stamps, $\{t_k\}_{k=1}^M$.
- A set of observations z_k and their noise covariance matrices $\Omega_k, k = 1, ...M$.

Outputs:
- A collection of equally-weighted samples $\{\boldsymbol{x}_k^i\}_{i=1}^N$ at each time $t_k,\ k = 1, \dots, M$.

Procedure: for each observation epoch t_k

Prediction

1. Propagate samples from time t_{k-1} to t_k, using the numerical scheme of choice to obtain
   $\tilde{\boldsymbol{x}}_k^i \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i),\ i = 1, ..., N.$

Update

2. Transform the samples $\{\tilde{\boldsymbol{x}}_k^i\}_{i=1}^N$ through the measurement function to obtain predicted observations
   $\boldsymbol{y}_k^i = \mathcal{M}_k(\tilde{\boldsymbol{x}}_k^i),\ i = 1, ..., N.$
3. Compute mean and covariance in the observation space:
   $\hat{\boldsymbol{y}}_k^N = \frac{1}{N}\sum_{i=1}^N \boldsymbol{y}_k^i$ and $\boldsymbol{C}_{y,k}^N = \Omega_k + \frac{1}{N-1}\sum_{i-1}^N(\boldsymbol{y}_k^i - \hat{\boldsymbol{y}}_k^N)(\boldsymbol{y}_k^i - \hat{\boldsymbol{y}}_k^N)^\top$
4. Compute the ensemble mean and cross-covariance matrix:
   $\hat{\boldsymbol{x}}_k^N = \frac{1}{N}\sum_{i=1}^N \tilde{\boldsymbol{x}}_k^i$ and $\boldsymbol{C}_{xy,k}^N = \frac{1}{N-1}\sum_{i-1}^N(\tilde{\boldsymbol{x}}_k^i - \hat{\boldsymbol{x}}_k^N)(\boldsymbol{y}_k^i - \hat{\boldsymbol{y}}_k^N)^\top.$
5. Compute the Kalman gain
   $K_G^N = \boldsymbol{C}_{xy,k}^N(\boldsymbol{C}_{y,k}^N)^{-1}.$
6. Update the ensemble samples
   $\boldsymbol{x}_k^i = \tilde{\boldsymbol{x}}_k^i + K_G^N(z_k - \boldsymbol{y}_k^i + \boldsymbol{s}_k^i),\ \boldsymbol{s}_k^i \sim \mathcal{N}(0, \Omega_k),\ i = 1, ..., N.$

Ensemble cubature Kalman filter

The ensemble cubature Kalman filter (EnCKF) is a hybrid algorithm which uses cubature points, which approximate Gaussian distributions and are known to perform well with highly nonlinear systems, in combination with ensemble sampling, like the EnKF, which is known to work well in high-dimensional state spaces [102]. In the regular EnKF, accuracy increases as ensemble size grows, but a maximum is rapidly reached, beyond which performance stops growing [98] [102]. Much like with the EnKF, when the EnCKF is used, estimation quality begins to decrease after a certain number of samples has been surpassed, so a point of diminishing returns is reached.

This approach involves generating an ensemble of cubature point sets of $d_{ckf} = 2d$ cubature points. The remainder of the method follows a similar procedure to that of the EnKF, with the key difference being that the ensemble states are constructed using the averages of the cubature points. Finally, the estimator of the posterior state is computed using the average of the ensemble states. Algorithm 5 presents the EnCKF formulation. Note that if $N_{ens} > d$, we must use the singular value decomposition (SVD) algorithm to obtain the $d \times d$ matrix $\boldsymbol{S}_{k|k}$.

Ensemble particle filter

In order to exploit the robustness of the EnKF scheme, next we propose a novel method that combines Kalman updates of the MC samples with the computation of importance weights, aimed at improving the accuracy of the filters. The new algorithm is denoted ensemble particle filter (EnPF) as it is a hybrid between the EnKF and the BPF. Algorithm 6 outlines the method when it is designed to track the state of an orbiting spacecraft. The prediction stage follows that of an EnKF, which propagates an ensemble through the dynamical model. The update stage, however, now consists of two parts:

- The computation of an MC estimate of the Kalman gain, to correct the particles based on the observation residuals, exactly as in step 2 of the EnKF.
- The computation of weights (similar to the BPF) associated with these particles, which we now denote $\check{\boldsymbol{x}}_k^i$, as well as a resampling step based on these weights. In other words, by assuming that the prior is given by $\{\check{\boldsymbol{x}}_k\}_{i=1}^N$, weights of the form $w_k^i \propto w_{k-1}^i p(\boldsymbol{z}_k|\check{\boldsymbol{x}}_k^i)$ are computed by calculating likelihoods, which allocate higher weights to samples that align more closely with the actual observation.

In Algorithm 6, resampling steps are taken adaptively, depending on the effective sample size (ESS), given by ESS $= \frac{1}{N\sum_{i=1}^N (w_k^i)^2}$, for $i = 1, ..., N$. The ESS is an approximate measure of sample diversity [122]. If the ESS falls below a given threshold φ, the particles are resampled and weights are reset.

The posterior estimate of the state can be computed as the weighted average of all particles, namely

$$\bar{\boldsymbol{x}}_k^N = \sum_{i=1}^{N} w^i \check{\boldsymbol{x}}_k^i. \tag{4.11}$$

Note that steps 1 to 6 in Algorithm 6 are exactly the same as Algorithm 4. It is important to remark that the EnPF is a "greedy" algorithm where each observation $\boldsymbol{z}_k$ is processed twice: once during the Kalman update, and again for the calculation of likelihoods that determine the weights and the resampling step. The algorithm proposed here, therefore lacks the theoretical guarantees of convergence of conventional PFs, as the weights do not necessarily follow the theoretical structure of Eq. (4.3).

```
Algorithm 5 Ensemble cubature Kalman filter (EnCKF)
Inputs:
- $N$ iid state samples $\{\hat{\boldsymbol{x}}_0^i\},\ i = 1, \dots, N$, from the prior pdf $\pi_0(\boldsymbol{x}_0)$ at time $t_0$.
- $M$ observation time-stamps $\{t_k\}_{k=1}^{M}$.
- A set of observations $z_k$ and their noise covariance matrices $\Omega_{z,k},\ k = 1, \dots, M$.
Outputs:
- Estimated state $\bar{\boldsymbol{x}}_k$ and covariance matrix $\boldsymbol{C}_{x,k}$ at each observation time-stamp.
Procedure: for each observation epoch $t_k$

                              Prediction

 1. Assume $\boldsymbol{S}_{k-1}$ is available from previous iteration and draw cubature points for each
    ensemble state $\boldsymbol{x}_{k-1}^{ij} = \boldsymbol{S}_{k-1}\eta^j + \hat{\boldsymbol{x}}_{k-1}^i$, for $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, d_{ckf}$.
 2. Propagate all cubature points:
    $\tilde{\boldsymbol{x}}_k^{ij} = f(\boldsymbol{x}_{k-1}^{ij}, \boldsymbol{v}_{k-1}^i)$
 3. Estimate predictive ensemble states (average of cubature points):
    $\tilde{\boldsymbol{x}}_k^i = \sum_{j=1}^{d_{ckf}} w^i \tilde{\boldsymbol{x}}_k^{ij},\ i = 1, 2, \dots, N.$
 4. Estimate predictive state (average of ensembles) as $\tilde{\boldsymbol{x}}_k^* = \frac{1}{N} \sum_{i=1}^{N} \tilde{\boldsymbol{x}}_k^i$
 5. Estimate the square-root predictive covariance (triangularisation):
    $\boldsymbol{S}_k = \frac{1}{\sqrt{N-1}} \left[ \tilde{\boldsymbol{x}}_k^i - \tilde{\boldsymbol{x}}_k^* \right]$

                                Update

 6. Draw cubature points around each ensemble state:
    $\boldsymbol{x}_k^{ij} = \boldsymbol{S}_k \eta^i + \tilde{\boldsymbol{x}}_k^i$, for $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, d_{ckf}$.
 7. Transform all cubature points by the measurement equation $\boldsymbol{y}_k^{ij} = \mathcal{M}_k(\boldsymbol{x}_k^{ij})$.
 8. Estimate the expected value of the cubature measurement transformations:
    $\hat{\boldsymbol{y}}_k^i = \sum_{j=1}^{d_{ckf}} w^i \boldsymbol{y}_k^{ij} + \boldsymbol{s}_k^i,\ i = 1, 2, \dots, N.$
 9. Estimate the expected value of the ensemble measurements:
    $\hat{\boldsymbol{y}}_k^* = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{y}_k^i$
10. Estimate the measurement covariance and cross covariance:
    $\boldsymbol{C}_{y,k} = \frac{1}{N-1} \left[ \boldsymbol{y}_k^i - \hat{\boldsymbol{y}}_k^{*T} \right] \left[ \boldsymbol{y}_k^i - \hat{\boldsymbol{y}}_k^{*T} \right]^T$ and $\boldsymbol{C}_{xy,k} = \frac{1}{N-1} \left[ \boldsymbol{x}_k^i - \hat{\boldsymbol{x}}_k^{*T} \right] \left[ \boldsymbol{y}^i - \hat{\boldsymbol{y}}_k^{*T} \right]^T$
11. Calculate the Kalman Gain $K_G$
    $K_G = \boldsymbol{C}_{xy,k} (\boldsymbol{C}_{y,k})^{-1}$
12. Compute updated ensemble state with actual measurement $z_k$:
    $\hat{\boldsymbol{x}}_k^i = \tilde{\boldsymbol{x}}_k^i + K_G(z_k - \boldsymbol{y}_k^i),\ i = 1, 2, \dots, N.$
```

Algorithm 6 Ensemble particle filter (EnPF)

Inputs:

- N iid samples $\{\boldsymbol{x}_0^i\}, i = 1, \ldots, N$, from the prior pdf $p(\boldsymbol{x}_0)$ at time t_0.
- M observation time-stamps, $\{t_k\}_{k=1}^M$
- A set of observations $\boldsymbol{z}_k$ and their noise covariance matrices $\Omega_k,\ k = 1, ...M$.
- The resampling threshold φ.

Outputs:

- A collection of weighted samples $\{\boldsymbol{x}_k^i, w^i\}_{i=1}^N$, at each time $t_k,\ k = 1, \ldots, M$.

Procedure: for each observation epoch t_k

Prediction

1. Propagate samples from time t_{k-1} to t_k, using the numerical scheme of choice to obtain
   $\tilde{\boldsymbol{x}}_k^i \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i),\ i = 1, ..., N.$

Update

2. Transform the samples $\{\tilde{\boldsymbol{x}}_k^i\}_{i=1}^N$ through the measurement function to obtain predicted observations
   $\boldsymbol{y}_k^i = \mathcal{M}_k(\tilde{\boldsymbol{x}}_k^i),\ i = 1, ..., N.$
3. Compute mean and covariance in the observation space:
   $\hat{\boldsymbol{y}}_k^N = \frac{1}{N}\sum_{i=1}^N \boldsymbol{y}_k^i$ and $\boldsymbol{C}_{y,k}^N = \frac{1}{N-1}\sum_{i-1}^N (\boldsymbol{y}_k^i - \hat{\boldsymbol{y}}_k^N)(\boldsymbol{y}_k^i - \hat{\boldsymbol{y}}_k^N)^\top + \boldsymbol{\Omega}_k$
4. Compute the ensemble mean and cross-covariance matrix:
   $\hat{\boldsymbol{x}}_k^N = \frac{1}{N}\sum_{i=1}^N \tilde{\boldsymbol{x}}_k^i$ and $\boldsymbol{C}_{xy,k}^N = \frac{1}{N-1}\sum_{i-1}^N (\tilde{\boldsymbol{x}}_k^i - \hat{\boldsymbol{x}}_k^N)(\boldsymbol{y}_k^i - \hat{\boldsymbol{y}}_k^N)^\top.$
5. Compute the Kalman gain $K_G^N = \boldsymbol{C}_{xy,k}^N(\boldsymbol{C}_{y,k}^N)^{-1}$.
6. Update the ensemble samples
   $\check{\boldsymbol{x}}_k^i = \tilde{\boldsymbol{x}}_k^i + K_G^N(\boldsymbol{z}_k - \boldsymbol{y}_k^i + \boldsymbol{s}_k^i),\ \boldsymbol{s}_k^i \sim \mathcal{N}(0, \Omega_k),\ i = 1, ..., N.$
7. Compute normalised importance weights in the form
   $w_k^i \propto w_{k-1}^i p(\boldsymbol{z}_k|\check{\boldsymbol{x}}_k^i),\ i = 1, ..., N.$
8. Compute the normalised ESS,
   $\text{NESS}_k = \frac{1}{N\sum_{i=1}^N (w_k^i)^2}.$
9. **If** $\text{NESS}_k < \varphi$, where φ is the resampling threshold:
   Resample N times with replacement to generate new particles $\boldsymbol{x}_k^i,\ i = 1, ..., N$ and set $w_k^i = \frac{1}{N},\ i = 1, ..., N$. Otherwise, set $\boldsymbol{x}_k^i = \check{\boldsymbol{x}}_k^i$.

4.3 Methodology: state tracking and parameter estimation

4.3.1. Hybrid filters

Ensemble Kalman filter with unknown parameters

Algorithm 7 follows [121] and has the same structure as the EnKF of section 4.2.3, except that it is adapted to simultaneously track the state and a set of unknown parameters. We define the extended state vector as

$$\chi_k = \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{\theta} \end{bmatrix},$$

which includes the 6-dimensional state $\boldsymbol{x}_k$ and the parameter vector $\boldsymbol{\theta}$, increasing the dimension d of the state to D. In our application, since we are interested in estimating the level of uncertainty in the system, the vector $\boldsymbol{\theta}$ contains, at least, the parameters of the diffusion coefficient $\sigma_W(\boldsymbol{x})$, but it may also include dynamical parameters such as the ballistic coefficient B_c or the SRP coefficient.

The update of the extended states of the ensemble is carried out with a variation of the classical Kalman update equations, given by

$$\chi_k^i = \tilde{\chi}_k^i + K_G^N(z_k - \boldsymbol{y}_k^i + \boldsymbol{s}_k^i), \ \boldsymbol{s}_k^i \sim \mathcal{N}(0, \Omega_k), \ i = 1, ..., N, \tag{4.12}$$

where $z_k - \boldsymbol{y}_k^i$ is the observation residual and $\boldsymbol{s}_k^i$ is the observation noise with covariance matrix $\boldsymbol{\Omega}_k$. Having updated the ensemble, the estimator of the extended state is simply the mean of all particles, given by

$$\bar{\chi}_k^N = \frac{1}{N}\sum_{i=1}^{N} \chi_k^i, \tag{4.13}$$

and an empirical covariance can be computed as

$$\boldsymbol{C}_k^N = \frac{1}{N}\sum_{i=1}^{N} (\chi_k^i - \bar{\chi}_k^N)(\chi_k^i - \bar{\chi}_k^N)^\top. \tag{4.14}$$

Then, the state estimate $\bar{\boldsymbol{x}}_k^N$ and the parameter estimate $\bar{\boldsymbol{\theta}}_k^N$ can simply be extracted from $\bar{\chi}_k^N$. These updated ensembles are then recursively propagated to the next time-step.

To the best of our knowledge, this variation of the EnKF has not been applied in astrodynamics.

Algorithm 7 Ensemble Kalman filter with unknown parameters (EnKFup)

Inputs:

- N iid samples $\boldsymbol{\chi}_0^i = \{\boldsymbol{x}_0^i, \boldsymbol{\theta}_0^i\}, i = 1, \ldots, N$, from the prior pdf $p(\boldsymbol{x}_0, \boldsymbol{\theta})$ at time t_0.
- M observation time-stamps, $\{t_k\}_{k=1}^M$.
- A set of observations $\boldsymbol{z}_k$ and their noise covariance matrices $\Omega_k, k = 1, ...M$.

Outputs:

- A collection of equally-weighted samples $\{\boldsymbol{\chi}_k^i\}_{i=1}^N$ at each time $t_k,\ k = 1, \ldots, M$.

Procedure: for each observation epoch t_k

Prediction

1. Propagate samples from time t_{k-1} to t_k, using the numerical scheme of choice to obtain $\tilde{\boldsymbol{\theta}}_k^i = \boldsymbol{\theta}_{k-1}^i$ and
   $\tilde{\boldsymbol{x}}_k^i \sim p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^i, \tilde{\boldsymbol{\theta}}_k^i),\ i = 1, ..., N$. Let $\tilde{\boldsymbol{\chi}}_k^i = \{\tilde{\boldsymbol{x}}_k^i, \tilde{\boldsymbol{\theta}}_k^i\}$.

Update

2. Transform the samples $\{\tilde{\boldsymbol{x}}_k^i\}_{i=1}^N$ through the measurement function to obtain predicted observations
   $\boldsymbol{y}_k^i = \mathcal{M}_k(\tilde{\boldsymbol{x}}_k^i),\ i = 1, ..., N$.

3. Compute mean and covariance in the observation space:
   $\hat{\boldsymbol{y}}_k^N = \frac{1}{N} \sum_{i=1}^N \boldsymbol{y}_k^i$ and $\boldsymbol{C}_{y,k}^N = \frac{1}{N-1} \sum_{i-1}^N (\boldsymbol{y}_k^i - \hat{\boldsymbol{y}}_k^N)(\boldsymbol{y}_k^i - \hat{\boldsymbol{y}}_k^N)^\top + \Omega_k$

4. Compute the ensemble mean and cross-covariance matrix:
   $\hat{\boldsymbol{\chi}}_k^N = \frac{1}{N} \sum_{i=1}^N \tilde{\boldsymbol{\chi}}_k^i$ and $\boldsymbol{C}_{\chi y,k}^N = \frac{1}{N-1} \sum_{i-1}^N (\tilde{\boldsymbol{\chi}}_k^i - \hat{\boldsymbol{\chi}}_k^N)(\boldsymbol{y}_k^i - \hat{\boldsymbol{y}}_k^N)^\top$.

5. Compute the Kalman gain $K_G^N = \boldsymbol{C}_{\chi y,k}^N (\boldsymbol{C}_{y,k}^N)^{-1}$.

6. Update the ensemble samples
   $\boldsymbol{\chi}_k^i = \tilde{\boldsymbol{\chi}}_k^i + K_G^N(\boldsymbol{z}_k - \boldsymbol{y}_k^i + \boldsymbol{s}_k^i),\ \boldsymbol{s}_k^i \sim \mathcal{N}(0, \Omega_k),\ i = 1, ..., N$.

Ensemble particle filter with unknown parameters

This method follows the same structure as the EnPF of Section 4.2.3, by combining the Kalman update of the MC samples with the calculation of importance weights. Described in Algorithm 8, it is adapted to track the state and estimate unknown parameters, similar to the EnKFup.

Here, the predictive samples *after* the EnKFup update are given by $\check{\chi}_k^i$, and weights are computed by processing these states mapped in the observation space, i.e., $w_k^i \propto w_{k-1}^i p(z_k|\check{x}_k^i)$, in the same way as for the BPF. The posterior mean of the extended state can be estimated as the weighted average of all particles, namely,

$$\bar{\chi}_k^N = \sum_{i=1}^{N} w^i \check{\chi}_k^i. \tag{4.15}$$

Then, the state estimate $\bar{\boldsymbol{x}}_k^N$ and the parameter estimate $\bar{\boldsymbol{\theta}}_k^N$ can simply be extracted from $\bar{\chi}_k^N$. Resampling may then be applied if the ESS is lower than the resampling threshold φ. Note that steps 1 to 6 in Algorithm 8 are exactly the same as Algorithm 7.

Nested hybrid filter

To tackle the simultaneous tracking of the states and uncertain parameters, a variation of the nested particle filter (NPF) [123][124] is implemented, which recursively computes the posterior distribution of the parameters, $p(\boldsymbol{\theta}_k|\boldsymbol{z}_{1:k}),\ k = 1, \ldots, M$. While the EnPFup is a “greedy" algorithm with no proven theoretical guarantees of convergence as of yet, we introduce a nested hybrid filter (NHF) for which a convergence analysis is available in [125].

The objective of the NHF is to approximate $p(\boldsymbol{\theta}|\boldsymbol{z}_{1:k})$ by employing a variation of the BPF. Using Bayes, we have

$$p(\boldsymbol{\theta}|\boldsymbol{z}_{1:k}) \propto p(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{z}_{1:k-1}). \tag{4.16}$$

Therefore, if we were able to sample from $p(\boldsymbol{\theta}|\boldsymbol{z}_{1:k-1})$ and evaluate $p(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1}, \boldsymbol{\theta})$, then we would be able to design a sequential importance sampler to approximate the posterior pdf, as per the methodology introduced in Section 4.2.2. Specifically, we would

1. sample $\boldsymbol{\theta}_k^i \sim p(\boldsymbol{\theta}|\boldsymbol{z}_{1:k-1})$,
2. compute weights $w_k^i \sim p(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1}, \boldsymbol{\theta}_k^i)$,
3. resample if necessary.

Unfortunately, neither step 1. nor step 2. can be performed exactly. However, they can be approximated.

```
Algorithm 8 Ensemble particle filter with unknown parameters (EnPFup)
Inputs:
- $N$ iid samples $\boldsymbol{\chi}_0^i = \{\boldsymbol{x}_0^i, \boldsymbol{\theta}_0^i\}, i = 1, \dots, N$, from the prior pdf $p(\boldsymbol{x}_0, \boldsymbol{\theta})$ at time $t_0$.
- $M$ observation time-stamps, $\{t_k\}_{k=1}^M$
- A set of observations $z_k$ and their noise covariance matrices $\Omega_k,\ k = 1, ...M$.
- Resampling threshold $\varphi$.
Outputs:
-A collection of weighted samples $\{\boldsymbol{\chi}_k^i, w_k^i\}_{i=1}^N$, at each time $t_k,\ k = 1, \dots, M$.
Procedure: for each observation epoch $t_k$

                              Prediction

  1. Propagate samples from time $t_{k-1}$ to $t_k$, using the numerical scheme of choice to
     obtain $\tilde{\boldsymbol{\theta}}_k^i = \boldsymbol{\theta}_{k-1}^i$ and
     $\tilde{\boldsymbol{x}}_k^i \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i, \tilde{\boldsymbol{\theta}}_k),\ i = 1, ..., N$. Let $\tilde{\boldsymbol{\chi}}_k^i = \{\tilde{\boldsymbol{x}}_k^i, \tilde{\boldsymbol{\theta}}_k^i\}$.

                              Update

  2. Transform the samples $\{\tilde{\boldsymbol{x}}_k^i\}_{i=1}^N$ through the measurement function to obtain predicted
     observations
     $\boldsymbol{y}_k^i = \mathcal{M}_k(\tilde{\boldsymbol{x}}_k^i),\ i = 1, ..., N$.

  3. Compute mean and covariance in the observation space:
     $\hat{\boldsymbol{y}}_k^N = \frac{1}{N} \sum_{i=1}^N \boldsymbol{y}_k^i$ and $\boldsymbol{C}_{y,k}^N = \frac{1}{N-1} \sum_{i-1}^N (\boldsymbol{y}_k^i - \hat{\boldsymbol{y}}_k^N)(\boldsymbol{y}_k^i - \hat{\boldsymbol{y}}_k^N)^\top + \Omega_k$

  4. Compute the ensemble mean and cross-covariance matrix:
     $\hat{\boldsymbol{\chi}}_k^N = \frac{1}{N} \sum_{i=1}^N \tilde{\boldsymbol{\chi}}_k^i$ and $\boldsymbol{C}_{\chi y,k}^N = \frac{1}{N-1} \sum_{i-1}^N (\tilde{\boldsymbol{\chi}}_k^i - \hat{\boldsymbol{\chi}}_k^N)(\boldsymbol{y}_k^i - \hat{\boldsymbol{y}}_k^N)^\top$.

  5. Compute the Kalman gain $K_G^N = \boldsymbol{C}_{\chi y,k}^N (\boldsymbol{C}_{y,k}^N)^{-1}$.

  6. Update the ensemble samples
     $\check{\boldsymbol{\chi}}_k^i = \tilde{\boldsymbol{\chi}}_k^i + K_G^N(z_k - \boldsymbol{y}_k^i + \boldsymbol{s}_k^i),\ \boldsymbol{s}_k^i \sim \mathcal{N}(0, \Omega_k),\ i = 1, ..., N$.

  7. Compute normalised importance weights in the form
     $w_k^i \propto w_{k-1}^i p(z_k|\tilde{\boldsymbol{\chi}}_k^i),\ i = 1, ..., N$.

  8. Compute the normalised ESS,
     $\text{NESS}_k = \frac{1}{N \sum_{i=1}^N (w_k^i)^2}$.

  9. If $\text{NESS}_k < \varphi$, where $\varphi$ is the resampling threshold:
     Resample $N$ times with replacement to generate new particles $\boldsymbol{\chi}_k^i,\ i = 1, ..., N$ and
     set $w_k^i = \frac{1}{N},\ i = 1, ..., N$. Otherwise, set $\boldsymbol{\chi}_k^i = \check{\boldsymbol{\chi}}_k^i$.
```

Step 1 is approximated by performing a “jittering" procedure, which consists of applying a controlled perturbation to the particles $\{\boldsymbol{\theta}_{k-1}^i\}_{i=1}^{N_1}$, resulting in $\{\tilde{\boldsymbol{\theta}}_k^i\}_{i=1}^{N_1}$. This perturbation is assumed to be Gaussian. It may be small and applied to many samples or it may be a relatively large perturbation applied to only a fraction of samples. See [123] for a detailed description and theoretical justification of the jittering step.

Step 2 is approximated by running a separate filter associated with each $\tilde{\boldsymbol{\theta}}_k^i$. Therefore, the algorithm can be visualised as a bank of N_1 filters (which may be PFs or Gaussian filters), one for each parameter $\tilde{\boldsymbol{\theta}}_k^i$, $i = 1, \dots, N_1$. In this implementation, the EnKF is used: at each time t_k, for each parameter sample $\tilde{\boldsymbol{\theta}}_k^i$, a collection of N_2 samples drawn from the state prior, is propagated from t_{k-1} to t_k, obtaining $\tilde{\boldsymbol{x}}_k^{i,j} \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \tilde{\boldsymbol{\theta}}_k^i)$, $i = 1, ..., N_1$ and $j = 1, ..., N_2$. Then, the EnKF update equations are employed to “move" the samples in the direction of the observation, resulting in N_1 ensembles of N_2 particles $\{\{\check{\boldsymbol{x}}_{\boldsymbol{k}}^{i,j}\}_{j=1}^{N_2}\}_{i=1}^{N_1}$ each.

The primary layer (i.e., the parameter layer) likelihood $p(z_k|z_{1:k-1}, \tilde{\boldsymbol{\theta}}_k^i)$ is then numerically approximated by the mean of the secondary layer (i.e., the state layer) likelihoods, as $\lambda_k^i = \frac{1}{N_2}\sum_{j=1}^{N_2} p(z_k|\check{\boldsymbol{x}}_k^{i,j})$, $i = 1, \dots, N_1$, thus obtaining N_1 likelihoods.

All that remains is to evaluate the ESS to determine whether to resample primary layer particles (and their associated state particles), in the same way as for the EnPFup. The NHF is featured in Algorithm 9.

4.4 Test cases: state-only tracking

4.4.1. LEO

The first test case used for the validation of the above algorithms, a LEO scenario with synthetic observations generated with a high-fidelity dynamical model, spanning a period of just over nine days. Its initial state $\boldsymbol{x}_0$ is given by

$$\boldsymbol{x}_0 = [3675.93\text{km} \quad \text{-}5401.93\text{km} \quad \text{-}2959.64\text{km} \quad 0.71\text{km/s} \quad \text{-}3.18\text{km/s} \quad 6.70\text{km/s}]^\top$$

and its initial covariance $\boldsymbol{\Sigma}_0$ is given by

$$\boldsymbol{\Sigma}_0 = \left[\begin{array}{cccccc} 0.07 & -0.08 & -0.04 & 0 & 0 & 0 \\ -0.08 & 0.13 & -0.07 & 0 & 0 & 0 \\ -0.04 & 0.07 & 0.05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.59\times10^{-8} & -3.63\times10^{-9} & 2.28\times10^{-8} \\ 0 & 0 & 0 & -3.63\times10^{-9} & 5.11\times10^{-8} & -9.26\times10^{-8} \\ 0 & 0 & 0 & 2.28\times10^{-8} & -9.26\times10^{-8} & 2.01\times10^{-7} \end{array}\right] \begin{array}{l} \text{km}^2 \\ \text{km}^2 \\ \text{km}^2 \\ \text{km}^2/\text{s}^2 \\ \text{km}^2/\text{s}^2 \\ \text{km}^2/\text{s}^2. \end{array}$$

The object’s physical parameters are outlined in Table 4.2. The object being tracked is termed "object C".

Algorithm 9 Nested hybrid filter (NHF)

```
Inputs:
- $N_1$ iid samples $\boldsymbol{\theta}_0^i \sim p(\boldsymbol{\theta})$ and $N_2$ iid samples $\boldsymbol{x}_0^{i,j} \sim p(\boldsymbol{x}_0, \boldsymbol{\theta}_0^i)$, for $i = 1, ..., N_1$ and $j = 1, \ldots, N_2$ at time $t_0$.
- Initial weights $w_0^i = \frac{1}{N_1}, i = 1, ..., N_1$.
- $M$ observation time-stamps, $\{t_k\}_{k=1}^M$
- A set of observations $\boldsymbol{z}_k$ and their noise covariance matrices $\Omega_k, k = 1, ...M$.
- Resampling threshold $\varphi$.
Outputs:
- A collection of weighted samples $\{\chi_k^i, w_k^i\}_{i=1}^{N_1},\ k = 1, \ldots, M$, where $\chi_k^i = \{\boldsymbol{\theta}_k^i, \boldsymbol{x}_k^{i,1:N_2}\}$, for $i = 1, \ldots, N_1$.
Procedure: for each observation epoch $t_k$

                              Prediction
 1. Jittering: Generate new particles in the parameter space by computing:
    $\tilde{\boldsymbol{\theta}}_k^{(i)} = \begin{cases} \boldsymbol{\theta}_{k-1}^{(i)} & \text{with probability } 1 - \epsilon_{N_1}, \\ \boldsymbol{\theta}_{k-1}^{(i)} + \kappa_k^{(i)} & \text{with probability } \epsilon_{N_1}, \end{cases}$ for $i = 1, \ldots, N_1$,
    where $\epsilon_{N_1} = N_1^{-1/2}$ and $\kappa_k^{(i)}$ is an independent noise term.
 2. Propagate samples $\boldsymbol{x}_k^{i,j}$ from time $t_{k-1}$ to $t_k$, using a numerical scheme of choice, obtaining $\tilde{\boldsymbol{x}}_k^{i,j} \sim p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^{i,j}, \tilde{\boldsymbol{\theta}}_k^i), i = 1, ..., N_1$ and $j = 1, ..., N_2$.

                              Update
    FOR $i = 1, \ldots, N_1$:
 3. Transform the samples $\{\tilde{\boldsymbol{x}}_k^{i,j}\}_{j=1}^{N_2}$ through the measurement function to obtain predicted observations $\boldsymbol{y}_k^{i,j} = \mathcal{M}_k(\tilde{\boldsymbol{x}}_k^{i,j}),\ j = 1, ..., N_2$.
 4. Compute mean and covariance in the observation space:
    $\hat{\boldsymbol{y}}_k^{i,N_2} = \frac{1}{N_2} \sum_{j=1}^{N_2} \boldsymbol{y}_k^{i,j}$ and $C_{y,k}^{i,N_2} = \frac{1}{N_2 - 1} \sum_{j-1}^{N_2} (\boldsymbol{y}_k^{i,j} - \hat{\boldsymbol{y}}_k^{i,N_2})(\boldsymbol{y}_k^{i,j} - \hat{\boldsymbol{y}}_k^{i,N_2})^\top + \Omega_k$
 5. Compute the ensemble mean and cross-covariance matrix:
    $\hat{\boldsymbol{x}}_k^{i,N_2} = \frac{1}{N_2} \sum_{j=1}^{N_2} \tilde{\boldsymbol{x}}_k^{i,j}$ and $C_{\chi y,k}^{i,N_2} = \frac{1}{N_2 - 1} \sum_{j=1}^{N_2} (\tilde{\boldsymbol{x}}_k^{i,j} - \hat{\boldsymbol{x}}_k^{i,N_2})(\boldsymbol{y}_k^{i,j} - \hat{\boldsymbol{y}}_k^{i,N_2})^\top$.
 6. Compute the Kalman gain $K_G^{i,N_2} = C_{\chi y,k}^{i,N_2} (C_{y,k}^{i,N_2})^{-1}$.
 7. Update the ensemble samples
    $\check{\boldsymbol{x}}_k^{i,j} = \tilde{\boldsymbol{x}}_k^{i,j} + K_G^{i,N_2} (\boldsymbol{z}_k - \boldsymbol{y}_k^{i,j} + \boldsymbol{s}_k^{i,j}),\ \boldsymbol{s}_k^{i,j} \sim \mathcal{N}(0, \Omega_k),\ j = 1, ..., N_2$.
 8. Compute primary (parameter) layer weights as
    $w_k^i \propto w_{k-1}^i \lambda_k^i$, where $\lambda_k^i = \frac{1}{N_2} \sum_{j=1}^{N_2} p(\boldsymbol{z}_k | \check{\boldsymbol{x}}_k^{i,j})$
    END FOR
 9. Let $\check{\chi}_k^i = [\tilde{\boldsymbol{\theta}}_k^i, \check{\boldsymbol{x}}_k^{i,1:N_2}]^\top$.
10. Compute the normalised ESS, $\text{NESS}_k = \frac{1}{N_1 \sum_{i=1}^{N_1} (w_k^i)^2}$.
11. If $\text{NESS}_k < \varphi$,
    Resample $N_1$ times with replacement to generate new particles $\{\chi_k^i\}_{i=1}^{N_1}$, where $\chi_k^i = \{\boldsymbol{\theta}_k^i, \boldsymbol{x}_k^{i,1:N_2}\}$ and set $w_k^i = \frac{1}{N_1}, i = 1, ..., N_1$. Otherwise, set $\chi_k^i = \check{\chi}_k^i$.
```

Table 4.2: Physical properties of object C

	Object C
Semi-major axis (km)	7173.058
Altitude (km)	794.921
Mass (kg)	3.472
$\mathbb{A}$ (m^2)	0.038
Drag coefficient (-)	1.737

The observation state space, as well as relevant information regarding observation sources can be found in Section 2.2.3. In LEO, these observations are assumed to come from radar measurements composed of rate,range rate, azimuth and elevation readings, which are irregularly spaced in time.

4.4.2. GEO

The second test case used for the validation of the above algorithms is a GEO scenario with synthetic observations generated with a high-fidelity dynamical model, spanning a period of just over six days. Its initial state $\boldsymbol{x}_0$ is given by

$$\boldsymbol{x}_0 = \text{[-26115.41km \quad -33131.04km \quad 41.98km \quad 2.41km/s \quad -1.90km/s \quad -0.01km/s]}^\top$$

and its initial covariance $\boldsymbol{\Sigma}_0$ is given by

$$\boldsymbol{\Sigma}_0 = \begin{bmatrix} 0.15 & 0.03 & -3.6\times10^{-4} & 0 & 0 & 0 \\ 0.03 & 0.1645 & 5.26\times10^{-5} & 0 & 0 & 0 \\ -3.55\times10^{-4} & 5.26\times10^{-5} & 0.0074 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6.86\times10^{-10} & -4.04\times10^{-10} & -2.02\times10^{-12} \\ 0 & 0 & 0 & -4.04\times10^{-10} & 4.92\times10^{-10} & 1.43\times10^{-12} \\ 0 & 0 & 0 & -2.02\times10^{-12} & 1.43\times10^{-12} & 3.98\times10^{-11} \end{bmatrix} \begin{matrix} \text{km}^2 \\ \text{km}^2 \\ \text{km}^2 \\ \text{km}^2/\text{s}^2 \\ \text{km}^2/\text{s}^2 \\ \text{km}^2/\text{s}^2 \end{matrix}$$

The object's physical parameters are outlined in Table 4.3. The object in question is denoted "object D".

Table 4.3: Physical properties of object D

	Object D
Semi-major axis (km)	42166.167
Altitude (km)	35807.980
Mass (kg)	166.193
$\mathbb{A}$ (m^2)	12.908
SRP coefficient (-)	1.415

The observation state space, as well as relevant information regarding observation sources can be found in Section 2.2.3. In GEO, these observations are assumed to come from telescope measurements composed of RA and DEC readings which are irregularly spaced in time.

4.5 Results: state-only tracking

4.5.1. Simulation setup

In this section, a selection of filters from Section 4.2 is chosen to study their performance on two different cases, object C in LEO, and object D in GEO. The simulation setups for this part of the proposed methodology have the following characteristics:

- The entire simulation time-span for both scenarios is just over five days.
- A reference orbit is simulated in HF by propagating the initial conditions through a deterministic integration model. This orbit is used for the generation of noisy observations at different time-stamps and for the calculation of error metrics.
- The HF model in this case involves an integration time-step of $h = 10$s, J_2 to J_4 gravitational potential terms, and the US76 atmospheric model.
- For the implementation of the filters, a stochastic propagation scheme is used with a low fidelity (LF) dynamical model, and an appropriately scaled diffusion coefficient is used to account for model mis-match.
- The LF model in this case involves an integration time-step of $h = 100$s, the J_2 gravitational potential term, and the US76 atmospheric model.

The metrics for performance evaluation include the RMSE, estimated uncertainty, and computational cost. For both scenarios, two tests are performed to determine optimal filter conditions[3].

The first test is the sample number (SN) test, i.e., the determination of an appropriate number of samples to be used with each algorithm, in order to achieve a sufficiently low RMSE. For this test, the number of samples is varied and the RMSE recorded for each algorithm. The second test is the minimum re-observation time (MROT) test, i.e., the study of performance with respect to the minimum time interval between observations. For this test, an appropriate number of samples from the SN test is chosen for each algorithm, and run with different minimum observation frequency setups. Each simulation for each test is run a total of 20 times, so that mean error values and standard deviations can be computed. The value of the diffusion coefficient is assumed to act only in the velocity components of the state (due to the acceleration term) and is assumed equal in all three directions. It is given by $\sigma_W = 5 \times 10^{-8}$ km^2/s^2 in LEO, and by $\sigma_W = 10^{-8}$ km^2/s^2 in GEO.

The computations for the proposed methods in both setups are run on a Macbook Pro with an Apple M1 processor.

[3] By optimal filter conditions, what is meant is the set of conditions which leads to the best results in terms of accuracy for each filter.

4.5.2. LEO

Sample number test

Figure 4.3 shows the results of the SN test. For each algorithm, the RMSE is shown for the corresponding sample number used. The number of simulations is 20 per algorithm, per set-up.

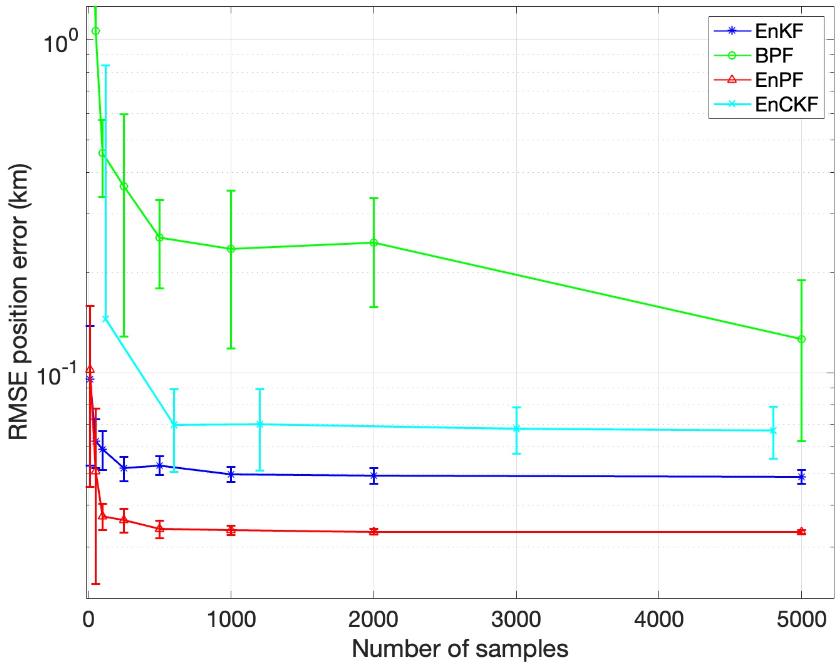


Figure 4.3: Position RMSE (km) against the number of samples for the LEO scenario, for the EnKF, EnPF, EnCKF and the BPF. A limit is reached for all but BPF beyond which the inclusion of more samples does not bring about any further improvements in RMSE.

The EnPF achieves the lowest RMSE, followed by the EnKF. All the algorithms except for the BPF reach a point where increasing the number of samples brings no additional improvements in the RMSE. Acceptable values are obtained (under 100 m) for all algorithms except for the BPF, which does not show appropriate results until N is at least 5,000 particles. The BPF does seem to achieve greater accuracy as N increases but the associated increase in computational cost must be considered.

Minimum re-observation time test

Figure 4.4 shows the results of the MROT test. For each algorithm, the RMSE is shown for the corresponding MROT used. The number of simulations is 20 per algorithm, per set-up. The optimal numbers of samples used for each algorithm, as per the SN test, are $N_{EnKF} = 500$, $N_{EnCKF} = 50$, $N_{EnPF} = 500$, $N_{BPF} = 5,000$, whilst no samples other than cubature points are required for CKF.

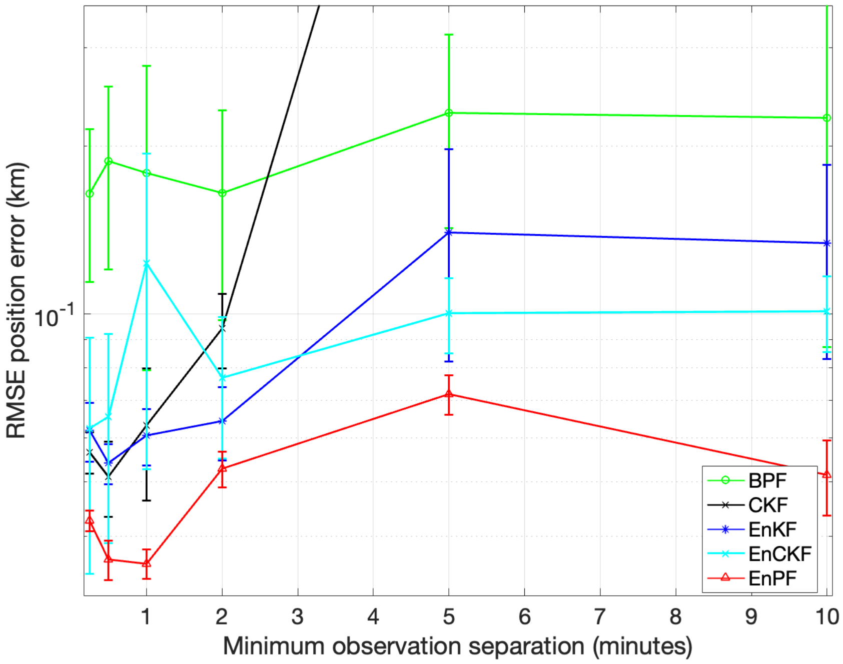


Figure 4.4: Position RMSE (km) against minimum observation separation of the simulated observation set in LEO. A weak upward correlation is generally observed between the RMSE and increasing separation values for the algorithms studied.

The lowest errors are observed with the EnPF. There is generally a very small rise in the RMSE values for most of the algorithms as the MROT is increased, up to a point of equilibrium. It is important to note that the CKF diverges once the MROT reaches two minutes. The CKF is the only pure Gaussian filter, and this is evidence that after a long-enough propagation period, the Gaussianity assumption in LEO ceases to hold and estimation accuracy diminishes to the point of filter divergence.

A minimum separation time of 30 seconds is chosen as the reference value with which to perform the experiments in the next section. Note that this is the minimum time separation between observations, but there are still observations which are separated by more than eight hours.

Performance comparison

Having obtained the optimal configuration for the filters to be studied, this section is devoted to assessing the RMSE values of the algorithms at each time-step throughout an entire simulation, and assessing the estimate covariance. In addition, the computational cost is examined to determine the applicability and suitability of each of the algorithms depending on the situation presented. Table 4.4 shows the RMSE and its standard deviation σ_ϵ computed over 20 independent simulations for the five algorithms studied in this section, followed by their computational cost.

Table 4.4: Estimation RMSE for the five algorithms applied to the LEO scenario. The metrics are the root mean squared error (RMSE) in position and velocity, the error standard deviations included as $\epsilon \pm \sigma_\epsilon$, and the mean run-time.

Algorithm	Position RMSE (m)	Velocity RMSE (m/s)	Run-time (mins)
CKF	52.35 ± 10.16	0.04 ± 0.01	0.71
EnKF	54.01 ± 4.46	0.08 ± 0.01	1.23
EnCKF	65.46 ± 18.67	0.05 ± 0.02	1.24
BPF	188.08 ± 65.75	0.25 ± 0.09	3.53
EnPF	36.18 ± 6.54	0.09 ± 0.01	1.63

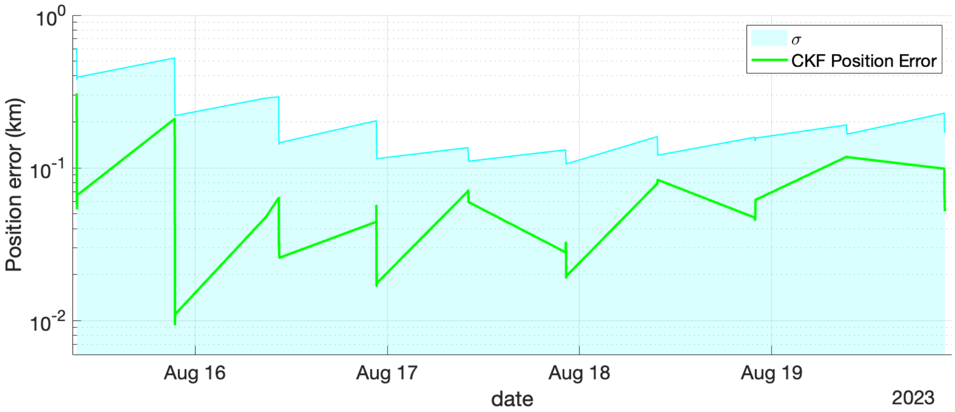


Figure 4.5: Position RMSE (km) and standard deviation of samples (shaded region) for the CKF. Simulation of five days.

In addition, the RMSE and the standard deviation of the collection of samples throughout the entire simulation is shown for all algorithms in Figures 4.5 to 4.9. Note that in all figures, several spikes in error can be observed, which correspond to the long re-observation times (of over eight hours at times) in between observation sets.

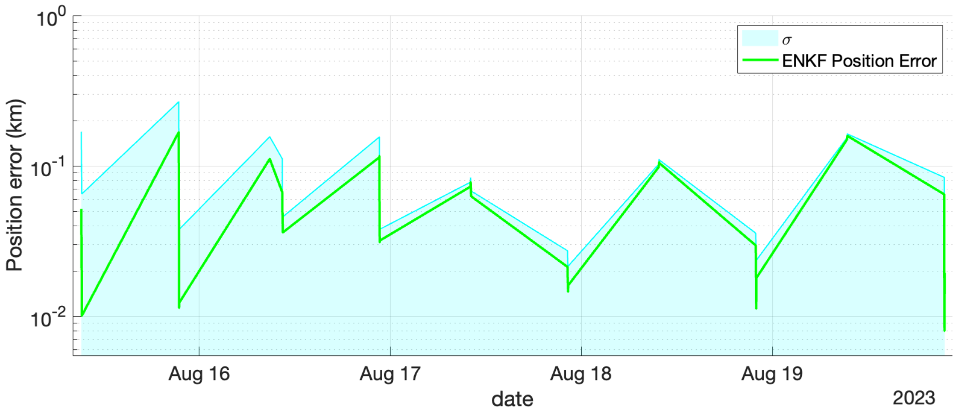


Figure 4.6: Position RMSE (km) and standard deviation of samples (shaded region) for the EnKF. Simulation of five days.

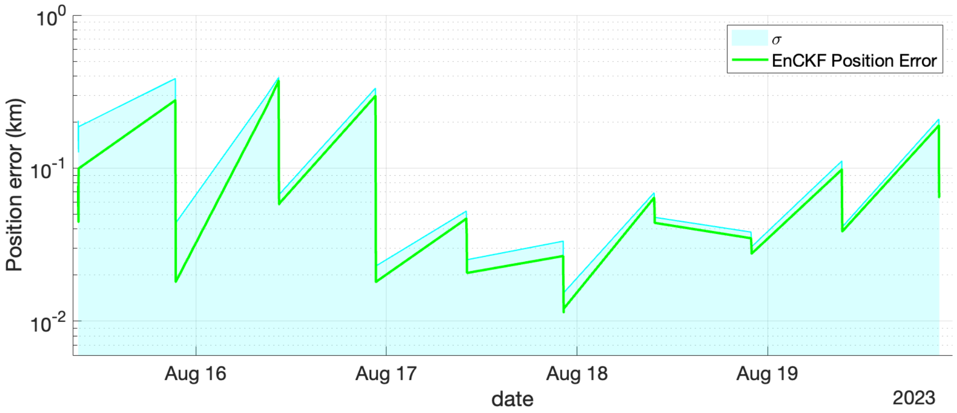


Figure 4.7: Position RMSE (km) and standard deviation of samples (shaded region) for the EnCKF. Simulation of five days.

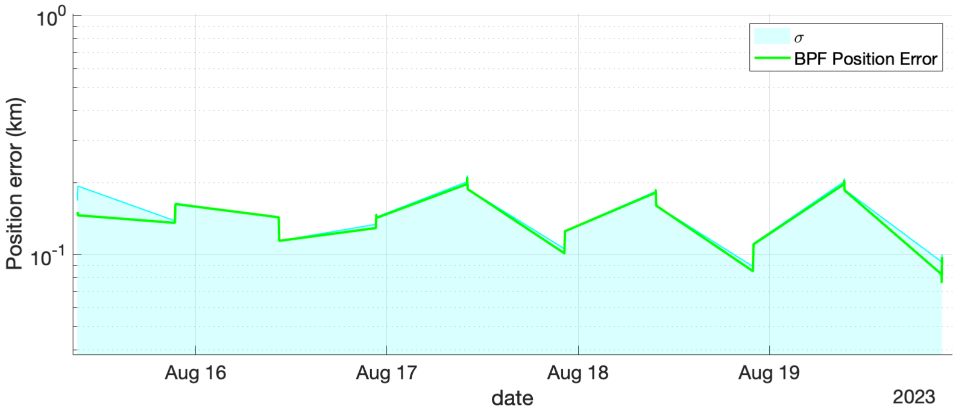


Figure 4.8: Position RMSE (km) and standard deviation of samples (shaded region) for the BPF. Simulation of five days.

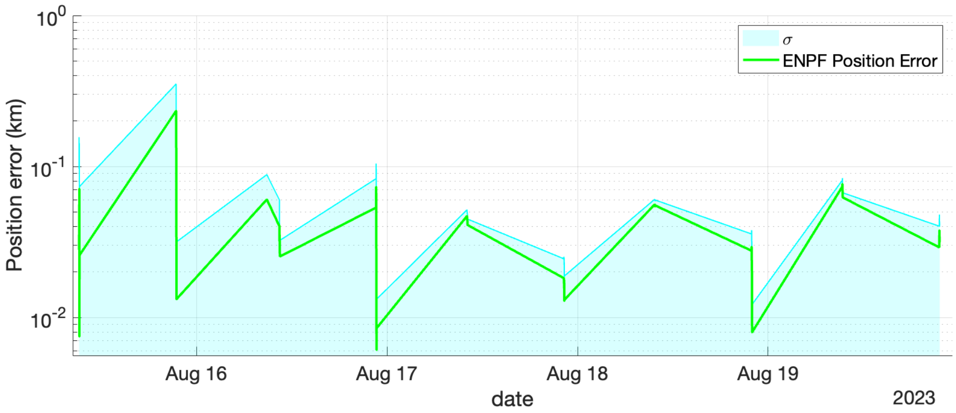


Figure 4.9: Position RMSE (km) and standard deviation of samples (shaded region) for the EnPF. Simulation of five days.

The CKF in Figure 4.5 attains a remarkably low RMSE throughout the entire simulation when the MROT is sufficiently small (30 seconds). The EnKF of Figure 4.6, and the EnPF of Figure 4.9, both of 500 particles, also attain a very low RMSE throughout the entire simulation, with more bounded standard deviations than the CKF. The EnCKF of 50 MC samples of Figure 4.7, which combines characteristics of both the CKF and the EnKF, shows large initial RMSE values which taper off mid-simulation, and achieves a remarkably low final RMSE. The BPF in Figure 4.8, of 5,000 particles, shows a higher RMSE than the rest of the algorithms, due to large error peaks after long propagation portions and a higher standard deviation among the particles.

4.5.3. GEO

Sample number test

Figure 4.10 shows the results of the SN test for the GEO scenario. For each algorithm, the RMSE is shown for the corresponding number of samples used over the entire simulation. The number of simulations is 20 per algorithm, per set-up.

The EnKF achieves the lowest RMSE, followed by the EnCKF. The EnPF achieves a lower RMSE than the BPF for relatively low SN but is overtaken at the $N = 1,000$ mark. This contrasts with the EnPF's performance in the LEO scenario, where it performs significantly better than the rest of the algorithms. All of the algorithms except for the BPF reach a point where increasing the number of samples brings no additional improvements in the RMSE. The BPF seems to achieve greater accuracy as N increases but the associated increase in computational cost must be considered. It is able to obtain good accuracy values in the $N = 1,000 - 5,000$ region.

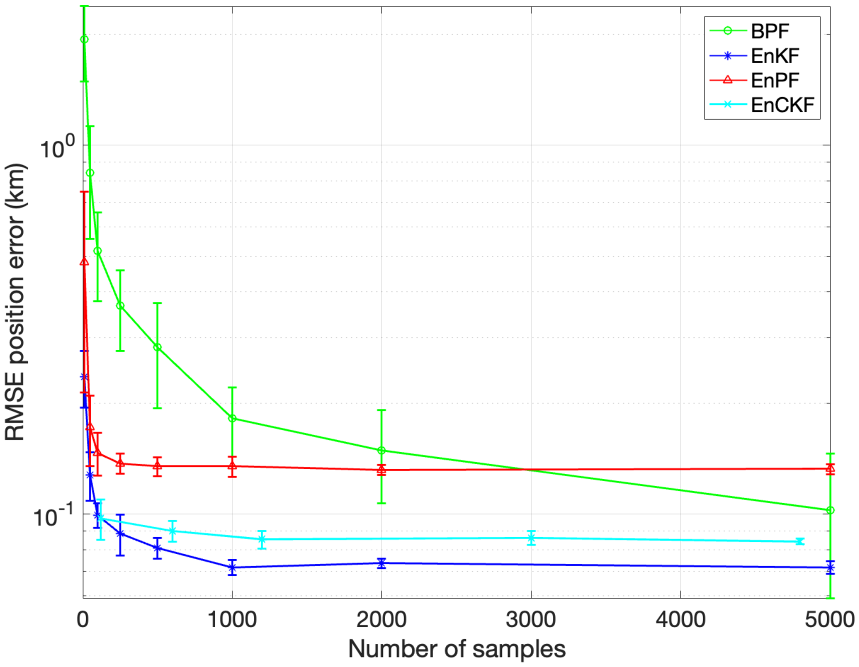


Figure 4.10: Position RMSE (km) against the number of samples for the GEO scenario, for the EnKF, EnPF, EnCKF and the BPF. A limit is reached for all but BPF beyond which the inclusion of more samples does not bring about any further improvements in RMSE.

Minimum re-observation time test

Figure 4.11 shows the results of the MROT test. For each algorithm, the RMSE is shown for the corresponding re-observation time used. The number of simulations is 20 per algorithm. The optimal numbers of MC samples used for each algorithm, as per the SN test, are $N_{EnKF} = 500$, $N_{EnCKF} = 50$, $N_{EnPF} = 500$, $N_{BPF} = 1,000$, whilst no samples other than cubature points are required for CKF.

The lowest errors are once again observed with the EnKF, whilst the other three do not show large differences. It seems like in GEO, there is little correlation between the RMSE obtained and minimum separation time, though, in general, RMSEs tend to decrease slightly as MROT increases, with the exception of EnKF, which exhibits the opposite trend. The decrease seems to lead to a similar value as that observed for very low MROT. A possible reason for this may be that in general, the Gaussian assumption holds well for long periods of time in GEO, meaning that fewer weighted updates may be necessary. This may in fact be beneficial due to the avoidance of degeneracy. A minimum separation time of five minutes is chosen as the reference value with which to perform the experiments in the next section. Note that this is the minimum time between observations, but there are intervals of more than ten hours without any observations, as is common in GEO.

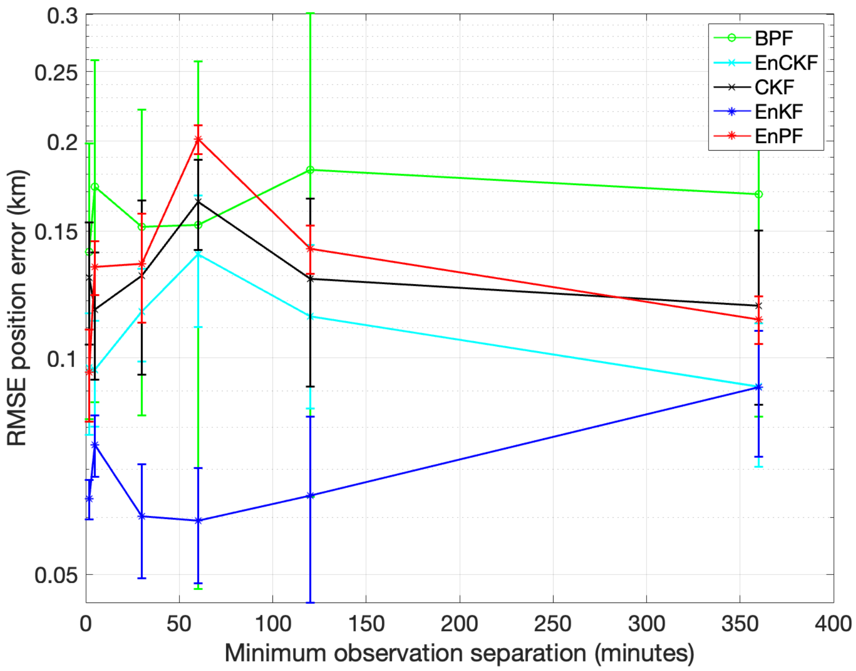


Figure 4.11: Position RMSE (km) against minimum observation time separation of the simulated observation set in GEO. A weak downward correlation can be observed for most algorithms, showing a decrease in RMSE as the MROT increases, except for the EnKF, which shows a slight upward trend.

Performance comparison

Having obtained the best configuration for the filters to be studied, this section is devoted to assessing the RMSE values of the algorithms at each time-step throughout an entire simulation, and assessing the estimate covariance. In addition, the computational cost is examined to determine the suitability of each of the algorithms depending on the situation presented. Table 4.5 shows the RMSE and the standard deviation σ_ϵ computed over 20 simulations for the five algorithms studied in this section, followed by their computational cost.

Table 4.5: Estimation RMSE for the five algorithms applied to the GEO scenario. The metrics are the root mean squared error (RMSE) in position and velocity, the error standard deviations included as $\epsilon \pm \sigma_\epsilon$, and the mean run-time.

Algorithm	Position RMSE (m)	Velocity RMSE (m/s)	Run-time (min)
CKF	116.67 ± 23.46	0.01 ± 0.005	0.35
EnKF	80.72 ± 7.35	0.01 ± 0.002	0.98
EnCKF	92.92 ± 5.91	0.01 ± 0.002	1.13
BPF	181.61 ± 38.49	0.07 ± 0.004	1.60
EnPF	135.71 ± 11.57	0.03 ± 0.01	1.11

In addition, the RMSE and the standard deviation of the collection of samples throughout an entire simulation is shown for all algorithms in Figures 4.12 to 4.16.

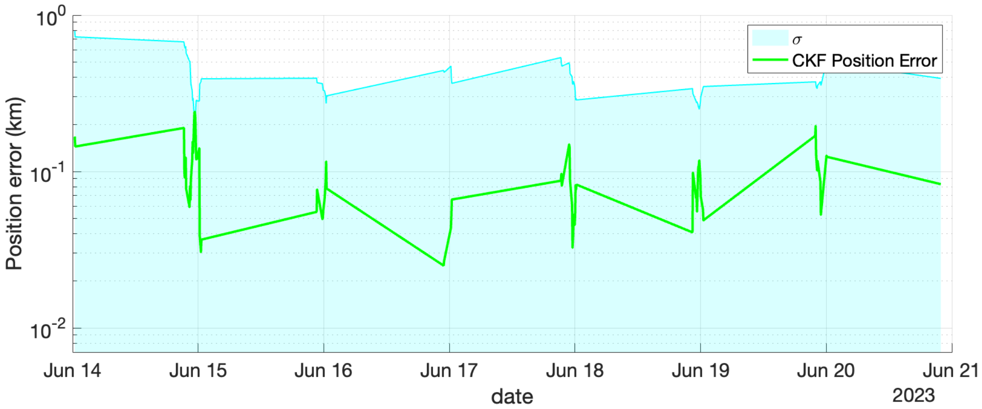


Figure 4.12: Position RMSE (km) and standard deviation of samples (shaded region) for the CKF. Simulation of five days.

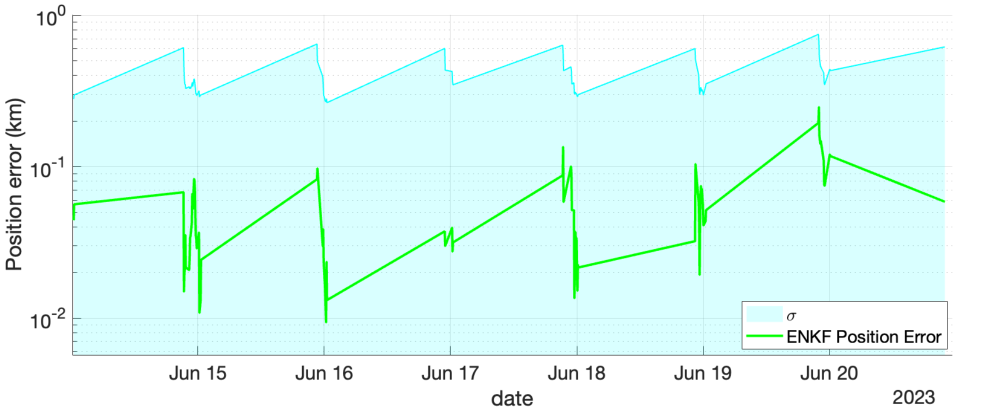


Figure 4.13: Position RMSE (km) and standard deviation of samples (shaded region) for the EnKF. Simulation of five days.

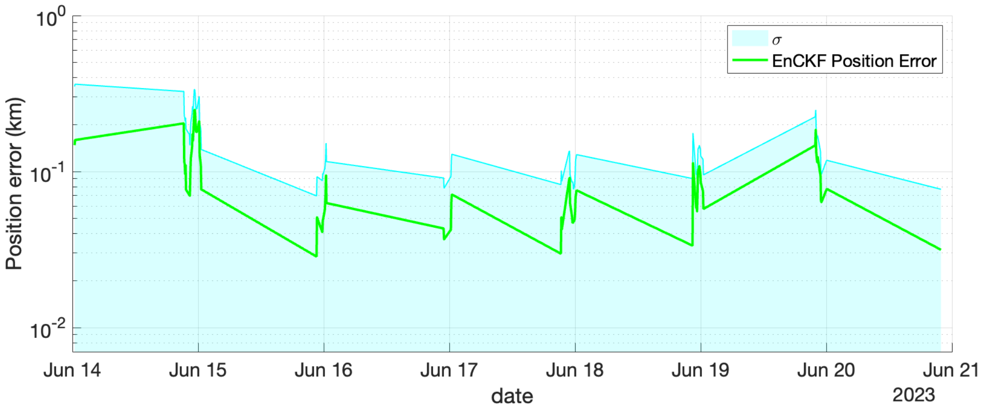


Figure 4.14: Position RMSE (km) and standard deviation of samples (shaded region) for the EnCKF. Simulation of five days.

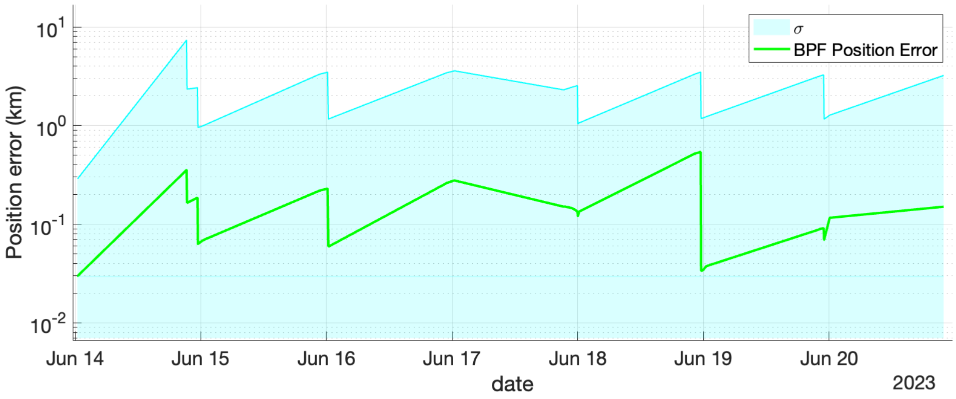


Figure 4.15: Position RMSE (km) and standard deviation of samples (shaded region) for the BPF. Simulation of five days.

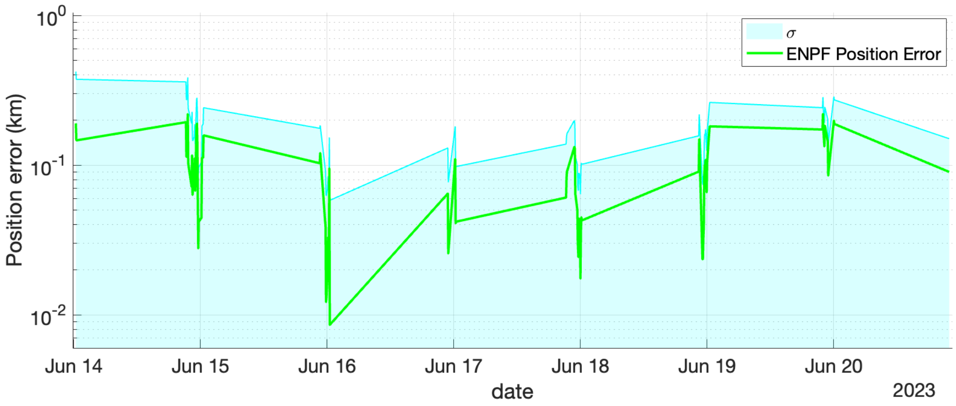


Figure 4.16: Position RMSE (km) and standard deviation of samples (shaded region) for the EnPF. Simulation of five days.

Note that in all figures, several spikes in error can be observed, which correspond to the long re-observation times (of over ten hours at times) in between observation sets. The CKF in Figure 4.12 and the EnKF (of 500 particles) in Figure 4.13 attain an acceptably low RMSE throughout the entire simulation, but exhibit a significant standard deviation in the samples, several times larger than the EnCKF of 50 MC samples (Figure 4.14), which, despite combining characteristics of both the CKF and the EnKF, achieves similar RMSE values, but with a greatly reduced covariance. The BPF in Figure 4.15, of 1,000 particles shows a higher RMSE than the rest of the algorithms, and a much higher covariance in the particles. However, it still represents an acceptable RMSE (under 300 m), enabling its use in GEO. The EnPF of 500 particles in Figure 4.16 attains low RMSE and low covariance values, making it a suitable option along with the EnKF, the CKF, and the EnCKF of 50 MC samples for the tracking of RSOs in GEO at an affordable cost. It is worth noting that the BPF, as reiterated in previous sections, is the simplest type of PF. A vast suite of techniques can be employed for the improvement of estimation error metrics, though these can at times be complicated to apply, or increase computational cost significantly.

4.6 Test case: state-tracking and parameter estimation

4.6.1. LEO

The first test case used for the validation of the above algorithms is a LEO scenario with synthetic observations generated with a high-fidelity dynamical model, spanning a period of just over nine days. Its initial state $\boldsymbol{x}_0$ is given by

$$\boldsymbol{x}_0 = \quad [\text{-2815.17km} \quad \text{6200.05km} \quad \text{-967.78km} \quad \text{0.15km/s} \quad \text{-1.09km/s} \quad \text{-7.53km/s}]^\top$$

and its initial covariance $\boldsymbol{\Sigma}_0$ is given by

$$\boldsymbol{\Sigma}_0 = \begin{bmatrix} 0.05 & -0.08 & 0.02 & 0 & 0 & 0 \\ -0.08 & 0.18 & -0.03 & 0 & 0 & 0 \\ 0.01 & -0.03 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.53\times10^{-8} & 4.81\times10^{-9} & -5.29\times10^{-9} \\ 0 & 0 & 0 & 4.81\times10^{-9} & 1.04\times10^{-8} & 3.46\times10^{-8} \\ 0 & 0 & 0 & -5.29\times10^{-9} & 3.46\times10^{-8} & 2.42\times10^{-7} \end{bmatrix} \begin{matrix} \text{km}^2 \\ \text{km}^2 \\ \text{km}^2 \\ \text{km}^2/\text{s}^2 \\ \text{km}^2/\text{s}^2 \\ \text{km}^2/\text{s}^2. \end{matrix}$$

The object being tracked is termed "object E", and its physical parameters are outlined in Table 4.6.

Table 4.6: Physical properties of object E

	Object C
Semi-major axis (km)	6879.602
Altitude (km)	501.466
Mass (kg)	2.678
$\mathbb{A}$ (m^2)	0.0402
Drag coefficient (-)	2.667

4.7 Results: state tracking and parameter estimation

4.7.1. Simulation configuration

The parameter θ introduced in Section 4.3 takes the form of σ_W in Section 2.4.2. Furthermore, it is assumed that the parameter is equivalent in all three directions of the velocity, i.e., $\sigma_W \equiv \sigma_{W(v_x)} = \sigma_{W(v_y)} = \sigma_{W(v_z)}$, and hence the problem is reduced to estimating a scalar, whose initial support is given as $[10^{-10},\ 10^{-6}]$, to account for the expected magnitude range of the truncated accelerations in the dynamical model.

A reference orbit is required to be able to generate synthetic observations. This orbit may be computed deterministically, or stochastically. In this section, we conduct validation experiments on two different simulation setups:

- The first objective is to correctly estimate the nominal parameter used in the generation of the reference orbit. In this case, HF stochastic propagation is used for the generation of the reference orbit, subject to a fixed stochastic diffusion coefficient, set to $\sigma_W = 5 \times 10^{-8}$. In the application of the filters, an HF model is used along with stochastic propagation, but the goal is to correctly estimate the diffusion coefficient in the SDE, by using the algorithms hereby presented.

- The second objective is to estimate the diffusion coefficient that is necessary to account for model mismatches. In this case, the reference HF orbit is computed deterministically (i.e., $\sigma_W = 0$), and in the application of the filters, a low-fidelity (LF) model is used to evaluate the ability of the algorithms to characterise the diffusion magnitudes required to offset the impact of reduced fidelity.

In all cases, the HF model used is the HPOP in MATLAB [20], which applies the NRLMSISE-00 drag model, the GGM03C model for Earth gravitational potential of up to order and degree ten, luni-solar perturbations, as well as other planets' gravitational influence and relativity perturbations. The LF model used in the second setup is the HPOP library, but applying only the NRLMSISE-00 drag model and the gravitational potential of up to order and degree five. The computations for the proposed methods in both setups are run on a Macbook Pro with an Apple M1 processor.

4.7.2. Estimating the nominal parameter

In this section, we tackle the first scenario described in Section 4.7.1. In particular, we assess whether the proposed filters can estimate the noise parameter, σ_W, with sufficient accuracy. This is done by comparing the estimate provided by each filter with the ground-truth parameter σ_W used to generate the synthetic data. Performance assessment is carried out in terms of:

- The RMSE in position (m) and velocity (m/s) of the algorithms described in Section 4.3. These are calculated for both predicted and estimated states with respect to the reference orbit, at each observation time-stamp.
- The computational run-time of each algorithm simulation.

To provide a benchmark, the EnKF and EnPF are run with models in which there are no unknown parameters. In addition, a nonlinear batch least squares (NBLS) is run a number of times with samples drawn from the initial distribution. The above metrics are helpful when determining the correct number of samples to be used in a trade-off between accuracy and speed. As long as none of the filters diverge computational speed takes precedence for the assessment of performance. The number of samples for each of the algorithms, following these criteria are $N_{\text{EnKFup}} = 450$, $N_{\text{EnPFup}} = 350$ and $N_{\text{NHF}} = 50 \times 50$ (i.e., $N_1 = N_2 = 50$). In the implementation of the NHF, the jittering step is performed by setting $\epsilon_{N_1} = N_1^{-\frac{1}{2}}$, and taking $\kappa_k^{(i)}$, the noise term driving the jittering procedure, to be Gaussian noise with zero mean and covariance set to 0.25.

Parameter estimation

The estimates of σ_W for each filter are shown in Figure 4.17, with the corresponding standard deviations computed over 30 independent simulations. The black dotted line is the ground truth, given by $\sigma_W = 5 \times 10^{-8}$ in units of acceleration, i.e., km^2/s^2. Initial values $\sigma^i_{W,0}$, $i = 1, \ldots, N$ are drawn from the parameter prior $p(\sigma_{W,0})$, with support $[10^{-10},\ 10^{-6}]$ km^2/s^2. All filters adjust the order of magnitude relatively quickly. Large fluctuations can be observed in the EnKFup (blue line) with a large variance (blue shaded region), indicating a relative lack of precision compared to the other two algorithms. Nevertheless, even with these fluctuations, all three filters are able to converge acceptably well to the correct value (black dotted line) and manage to achieve sufficiently low error rates in position. The most precise estimator of σ_W is the NHF, as seen by the standard deviation (green) around the estimate, as well as the estimator with the lowest bias throughout the studied time-span. However, the EnPFup achieves a reasonably accurate estimate (though it requires a longer simulation time to do so), and its overall computational cost is considerably lower than that of the NHF.

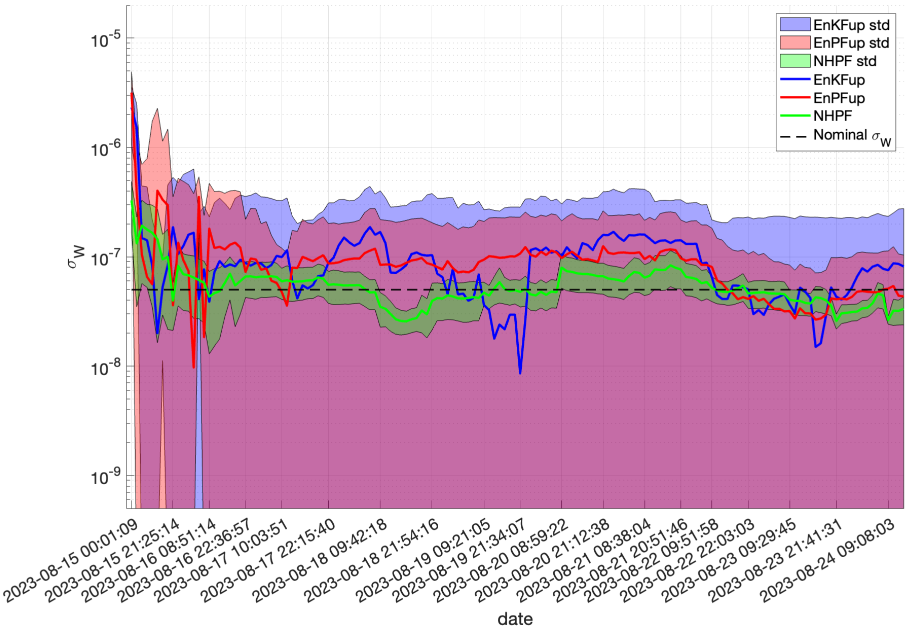


Figure 4.17: Parameter estimates shown in solid lines for the EnKFup (blue), the EnPFup (red) and the NHF (green), and nominal values (dotted line). Shaded areas representing the standard deviations are shown with the corresponding colours of each algorithm.

In Figure 4.18, an estimate of the parameter posterior pdf for all three algorithms is shown at two different instants: in cyan, the particles before the algorithm has converged (corresponding to one of the initial time-steps in Figure 4.17), and in green, the particles at some time step after convergence near the ground-truth value. A visibly larger spread towards large values can be observed for the cyan curve, and a more concentrated area can be observed for the green curves, indicating a denser concentration of samples around the ground-truth value (dotted line). While for the EnKFup the increase in concentration of probability mass is less distinguishable between the two time steps, the EnPFup and NHF show a more pronounced difference in curve peak location and concentration.

Position and velocity errors

In this subsection, the position and velocity RMSEs are shown for the entirety of the simulation period (nine days), in order to observe the trend in the accuracy of the algorithms as the parameter is adjusted in the background. Figure 4.19 shows the RMSEs in position, while Figure 4.20 shows the RMSEs in velocity of the three algorithms. The EnKFup is shown as a solid blue line with star pointers, the EnPFup as a solid red line with triangle pointers, and the NHF is shown as a thick solid green line.

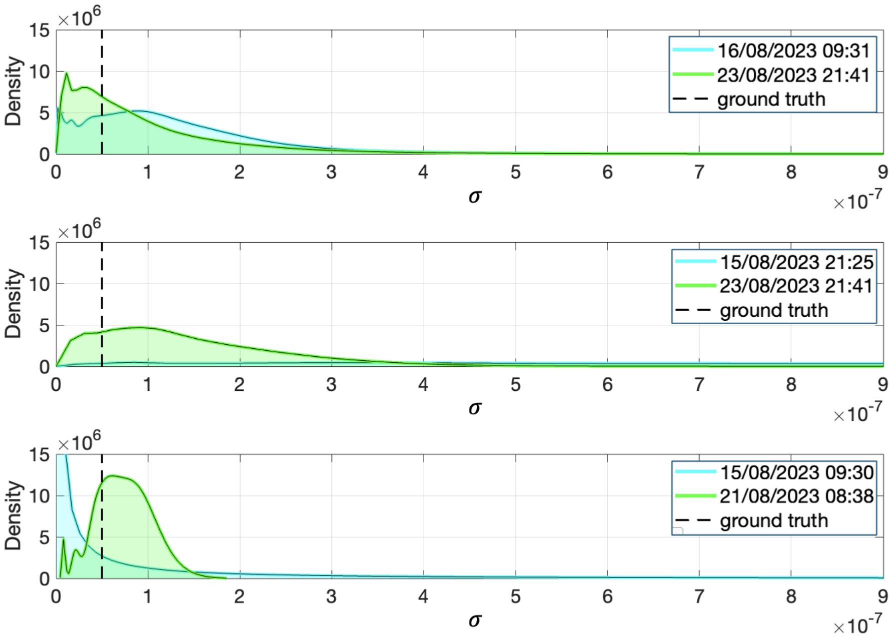


Figure 4.18: Estimated posterior pdf of σ_W for the EnKFup (top), the EnPFup (middle) and the NHF (bottom). Two time-steps are shown: in cyan, a time step before convergence (15 August 2023, ~ 21:00), where a larger spread can be seen, and in green, a time step after convergence (23 August 2023, ~ 21:40), with more concentration of samples around the nominal value (dotted line).

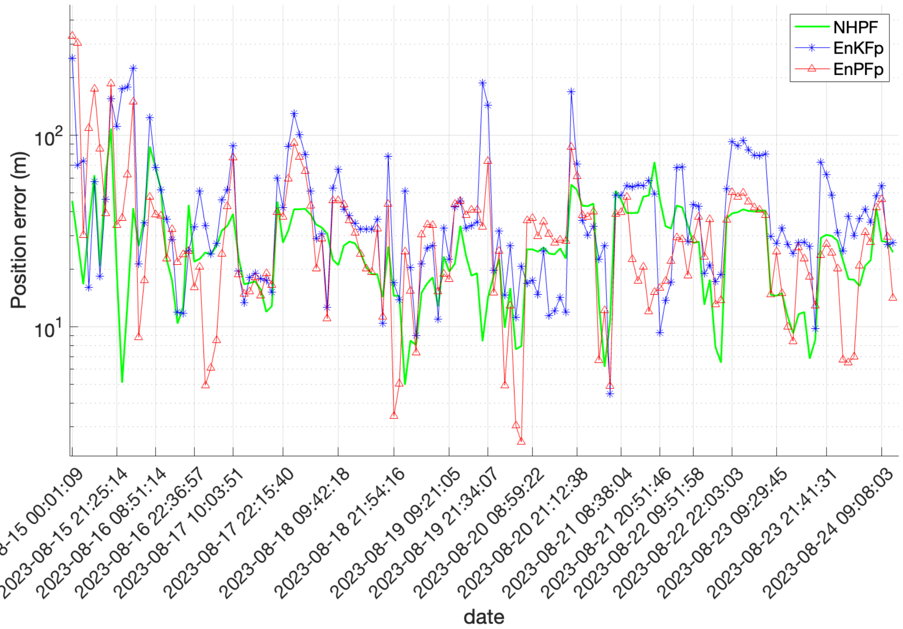


Figure 4.19: RMSEs in position for the three proposed algorithms which estimate the given σ_W. The solid blue line shows the EnKFup errors, the red line shows the EnPFup errors, and the green line shows the NHF errors.

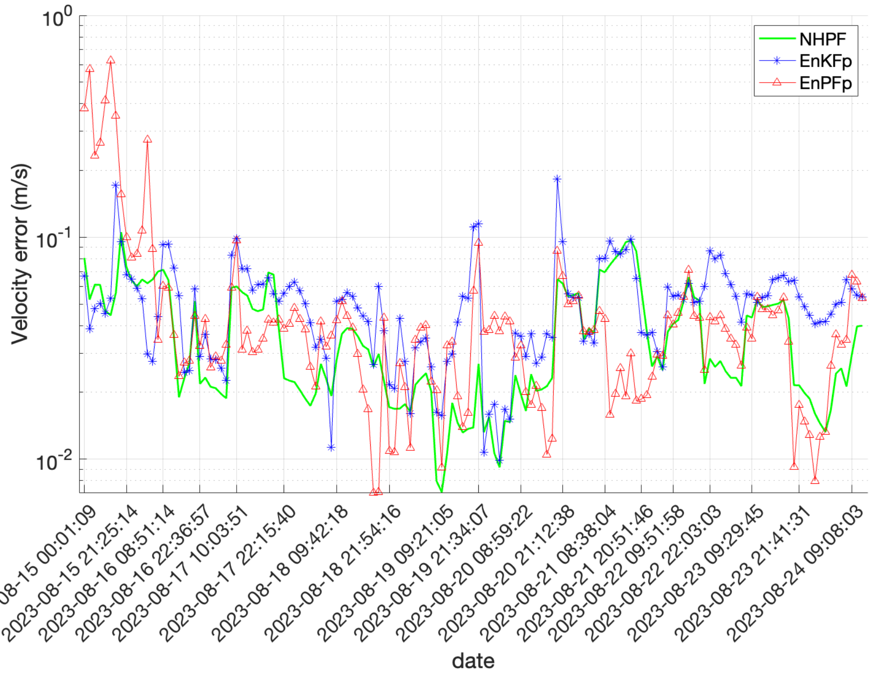


Figure 4.20: RMSEs in velocity for the three proposed algorithms which estimate the given σ_W. The solid blue line shows the EnKFup errors, the red line shows the EnPFup errors, and the green line shows the NHF errors.

For the three algorithms, the estimated errors fluctuate significantly throughout the propagation due to relatively long re-observation times. This means a much longer observation free propagation between tracklets than between observations in the same tracklet. A slight downward tendency can be observed, both in the position and in the velocity errors, indicating not only a correct and satisfactory performance of the filter, but also an adjustment of the unknown parameter as it is being estimated in the background. Both the EnKFup and EnPFup begin with large errors in position (over 120 m), which are in stark contrast to the final values (20-30 m), but achieve errors as low as 3 m throughout the simulation.

Performance comparison

The computational run-time for each algorithm is shown in Table 4.7 below. Note that the period of simulation is nine days, which is a significant propagation period, and the reason why the run-times are relatively high. The results shown are the averages of 10 simulations for each algorithm. Shown as well, are the RMSEs in the form $\epsilon \pm \sigma_\epsilon$, where ϵ is the mean error and σ_ϵ is the standard deviation. The table also includes results for the ensemble Kalman filter (EnKF) and the ensemble particle filter (EnPF) with known parameter σ_W. These algorithms are implemented as in Algorithm 4 and Algorithm 6, respectively, which simply track the state, and assume the correct σ_W value. Additionally, Table 4.7 also shows the errors obtained by a nonlinear batch least squares (NBLS) algorithm applied to the same initial conditions [4].

Table 4.7: Performance comparison for the three algorithms. The metrics are the root mean squared error (RMSE) in position and velocity of the estimated trajectory compared to the reference trajectory, the error standard deviations included as $\pm\sigma_\epsilon$ and the mean run-time.

Algorithm	Position RMSE (m)	Velocity RMSE (m/s)	Run-time (min)
EnKFup	44.31 ± 9.12	0.07 ± 0.004	35.32
EnKF	10.15 ± 3.52	0.01 ± 0.001	34.50
EnPFup	40.19 ± 11.13	0.06 ± 0.002	41.51
EnPF	14.29 ± 3.52	0.02 ± 0.002	39.42
NHF	17.71 ± 7.88	0.01 ± 0.002	351.29
EnKF(50)*	10.77 ± 4.79	0.01 ± 0.001	12.21
NLBLS	54.38 ± 16.81	0.06 ± 0.04	12.44

*The EnKF(50) run involves using the same number of samples for the state, $N = 50$, as the NHF run.

The NHF is by far the most costly, showing a run-time one order of magnitude higher than the rest. The EnKFup and EnPFup show similar run-times, with the EnPFup attaining slightly smaller errors. Together with the results in Section 4.7.2, this seems to indicate a better trade-off between computational cost and accuracy for the EnPFup, achieving errors which are comparable to a nonlinear batch least squares filter with known parameters, an industry standard. The nominal parameter versions of the algorithms (EnKF and EnPF) achieve a lower RMSE, as expected.

The errors in Table 4.7 represent the estimation RMSE of the corresponding trajectory, upon being updated by each incoming measurement. However, prediction errors (before the measurement update) can be calculated by using a single forward pass of a propagation model using the Qlaw method [126]. The idea behind this is to obtain a dynamics-informed interpolation between estimated values, by calculating appropriate acceleration magnitudes in orbital elements at t_{k-1} in order to drive the state towards the estimated target value at time t_k. An RMSE value can be calculated at this point between this corrected trajectory and the reference trajectory.

The results for each algorithm are shown in Table 4.8, and are given as the average errors throughout the entire run. These are inevitably higher than the estimation errors, as the latter are calculated after the processing of observations. The relative performance of the filters is the same as described for Table 4.7, with the NHF attaining the lowest errors (at the highest computational cost) and the EnPFup achieving the best trade-off between accuracy and computational cost.

[4] The NHF is easily paralellizable using MATLAB's parfor capabilities for parallel computing. In this case, it is done over the parameter space, so that the state space EnKFs run in parallel.

Table 4.8: Prediction errors for the three algorithms. The metrics are the root mean squared error (RMSE) in position and velocity and the error standard deviations included as $\epsilon \pm \sigma_\epsilon$.

Algorithm	Position pred. RMSE (m)	Velocity pred. RMSE (m/s)
EnKFup	110.21 ± 12.47	0.09 ± 0.02
EnPFup	104.34 ± 10.11	0.08 ± 0.03
NHF	102.41 ± 11.29	0.07 ± 0.02

4.7.3. Estimating the parameter with simplified dynamics

In this section, the performance of the proposed filters is evaluated using a simplified dynamical model. To do this, an HF reference orbit is generated deterministically. This reference orbit is also used to produce synthetic radar observations. However, the different filters (EnKFup, EnPFup, NHF) are built around a stochastic LF representation of the orbital dynamics. The goal is to assess whether the filtering algorithms can estimate a suitable noise parameter σ_W that accounts for the difference between the LF model used by the algorithms and the HF model that generates the reference orbit. This is done by computing the RMSEs in position (m) and velocity (m/s) of all three filters. In addition, the computational run-times are assessed to determine the optimal trade-off of accuracy and speed. The number of samples used for the EnKFup, the EnPFup, and the NHF are $N_{\text{EnKFup}} = 450$, $N_{\text{EnPFup}} = 450$ and $N_{\text{NHF}} = 50 \times 50$ (i.e., $N_1 = N_2 = 50$).

Position and velocity errors

In this subsection, the position and velocity RMSEs are shown for the entirety of the simulation period (nine days), to observe the trend in the accuracy of the algorithms as the parameter is simultaneously adjusted. Figure 4.21 shows the RMSEs in position and Figure 4.22 shows the RMSEs in velocity of the three algorithms. Compared to Figures 4.19 and 4.20, the estimation errors display larger fluctuations over time, due to the simultaneous adjustment of the process noise to account for the simplified model being used. A clear adjustment takes place almost immediately after the start of the propagation period. The filter appears to have somewhat converged after two days and remains at low error values for most of the simulation. These figures demonstrate the ability to use LF models and achieve sufficiently low RMSEs when using a filter which estimates σ_W, hence decreasing the computational cost associated with tracking an object online.

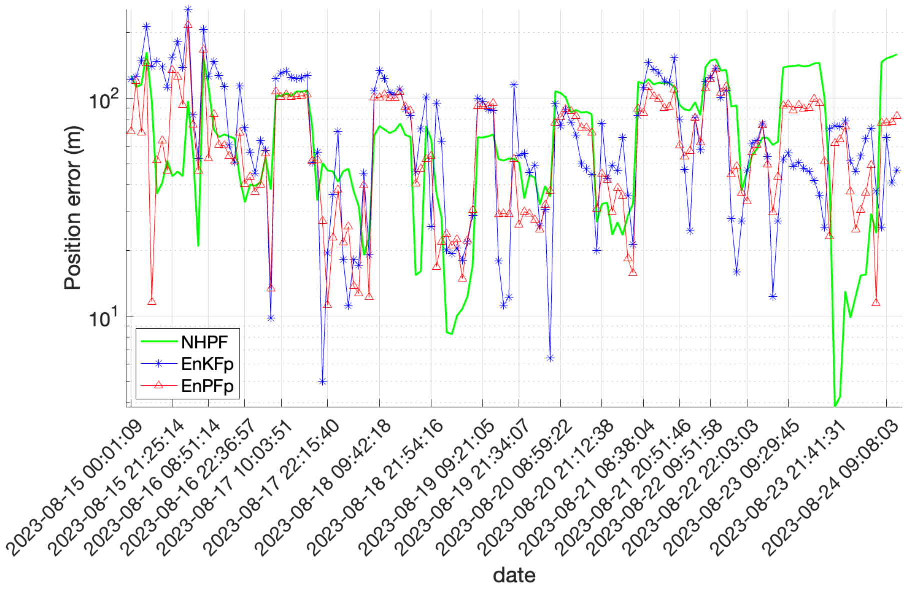


Figure 4.21: RMSEs in position for the three proposed algorithms which adjust the process noise magnitude, σ_W to account for the difference between the HF model used to generate the reference orbit and the LF model used by the filters. The solid blue line shows the EnKFup errors, the red line shows the EnPFup errors, and the green line shows the NHF errors.

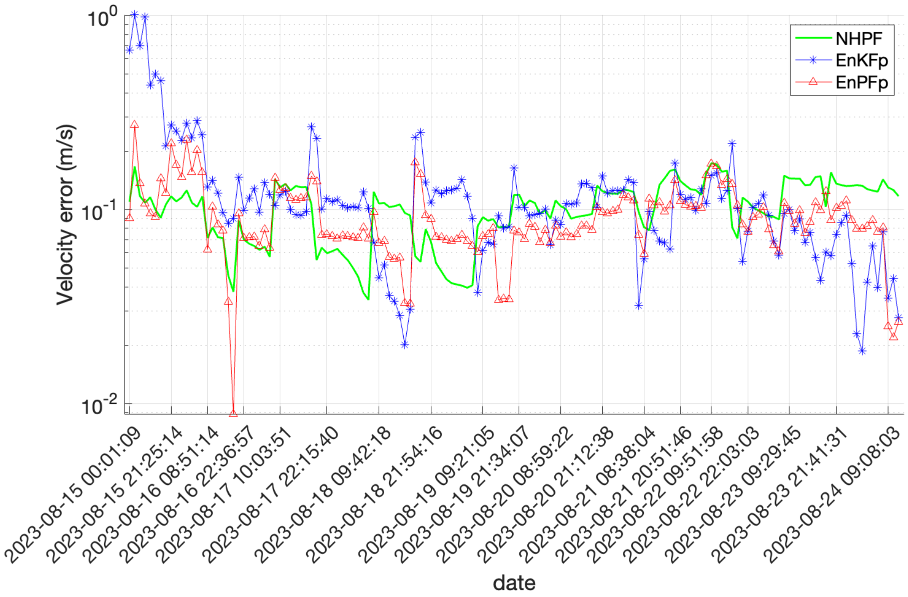


Figure 4.22: RMSEs in velocity for the three proposed algorithms which adjust the process noise magnitude, σ_W, to account for the difference between the HF model used to generate the reference orbit and the LF model used by the filters. The solid blue line shows the EnKFup errors, the red line shows the EnPFup errors, and the green line shows the NHF errors.

Performance comparison

The RMSEs and computational run-time for each algorithm are shown in Table 4.9. Note that the period of simulation is nine days, which is a significant propagation period, and the reason why the run-times are relatively high. The results presented in the table are the averages over 10 simulations of each of the algorithms, showing, in addition, the standard deviations of the position and velocity RMSEs, and the mean run-time. Prediction errors are shown in Table 4.10. They are computed in the same way as for Table 4.8.

Table 4.9: Performance comparison for the three algorithms. The metrics are the RMSE in position and the RMSE in velocity of the estimated trajectory compared to the reference trajectory, the standard deviations of the obtained errors, and the run-time (in minutes).

Algorithm	Position RMSE (m)	Velocity RMSE (m/s)	Run-time (min)
EnKFup	70.71 ± 13.19	0.10 ± 0.05	17.50
EnPFup	61.60 ± 9.29	0.09 ± 0.04	18.21
NHF	69.18 ± 7.81	0.10 ± 0.01	141.36

The NHF is still by far the most computationally costly of the three methods. Both the EnKFup and EnPFup show similar run-times, and therefore, given also the results in Section 4.7.2, the EnPFup attains the best trade-off between computational efficiency and accuracy, as it achieves the lowest errors in position and a run-time one order of magnitude lower than the NHF.

Table 4.10: Prediction errors for the three algorithms. The metrics are the RMSE in position and velocity. The error standard deviations are indicated as $\pm\sigma_\epsilon$.

Algorithm	Position pred. RMSE (m)	Velocity pred. RMSE (m/s)
EnKFup	169.13 ± 21.18	0.14 ± 0.06
EnPFup	159.81 ± 13.33	0.11 ± 0.04
NHF	174.19 ± 10.50	0.14 ± 0.03

4.7.4. Estimating a 3-dimensional parameter

In this section, the noise parameter σ_W of the stochastic LF model consists of three different quantities to be estimated: $\sigma_W(v^{\mathrm{R}})$, $\sigma_W(v^{\mathrm{T}})$ and $\sigma_W(v^{\mathrm{N}})$, i.e., the diffusion coefficients in the radial, transversal, and normal components of the velocity. In this case, the NHF is run with $N_1 = 100$ and $N_2 = 100$ samples. Table 4.11 shows the RMSEs and run-times of the algorithms using this set-up.

From Table 4.11, it can be seen that the estimation performance does not improve compared to Table 4.9. Despite working with more degrees of freedom (d.o.f.) in the parameter space, the fact that more parameter components need to be adjusted causes uncertainty to increase. The NHF is the algorithm with the largest errors, whilst the EnPFup seems to be the most stable of the three. Figure 4.23 shows the estimated RTN components in $\bar{\sigma}_W$ for the EnKFup, EnPFup and NHF.

Table 4.11: Performance comparison for the three algorithms. The metrics are the RMSE in position and the RMSE in velocity of the estimated trajectory compared to the reference trajectory, the standard deviations of the obtained errors, and the run-time.

Algorithm	Position RMSE (m)	Velocity RMSE (m/s)	Run-time (min)
EnKFup	86.90 ± 11.24	0.12 ± 0.2	16.31
EnPFup	73.84 ± 11.22	0.09 ± 0.03	15.19
NHF	111.75 ± 25.18	0.15 ± 0.06	243.51

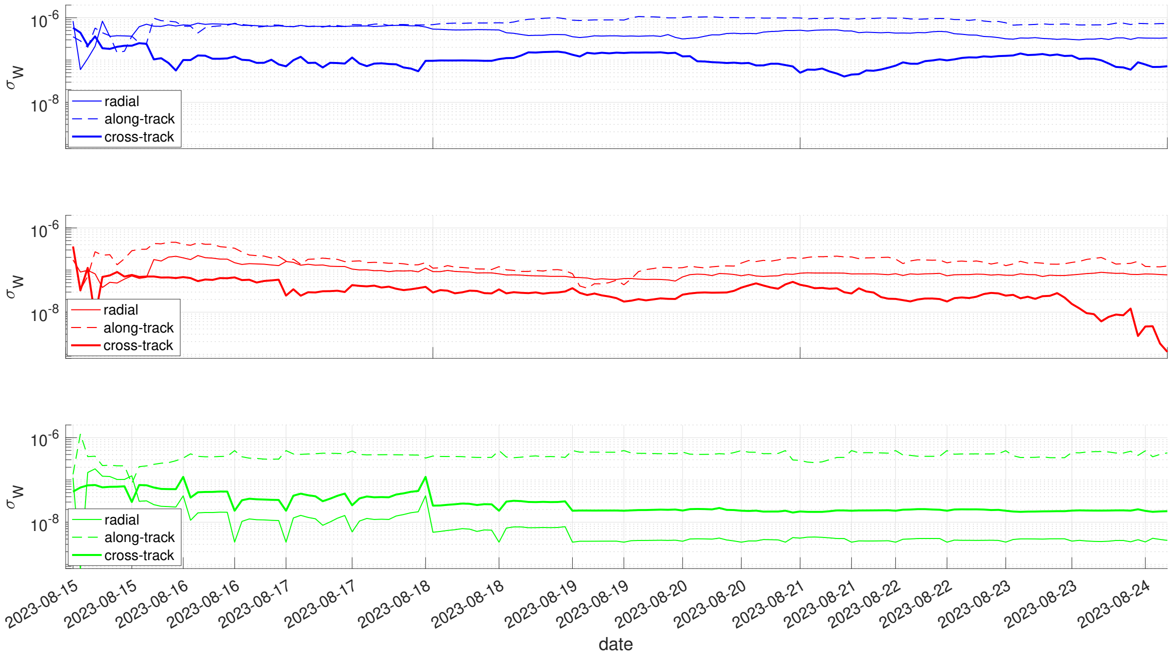


Figure 4.23: In this figure, the $\bar{\sigma}_W$ estimated by the EnKFup (in blue), the EnPFup (in red) and the NHF (in green) algorithms , in the presence of a simplified dynamical model. The thick solid line shows the cross-track component, the thin solid line shows the radial component, and the dashed line shows the along-track component.

From Figure 4.23, it is shown that both the EnKFup and EnPFup estimate the along-track component to be the largest and the cross-track component to be the smallest. The NHF shows agreement on the along-track component, but estimates the radial component to be the smallest. Note that the truncation of certain acceleration terms in the LF dynamical model may induce error in particular directions. LEO trajectories typically show a larger amount of uncertainty in the along-track component during tracking, while the cross-track and radial components typically show lower levels of error. The interchangeability between the latter two, as shown by the algorithms in Figure 4.23 may arise due to variations in the magnitudes of the truncated accelerations within the simplified dynamical model.

4.8 Concluding remarks

In the first part of this chapter, a set of filters is studied to compare their performance as a function of the number of samples and minimum re-observation time in two different scenarios, a LEO and a GEO spacecraft. The filters studied are the CKF, the EnKF, the EnCKF, the EnPF and the BPF. In LEO, the best performing method is the EnPF, showing robustness with regard to the sample number (SN) and minimum re-observation time (MROT), and showing a low computational cost, compared to the BPF. The BPF fails to achieve appropriate RMSEs unless sample number is very large and the CKF diverges when the MROT exceeds the two-minute mark. The results show that purely Gaussian filters are inadequate in LEO when observations are sparse, whereas sample-based filters are able to sufficiently cover the state-space to provide accurate estimations, in addition to being computationally affordable. In GEO, the EnPF performs worse than the EnKF (as opposed to the LEO scenario), and the rest of the algorithms, including the BPF represent appropriate filters to be applied in GEO due to their low RMSEs and relatively low computational costs. In the second part of the chapter, three recursive filters, the EnKFup, the EnPFup, and the NHF are presented. These incorporate notions of PFs and Gaussian filters, to track a LEO spacecraft for a period of ~9 days. By modelling orbital dynamics as an SDE, we are able to simultaneously track the state (position and velocity) of a spacecraft and estimate the diffusion coefficient magnitude in the SDE, i.e., the process noise magnitude.

In order to validate the proposed algorithms, two scenarios are tested. The first uses a set of measurements computed from a HF stochastically propagated reference orbit with a given diffusion coefficient σ_W. To assess whether the algorithms are capable of correctly estimating the parameter, these are run with the same dynamical model with the aim of estimating σ_W. The second scenario uses measurements computed from an HF deterministic reference orbit. In order to assess whether the algorithms are capable of performing well while using a very simplified dynamical model, these are run on a stochastic LF model, and the performance of stochastic parametrisation is assessed, i.e., determining how well the unknown accelerations in the dynamical model are accounted for by estimating the parameter σ_W.

In the first test, the three algorithms achieve low errors in both position and velocity compared to an NBLS method, and can estimate the magnitude of the nominal parameter σ_W accurately, with the NHF achieving the lowest estimation errors in the parameter space, but showing the highest computational cost. The EnPFup is chosen over the EnK-

Fup due to its lower errors in position, but similar run-time. For the second test, all three algorithms achieve low errors in position, and velocity, so in terms of computational cost, the EnPFup is determined to achieve the best trade-off between cost and accuracy out of the three algorithms. The algorithms are therefore able to track a spacecraft in the realistic case where there are significant sources of uncertainty, and/or the model used is simplified in order to cut run-time costs. The result is an ability to “estimate what you do not know" online and appropriately characterise the uncertainty of the system, whilst performing satisfactory tracking of a spacecraft in the presence of observations.

The approaches for OD described in this chapter serve as an adequate starting point for the introduction of the following chapter. Having obtained a sufficiently accurate OD, with appropriate process noise scaling, observation-free propagation can be carried out with a higher degree of reliability. The next chapter revolves around an application where uncertainty characterisation plays a critical role in predicting safe, yet informative re-entry time windows for decaying spacecraft.

5

SPACECRAFT RE-ENTRY PREDICTION

I'm stranded in space, I'm lost without trace
I haven't a chance of getting away
Too close to the Sun, I surely will burn
Like Icarus before me, or so legend goes

Iron Maiden
Satellite 15... The Final Frontier (2010)

This chapter mainly concerns the development of an alternative procedure to compute re-entry time windows. As described in Chapter 1, commonly used methods to assess re-entry events involve propagation assumptions, such as Gaussian state uncertainties, that cease to hold true over long propagation periods, especially in decaying LEO orbits. In addition, remaining spacecraft lifetime (RSL) windows are, in an attempt to preserve safety when predicting probable re-entry times, overly conservative, and set to follow general guidelines. This comes at the expense of accuracy. The goal of the method proposed in this chapter is to bypass limiting assumptions in an efficient way by employing a multifidelity Monte Carlo approach, a CMC variant, aimed at obtaining accurate estimates of the re-entry times, as well as estimators for the upper and lower bounds of the re-entry time, enabling a representation of the likely re-entry time window which is non-necessarily symmetric.

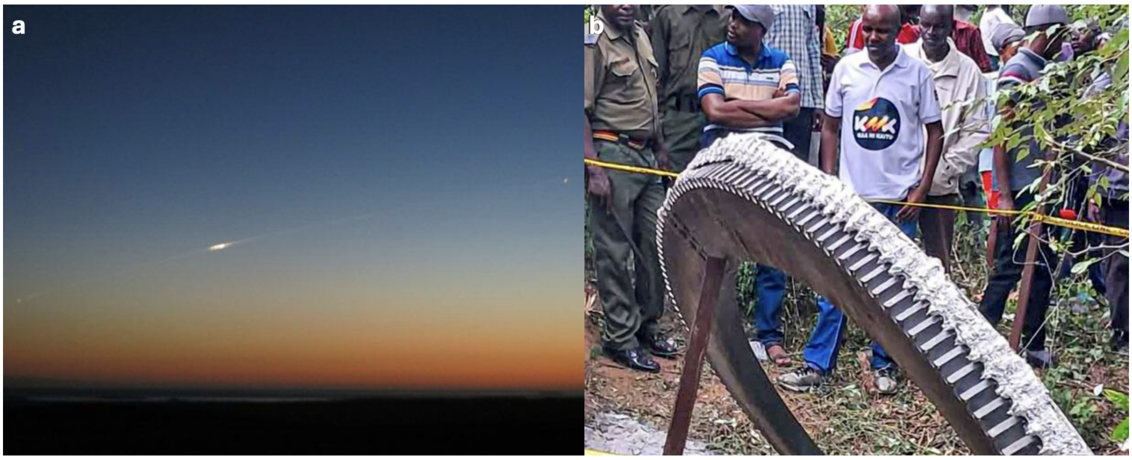


Figure 5.1: a) The moment the Gravity Field and Steady-State Ocean Circulation Explorer (GOCE) was observed over the Falkland islands by locals, on 11 November 2013. b) The piece of space debris that landed in the Kenyan village of Mukuku, on 12 January 2025.

5.1 State of the art

5.1.1. Background

RSOs often remain in orbit long after their operational lifetimes have ended. One of the key consequences of a highly saturated near-Earth space environment is a higher proportion of these vagrant objects. These defunct objects face one of two outcomes: they either persist in orbit, posing a long-term collision risk to active spacecraft, or eventually undergo uncontrolled re-entry into Earth's atmosphere. Many of these re-entering objects disintegrate upon contact with the lower layers of the Earth's atmosphere, but larger bodies can traverse the lower layers of the atmosphere and put populated areas of the world in danger. Controlled re-entries refer to the active de-orbiting of large spacecraft or space stations, to ensure that their decay occurs over largely uninhabited areas. Uncontrolled re-entries are those where the final location cannot be controlled but predicted. During uncontrolled events, although catastrophe may be rare due to the low probability of impact on populated areas, it is possible for uncontrolled objects to cause major disruption in air traffic management and communications. In recent times, the world has not been short of examples of this phenomenon. For example, citizens of the Kenyan village Mukuku witnessed in 2025 the impact of a 2.5 m tall, 500 kg space debris rocket separation ring (see Figure 5.1b). Although this type of structure is designed to burn up upon re-entry, sometimes this is not the case. No villagers were injured, but damage to nearby properties has been reported [127]. When properly tracked and controlled, decaying objects usually fall over empty areas, such as the Pacific ocean (a common choice for such events). As an example, the gravity field and steady-state ocean circulation explorer (GOCE) mission culminated in an uncontrolled, though adequately tracked re-entry, and hence was determined to pose no significant threat to ground populations. It was seen re-entering over the Falkand islands on 11 November 2013 [128]. Other times, accurate tracking information is unavailable or undisclosed, and there can be significant uncertainty with regard to the time or location of the expected impact, as was the case with the Chinese Long March-5B rocket thruster which underwent uncontrolled re-entry on 31 October 2022 over the Pacific ocean. However, its descent trajectory flew over much of Europe and caused general unease to authorities in several countries, even forcing the complete shutdown of the Spanish airspace over specific locations in fear of debris impact [129].

In 2018, the authors of [130] suggested that the probability that a person is struck by falling debris in their lifetime is approximately 10^{-12}, while that of being hit in a car accident is approximately 10^{-2} in industrialised countries. Since this observation almost eight years ago, the number of satellites has almost doubled, with over 38,000 RSOs, of which only about a third are operational spacecraft [3]. The remainder constitutes unusable and likely uncontrollable space debris and defunct rocket bodies, of which over 200-400 are estimated to re-enter annually [3].

5.1.2. Dynamical modelling standards

Spacecraft in LEO are subject to atmospheric drag forces, resulting in constant reductions to the orbit's semi-major axis, provided no additional propulsion or control manoeuvre is performed. Unless the effect of drag is accurately modelled at critical times throughout atmospheric re-entry, empirical density models may induce significant uncertainty in the prediction of orbital objects' trajectories in LEO. Models often assume a relatively constant mass and cross-sectional area down to an altitude of 80 km [131], whereas the drag coefficient C_D varies significantly as the spacecraft ventures toward lower altitudes. The estimation of C_D is usually carried out by fitting arcs of varying length depending on the proximity to re-entry altitudes. Standard methods that provide high-quality estimates of re-entry trajectories rely on accurate aerodynamic modelling [132]. Instead of modelling a point mass, methods such as SCARAB [133] or DRAMA's SARA package [134] compute changes in spacecraft attitude, orientation (endowing them with 6 d.o.f), dynamical motion, and aerothermal dynamics. However, these tools incur higher computational costs and implementation complexity due to the precise modelling of spacecraft structures and their interaction with the space environment [132]. Ref. [128] presents statistical reentry predictions of the GOCE mission with an improved version of the 6 d.o.f. model used in [135], by modelling atmospheric density, and orientation, as well as testing different dynamical configurations to determine which modelling setup achieves the most accurate predictions.

Many propagation schemes involve Gaussian representations of the state uncertainty; that is, they assume that the state is modelled with an r.v. that can be completely described by a mean and a covariance matrix. However, since the equations of motion in orbital dynamics are highly nonlinear, it has been shown that in ECI coordinates the Gaussian uncertainty assumption breaks down after a long-enough propagation period [136], [137]. This is especially true for decaying objects, as the exponential growth of drag forces as the object enters denser atmospheric layers deforms the distribution in non-straightforward ways.

5.1.3. Re-entry time window predictions

In order to ensure a safe space environment, predicting the time and possible location of re-entry events is crucial. This includes analysing scenarios where data may be limited or non-existent, such as the final time window of a decaying spacecraft. Therefore, accurate orbit propagation models and efficient estimation frameworks become pivotal. For instance, the Russian space station MIR had a predicted area of impact of 2500 km^2 ± 100 km^2 upon re-entry, which highlighted the need for improved approaches to compute impact time and location predictions [138]. ESA has set standards for methodologies dedicated to short, medium and long-term prediction of uncontrolled events [131]. For longer-term predictions, windows are necessarily larger than for short-term predictions, which may be spaced only hours from a re-entry event. The literature suggests that semi-analytical propagation schemes should be used for long-term predictions, and numerical propagation for short-term predictions, at altitudes below 200 km [130].

Due to the expected level of uncertainty, instead of precise re-entry date estimates, reentry windows as a function of RSL are usually computed. Traditionally, standard RSL windows assume a static symmetric uncertainty of 20%, as is the case for ESA's recommendations [131]. This covers the majority of cases, but for 5% of re-entry predictions, the true decay time has been shown to lie outside the predicted RSL bounds. There has generally been limited research validating such assumptions [128]. Ref. [130], one of the main statistical works of research in the topic suggests that an uncertainty window able to guarantee a confidence level of 90% should generally adopt an amplitude of about ±20% RSL around the estimated nominal re-entry time, to be raised to about ±25% in the couple of days ahead of re-entry. According to the authors of [139], based on atmospheric model uncertainty alone, RSL errors are generally smaller than 14% for long-term (one month) predictions. If solar and geomagnetic uncertainties are added, windows up to 25% cover more than 90% of cases.

5.1.4. Monte Carlo-based methods

CMC is often used as a benchmark against which to compare newly proposed re-entry prediction methodologies. CMC avoids the need to make assumptions about the distribution of the target state [128][135], and can therefore be statistically reliable. The drawback is that, for the desired level of accuracy, it is a rather inefficient method, as trajectories are drawn from the initial distribution at random and sufficient coverage of the areas of interest in the state space incurs a potentially prohibitive computational cost.

Works such as [140] introduce approaches which leverage the use of expensive, but accurate HF models, as well as cheap LF models. The authors propose a bi-fidelity variation of IS for efficient uncertainty quantification, which is able to provide an alternative to CMC at a reduced cost. The multifidelity Monte Carlo (MFMC) scheme of [141] aims to achieve the same level of accuracy as the CMC method with a HF propagator at a lower computational cost, i.e., to produce estimates with reduced variance, by optimally combining estimators of different fidelities. Other MFMC adaptations have been proposed, such as Ref. [142], which builds on the bi-fidelity method to develop a mixture multifidelity variation of IS for rare event analysis, optimally combining an arbitrary number of surrogate models. For an in-depth review of multifidelity approaches to uncertainty quantification and optimisation problems, see [143]. The MFMC has been used in a variety of contexts outside astrodynamics, including works such as [144], which uses MFMC to improve the efficiency of uncertainty quantification in climate-related models, Ref. [145], which employs MFMC to address the computational challenges of performing uncertainty quantification in reservoir management, and Ref. [146], in the assessment of uncertainties in spent nuclear fuel characterization.

The core idea of MFMC is to optimally combine, for a given QoI, the estimator outputs of models of varying fidelity, where the set of models must include the HF model. This leverages the low computation costs of LF models as well as the accuracy of HF models in an optimal way by using control variates, which allow us to determine the optimal number of evaluations required for each model a priori, given a fixed computational budget.

5.2 Test cases

In this section, a brief description of the initial conditions, physical characteristics and data available are provided for two LEO scenarios featuring decaying spacecraft, as well as an outline of the simulation configuration for this work.

Roseycubesat-1

Roseycubesat-1 was an educational spacecraft launched onboard SpaceX's Falcon 9 in 2023, aimed at taking simple photographs, providing telemetry, and using amateur radio equipment for transmission [147]. The time at which the spacecraft was last observed was $t_c \equiv$ 31 August 2024, 06:11:00 UTC [148][149]. The state vector of the spacecraft at $t_0 \equiv$ 22 August 2024, 21:49:00 UTC (approximately eight days prior to decay), containing the position and velocity expressed in ECI coordinates was

$$\boldsymbol{x}_0 = \ [-5208.06\text{km},\ 4172.09\text{km},\ 12.29\text{km},\ 0.64\text{km/s},\ 0.76\text{km/s},\ 7.66\text{km/s}]^\top.$$

The position and velocity components of the initial covariance matrix are empirically estimated from multiple sets of two-line elements (TLEs) —a standard approach for covariance estimation when no other sources are available [150]— and are given by

$$\boldsymbol{\Sigma}_0 = \begin{array}{l} \left[\begin{array}{cccccc} 0.0918 & 0.0022 & 0.0220 & 0 & 0 & 0 \\ 0.0022 & 0.0927 & 0.0267 & 0 & 0 & 0 \\ 0.0220 & 0.0267 & 0.3555 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9.1825\times10^{-8} & 2.2138\times10^{-9} & 2.2014\times10^{-8} \\ 0 & 0 & 0 & 2.2138\times10^{-9} & 9.2685\times10^{-8} & 2.6698\times10^{-8} \\ 0 & 0 & 0 & 2.2014\times10^{-8} & 2.6698\times10^{-8} & 3.5549\times10^{-7} \end{array}\right] \end{array} \begin{array}{l} \mathrm{km}^2 \\ \mathrm{km}^2 \\ \mathrm{km}^2 \\ \mathrm{km}^2/\mathrm{s}^2 \\ \mathrm{km}^2/\mathrm{s}^2 \\ \mathrm{km}^2/\mathrm{s}^2. \end{array}$$

The assumed physical properties of the satellite are detailed in Table 5.1.

Table 5.1: Physical properties of the decaying Roseycubesat-1 spacecraft

	Roseycubesat-1
Mass (kg)	1
$\mathbb{A}$ (m^2)	0.01
C_D (-)	1.6

Furthermore, the stochastic diffusion coefficient $\boldsymbol{\sigma}_W$ introduced in Section 2.5 is assumed to be constant, with a magnitude proportional to the value of the acceleration output at which the HF dynamical model is truncated. The three velocity components of $\boldsymbol{\sigma}_W$ are given as $\sigma_W(v_x) = \sigma_W(v_y) = \sigma_W(v_z) = 10^{-7}$ km^2/s^2.

GOCE

GOCE was a scientific mission launched by ESA in 2009, and was designed to carry out measurements of the Earth's gravitational field with extreme accuracy. GOCE entered a decaying trajectory on 11 November 2013, 00:00:00 UTC. The state vector of the spacecraft at $t_0 \equiv$ 20 October 2013, 00:00:00 UTC (approximately 21 days prior to decay), containing the position and velocity expressed in ECI coordinates are

$$\boldsymbol{x}_0 = [-817.38\mathrm{km},\ -11.09\mathrm{km},\ 6547.01\mathrm{km},\ -3.0830\mathrm{km/s},\ 7.1220\mathrm{km/s},\ -0.3669\mathrm{km/s}]^\top.$$

The position and velocity components of the initial covariance matrix are taken as conservative values commonly assumed in the low Earth orbit domain, and are given by

$$\boldsymbol{\Sigma}_0 = \left[\begin{array}{cccccc} 0.0100 & 0.0005 & -0.0002 & 7.2000\times10^{-7} & 1.3500\times10^{-6} & 2.5000\times10^{-7} \\ 0.0005 & 0.0096 & -0.0004 & 1.1000\times10^{-6} & 6.7700\times10^{-6} & 6.3800\times10^{-6} \\ -0.0002 & -0.0004 & 0.0104 & -3.8000\times10^{-7} & 6.6400\times10^{-6} & -7.6500\times10^{-6} \\ 7.2000\times10^{-7} & 1.1000\times10^{-6} & -3.8000\times10^{-7} & 1.0000\times10^{-8} & 0 & 0 \\ 1.3500\times10^{-6} & 6.7700\times10^{-6} & 6.6400\times10^{-6} & 0 & 1.0000\times10^{-8} & 0 \\ 2.5000\times10^{-7} & 6.3800\times10^{-6} & -7.6500\times10^{-6} & 0 & 0 & 1.0000\times10^{-8} \end{array}\right] \begin{array}{l} \mathrm{km}^2 \\ \mathrm{km}^2 \\ \mathrm{km}^2 \\ \mathrm{km}^2/\mathrm{s}^2 \\ \mathrm{km}^2/\mathrm{s}^2 \\ \mathrm{km}^2/\mathrm{s}^2. \end{array}$$

The assumed physical properties of the satellite are detailed in Table 5.2. The attitude control system maintained the spacecraft in a minimum drag configuration during the re-entry portion, meaning that the re-entry was equivalent to that of an object with a constant area-to-mass ratio, contradicting original expectations [151].

Table 5.2: Physical properties of the decaying GOCE spacecraft. Although launch mass was 1050 kg, over 200 kg of fuel were used up throughout mission. $\mathbb{A}$ is the cross-sectional area presented in the afore-mentioned minimum drag configuration.

	GOCE
Mass (kg)	850
$\mathbb{A}$ (m^2)	1.1
C_D (-)	2.5

The stochastic diffusion coefficient $\boldsymbol{\sigma}_W$ introduced in Section 2.5 is assumed to be constant, with a magnitude proportional to the value of the acceleration output at which the HF dynamical model is truncated. The three velocity components of $\boldsymbol{\sigma}_W$ for the stochastic propagation are given as $\sigma_W(v_x) = \sigma_W(v_y) = \sigma_W(v_z) = 10^{-8}$ km^2/s^2.

5.3 Methodology

5.3.1. State-space definition

Orbital dynamics are commonly modelled as a deterministic system of equations of motion. However, in order to appropriately represent the uncertainty that arises due to model mismatch with reality, or random numerical error accrued during propagation, we model the propagation of the state by means of SDEs. Assume a Gaussian prior for the spacecraft states at t_0, $\pi(\boldsymbol{x}(t_0)) \equiv \mathcal{N}(\hat{\boldsymbol{x}}(t_0), \boldsymbol{\Sigma}_0)$, given by the mean $\hat{\boldsymbol{x}}(t_0) = [\hat{\boldsymbol{r}}(t_0)^\top, \hat{\boldsymbol{v}}(t_0)^\top]^\top$ and covariance matrix $\boldsymbol{\Sigma}_0$; that is, assuming that a previous OD procedure has already been performed. The governing SDE is then given by Eq. (2.13), where $f(\boldsymbol{x}, t)$ represents the deterministic drift of the stochastic state $\boldsymbol{x}(t)$. A noise term is also incorporated, driven by a d-dimensional Wiener process $\boldsymbol{W}(t)$, and scaled by the 6×6 diffusion coefficient $\boldsymbol{\sigma}_W$ matrix, which represents the magnitude of the assumed uncertainty. This stochastic parametrisation of unknown model accelerations captures the uncertainty in the dynamical model due to unmodelled dynamics or errors [25] and provides a potentially more realistic characterisation of spacecraft motion under uncertainty. The diffusion coefficient is assumed to act in the three velocity components of the state and it is given by

$$\boldsymbol{\sigma}_W(\boldsymbol{x}) = \operatorname{diag}\Big(0,\ 0,\ 0,\ \sigma_W(v_x),\ \sigma_W(v_y),\ \sigma_W(v_z)\Big) \in \mathbb{R}^{6\times 6}.$$

For the rest of this work, we assume that $\boldsymbol{\sigma}_W(\boldsymbol{x}) = \boldsymbol{\sigma}_W$ is independent of $\boldsymbol{x}$.

To integrate the SDE (2.13) we adopt the stochastic Runge-Kutta scheme introduced in Section 2.4.2. Assume a time grid $\boldsymbol{T}' = \{t'_l\}_{l=0}^{L}$, where t'_0 is the initial time and $t'_1, \ldots, t'_L$ are time instants at which the value of the state $\boldsymbol{x}(t'_l)$ must be approximated, i.e., $\boldsymbol{x}_l \approx \boldsymbol{x}(t'_l)$. Assume also that we have $n = 1, \ldots, K$ different dynamical models, possibly of different fidelities. We define the nth propagation function as

$$\begin{aligned} \Phi_l^n &: \mathbb{R}^6 \mapsto \mathbb{R}^6, \\ \boldsymbol{x}_{l-1} &\rightsquigarrow \boldsymbol{x}_l = \Phi_l^n(\boldsymbol{x}_{l-1}); \end{aligned} \tag{5.1}$$

which consists of the numerical integration scheme of the nth dynamical model and is applied to the states at time t'_{l-1} to produce states at time t'_l. We denote with $\Phi_l^1(\cdot)$ the HF model, and with $\Phi_l^2, \ldots, \Phi_l^K$ the LF models.

5.3.2. Re-entry event definition

To estimate a spacecraft's re-entry time, we assess its altitude as a function of time, by propagating its initial state $\boldsymbol{x}(t'_0)$ to some later time t'_L using the propagation functions $\Phi_1^n(\cdot), \ldots, \Phi_L^n(\cdot)$ in Eq. (2.12). We compute altitudes at each integration time-step $l = 0, \ldots, L$, i.e., $\{\mathcal{A}^n(t'_0), \ldots, \mathcal{A}^n(t'_L)\}$, using the nth model as

$$\mathcal{A}^n(t'_l) = \|\boldsymbol{r}^n(t'_l)\| - R_E, \tag{5.2}$$

where $\|\boldsymbol{r}^n(t'_l)\|$ is the magnitude of the position vector of the spacecraft, i.e., the radial distance between the spacecraft and the centre of the Earth, computed via de nth model, and R_E is the Earth's radius.

The spacecraft state is propagated until it has decayed or the simulation time span t'_L has been reached. By defining β as a suitable altitude threshold (in km), a re-entry event is detected at time t'_l during propagation when $\mathcal{A}^n(t'_l) < \beta$, point at which the propagation is halted, and the re-entry time T_β^n is recorded as

$$T_\beta^n = \begin{cases} t'_l, & \text{if } \mathcal{A}^n(t'_l) < \beta \\ t'_L, & \text{otherwise} \end{cases}. \tag{5.3}$$

The function for the nth model which computes the re-entry time T_β^n as a function of the propagated spacecraft state is given by

$$\begin{aligned} \tau^n(\cdot) &: \mathbb{R}^6 \to \mathbb{R}, \\ \boldsymbol{x}_l &\rightsquigarrow \tau^n(\boldsymbol{x}_l, t'_l) := T_\beta^n. \end{aligned} \tag{5.4}$$

If many samples are used, a window is typically defined by a lower bound and an upper bound of re-entry times, which are given by the decay time of the first and last samples, respectively. For this purpose, a CMC method can be used.

5.3.3. High-fidelity crude Monte Carlo

In the context of computing the estimated re-entry time of a spacecraft, the CMC method consists of generating a collection of sample spacecraft trajectories and inferring relevant information from their probability distribution. The procedure consists of the following four steps:

1. Draw N independent samples $\{\boldsymbol{x}_0^i\}_{i=1}^N$ from the prior distribution π_0.
2. Select a fine time grid $0 = t'_0 \leq t'_1 \leq \ldots \leq t'_L$ and propagate each sample using a high-fidelity (HF) numerical scheme via Eq. (2.12), yielding $\boldsymbol{x}_l^i = \Phi_l^1(\boldsymbol{x}_{l-1}^i),\ l = 1, \ldots, L$.
3. Compute the altitude of each propagated sample at time t'_l using Eq. (5.2), resulting in values $\mathcal{A}^{i,1}(t'_l)$ for $i = 1, \ldots, N$.
4. Apply Eq. (5.3) to each sample to compute the re-entry time $T_\beta^{i,n}$. This results in a CMC approximation of the re-entry time distribution, $\{T_\beta^{i,n}\}_{i=1}^N$, from which QoIs can be estimated—such as the minimum, maximum, quantiles, or mean re-entry time under the HF model ($n = 1$) —via the CMC estimator

$$\hat{s}_N^n = \frac{1}{N} \sum_{i=1}^{N} \tau^n(\boldsymbol{x}_L^i, t'_L) = \frac{1}{N} \sum_{i=1}^{N} T_\beta^{i,n}. \tag{5.5}$$

CMC provides an unbiased estimator of the QoI using the HF model and is a method that does not make simplifying assumptions about a spacecraft's state probability distributions. However, in order to achieve accurate and statistically significant results, a large number of samples must be used, which can incur a high computational cost. This makes CMC a costly alternative to compute QoIs.

5.3.4. Stochastic re-entry analysis

The use of deterministic propagation models alone is equivalent to assuming that the only noise present in the system is due to the uncertainty in the initial conditions. Randomness, however, as reiterated throughout the previous chapters, is always present, whether in the form of aleatory error, or systemic error. When computing RSL windows of decaying spacecraft, it is important to understand that the dynamical model can never fully capture reality to its full extent; the expected spacecraft states and window location of the re-entry time are not exempt from this uncertainty. Employing stochastic propagation ensures that we are modelling the system as one with an element of randomness, giving us some informed leeway in the prediction of re-entry windows and spacecraft location throughout propagation.

In this subsection, CMC simulations are performed for the GOCE scenario described in Section 5.2, with sample size $N = 1,000$. Re-entry time histograms are shown for the comparison between deterministic and stochastic propagation at different diffusion scales. This is done to determine the effect of varying the magnitudes of the diffusion coefficient on the re-entry window size.

The diffusion coefficient is assumed to act in the three velocity components of the state. The coefficient is also assumed to be independent of the state and is given by $\sigma_W = \sigma_W(v_x) = \sigma_W(v_y) = \sigma_W(v_z)$. For the test, R different $\boldsymbol{\sigma}_W$ candidate matrices are used, characterised by R coefficients $\sigma_W^1, \sigma_W^2, \ldots, \sigma_W^R$. These are then inputted into Eq. (2.13) to run independent simulations, resulting in R RSL window predictions. The $\boldsymbol{\sigma}_W$ components used in this experiment are given in the set

$$\sigma_W \in \left\{ \begin{array}{llll} 1\times10^{-13}, & 1\times10^{-12}, & 1\times10^{-11}, & 1\times10^{-10}, \\ 1\times10^{-9}, & 1\times10^{-8}, & 5\times10^{-8}, & 1\times10^{-7}, \\ 2.5\times10^{-7}, & 5\times10^{-7}, & 7.5\times10^{-7}, & 1\times10^{-6}, \\ 2.5\times10^{-6}, & 5\times10^{-6}, & 7.5\times10^{-6}, & 1\times10^{-5}, \\ 2.5\times10^{-5} & & & \end{array} \right\}. \tag{5.6}$$

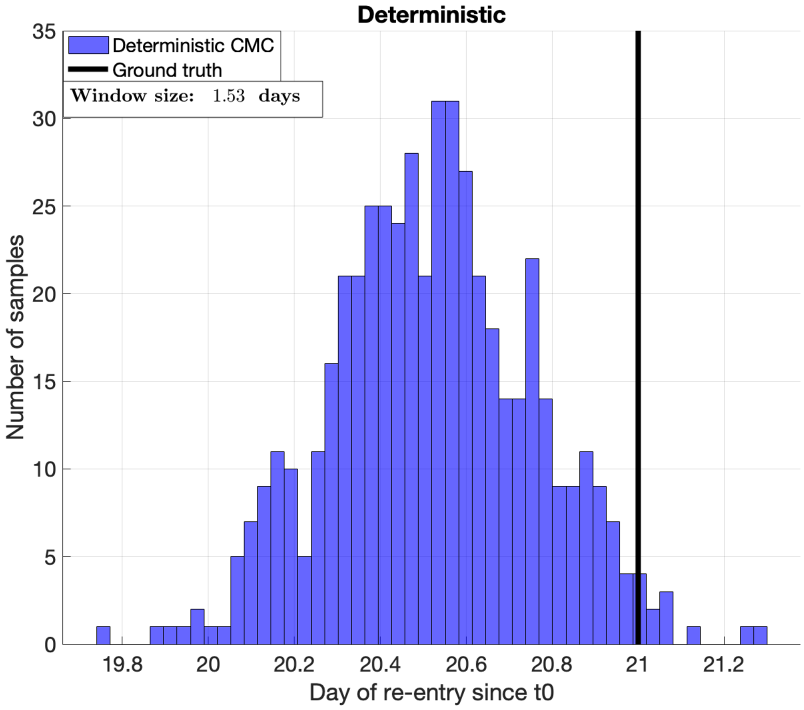


Figure 5.2: Deterministic histogram of re-entry times for the GOCE scenario (in blue), and the real re-entry time (shown as a thick black line).

Figure 5.2 shows the deterministic benchmark histogram for the GOCE reentry scenario, attaining a baseline window size of 1.53 days and overlapping with the real re-entry time at the right tail. The spread in re-entry times in this case is exclusively attributed to the scale of the state's initial covariance matrix.

Figure 5.3 shows the subset of the stochastic histograms associated with visible window size changes. Values of the diffusion coefficient higher than those shown exhibit windows that are too large (and hence prove uninformative) and exceed the simulation time-span.

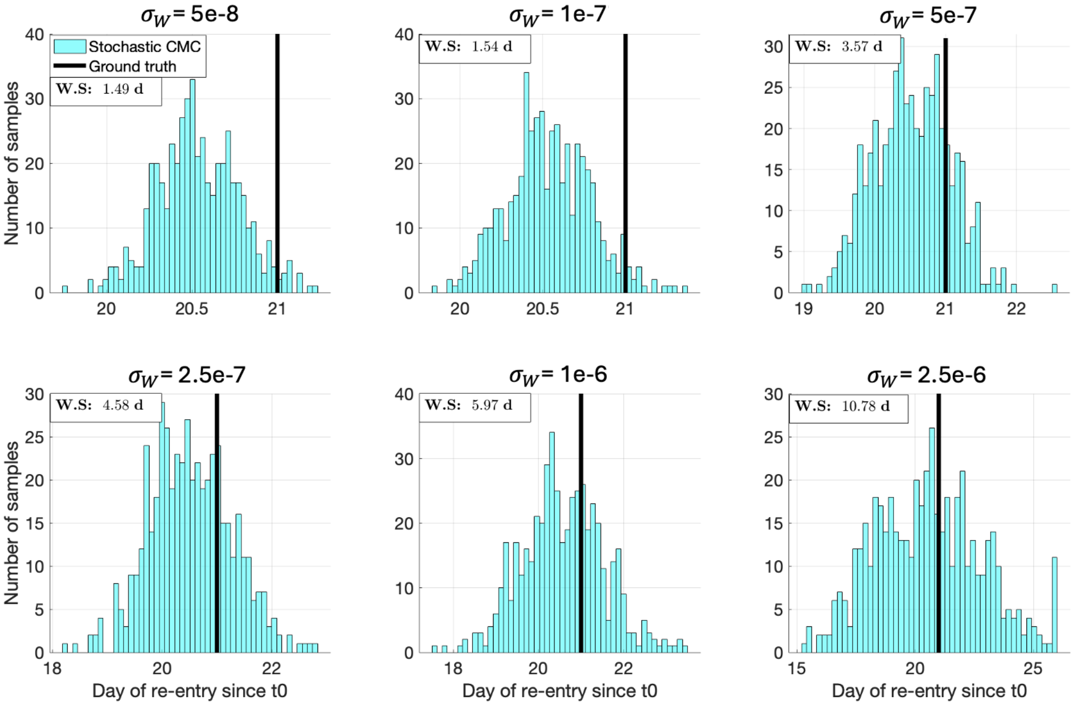


Figure 5.3: A set of histograms showing the different stochastic diffusion coefficient scales and their corresponding window sizes.

From left to right, it can be seen that increasing the scale of the diffusion coefficient in this range leads to significant increases in window size. From a safety standpoint, 21 days out from the predicted re-entry, the literature suggests conservative approaches with regard to window size, leading us to believe that a σ_W value of 1×10^{-6} may satisfy safety needs, though precision may be sacrificed.

The extent of variation in window size can also be analysed in terms of correlation with the σ_W values, as shown in Figure 5.4. The graph shows that increasing the diffusion scale from deterministic to a value of $\sim 1 \times 10^{-7}$ does not bring about any substantial changes in window size, but the change becomes almost exponential on approach to 1×10^{-6}. If we arbitrarily inflate the diffusion coefficient (perhaps in an unjustified manner) for the sake of safety when predicting re-entry times, we are simply assuming an arbitrarily enlarged window, akin to the 20% RSL window commonly used in the industry. The goal, however, is to use a diffusion scale which is consistent with the truncation of the dynamical model. In addition, we aim to estimate the re-entry window in an informed way, instead of providing overly enlarged values. The next section deals with a CMC variant which is able to estimate a mean re-entry time prediction, but also a properly sized RSL window.

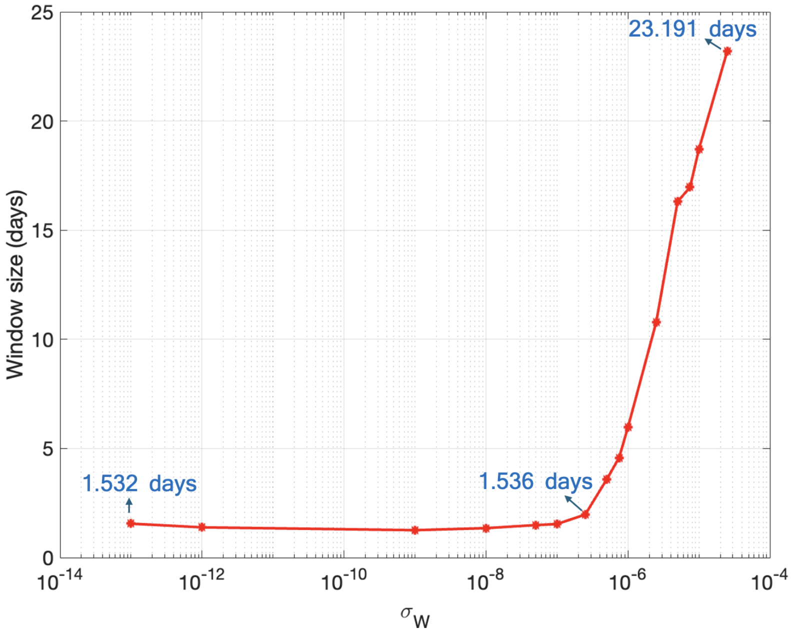


Figure 5.4: The variation of window sizes (in days) with respect to σ_W. The starting point, the ending point, and the point at which window size begins to change significantly are singled out.

5.3.5. Multifidelity Monte Carlo

Multifidelity approaches to Monte Carlo estimation strike a balance between computational cost and accuracy by employing a hierarchy of models. This is done by exploiting lower-fidelity models to improve the accuracy of a HF estimator with a given computational budget. The multifidelity Monte Carlo (MFMC) scheme of [141], which we implement in this work, solves an optimisation problem to efficiently distribute computational resources to each of the models such that the mean squared error of the estimator is minimised for a given computational budget. The MFMC method calculates the optimal number of model evaluations that can be computed for each surrogate by employing control variates. MFMC is applicable to an arbitrary number of surrogate models, obtained in a variety of different ways and is able to achieve an unbiased estimator of the QoI (in this case, the re-entry time) due to occasional evaluations of the HF model.

Suppose that we have access to a HF propagation model Φ^1 as defined in Eq. (2.12), and to $K-1$ lower-fidelity models, denoted $\Phi^2, \dots, \Phi^K : \mathbb{R}^6 \to \mathbb{R}^6$ with $K \in \mathbb{N}$ and endowed with different computational costs. The real re-entry time as a function of the state $\boldsymbol{x}_l$ at time t'_l is given by $s = \tau(\boldsymbol{x}_l, t'_l)$. The CMC estimator is the unbiased estimator of $\mathbb{E}[\tau(\boldsymbol{x}_l, t'_l)]$, where $\tau(\cdot, t')$ is defined in Eq. (5.4). A CMC estimate can be computed for the nth model using the procedure of Section 5.3.3.

To compute the MFMC estimator, let $\boldsymbol{m} = [m_1, \dots, m_K]^\top \in \mathbb{N}^K$ denote the vector specifying the number of samples used for each of the K models, ordered such that $0 < m_1 \leq m_2 \leq \cdots \leq m_K$. Let $\{\boldsymbol{x}_0^i\}_{i=1}^{m_K} \subset \mathcal{D}$ be m_K i.i.d. samples drawn from the initial distribution of the state variable $\boldsymbol{x}_0$. For each model $n = 1, \ldots, K$, define the sample set

$$\mathbb{X}_n := \{\boldsymbol{x}_0^i\}_{i=1}^{m_n},$$

so that the samples are *shared* across models, satisfying the nesting structure

$$\mathbb{X}_1 \subset \mathbb{X}_2 \subset \cdots \subset \mathbb{X}_K,$$

with $\mathbb{X}_n \subset \mathbb{X}_{n+1}$ and $\mathbb{X}_n \cap \mathbb{X}_{n+1} = \{\boldsymbol{x}^i\}_{i=1}^{m_n}$.

Then, we compute the estimate $\hat{s}_{m_n}^n$ as in Eq. (5.5) with m_n evaluations, and also compute $\hat{s}_{m_{n-1}}^n$ with m_{n-1} evaluations. Since we reuse the m_{n-1} samples used by the previous model, the two estimators are dependent. The MFMC estimate of s is then

$$\hat{s}_{\text{MFMC}} := \hat{s}_{m_1}^1 + \sum_{n=2}^{K} \left(\hat{s}_{m_n}^n - \hat{s}_{m_{n-1}}^n\right) \boldsymbol{Q}^n \approx \mathbb{E}[\tau^1(\boldsymbol{x}_l, t_L')], \tag{5.7}$$

where $\boldsymbol{Q}^2, \ldots \boldsymbol{Q}^K \in \mathbb{R}$ are coefficients which weigh the differences $\hat{s}_{m_n}^n - \hat{s}_{m_{n-1}}^n$.

The goal of the MFMC is to use Eq. (5.7) by optimally exploiting surrogate model evaluations $\boldsymbol{T}_\beta^2, \ldots, \boldsymbol{T}_\beta^K$ to achieve higher accuracy than CMC at lower cost. These surrogate models are correlated. In general, distributing work across multiple surrogate models is challenging due to a lack of accurate correlation information of the models relative to the HF model. Hence, an optimisation problem is posed to balance these correlations and the computational costs of the models, given a fixed computational budget B_c, to compute the optimal number of model evaluations for each surrogate model.

The costs to simulate these models are captured in the vector $\boldsymbol{c} = [c_1, \ldots, c_K]^\top \in \mathbb{R}_+^K$, containing the averaged computational cost of propagating one sample with the relevant dynamical model. Given a computational budget B_c, we minimise the variance of the MFMC estimator, which is the cost function. To do this, we simulate N samples from the prior $\boldsymbol{x}_0^i \sim \pi_0$ and pass them through propagation functions $\{\Phi_{1:L}^n\}_{n=1}^K$, where $\Phi_{1:L}^n = \Phi_L^n \circ \Phi_{L-1}^n \circ \ldots, \circ \Phi_1^n$. The altitudes $\{\mathcal{A}^{i,n}\}_{i=1}^N$, for all $n = 1, \ldots, K$ models, and for all samples can now be computed. With these K altitude vectors, the correlation vectors, and the cost vectors, we minimise the variance of $\hat{s}_{\text{MFMC}}$ [152], given by

$$\text{Var}[\hat{s}_{\text{MFMC}}] = \frac{(\sigma^1)^2}{m_1} + \sum_{n=2}^{K} \left(\frac{1}{m_{n-1}} - \frac{1}{m_n}\right) (\, (\boldsymbol{Q}^n)^2 (\sigma^n)^2 - 2\boldsymbol{Q}^n \rho^n \sigma^1 \sigma^n \,), \tag{5.8}$$

where σ^n is the standard deviation of the samples $\{\boldsymbol{x}_L^{i,n}\}_{i=1}^N$, and ρ^n is the Pearson correlation coefficient, which measures the linear correlation between sample altitudes of the nth surrogate model $\{\mathcal{A}^{i,n}\}_{i=1}^N$ and the sample altitudes of the HF model, $\{\mathcal{A}^{i,1}\}_{i=1}^N$.
Let

$$\mathcal{B}(\boldsymbol{m}, \boldsymbol{Q}^2, \ldots, \boldsymbol{Q}^K) := \text{Var}[\hat{s}_{\text{MFMC}}], \tag{5.9}$$

and let $\boldsymbol{m}^* \in \mathbb{R}^K$ and $Q^{2*}, \dots, Q^{K*} \in \mathbb{R}^K$ be the solution to the optimisation problem

$$\begin{aligned} \{Q^{2*}, \dots, Q^{K*}, \boldsymbol{m}^*\} = \underset{\boldsymbol{m} \in \mathbb{R}^K,\ Q^2, \dots, Q^K \in \mathbb{R}}{\arg\min} \quad & \mathcal{B}(\boldsymbol{m}, Q^2, \dots, Q^K) \\ \text{subject to} \quad & m_{n-1} - m_n \leq 0, \quad n = 2, \dots, K, \\ & -m_n \leq 0, \quad n = 2, \dots, K, \\ & \boldsymbol{c}^\top \boldsymbol{m} \leq B_c, \end{aligned} \tag{5.10}$$

where $\boldsymbol{c}$ is the cost vector. A global solution for the optimisation problem (5.10) can be found under two assumptions:

1. The K models have correlations ordered as $|\rho^1| > \ldots > |\rho^K|$.
2. The vector $\boldsymbol{c}$ satisfies the $K - 1$ inequalities

$$\frac{c_{n-1}}{c_n} > \frac{(\rho^{n-1})^2 - (\rho^n)^2}{(\rho^n)^2 - (\rho^{n+1})^2}, \text{ for } n = 2, \dots, K. \tag{5.11}$$

Under the assumptions above, it can be shown [141] that $Q^{n*} = \frac{\rho^n \sigma^1}{\sigma^n}$, for $n = 2, \dots, K$. Moreover, if we construct the vector $\boldsymbol{g}^* = [g^{1*}, \dots, g^{K*}]^\top$ with elements defined as

$$g^{n*} := \sqrt{\frac{c_1[(\rho^n)^2 - (\rho^{n+1})^2]}{c_n(1 - (\rho^2)^2)}}, \quad n = 1, \dots, K, \tag{5.12}$$

then we can compute the vector of optimal sample numbers $\boldsymbol{m}^* = [m_1^*, \dots, m_K^*]$, where $m_n^* = m_1^* g^{n*}$ for $n = 2, \dots, K$, and $m_1^* = \frac{B_c}{\boldsymbol{c}^\top \boldsymbol{g}^*}$, where B_c is the computational budget. Once $\boldsymbol{m}^*$ and $Q^{2*}, \dots, Q^{K*}$ have been computed, the models can be evaluated with the corresponding number of samples, and Eq. (5.7) can be used to estimate the QoI.

In our application, there are three QoIs to be estimated. In particular, the lower and upper bound of the re-entry window and the expected value of the re-entry time, given by $\hat{s}^{lb}_{\text{MFMC}}$, $\hat{s}^{ub}_{\text{MFMC}}$, and $\hat{s}^{avg}_{\text{MFMC}}$, respectively. Specific details are provided in Algorithm 10. The intuitive idea is that once the vector of optimal evaluation number $\boldsymbol{m}^*$ for a given computational budget B_c is obtained, we can simply evaluate the re-entry times of the samples, by performing a forward propagation of m_n^* (and where applicable m_{n-1}^*) samples with each of the respective $n = 1, \dots, K$ fidelities, obtaining the MFMC estimators for each of the quantities by simply saving the nth estimator of the average $\hat{s}$, the 1% and the 99% quantiles of the re-entry time distribution.

5.4 Results and discussion

This section is divided into two parts, showing the results for each of the test cases. In particular, the metrics explored to evaluate performance revolve around the computation of the expected time of re-entry, and predicting a time window for probable re-entry.

Algorithm 10 Multifidelity Monte Carlo

Inputs:

- A HF model and $K - 1$ surrogate models.
- A computational budget B_c.
- The altitude threshold, β.
- The initial distribution π_0 with mean state $\hat{\boldsymbol{x}}_0 \equiv \hat{\boldsymbol{x}}(0)$ and covariance matrix $\boldsymbol{\Sigma}_0$.

Outputs:

- Re-entry time window estimators $\hat{s}^{lb}_{\text{MFMC}}$, $\hat{s}^{ub}_{\text{MFMC}}$, and $\hat{s}^{avg}_{\text{MFMC}}$, i.e., the lower bound, upper bound, and mean estimates of the re-entry window.

Procedure:

Optimisation

1. Draw N samples $\boldsymbol{x}^i_0 \sim \mathcal{N}(\hat{\boldsymbol{x}}_0, \boldsymbol{\Sigma}_0)$.

2. Propagate these samples through $\Phi^n_{1:L}(\cdot)$, $n = 1, \ldots, K$, from t'_0 to some later time $t'_L = T'$.

3. Compute final altitudes of all samples across the models as $\mathcal{A}^{i,n}$, for $i = 1, \ldots, N$ and $n = 1, \ldots, K$, following Eq. (5.2).

4. For each model, compute and save the standard deviations between the samples, σ^n, as well as the Pearson correlation coefficients ρ^n between the surrogate samples and those computed with the HF model. Save also the computational cost per sample c_n for $n = 1, \ldots, K$.

5. As long as conditions 1 and 2 of Eq. (5.11) are satisfied, Eq. (5.12) is applied to compute the set $\{g^{n*}\}^K_{n=1}$.

6. Compute global optimum vector $\boldsymbol{m}^*$, for the number of model evaluations and $\boldsymbol{Q}^{n*} = \frac{\rho^n \sigma^1}{\sigma^n}$, for each model n.

MFMC

7. For each model $n = 1, \ldots, K$, and for the time span $[t'_0, t'_L]$, run $\Phi^n_{1:L}$ on m^*_n samples, out of which m^*_{n-1} are those evaluated on the previous model.

8. Compute the vector of re-entry times $\boldsymbol{T}^n_\beta$, by saving, for each sample $i = 1, \ldots, m^*_n$, the time-stamp of each sample before its altitude $\mathcal{A}^{i,n}(t_l)$ reaches below the threshold β.

9. Compute the QoIs, in this case $\hat{s}^{lb}$, $\hat{s}^{ub}$, and $\hat{s}^{avg}$, following Eq. (5.7)

$$\hat{s}_{\text{MFMC}} = \hat{s}^1_{m^*_1} + \sum_{n=2}^{K} \left(\hat{s}^n_{m^*_n} - \hat{s}^n_{m^*_{n-1}} \right) \boldsymbol{Q}^{n*}$$

In the implementation of the MFMC in this work, the computational budget, as well as the computational costs of the different models are handled in the form of computational run-time in seconds. It follows that different implementations of the algorithm or the use of different platforms may incur different processing times. It is important to note that if computational run-time is used, the costs vector needs to be re-initialized whenever a different platform is used.

In this chapter, the abbreviation for the quantity of seconds is expressed as [sec] instead of [s], to avoid confusion with the QoI $\hat{s}$.

5.4.1. Propagation models

This section contains the description of the dynamical models and propagation methods used for the generation of the results in this paper. In fact, any dynamical model can be used, provided conditions 1 and 2 in Section 5.3.5 hold. It is important to discard any model that does not satisfy them, as the MFMC estimator equations and its optimal parameters may not otherwise hold.

The HF model is the stochastic Runge Kutta of Section 2.4.2, which employs a Cowell formulation with gravitational perturbations J_2 to J_4, the NRLMSISE-00 atmospheric drag model, and integration step-size $h = 10$ sec. Different fidelities may be included by making adjustments to the described model by either removing modelled forces, or decreasing/increasing the step-size of integration for the stochastic Runge-Kutta. Surrogate model number one is the Cowell formulation, with all of the mentioned perturbations, but with integration time step $h = 100$ sec. The second surrogate model is the Cowell formulation with only J_2 and the NRLMSISE-00 drag model, as well as integration step-size $h = 100$ sec. For the simulations in this work, the number of samples for the optimisation problem is $N = 250$.

The simulations for the proposed method were run on an Apple Macbook Pro with an M1 chip, an 8-core CPU, and 16GB of unified RAM memory.

5.4.2. Roseycubesat-1

We show the performance of the MFMC estimator with three different propagation models compared to i) a CMC that samples directly from the HF model, and ii) a recorded altitude obtained from real observations around the time of decay. The metrics to evaluate performance are the expected time of re-entry, and the time window for probable re-entry. The last available observation of the cubesat places it at an altitude of 200 km on the morning of 31 August 2024. We use MFMC and CMC to estimate the time at which the 200 km mark is surpassed and to perform a comparison. The ability of the MFMC method to achieve convergence with reduced variance compared to CMC is assessed first, by performing 50 independent simulations of both methods, enabling the computation of the mean target time as calculated by the MFMC estimator, $\hat{s}_{\text{MFMC}}$ and by the CMC estimator $\hat{s}_N$. Their empirical standard deviations are also evaluated, as a function of the computational budget. Additionally, we validate our predictions with the time of the last observed state (at 200 km), taken as the ground truth. However, though one might take the last observation as the time of decay, sources such as [149] confirm that the re-entry was in fact almost a day later, as spacecraft may undergo additional revolutions after they are last seen. Therefore, the simulation is extended beyond the 200 km mark, until the point where the propagation models fail or begin behaving erratically and unpredictably. We experimentally determine this point to be 106 km, after which we can say that a real decay has occurred. Figure 5.5 shows the altitude profile of the spacecraft for the duration of a simulation using HF propagation. Both the 200 km and the 106 km marks are included for clarity.

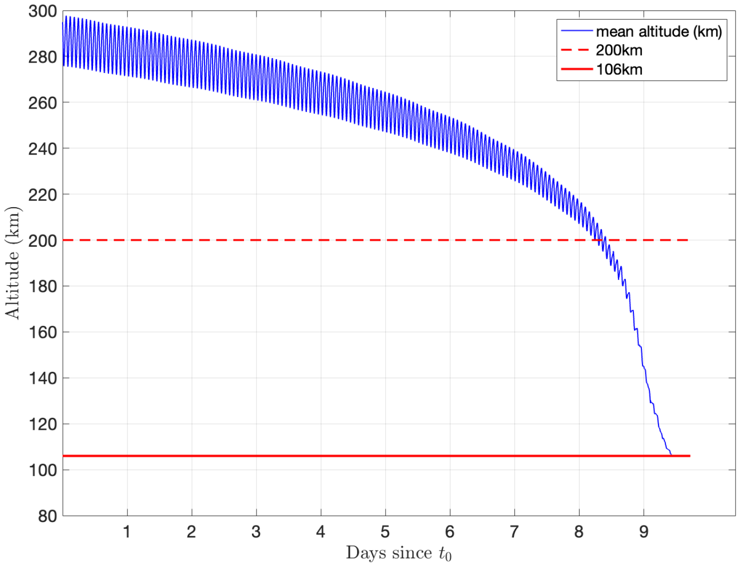


Figure 5.5: The mean altitude of the collection of HF samples used in the MFMC algorithm is shown in this figure. The 200 km mark is the altitude at which the final observation occurred, and the 106 km mark represents the onset of numerical instabilities: the decay event is declared.

Variance analysis

We compute the mean estimators $\hat{s}_{\text{MFMC}}$ and $\hat{s}_N$ and their 3σ empirical standard deviations over 50 simulations, and vary the computational budget to take values $B_c \in \{25;\ 100;\ 500;\ 1,000;\ 2,500\}$ [sec]. Figure 5.6 shows the above quantities, as well as the ground truth (black dotted line), showing the days after time t_0 at which the spacecraft surpassed 200 km, which is the last available measurement, on 31 August 2024, at 06:11.

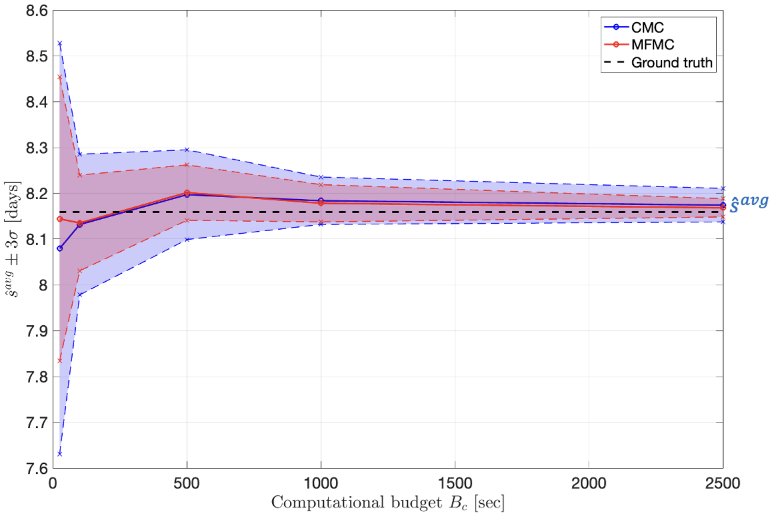


Figure 5.6: MFMC (red) and CMC (blue) estimators for the mean re-entry time for Roseycubesat-1, for different computational budgets. Their 3σ values are plotted in shaded regions of their respective colours. The time of the last available measurement is shown as a black dotted line.

We can see that the MFMC achieves asymptotic unbiasedness with respect to the HF CMC, as the computational budget increases. In addition, the variance of the MFMC estimator is lower than the CMC and remains so for all values of B_c. The models do not achieve full convergence to the exact ground truth value, due to mismatches between the dynamical model used and the true dynamics involved.

Table 5.3: Performance comparison of MFMC and CMC on the Roseycubesat-1 test case, in terms of computational cost (in seconds) and standard deviation when estimating reentry time (in days since t_0).

B_c [sec]	σ $(MFMC)$ [days]	σ (CMC) [days]
25	0.1032	0.1496
100	0.0348	0.0512
500	0.0202	0.0326
1000	0.0136	0.0173
2500	0.0067	0.0127

Table 5.3 shows the computational cost B_c and empirical standard deviations σ of the proposed MFMC implementation, compared with the CMC simulations. Given a computational budget, the MFMC estimator is able to achieve a consistently lower variance than that of CMC, which is the principal objective of using MFMC.

Figure 5.7 shows the empirical standard deviations in hours, for both MFMC and CMC.

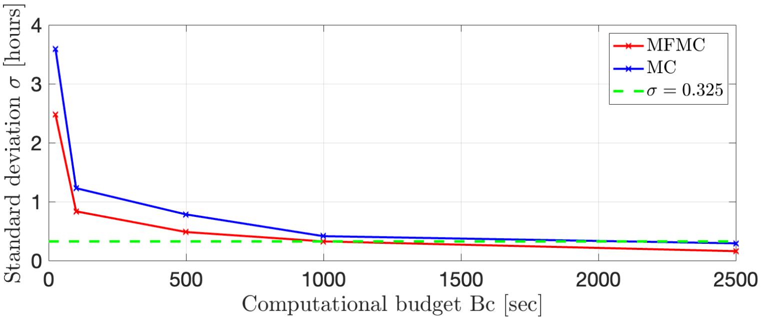


Figure 5.7: The standard deviations of the multifidelity Monte Carlo (MFMC) estimator (red) and crude Monte Carlo (CMC) estimator (blue), for different computational budgets. The MFMC's $B_c = 1,000$ sec is chosen as reference (green dotted line) for the variance equivalent CMC budget.

A green horizontal line is plotted to show, as further contrasted by Table 5.3, that an MFMC with $B_c = 1,000$ sec achieves approximately equal variance values as a CMC estimator with $B_c = 2,500$ sec.

Following this observation, we can set the computational budget required for a fair comparison of the two methods, when estimating the real re-entry window, as a function of the spacecraft's RSL in days.

Re-entry window prediction

In the previous section, we compared the MFMC estimator with real data corresponding to the last available observation. The re-entry time as published in [149] was 1 September 2024 at 00:00 (almost a day later). We therefore extend the propagation to simulate until the spacecraft undergoes no more full revolutions around the Earth (actual decay). The validation metric for the MFMC in this extended case is the CMC simulation. In order to obtain the lower and upper bounds of the re-entry window, the decay times of all samples are recorded for each of the surrogate models. We therefore obtain a distribution of re-entry times from which we extract quantiles. The lower bound is chosen as the 1% percentile, whilst the upper bound is chosen to be the 99% percentile. The MFMC then computes estimates of these quantities, as well as the mean, to obtain a singular averaged value for each of these QoIs. Figure 5.8 shows the re-entry time window spanned by QoIs $\hat{s}^{\text{lb}}$ and $\hat{s}^{\text{ub}}$ (cyan), with computational budget $B_c = 1,000$ sec. Based on this, the MFMC determines the optimal number of model evaluations to be $\boldsymbol{m}^* = [118,\ 820,\ 1540]$, where each entry is the optimal number of evaluations of models one, two and three, respectively. Figure 5.8 also shows the histogram of re-entry times (green) of an independent CMC run with $B_c = 2,500$ sec. The mean MFMC estimator $\hat{s}^{\text{avg}}$ is plotted as a single dotted red line. In addition, the industry standard window of width ±20% of the RSL is shown for comparison (pink).

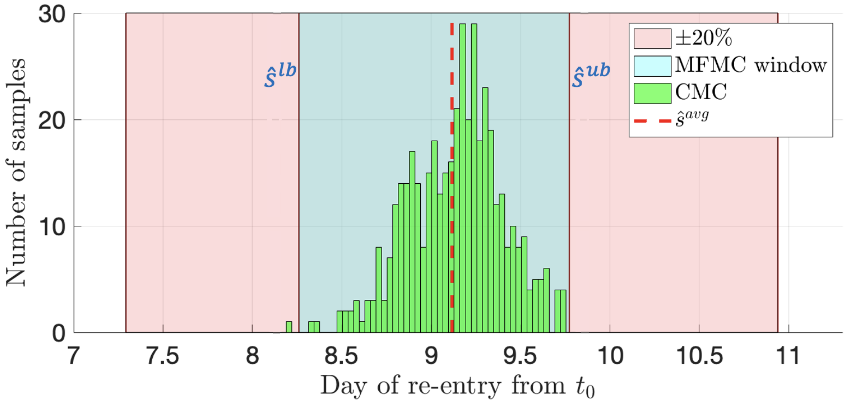


Figure 5.8: Multifidelity Monte Carlo (MFMC) re-entry time window shown in cyan for Roseycubesat-1, bounded by the estimators $\hat{s}^{lb}$ and $\hat{s}^{ub}$. The MFMC's computational budget was $B_c = 1,000$ sec. The figure also shows the average estimate $\hat{s}^{avg}$ (red dotted line), and a CMC simulation of $B_c = 2,500$ sec (green histogram), and the ±20% RSL recommended in the literature.

We observe that the asymmetrical window spanned by the lower and upper bounds contains approximately 99.8% of the CMC samples, ensuring operational safety when predicting a window of probable decay time, and providing a sensible shortening to the standard symmetrical window of 20% RSL. The $\hat{s}^{\text{avg}}$ is close in value to the centre of the CMC distribution, showcasing the unbiasedness of this estimator, as compared with the CMC.

5.4.3. GOCE

We show the performance of the MFMC estimator with three different propagation models compared to i) a CMC that samples directly from the HF model, and ii) a recorded altitude obtained from real observations around the time of decay. The metrics to evaluate performance are the expected time of re-entry, and the time window for probable re-entry. GOCE was controlled in attitude for most of its mission, but is considered an uncontrolled decay event due to its uncontrolled decay location [128], which took longer than expected due to the atmospheric density being lower than expected. The ability of the MFMC to achieve convergence to the HF estimate with reduced variance compared to the CMC is assessed first by computing the average of the mean, upper bound and lower bound MFMC and CMC estimators of the target time and their standard deviations, as a function of the computational budget (over many simulations). Then, the MFMC re-entry window prediction is compared with the CMC distribution, as well as the real re-entry time.

Variance analysis

We compute the mean estimators $\hat{s}_{\text{MFMC}}$ and $\hat{s}_N$ (CMC) and their 3σ empirical standard deviations over 50 simulations, and vary the computational budget to take values $B_c \in \{25;\ 100;\ 500;\ 1,000;\ 2,500;\ 10,000\}$ [sec]. Figure 5.9 shows the above quantities, as well as the ground truth (black dotted line), showing the days after time t_0 at which the spacecraft surpassed 80 km, which is the last available measurement, on 11 November 2013, at 00:18.

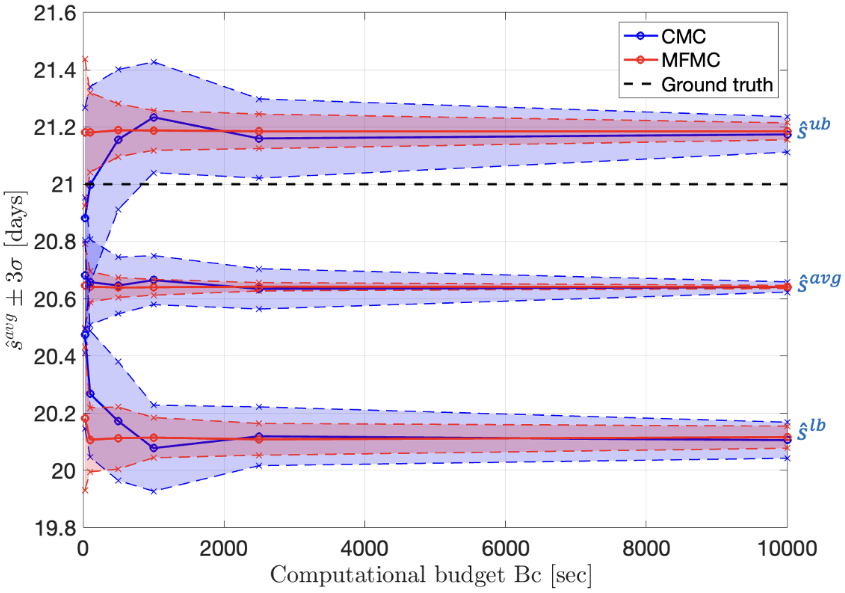


Figure 5.9: MFMC (red) and CMC (blue) estimators for the mean target time (last available measurement) for GOCE, for different computational budgets. Their 3σ values are plotted in shaded regions of their respective colours.

We can see that MFMC achieves asymptotic unbiasedness relative to HF CMC as the computational budget increases. In addition, the variance of the MFMC estimator is lower than CMC and remains so for all values of B_c. The lower and upper bound MFMC estimators, $\hat{s}^{lb}$ and $\hat{s}^{ub}$, are compared with the respective lower and upper bounds of the CMC simulation. Again, there is a bias in the estimated mean time of re-entry, due to differences between the dynamical model used and the true dynamics, but the ground truth value is contained within the MFMC bounds.

Table 5.4 shows the computational cost B_c and empirical standard deviations σ of the proposed MFMC implementation, in comparison with that of an independent CMC run.

We see that with a given computational budget the variance of the MFMC estimator is consistently lower than that of the CMC, which is the principal objective of using such an estimator.

Table 5.4: Performance comparison of MFMC and CMC on the GOCE test case, in terms of computational cost (in seconds) and standard deviation when estimating reentry time (in days since t_0).

B_c [sec]	σ $(MFMC)$ [days]	σ (CMC) [days]
25	0.0488	0.0904
100	0.0179	0.0496
500	0.0113	0.0329
1000	0.0092	0.0288
2500	0.0050	0.0233
10000	0.0018	0.0060

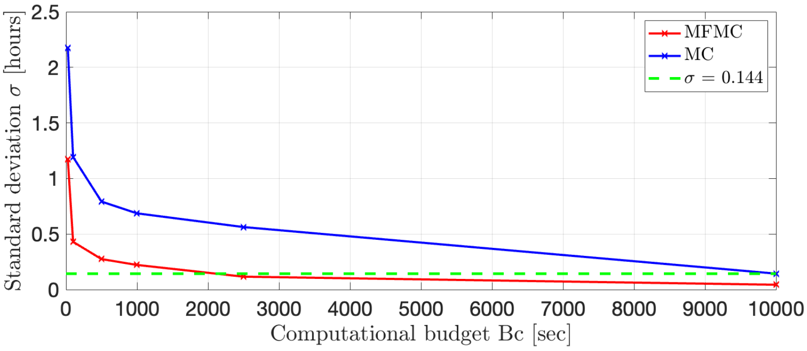


Figure 5.10: Standard deviations of the multifidelity Monte Carlo (MFMC) estimator (red) and crude Monte Carlo (CMC) estimator (blue) for different computational budgets. The MFMC's $B_c = 2,000$ sec is chosen as reference (green dotted line) for the variance equivalent CMC budget.

Figure 5.10 shows the empirical standard deviations in hours, for both MFMC and CMC. A green horizontal line is plotted to show, as further contrasted by Table 5.4, that an MFMC of $B_c = 2,000$ sec achieves approximately equal variance values as a CMC simulation of $B_c = 10,000$ sec. Following this observation, we can set the computational budget required for a fair comparison of the two methods when estimating the real re-entry window as a function of the spacecraft's RSL in days.

Re-entry window prediction

The re-entry condition for this case was the surpassing of altitude threshold $\beta = 80$km. The validation metric for the MFMC in this extended case is the CMC simulation, as well as comparison with the ground truth re-entry date. In order to obtain the lower and upper bounds of the re-entry window, the decay times of all samples are recorded for each of the surrogate models. We therefore obtain a distribution of re-entry times from which we extract quantiles. The lower bound is chosen as the 1% percentile, whilst the upper bound is chosen to be the 99% percentile. The MFMC then computes estimates of these quantities, as well as the mean, to obtain a singular averaged value for each of these QoIs.

Figure 5.11 shows the re-entry time window spanned by QoIs $\hat{s}^{\mathrm{lb}}$ and $\hat{s}^{\mathrm{ub}}$ (cyan), with $B_c = 2,000$ sec. Based on this, the MFMC determines the optimal number of model evaluations to be $\boldsymbol{m}^* = [179,\ 482,\ 4621]$, where each entry is the optimal number of evaluations of models one, two and three, respectively. Figure 5.11 also shows the histogram of re-entry times (green) of an independent CMC run with $B_c = 10,000$ sec. The mean MFMC estimator $\hat{s}^{\mathrm{avg}}$ is plotted as a single dotted red line. Note that the industry standard window of width ±20% of RSL is not shown, as the window is computed with data 21 days prior, hence, the ±20% window is somewhat uninformative.

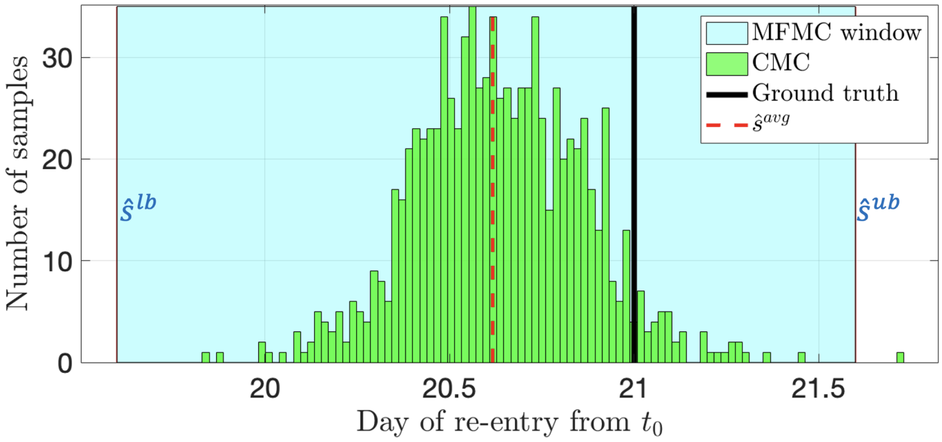


Figure 5.11: Multifidelity Monte Carlo re-entry time window shown in cyan for GOCE, bounded by $\hat{s}^{lb}$ and $\hat{s}^{ub}$. The computational budget for the MFMC is $B_c = 2,000$ sec. The figure also shows the average estimate $\hat{s}^{avg}$ (red dotted line) and a crude Monte Carlo simulation of $B_c = 10,000$ sec (green histogram).

We observe that the asymmetrical window spanned by the lower and upper bounds contains approximately 99.8% of the CMC samples, ensuring operational safety when predicting a window of probable decay time. The $\hat{s}^{\mathrm{avg}}$ is close in value to the centre of the CMC distribution, showcasing the relative unbiasedness of the MFMC estimator, compared with the CMC estimator.

5.5 Concluding remarks

The work presented here aims to provide a computationally efficient method with which to attain a level of estimation accuracy similar to CMC simulations. In order to account for dynamical model mismatch, stochastic propagation is used in the numerical integration scheme. This enables us to appropriately quantify the uncertainty present in the propagation of spacecraft states. The test performed for the appropriate assessment of the diffusion coefficient scale for this problem reveals that only relatively large magnitudes (in the range 1×10^{-7} to 1×10^{-6} and upwards) produce a significant increase in window size. Although a higher level of uncertainty may produce more conservative predictions, favouring safety when predicting re-entry windows, it is important to choose the diffusion scaling based on the dynamical model truncation or on other parameters which are expected to drive the uncertainty.

We apply an MFMC approach to two re-entry scenarios: the Roseycubesat-1 mission which decayed on 1 September 2024 [149], and the GOCE mission re-entry, which was considered terminated on 21 October 2013, when it ran out of fuel and entered a decaying trajectory. The official re-entry time was 11 November 2013, when it crossed below the altitude of 80 km [128]. The last known observation of the Roseycubesat-1 places it at an altitude of 200 km on 31 August 2024, and the MFMC estimator is able to predict the time of arrival at the 200 km mark with a precision of merely a few hours, and this bias is due to the dynamical model's inaccuracies. In both cases, by comparing the MFMC with a CMC simulation of sufficiently many samples, we show that the variance of the MFMC estimator is greatly reduced compared to that of the CMC. The MFMC is provably asymptotically unbiased with respect to the HF model [152][144], as also seen in the numerical results in Sections 5.4.2 and 5.4.3.

Aside from the mean re-entry time estimate, the MFMC can provide estimates for lower and upper bounds of re-entry times, which can render a re-entry RSL window. The window predicted by the proposed method contains the CMC probability distribution almost in its entirety for both scenarios, accounting for not only the most probable re-entry times, but also for those at the tails of the distribution, which favours operational safety when dealing with uncontrolled re-entry events.

The proposed MFMC approach produces estimates which i) are directly comparable with real observations, and ii) show a significant reduction in variance relative to an independent CMC simulation of sufficient samples.

6

CONCLUSIONS AND FUTURE WORK

So we can take the world back from a heart attack
One maniac at a time we will take it back
You know time crawls on when you're waiting for the song to start
So dance alone to the beat of your heart
Hey young blood, doesn't it feel like our time is running out?

Fallout Boy
The Phoenix (2013)

In this chapter, conclusions about the three principal topics of study in this thesis are presented and summarised, followed by a brief discussion of future research directions —including some preliminary results.

6.1 Summary and conclusions

In Chapter 3, a probability of collision (PoC) computation method which builds on a novel collision risk metric for space objects is proposed, providing means to bypass common simplifying assumptions involving their state distributions and the relevant dynamics at the time of collision. This is complemented by employing importance sampling (IS), a Monte Carlo method aimed at reducing estimation variance and hence, computation costs. Concretely, the techniques presented rely on a reduced-dimension orbital propagation model for assessment of collision states that converts the full dimensional problem into one in two dimensions. Based on this 2D model, the PoC is approximated using IS, which enables the approximation of potentially very small PoCs with a relatively small number of samples compared to standard crude Monte Carlo (CMC). In this work, a Gaussian proposal is used, which refers to states at the initial time t_0, but no assumptions are imposed on the distribution of states at time of closest approach (TCA). The new method is shown to reliably attain similar or better accuracy than a CMC simulation with a fraction of the computational cost in two different scenarios: a low-Earth orbit (LEO) conjunction and a geostationary orbit (GEO) conjunction. In light of the results obtained, this method could satisfy the need to provide reliable safety guidelines for satellite operators for the prevention of collisions, as well as for the prevention of unnecessary collision avoidance manoeuvres, though it is still dependent on the uncertainty in the initial states. This uncertainty can be reduced with improved orbit determination procedures.

In Chapter 4, we aim to satisfy this need by first introducing a set of filtering algorithms and comparing their performance as a function of sample number and minimum re-observation time in two different scenarios involving LEO and GEO spacecraft. The filters studied are the CKF, the EnKF, the EnCKF, the EnPF and the BPF. The EnPF attains the best performance in LEO, whilst the BPF requires too large a sample size to reach sufficient accuracy. The CKF completely diverges when the minimum time between observations exceeds two minutes. All of the methods work sufficiently well in GEO.

Next, we present three recursive filters, the EnKFup, the EnPFup and the NHF, which incorporate elements of PFs and Gaussian filters, to track a LEO spacecraft for a period of approximately nine days, whilst simultaneously estimating the diffusion coefficient required for the proper characterisation of the uncertainty. The three algorithms achieve low errors in both position and velocity compared to a nonlinear batch least-squares method. The ability to estimate model parameters is leveraged to implement a stochastic parameterisation of the unknown accelerations in the system. The algorithms are shown to be able to estimate the magnitude of the nominal parameter σ_W accurately (test 1), and to track a spacecraft in the case where the model used is simplified in order to cut run-time costs (test 2). In terms of performance, the EnPFup achieves the best trade-off between cost and accuracy out of the three algorithms. The result is an ability to appropriately characterise the uncertainty of the system whilst performing satisfactory tracking of a spacecraft in the presence of observations, i.e., to “estimate what you do not know" online.

Chapter 5, proposes a multifidelity Monte Carlo (MFMC) approach, which provides a computationally efficient way to attain a level of estimation accuracy akin to CMC simulations when predicting re-entry spacecraft windows. This task is necessary in order to avoid over-enlarged and uninformative re-entry time windows simply to preserve safety. By applying MFMC to two decaying spacecraft scenarios —Roseycubesat-1 and GOCE— in which a reference altitude threshold is available, we are able to directly compare our estimations with reality, as well as with an independent CMC simulation of sufficient samples, used as a benchmark experiment. We show that the variance of the MFMC estimator is greatly reduced compared to that of the CMC for both cases. The MFMC method achieves asymptotic unbiasedness relative to the high-fidelity (HF) CMC as evidenced by the improved accuracy as the computational budget is increased. The results obtained are validated with real data with information about the decay phase. Aside from the mean re-entry time estimate, the MFMC algorithm can provide estimates for the lower and upper bounds of re-entry times, which can render a remaining spacecraft lifetime (RSL) window. The windows predicted by our method contain the CMC probability distributions of the two test cases almost in their entirety, accounting for not only the most probable re-entry times, but also for those at the tails of the distribution, favouring operational safety when dealing with uncontrolled re-entry events.

The topics studied in this thesis lie at the core of safe space traffic management. The study of collision risk, orbit determination, and re-entry window analysis are fundamental aspects of ensuring operational awareness in space. The overarching challenge in the traditional counterpart of these methods is that both state distribution and dynamical model simplifications are usually made, mainly by assuming Gaussianity, and procedures to bypass them can be computationally prohibitive. In addition, uncertainty in the propagation of spacecraft states is often treated in an ad-hoc or arbitrary manner, and the growing resident space object population demands improvements in its characterisation and the resulting refinements in the very methodologies aimed at keeping spacecraft safe. This thesis provides alternative methods in an attempt to bypass common assumptions and appropriately account for the uncertainty present in various space traffic management prediction and estimation processes, in a more efficient manner. It does not claim to fully solve the problem in the application areas studied, but rather open up new paths for deeper development of more advanced techniques to deal with the ever-increasing population of space objects. The next section aims to provide a bridge between the contents of this thesis and potential operational frameworks.

6.2 Future work

This section is devoted to the description of concepts and methodologies which stem from those studied in this thesis, but have been left for further development in the future.

6.2.1. Orbital collision risk assessment

Uncertainties in dynamical parameters, such as the drag coefficient, or the SRP coefficient, can also affect the PoC. The methodology for the calculation of PoC, in its present form, assumes nominal values for C_{SRP} and C_D, and therefore uncertainty in these parameters is not explicitly treated. The inclusion of uncertainty in said parameters, by simply replacing deterministic propagation by stochastic propagation could potentially incur a degree of dilution by producing significant variations in the predicted trajectories. In the case of re-entry prediction, fine-tuning the stochastic parameters is less critical (and a larger uncertainty may at times even prove to be the safer option), but with PoC calculations, the maximum precision whilst taking uncertainty into account is desired. The methodology in Chapter 3 could therefore be extended in the following ways:

- Incorporating the uncertain parameters (say, C_D) into the joint state vector: This would imply transforming the 12-dimensional state vector into a 14-dimensional vector, and applying IS on the augmented state. This would involve propagating the ith 12-dimensional state vector, conditional on the corresponding ith value for the parameters, implicitly treating the uncertainty. The colliding samples would still be evaluated the same way, so the methodology need not be modified, in principle, other than to take care of propagating samples with the corresponding parameter values. The trade-off could potentially involve an increase in computational cost in achieving convergence and working with a larger dimensional state-vector.
- Implementing stochastic propagation in a stochastic framework: When we model orbital dynamics as a stochastic dynamical system, we can incorporate process noise, with a magnitude proportional to the uncertainty incurred by our knowledge (or lack thereof) of the relevant parameter. This involves incorporating stochastic propagation of samples, whilst retaining parts of the methodology proposed in Chapter 3, Section 3.2.3. The following subsection aims to expand on this concept, to ensure the foundations are laid out properly.

A stochastic PoC computation framework

The proposed framework for the calculation of PoC consists of employing the concept behind IS, with the inclusion of stochastic dynamics, and a more sophisticated procedure for the generation of the proposal distribution. This transforms the dimensionality reduction surrogate method used in Chapter 3 into one which accounts for uncertainty in unknown parameters. The proposed stochastic framework is given in Algorithm 11. As such, the proposed method aims to tackle three different problems:

- The first one involves notions of Chapter 4, for the sequential tracking of spacecraft states $\boldsymbol{x}_k$ which lead to collision at a future time. Assume that “fictitious" observations are collected at times $\{t_k\}_{k=1}^{M}$ and denoted z_k for conciseness. These observations are centred on $z_k = [0,\ 0]$, and approximately spanning the collision area in $\mathbb{R}^2$. The collection of samples $\boldsymbol{x}_k$ is resampled based on a likelihood function describing the similarity between the observations and the samples. To calculate the likelihood, the samples must first be mapped into the surrogate reduced dimension space defined in Section 3.2.4. Resampling (see Section 4.2.2) is then performed to replicate samples with high likelihood values.

- The second problem involves stochastic propagation of the initial distribution — independently with respect to the proposal distribution of step one — to obtain a propagated baseline distribution at some later time $t < T$ at which the PoC is calculated. The stochastic diffusion scale is set to an informed value inferred from the sample scenarios in Chapters 4 and 5.

- The third one involves the actual calculation of the PoC. The calculation of importance weights must be carried out in a careful manner, since it involves the approximation of the multidimensional (12D) densities of the proposal and target distributions. Several methods exist for this, though in this work, these densities are approximated by means of the K nearest neighbours (KNN) algorithm (see [153] and [154])

Preliminary results

Preliminary results obtained with the described methodology are shown in this section. The test-case used is the LEO scenario of Section 3.4.1, though computed using the HPOP library in MATLAB (see Section 2.4), which applies the USSA76 drag model, the GGM03C model for Earth gravitational potential up to degree and order (70,70) and Luni-Solar perturbations. The numerical scheme used is the SRK of Section 2.4.2. The computations for the proposed methods in both setups are run on a Macbook Pro with an Apple M1 processor.

Figure 6.1 shows the sequential construction of the proposal distribution μ_k at different time-stamps throughout propagation, mapped onto the collision 2D space. Samples in green are those which are stochastically propagated and mapped into $\mathbb{R}^2$, whilst samples in blue are those which are resampled after processing the fictitious observations in 2D space. It can be seen that samples effectively concentrate in and around the collision threshold, enabling the computation of final IS weights and consequently, a PoC value.

Algorithm 11 Stochastic IS PoC computation

```
Inputs:
  - The HBR, $\gamma$ and collision threshold $\gamma'$.
  - The number of samples $N$.
  - The initial distribution $\pi_0$ with mean $\hat{\boldsymbol{x}}(0)$ and covariance matrix $\boldsymbol{\Sigma}_0$.
  - $M$ observation time-stamps, $\{t_k\}_{k=1}^{M}$ and a set of “fictitious" observations with mean $z_k = [0,\ 0]$ and their noise covariance matrices $\Omega_k,\ k = 1, ...M$.
Outputs:
  - The IS estimator of the PoC at time $t_{M-1}$, denoted $\bar{P}^{\mathrm{IS},N}_{T_{M-1},\gamma}$.
Procedure:

  1. To obtain the $\pi_{M-1}$ distribution, propagate samples from $\pi_0$ stochastically from time $t_0$ to $t_{M-1}$, using the numerical scheme and dynamical model of choice, obtaining
     $\tilde{\boldsymbol{x}}^i_{M-1} \sim p(\boldsymbol{x}_{M-1}|\boldsymbol{x}^i_0),\ i = 1, ..., N_1$.
  2. Let $\mu_0 = \pi_0$.

FOR $t_k = t_1, \ldots, t_{M-2}$

                              Prediction

  2. Propagate samples from $\mu_{k-1}$ from time $t_{k-1}$ to $t_k$, using the stochastic numerical scheme and dynamical model of choice, obtaining
     $\tilde{\boldsymbol{x}}^i_k \sim p(\boldsymbol{x}_k|\boldsymbol{x}^i_{k-1}),\ i = 1, ..., N$.

                              Update

  3. Set $\boldsymbol{x}_{\mathrm{ref},0}$ to be the mean of the distribution $\{\tilde{\boldsymbol{x}}^i_k\}_{i=1}^{N}$ to compute $\boldsymbol{\epsilon}_{\mathrm{ref},k}$ in Eq. (3.16), and Jacobian $\boldsymbol{J}^{\mathrm{HF}}_k$ for node passages $t_k,\ k = 1, \ldots, c$.
  4. Compute predicted measurements $\boldsymbol{y}^i_k = \mathcal{M}_k(\tilde{\boldsymbol{x}}^i_k) \equiv \tilde{\boldsymbol{\xi}}^i_k$ using the surrogate model of Eq. (3.17) and evaluate likelihoods $L^i_k \propto p(z_k|\tilde{\boldsymbol{x}}^i_k),\ i = 1, \ldots, N$.
  5. Compute normalised importance weights:
     $w^i_k \propto L^i_k,\ i = 1, ..., N$.
  6. Resample the weighted set $\{\tilde{\boldsymbol{x}}^i_k, w^i_k\}_{i=1}^{N_1}$ $N_1$ times with replacement to generate new particles $\boldsymbol{x}^i_k,\ i = 1, ..., N$.
  7. The proposal function at time $t_k$ is given by $\mu_k \equiv \{\boldsymbol{x}^i_k\}_{i=1}^{N}$, i.e., the resampled particles.

END FOR

  8. Compute normalised importance weights $w^i = \frac{w^{*,i}}{\sum_{l=1}^{N_1} w^{*,l}},\ i = 1, \ldots, N$, where $w^{*,i} = \frac{\pi_{M-1}(\boldsymbol{x}^i_{M-1})}{\mu_{M-1}(\boldsymbol{x}^i_{M-1})}$ are the non-normalised weights.
  9. Propagate $\mu_{M-1}$ to TCA, obtaining
     $\tilde{\boldsymbol{x}}^i_M \sim p(\boldsymbol{x}_M|\boldsymbol{x}^i_{M-1}),\ i = 1, ..., N$.
  10. Compute $d^i_c = \|\boldsymbol{r}_{M,B} - \boldsymbol{r}_{M,A}\|$ for $i = 1, \ldots, N$, then approximate the PoC at $t_{M-1}$ as
      $\bar{P}^{\mathrm{IS},N}_{T_{M-1},\gamma} = \sum_{i=1}^{N} w^i I_\gamma(d^i_c)$.
```

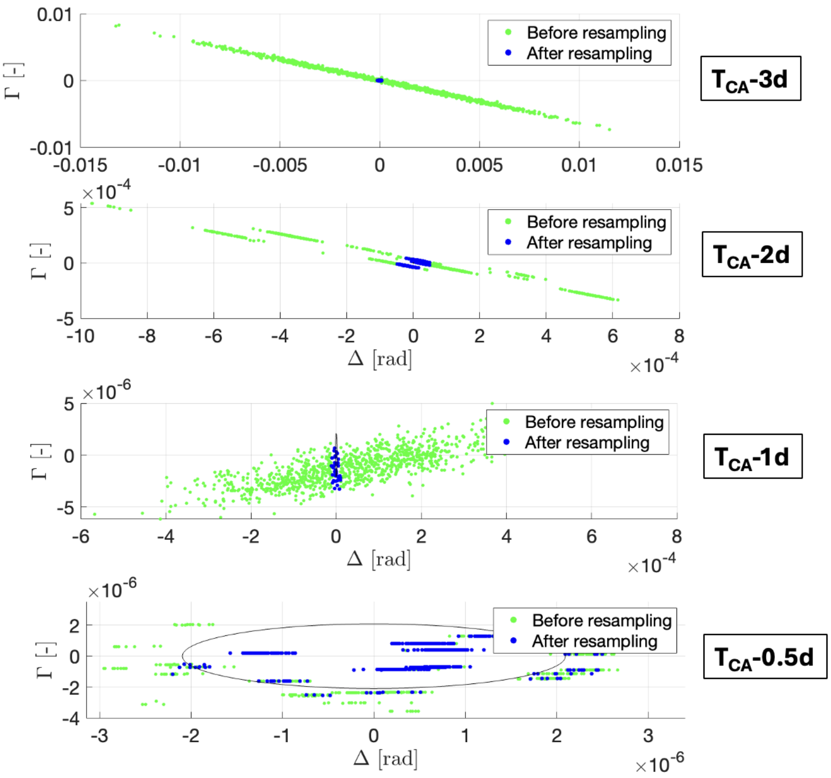


Figure 6.1: Shown is the sequential construction of the proposal distribution, as the resampling (shown in blue) drives the concentration of samples around collision. Those samples which are not resampled appear in green. In some instances, clouds of particles with drift can be observed, caused by the stochastic propagation.

Preliminary comparisons with a stochastic CMC simulation of $N = 10,000$ samples show a noticeable underestimation of the PoC as shown in Table 6.1. Further research into the possible causes of this difference is required.

Table 6.1: Performance comparison for LEO conjunction using stochastic framework: PoC estimates and run-time.

	PoC estimate	Run-time (s)
CMC simulation (HF)	3.20×10^{-3}	8.2×10^{4}
Stochastic IS	9.92×10^{-4}	581.2

6.2.2. Spacecraft re-entry prediction

The methodology followed in this chapter can be studied in more depth with regards to the modelling of uncertainty and fidelity in the propagation process. Regarding fidelity, a detailed comparison could be carried out between a HF 6 d.o.f propagation involving accurate attitude dynamical models as well as precise aerodynamic models, and a stochastic propagation model as that introduced in Chapter 5, Section 5.3.4. It is easy to see that stochastic propagation may never produce precision levels akin to 6 d.o.f. HF propagation, but that is not the objective of stochastic propagation, but rather to take proper uncertainty levels into account. In fact, 6 d.o.f. and stochastic propagation can be used together; the only implication is that the scale of the diffusion coefficient may be smaller, as the overall uncertainty will be lower.

Regarding the modelling of uncertainty in the MFMC, it is worth mentioning that an arbitrary number of dynamical models can be used, which enables the exploration and the inclusion of “exotic” uncertainty sources through the incorporation of different models. Specifically, the inclusion of models which assume different values for the C_D may aid in the proper characterisation of the uncertainty associated to drag effects, which are crucial in the propagation of re-entering spacecraft states. According to [141], using surrogate models which inform of different aspects of the system but may be less accurate is often more beneficial than only using accurate surrogates which lack diversity.

In addition, as it is the case for the other topics studied in this thesis, an expanded analysis of the methods proposed with a more extensive test case dataset would help to further validate the results obtained. One interesting case which could be used in this regard is the recent re-entry of the Kosmos 482 descent craft, a vessel designed to withstand atmospheric descent onto the surface of Venus. The case received significant media attention, and key space agencies and companies were unable to produce precise estimates of the time of re-entry. The spacecraft’s official time of re-entry was 10 May 2025, 06:24 UTC. By using TLE data from 6 May 2025, 09:39 UTC, and assuming a standard, but large initial covariance matrix, a HF propagation produces predictions which are in good agreement with the TLE data, as seen in Figure 6.2.

Furthermore, a preliminary CMC deterministic simulation of $N = 1,000$ samples generates the histogram shown in Figure 6.3. What is shown is that approximately four days out from decay, in the absence of process noise, a window of one day is obtained, but the mean estimate lies somewhat close to the real re-entry time. It is worth noting that unless focus is placed on improving the dynamical model in the HF propagation, we are unable to achieve a more *precise* estimate, but the application of MFMC is likely to exhibit an accurate time window of appropriate width corresponding to the uncertainty present, and a mean estimate which is at least not far from predictions obtained by key players in the field.

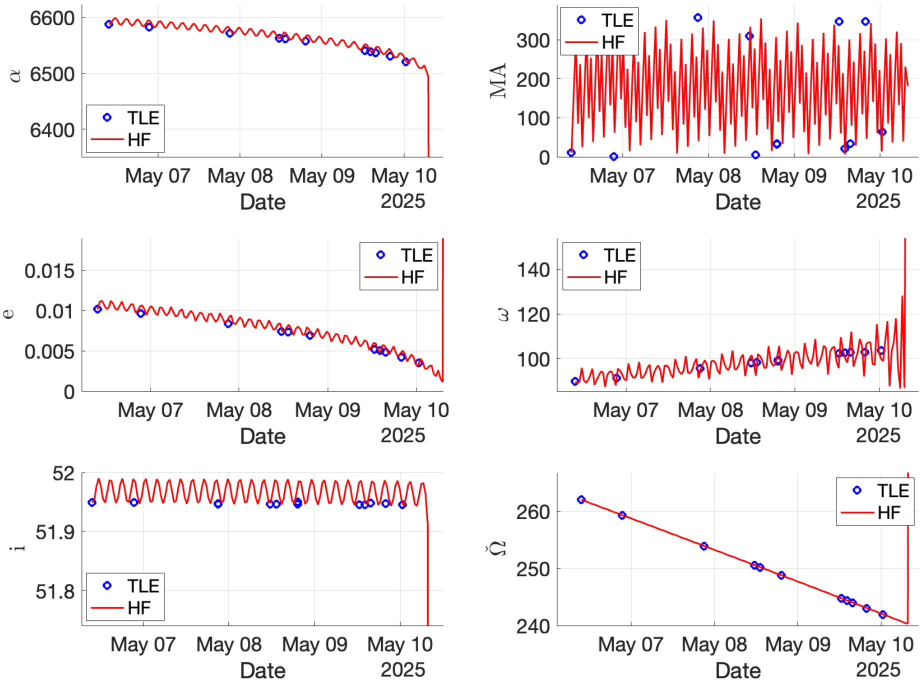


Figure 6.2: Comparison of the 6 orbital elements α, e, i, MA, ω and $\check{\Omega}$ retrieved from TLE data (in blue) and from a HF propagation of the initial conditions (in red).

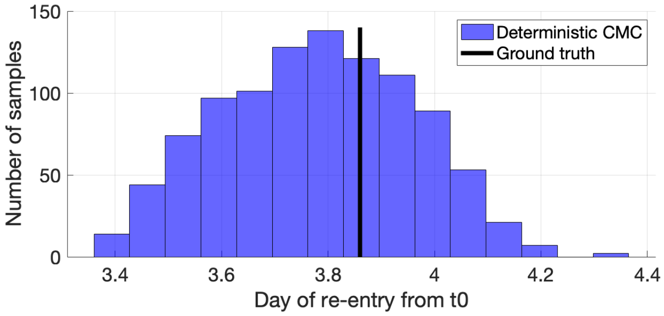


Figure 6.3: Histogram of re-entry times obtained by means of a HF deterministic crude Monte Carlo simulation of 1,000 samples (blue), as well as the real re-entry time (solid black line).

BIBLIOGRAPHY

[1] B. Turner, “Shattered russian satellite forces iss astronauts to take shelter in stricken starliner capsule,” *Live Science*, Jun. 2024. [Online]. Available: https://www.livescience.com/space/shattered-russian-satellite-forces-iss-astronauts-to-take-shelter-in-stricken-starliner-capsule.

[2] J. Hublin *et al.*, “New fossils from Jebel Irhoud, Morocco and the pan-African origin of Homo sapiens,” *Nature*, vol. 546, no. 7657, pp. 289–292, 2017. DOI: 10.1038/nature22336. [Online]. Available: https://doi.org/10.1038/nature22336.

[3] Space Debris Office, ESA, *ESA’s annual space environment report*, Issue 9, 2025. [Online]. Available: https://www.sdo.esoc.esa.int/environment_report/Space_Environment_Report_latest.pdf.

[4] Secure World Foundation, “2007 chinese anti-satellite test fact sheet,” Secure World Foundation, Tech. Rep., Nov. 2010. [Online]. Available: https://swfound.org/media/9550/chinese_asat_fact_sheet_updated_2012.pdf.

[5] S. Bugos, “Russian ASAT test creates massive debris,” *Arms Control Today*, Dec. 2021. [Online]. Available: https://www.armscontrol.org/act/2021-12/news/russian-asat-test-creates-massive-debris.

[6] Space.com, “SpaceX Starlink collision avoidance maneuver growth stalls,” *Space.com*, 2024. [Online]. Available: https://www.space.com/spacex-starlink-collision-avoidance-maneuver-growth-stalls.

[7] Kratos Defense, “SpaceX semi-annual update on Starlink network health, failure rate, collision risk,” *Kratos Defense Constellations*, 2024. [Online]. Available: https://www.kratosdefense.com/constellations/articles/spacex-semi-annual-update-on-starlink-network-health-failure-rate-collision-risk.

[8] D. J. Kessler, “The Kessler syndrome: Implications to future space operations,” *Advances in Astronautical Sciences*, no. 138, 2010.

[9] Y.-z. Luo and Z. Yang, “A review of uncertainty propagation in orbital mechanics,” *Progress in Aerospace Sciences*, vol. 89, pp. 23–39, 2017.

[10] Encyclopaedia Britannica, *Kepler's laws of planetary motion*, Accessed: 2025-03-28, 2021. [Online]. Available: https://www.britannica.com/science/Keplers-laws-of-planetary-motion.

[11] P. Fortesque, G. Swinerd, and J. Stark, *Spacecraft Systems Engineering*. Wiley, vol. 1.

[12] H. Curtis, *Orbital Mechanics for Engineering students*. 2005.

[13] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*, 4th. Microcosm Press and Springer, 2013.

[14] J. Emmert and J. Picone, "NRLMSISE 2.0: A whole-atmosphere empirical model of temperature and neutral species densities," *Earth and Space Science*, vol. 8, no. 3, 2020.

[15] A. Poore, J. Aristoff, and J. Horwood, *Covariance and uncertainty realism in space surveillance and tracking*, Air Force Space Command Astrodynamics Innovation Committee, 2016.

[16] J. G. Palacios, *Introduction to the theory of stochastic processes and brownian motion problems*, 2004.

[17] D. Williams, *Probability with Martingales*. Cambridge: Cambridge University Press, 1991.

[18] G. Casella and R. Berger, *Statistical Inference*, 2nd. Duxbury Press, 2002.

[19] W. E. Wiesel, *Modern Orbit Determination*. CreateSpace Independent Publishing Platform, 2010. [Online]. Available: https://books.google.com/books/about/Modern_Orbit_Determination.html?id=Q47CSgAACAAJ.

[20] M. Mahooti, "High precision orbit propagator," *MATLAB Central File Exchange*, 2024. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/55167-high-precision-orbit-propagator.

[21] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, 2nd. John Wiley & Sons, 2008.

[22] L. F. Shampine and M. W. Reichelt, *The matlab ode suite*, 1, 1997, pp. 1–22. DOI: 10.1137/S1064827594276424.

[23] F. R. Hoots and R. L. Roehrich, "Spacetrack report no. 3: Models for propagation of norad element sets," U.S. Air Force Aerospace Defense Command, Tech. Rep., Dec. 1980. [Online]. Available: https://celestrak.org/publications/AIAA/1980-0936/.

[24] D. A. Vallado, P. Crawford, R. Hujsak, and T. S. Kelso, "Revisiting spacetrack report #3," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, vol. AIAA 2006-6753, Aug. 2006. [Online]. Available: https://celestrak.org/publications/AIAA/2006-6753/.

[25] J. M. S. Perez-Vieites I. P. Mariño, “Probabilistic scheme for joint parameter estimation and state prediction in complex dynamical systems,” *Physical Review*, no. 98, 2018.

[26] W. Rümelin, “Numerical treatment of stochastic differential equations,” *SIAM Journal on Numerical Analysis*, vol. 19, no. 3, pp. 604–613, 1982. DOI: `10.1137/0719040`.

[27] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations* (Stochastic Modelling and Applied Probability). Berlin, Germany: Springer, 1992, vol. 23. DOI: `10.1007/978-3-662-12616-5`.

[28] S. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Prentice Hall, 1993.

[29] S. Alfano, “A numerical implementation of spherical object collision probability.,” *Journal of Astronautical Sciences*, vol. 53, no. 1, pp. 103–109, 2005.

[30] J. Foster and H. Estes, “A parametric analysis of orbital debris collision probability and maneuver rate for space vehicles,” *JSC 25898*, 1992.

[31] M. Akella and K. Alfriend, “Probability of collision between space objects,” *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 5, pp. 769–772, 2000. DOI: `https://doi.org/10.2514/2.4611`.

[32] R. Patera, “Satellite collision probability for nonlinear relative motion,” *Journal of Guidance, Control and Dynamics*, vol. 26, no. 5, 2003. DOI: `https://doi.org/10.2514/2.5127`.

[33] K. Chan, “Collision probability analyses for Earth-orbiting satellites,” *Proceedings of the 7th International Space Conference of Pacific Basin Societies, Nagasaki, Japan*, 1997.

[34] R. Garcıa-Pelayo and J. Hernando-Ayuso, “Series for collision probability in short-encounter model,” *Journal of Guidance, Control, and Dynamics*, pp. 1908–1916, 2016. [Online]. Available: `https://indico.esa.int/event/111/contributions/411/attachments/550/595/71_Garcia-Pelayo_Paper.pdf`.

[35] B. Jones and A. Doostan, “Satellite collision probability estimation using polynomial chaos expansions,” *Astrodynamics, Springer*, vol. 6, pp. 95–120, 2022.

[36] K. DeMars, Y. Cheng, and M. Jah, “Collision probability with Gaussian mixture orbit uncertainty,” *Journal of Guidance, Control and Dynamics*, vol. 38, no. 3, pp. 979–985, 2014. DOI: `https://doi.org/10.2514/1.62308`.

[37] V. Vittaldev and R. Russel, “Collision probability for space objects using Gaussian mixture models,” *Proceedings of the 23rd AAS/AIAA Space Flight Mechanics Meeting*, vol. 148, pp. 2339–2358, 2013. DOI: `https://doi.org/10.2514/1.G001610`.

[38] R. Patera, “Satellite collision probability for nonlinear relative motion,” *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 5, pp. 728–733, 2003.

[39] J. Carpenter, "Conservative analytical collision probability for design of orbital formations," *Astrodynamics, Springer*, 2004.

[40] C. Shelton and J. Junkins, "Probability of collision between space objects including model uncertainty," *Acta Astronautica*, vol. 155, pp. 462–471, 2019. DOI: https://doi.org/10.1016/j.actaastro.2018.11.051.

[41] M. Maestrini, A. D. Vittori†, P. D. Lizia‡, and C. Colombo, "Dynamics-based uncertainty propagation with low-thrust," *AAS/AIAA Astrodynamics Specialist Conference*, no. 634, 2022. [Online]. Available: https://www.researchgate.net/publication/363056209_Dynamics-Based_Uncertainty_Propagation_with_Low-Thrust.

[42] Y. Khatri and D. J. Scheeres, "Near-earth semi-analytical uncertainty propagation toolkit for conjunction analysis," *AMOS Conference*, 2023. [Online]. Available: https://www.researchgate.net/publication/375891514_Near-Earth_Semi-Analytical_Uncertainty_Propagation_Toolkit_for_Conjunction_Analysis.

[43] Y. Khatri and D. J. Scheeres, "Nonlinear semi-analytical uncertainty propagation for conjunction analysis," *Acta Astronautica*, no. 203, pp. 568–576, 2023. DOI: https://doi.org/10.1016/j.actaastro.2022.12.009.

[44] S. Alfano, "Review of conjunction probability methods for short-term encounters," *Journal of the Astronautical Sciences*, vol. 55, no. 2, pp. 193–226, 2007. DOI: 10.1007/BF03256497.

[45] J.-S. Li, Z. Yang, and Y.-Z. Luo, "A review of space-object collision probability computation methods," *Astrodynamics, Springer*, vol. 6, pp. 95–120, 2022. DOI: DOI:10.1007/s42064-021-0125-x.

[46] B. Jones and A. Doostan, "Satellite collision probability estimation using polynomial chaos expansions," *Advances in Space Research*, vol. 52, no. 11, pp. 1860–1875, 2013. DOI: https://doi.org/10.1016/j.asr.2013.08.027.

[47] S. Juneja and P. Shahabuddin, *Rare-Event Simulation Techniques: An Introduction and Recent Advances* (Handbook in OR). Elseiver, 2006, vol. 13, ch. 11.

[48] J. Morio, M. Balesdent, D. Jacquemart, and C. Vergé, "A survey of rare event simulation methods for static input–output models," *Simulation Modelling Practice and Theory*, vol. 49, pp. 287–304, 2014. DOI: https://doi.org/10.1016/j.simpat.2014.10.007.

[49] H. Farah and C. L. Azevedo, "Safety analysis of passing maneuvers using extreme value theory," *IATSS Research*, vol. 41, no. 1, pp. 12–21, 2017. DOI: https://doi.org/10.1016/j.iatssr.2016.07.001.

[50] F. Orsini, G. Gecchele, M. Gastaldi, and R. Rossi, "Collision prediction in roundabouts: A comparative study of extreme value theory approaches," *Transport Science*, vol. 15, no. 2, pp. 556–572, 2019. DOI: https://doi.org/10.1080/23249935.2018.1515271.

[51] H. Blom, J. Krystul, G. Bakker, M. Klompstra, and B. K. Obbink, "Free flight collision risk estimation by sequential MC simulation," *Stochastic Hybrid Systems*, pp. 249–281, 2007. DOI: DOI:10.1201/9781420008548.ch10.

[52] K. Youngjun and M. Kochenderfer, "Improving aircraft collision risk estimation using the cross-entropy method," *Journal of Air Transportation*, vol. 24, no. 2, 2016. DOI: https://doi.org/10.2514/1.D0020.

[53] M. Wang, Y. Wang, E. Cui, and X. Fu, "A novel multi-ship collision probability estimation method considering data-driven quantification of trajectory uncertainty," *Ocean Engineering*, vol. 272, 2023. DOI: https://doi.org/10.1016/j.oceaneng.2023.113825.

[54] R. Pastel, "Estimating satellite versus debris collision probabilities via the adaptive splitting technique," in *Proc. 3rd Int. Conf. Comp. Modeling Simulation*, 2011, pp. 1–6.

[55] C. Vergé, J. Morio, P. D. Moral, and J. C. D. Pérez, "Probabilistic safety analysis of the collision between space debris and satellite with an island particle algorithm," in *Space Engineering*, 2016, pp. 443–457. DOI: DOI:10.1007/978-3-319-41508-6_17.

[56] C. Vergé, J. Morio, and P. D. Moral, "An island particle algorithm for rare event analysis," *Journal of Reliability Engineering and System Safety*, vol. 149, pp. 63–75, 2015. DOI: https://doi.org/10.1016/j.ress.2015.11.017.

[57] N. Chopin, P. E. Jacob, and O. Papaspiliopoulos, "SMC2: An efficient algorithm for sequential analysis of state space models," *Journal of the Royal Statistical Society*, vol. Statistical Methodology B, 2012. DOI: https://doi.org/10.1111/j.1467-9868.2012.01046.x.

[58] M. Losacco *et al.*, "Advanced Monte Carlo sampling techniques for orbital conjunctions analysis and near Earth objects impact probability computation," *Proceedings 1st NEO and Debris Detection Conference, Darmstadt, Germany*, 2019. [Online]. Available: https://conference.sdo.esoc.esa.int/proceedings/neosst1/paper/482/NEOSST1-paper482.pdf.

[59] U. E. N. Garzon and E. G. Lightsey, "Mahalanobis shell sampling (MSS) method for collision probability computation," in *AIAA Scitech 2021 Forum*, 2021, pp. 1–25.

[60] B. Jones, A. Doostan, and G. Born, "Nonlinear propagation of orbit uncertainty using non-intrusive polynomial chaos," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 2, pp. 430–444, 2013.

[61] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods* (Monte Carlo Statistical Methods). Springer, 2004.

[62] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric, "Adaptive importance sampling: The past, the present, and the future," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 60–79, 2017.

[63] T. Kelso, "Analysis of the Iridium 33-Cosmos 2251 collision," *Advances in the Astronautical Sciences*, pp. 1099–1112, 2009.

[64] A. Gelb, *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.

[65] P. S. Maybeck, *Stochastic Models, Estimation, and Control, Volume 1*. New York, NY: Academic Press, 1979.

[66] G. E. Fraser, *The Estimation of Dynamic Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1969.

[67] P. Patil and S. K. T, "Orbit determination using batch sequential filter," *International Journal of Engineering Research and Technology*, vol. 1, no. 4, pp. 1–5, 2013.

[68] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, no. 82, pp. 35–45, 1960.

[69] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings F - Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993. DOI: `10.1049/ip-f-2.1993.0015`.

[70] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, 1972. DOI: `10.1109/TAC.1972.1100103`.

[71] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000. DOI: `10.1023/A:1008935410038`.

[72] B. D. O. Anderson and J. B. Moore, *Optimal Filtering* (Prentice-Hall Information and System Sciences Series). Englewood Cliffs, NJ: Prentice-Hall, 1979.

[73] S. J. Julier and J. K. Uhlmann, "The unscented Kalman filter for nonlinear estimation," *Proceedings of the 1995 IEEE Aerospace Conference*, vol. 6, pp. 3–9, 1995. DOI: `10.1109/AERO.1995.498946`.

[74] H. Arasaratnam and S. Haykin, "Cubature Kalman Filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[75] S. Segan, "Orbit determination and parameter estimation: Extended Kalman filter (EKF) versus least squares orbit determination," *Celestial Mechanics and Dynamical Astronomy*, vol. 129, pp. 345–368, 2017.

[76] A. Mashiku, "Statistical orbit determination using the particle filter for incorporating non-Gaussian uncertainties," *Acta Astronautica*, vol. 150, pp. 129–140, 2022.

[77] T. V. Ramanathan and R. A. Chipade, *Comparative evaluation of statistical orbit determination algorithms for short-term prediction of geostationary and geosynchronous satellite orbits in navic constellation*, https://arxiv.org/abs/2111.12348, Accessed: 2025-03-28, 2021.

[78] Q. Lam, D. Junker, and D. Vallado, "Analysis of an extended Kalman filter based orbit determination system," 2010.

[79] X. Sun, P. Chen, and C. Macabiau, "Low-Earth orbit determination from gravity gradient measurements," *Acta Astronautica*, vol. 123, pp. 350–362, 2016.

[80] Z. Li, "A novel fifth-degree cubature Kalman filter for real-time orbit determination by radar," *Aerospace Science and Technology*, vol. 89, pp. 12–20, 2021.

[81] P. C. Pardal, "An investigation on cubature Kalman filter performance for orbit determination application," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 2019.

[82] E. Wan and R. V. D. Merwe, "The unscented Kalman filter for nonlinear estimation," *IEEE Xplore*, no. 10, pp. 3460–3477, 2000.

[83] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice* (Statistics for Engineering and Information Science), A. Doucet, N. de Freitas, and N. Gordon, Eds. New York, NY: Springer, 2001. DOI: 10.1007/978-1-4757-3437-9.

[84] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003. DOI: 10.1109/MSP.2003.1233220.

[85] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, 2007.

[86] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear time series," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996. DOI: 10.1080/10618600.1996.10474713.

[87] S. Maskell, M. Briers, R. Wright, and P. Horridge, "Tracking using a radar and a problem specific proposal distribution in a particle filter," in *Proceedings of the Fifth International Conference on Information Fusion*, IEEE, vol. 2, 2002, pp. 867–872.

[88] T. Li, M. Bolic, and P. M. Djuric, "Resampling methods for particle filtering," *IEEE Signal Processing Magazine*, pp. 70–86, 2015.

[89] C. Snyder, T. Bengtsson, P. Bickel, and J. L. Anderson, "Obstacles to high-dimensional particle filtering," *Monthly Weather Review*, vol. 136, no. 12, pp. 4629–4640, 2008. DOI: 10.1175/2008MWR2529.1.

[90] P. C. P. M. Pardal, “The particle filter sample impoverishment problem in the orbit determination application,” *Aerospace Science and Technology*, vol. 104, pp. 20–32, 2021.

[91] J. S. McCabe, “Particle filter methods for space object tracking,” *Acta Astronautica*, vol. 125, pp. 50–62, 2016.

[92] H. Shen, R. Russel, and V. Vittaldev, “Parallelized sigma point and particle filters for navigation problems,” *Advances in Astronautical Sciences*, vol. 13, no. 034, 2013.

[93] S. Yun and R. Zanetti, “Kernel-based ensemble gaussian mixture filtering for orbit determination with sparse data,” *Advances in Space Research*, vol. Pre-Proofs, 2021.

[94] P. C. Pardal and H. Kuga, “The particle filter sample impoverishment problem in orbit determination applications,” *Mathematical Problems In Engineering*, vol. 2015, 2015.

[95] C. Barmann and U. Hugentobler, “A particle filter for orbit determination of space debris based on mono and multi-static laser ranging,” no. 16, p. 437, 2016.

[96] G. Escribano and M. S. Rivo, “Automatic maneuver detection and tracking of space objects in optical survey scenarios based on stochastic hybrid systems formulations,” *Advances in Space Research*, no. 69, pp. 3460–3477, 2022.

[97] G. Evensen, “The ensemble Kalman filter: Theoretical formulation and practical implementation,” *Ocean Dynamics*, vol. 53, no. 4, pp. 343–367, 1996. DOI: 10.1007/s10236-003-0036-9.

[98] E. Gamper, *Statistical orbit determination using ensemble kalman filter*, 2019.

[99] K. DeMars, “Probabilistic initial orbit determination using Gaussian mixture models,” *Celestial Mechanics and Dynamical Astronomy*, vol. 118, no. 3, pp. 171–184, 2014.

[100] S. Yun, “Kernel-based ensemble Gaussian mixture filtering for orbit determination with sparse data,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 2, pp. 845–858, 2019.

[101] D. Raihan and S. Chakravorty, “A UKF-PF based hybrid estimation scheme for space object tracking,” *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 8, pp. 1456–1470, 2021.

[102] Q. Meng and H. Leib, “Cubature ensemble Kalman filter for highly dimensional strongly nonlinear systems,” *IEEE*, vol. 6, 2016.

[103] L. Zhang, D. Sidoti, A. Bienkowski, K. R. Pattipati, Y. Bar-Shalom, and D. L. Kleinman, “On the identification of noise covariances and adaptive Kalman filtering: A new look at a 50 year-old problem,” *IEEE Access*, vol. 8, pp. 59 362–59 388, 2020.

[104] L. Ljung, *System Identification: Theory for the User*, 2nd. Upper Saddle River, NJ: Prentice Hall, 1999.

[105] J. Duník, O. Straka, O. Kost, and J. Havlík, "Noise covariance matrices in state-space models: A survey and comparison of estimation methods—part I," *International Journal of Adaptive Control and Signal Processing*, vol. 31, no. 11, pp. 1505–1543, 2017.

[106] W. G. on Covariance Realism, "Covariance and uncertainty realism in space surveillance and tracking," Astrodynamics Innovation Committee, Tech. Rep., 2016.

[107] N. Stacey, R. Furfaro, and E. J. Fossum, "Time-adaptive process noise estimation for orbit determination," *ArXiV*, 2020, Available at: `https://arxiv.org/abs/2001.03273`. eprint: `arXiv:2001.03273`.

[108] N. Stacey, R. Furfaro, and E. J. Fossum, "Adaptive and dynamically constrained process noise estimation for orbit determination," *ArXiV*, 2019, Available at: `https://arxiv.org/abs/1909.07921`. eprint: `arXiv:1909.07921`.

[109] A. Cano, A. Pastor, D. Escobar, J. Míguez, and M. Sanjurjo-Rivo, "Covariance determination for improving uncertainty realism in orbit determination and propagation," *Advances in Space Research*, vol. 72, pp. 2759–2777, 2023. DOI: `10.1016/j.asr.2023.06.010`.

[110] A. Cano, A. Pastor, S. Fernandez, J. Míguez, M. Sanjurjo-Rivo, and D. Escobar, "Improving orbital uncertainty realism through covariance determination in GEO," *The Journal of the Astronautical Sciences*, vol. 69, no. 5, pp. 1394–1420, 2022. DOI: `10.1007/s40295-022-00343-x`.

[111] M. Arulampalam and S. Maskell, "A tutorial on particle filter for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, 2002.

[112] G. Welch and G. Bishop, *An introduction to the Kalman filter*, `https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf`, Accessed: 2025-03-28, 2003.

[113] I. Arasaratnam, *Cubature Kalman filtering*, PhD Dissertation, 2009.

[114] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, p. 1254, 2009.

[115] N. Branchini and V. Elvira, "Optimized auxiliary particle filters: Adapting mixture proposals via convex optimization," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, ser. Proceedings of Machine Learning Research (PMLR), 2021.

[116] Ö. D. Akyildiz and J. Míguez, "Nudging the particle filter," *Statistics and Computing*, vol. 30, no. 2, pp. 305–330, 2020.

[117] R. Chen and J. S. Liu, "Mixture Kalman filters," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 62, no. 3, pp. 493–508, 2000.

[118] M. Zhu, P. J. Van Leeuwen, and J. Amezcua, “Implicit equal-weights particle filter,” *Quarterly Journal of the Royal Meteorological Society*, vol. 142, no. 698, pp. 1904–1919, 2016.

[119] M. Roth, G. Hendeby, C. Fritsche, and F. Gustafsson, “The ensemble Kalman filter: A signal processing perspective,” *EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 56, pp. 1–16, 2017. DOI: `10.1186/s13634-017-0492-2`.

[120] A. N. Bishop and P. D. Moral, “On the mathematical theory of ensemble (linear-Gaussian) Kalman–Bucy filtering,” *Mathematics of Control, Signals, and Systems*, vol. 35, pp. 1–83, 2023. DOI: `10.1007/s00498-023-00357-2`. [Online]. Available: `https://link.springer.com/article/10.1007/s00498-023-00357-2`.

[121] S. Pérez-Vieites, I. P. Mariño, and J. Míguez, “A probabilistic scheme for joint parameter estimation and state prediction in complex dynamical systems,” *Physical Review E*, vol. 98, no. 6, 2018. DOI: `10.1103/PhysRevE.98.063305`.

[122] V. Elvira and J. Miguez, “On the performance of particle filters with adaptive number of particles,” *Statistics and Computing*, vol. 31, no. 81, 2021.

[123] D. Crisan and J. Míguez, “Nested particle filters for online parameter estimation in discrete-time state-space Markov models,” *Bernoulli*, vol. 23, no. 4A, pp. 2672–2714, 2017.

[124] D. Crisan and J. Míguez, “Uniform convergence over time of a nested particle filtering scheme for recursive parameter estimation in state-space Markov models,” *Advances in Applied Probability*, vol. 49, no. 4, pp. 1148–1175, 2017.

[125] S. Pérez-Vieites and J. Míguez, “Nested Gaussian filters for recursive Bayesian inference and nonlinear tracking in state space models,” *Signal Processing*, vol. 189, Dec. 2021. DOI: `10.1016/j.sigpro.2021.108295`. [Online]. Available: `https://doi.org/10.1016/j.sigpro.2021.108295`.

[126] A. Petropoulos, “Refinements to the Q-law for the low-thrust orbit transfers,” *Jet Propulsion Laboratory, NASA*, 2005. DOI: `10.1016/j.asr.2023.06.010`.

[127] BBC, *Re-entry event coverage*, Accessed: February 2025, 2025. [Online]. Available: `https://www.bbc.com/news/articles/clyn9dgdwe3o`.

[128] N. Geul, S. Lemmens, and B. Bastida Virgili, “Analysis of uncertainties and modeling in short-term reentry predictions,” *Journal of Space Safety Engineering*, vol. 7, no. 3, pp. 215–222, 2020. DOI: `10.1016/j.jsse.2020.08.002`.

[129] CNN, *China rocket booster long march reentry*, Accessed: February 2025, 2022. [Online]. Available: `https://edition.cnn.com/2022/11/04/world/china-rocket-booster-long-march-reentry-scn/index.html#`.

[130] C. Pardini and L. Anselmo, “Assessing the risk and the uncertainty affecting the uncontrolled re-entry of manmade space objects,” *Acta Astronautica*, vol. 146, pp. 388–400, 2018. DOI: 10.1016/j.actaastro.2018.02.024.

[131] H. Klinkrad, “Methods and procedures for re-entry predictions at ESA,” in *Proceedings of the 4th European Conference on Space Debris*, 2005, pp. 643–650.

[132] E. J. Cho, J. H. Jo, and S. D. Park, “A study on re-entry predictions of uncontrolled space objects for space situational awareness,” *Journal of Astronomy and Space Sciences*, vol. 35, no. 1, pp. 33–42, 2018. DOI: 10.5140/JASS.2018.35.1.33.

[133] T. Lips, B. Fritsche, M. Homeister, G. Koppenwallner, H. Klinkrad, and M. Toussaint, “Re-entry risk assessment for launchers – development of the new scarab 3.1l,” in *Proceedings of the Second IAASS Conference*, ser. ESA SP-645, ESA Communication Production Office, 2007, pp. 2387–2396. [Online]. Available: https://www.fzt.haw-hamburg.de/pers/Scholz/ewade/2007/CEAS2007/papers2007/ceas-2007-360.pdf.

[134] V. Braun, J. Gelhaus, C. Kebschull, N. Sanchez-Ortiz, J. Oliveira, and H. Klinkrad, “DRAMA 2.0 – ESA’s space debris risk assessment and mitigation analysis tool suite,” in *Proceedings of the 64th International Astronautical Congress (IAC)*, Beijing, China, 2013, pp. 1–10.

[135] A. Ronse and E. Mooij, “Statistical impact prediction of decaying objects,” *Journal of Spacecraft and Rockets*, vol. 51, no. 6, pp. 1797–1810, 2014.

[136] C. Yanez, M. Gupta, V. Morand, and J. Dolado, “On the Gaussianity validity time for orbital uncertainty propagation,” in *Proceedings of the 1st NEO and Debris Detection Conference*, 2021. [Online]. Available: https://conference.sdo.esoc.esa.int/proceedings/neosst1/paper/408/NEOSST1-paper408.pdf.

[137] S. Bhattacharjee, R. S. Erwin, and M. J. Holzinger, “Understanding the effect of perturbations on the gaussianity of uncertainty propagation in orbital mechanics,” in *Proceedings of the 2018 AMOS Technical Conference*, 2018. [Online]. Available: https://amostech.com/TechnicalPapers/2018/Astrodynamics/Bhattacharjee.pdf.

[138] M. Surratt, J. Martin, and D. Lyons, “A novel method for prediction and warning for uncontrolled re-entry object impact,” in *Proceedings of the 8th International Association for the Advancement of Space Safety (IAASS) Conference*, 2016, pp. 345–354.

[139] L. Anselmo and C. Pardini, “Computational methods for reentry trajectories and risk assessment,” *Advances in Space Research*, vol. 62, no. 6, pp. 1266–1286, 2018. DOI: 10.1016/j.asr.2018.04.006.

[140] B. Peherstorfer and K. Willcox, “Multifidelity importance sampling,” *Computer Methods in Applied Mechanics and Engineering*, vol. 300, pp. 490–509, 2016.

[141] B. Peherstorfer, K. Willcox, and M. Gunzburger, “Optimal model management for multifidelity Monte Carlo estimation,” *SIAM Journal on Scientific Computing*, vol. 38, no. 5, A3163–A3194, 2016. DOI: 10.1137/15M1046472. [Online]. Available: https://kiwi.oden.utexas.edu/papers/multi-fidelity-monte-carlo-peherstorfer-willcox-gunzburger.pdf.

[142] P. Peherstorfer, B. Kramer, and K. Willcox, “Combining multiple surrogate models to accelerate failure probability estimation with expensive high-fidelity models,” *Journal of Computational Physics*, vol. 341, pp. 61–75, 2017. DOI: https://doi.org/10.1016/j.jcp.2017.04.012.

[143] B. Peherstorfer, K. Willcox, and M. Gunzburger, “Survey of multifidelity methods in uncertainty propagation, inference, and optimization,” *SIAM Review*, vol. 60, no. 3, pp. 489–776, 2018.

[144] A. Gruber, M. Gunzburger, L. Ju, R. Lan, and Z. Wang, “Multifidelity Monte Carlo estimation for efficient uncertainty quantification in climate-related modeling,” *Geoscientific Model Development*, vol. 16, pp. 1213–1232, 2023. DOI: 10.5194/gmd-16-1213-2023. [Online]. Available: https://gmd.copernicus.org/articles/16/1213/2023/.

[145] M. Mehana, A. Pachalieva, A. Kumar, J. Santos, and D. O’Malley, “Prediction and uncertainty quantification of shale well performance using multifidelity Monte Carlo,” *Journal of Petroleum Science and Engineering*, vol. 220, p. 111 127, 2023. DOI: 10.1016/j.petrol.2023.111127. [Online]. Available: https://doi.org/10.1016/j.petrol.2023.111127.

[146] A. Albà, A. Adelmann, and D. Rochman, “Uncertainty quantification of spent nuclear fuel with multifidelity Monte Carlo,” *Annals of Nuclear Energy*, vol. 211, p. 110 892, 2025. DOI: 10.1016/j.anucene.2023.110892. [Online]. Available: https://doi.org/10.1016/j.anucene.2023.110892.

[147] G. S. Page, *Roseycubesat-1*, 2025. [Online]. Available: https://www.nanosats.eu/sat/roseycubesat-1.

[148] E. Kulu, *Roseycubesat-1 spacecraft*, Accessed: February 2025, 2025. [Online]. Available: https://www.nanosats.eu/sat/roseycubesat-1.

[149] Space-track.org, *Decays/re-entries*, Space-track, Accessed: March 2025, 2024. [Online]. Available: https://www.space-track.org/#decay.

[150] V. P. Osweiler, “Covariance estimation and autocorrelation of NORAD two-line element sets,” Master’s thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 2006. [Online]. Available: https://scholar.afit.edu/etd/3531/.

[151] B. Bastida Virgili and H. Krag, “GOCE re-entry campaign,” *Acta Astronautica*, vol. 117, pp. 149–159, 2015. DOI: 10.1016/j.actaastro.2015.07.029.

[152] B. Peherstorfer, "Multifidelity Monte Carlo estimation with adaptive low-fidelity models," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 7, no. 2, pp. 648–671, 2019. DOI: 10.1137/17M1159208. [Online]. Available: https://epubs.siam.org/doi/10.1137/17M1159208.

[153] Z. Zhang, "Introduction to machine learning: K-nearest neighbors," *Annals of Translational Medicine*, vol. 4, no. 11, p. 218, 2016. DOI: 10.21037/atm.2016.03.37. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4916348/.

[154] M. K. Attouch and W. Bouabça, "The K-nearest neighbors estimation of the conditional mode for functional data," *Revue Roumaine de Mathématiques Pures et Appliquées*, vol. 58, no. 4, pp. 329–343, 2013. [Online]. Available: https://imar.ro/journals/Revue_Mathematique/pdfs/2013/4/2.pdf.