



School of Mechanical and Manufacturing Engineering

Faculty of Engineering

UNSW Sydney

BY

Stella Edmondston

**Optimal Filtering Algorithms For Orbit Determination In Near Rectilinear
Halo Orbit Applications**

Thesis submitted as a requirement for the degree of Bachelor of Engineering in
Aerospace Engineering

Submitted: 08/08/2025
Supervisor: Yang Yang (UNSW)

Student zID: z5363728

ORIGINALITY STATEMENT

'I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.'

Signed

X

Stella Edmondston
Student

Date 08/08/2025

Abstract

Orbit determination (OD) is concerned with accurately estimating a spacecraft's state; however, difficulties arise in the case of near rectilinear halo orbits (NRHOs) due to instability at perilune passage. In this study, three classical filters were compared in the NRHO context (extended (EKF), cubature (CKF) and unscented (UKF) Kalman filters) and hybrid variants were tested. Overall, the CKF/UKF outperformed in accuracy, the UKF was most consistent, the EKF was most efficient, and the hybrids achieved balance. These results varied depending on timestep. This suggests that hybrid filters may provide a balanced solution with minimal tradeoff between accuracy and efficiency.

Acknowledgements

I would like to thank my supervisor, Yang Yang, for his guidance and support throughout this project. This unique topic was both a challenging and rewarding learning experience.

Contents

Abstract.....	iii
Acknowledgements	iv
List of figures.....	vi
List of tables.....	viii
Nomenclature.....	ix
I. Introduction	1
II. Literature review.....	2
A. Orbit Determination Fundamentals.....	4
B. Challenges Regarding NRHO Applications	6
C. Filtering Approaches For NRHO OD	8
D. Filter Performance Comparison.....	12
E. Measurement/Navigation-Targeted Solutions	13
III. Methodology	16
A. Methodological Approach & Filter Selection.....	16
B. Filter Algorithm Implementation.....	18
C. Simulation Framework & Data Generation.....	22
D. Program Architecture & Implementation.....	25
E. Filter Hybridisation.....	28
F. Validation & Testing.....	30
IV. Results & Discussion	32
A. Results.....	32
B. Discussion	42
V. Conclusions	50
References.....	51
Appendices.....	54
A. Coordinate Transformations.....	54
B. Position Error Plots For All Runs	56

List of figures

Figure 1. Lunar Synodic Resonant NRHO in Three Coordinate Frames [1].....	2
Figure 2. Example Accuracy Degradation With Multiple Perilune Passages (UKF Performance, Mahalanobis Distance Metric) [5].....	3
Figure 3. Relationship Between Navigation, Reference & Truth Trajectories [6].....	4
Figure 4. Kepler's Second Law Visualised [9].....	6
Figure 5. Uncrewed & Crewed Mission Position Error & Uncertainty Using Simulated DSN Measurements [12]	8
Figure 6. Fitting Gaussian PDFs To Approximate A Non-Gaussian PDF [19].....	10
Figure 7. Filter RMSE For EKF, UKF, SRCKF-KF	11
Figure 8. Filter Ranking In Terms of Accuracy & Efficiency	12
Figure 9. Filter RMSE for Conventional Kalman Filter, EKF & UKF	13
Figure 10. Filtering Process	17
Figure 11. Elevation Angle Diagram [35]	24
Figure 12. Software Framework.....	26
Figure 13. Mission Trajectory, BCI (left), BCR (right).....	28
Figure 14. Hybridisation Algorithm	30
Figure 15. Example UKF RMSE State Only Debugging	31
Figure 16. EKF 3,000-Second Time Step Error Plots, Position (left), Velocity (right)	38
Figure 17. CKF 3,000-Second Time Step Error Plots, Position (left), Velocity (right)	39
Figure 18. UKF 3,000-Second Time Step Error Plots, Position (left), Velocity (right).....	39
Figure 19. EKF-CKF (Time Based) 3,000-Second Time Step Error Plots, Position (left), Velocity (right)	39
Figure 20. EKF-CKF (Entropy Based) 3,000-Second Time Step Position Error Plot	40
Figure 21. EKF-UKF (Time Based) 3,000-Second Time Step Position Error Plot.....	40
Figure 22. EKF-UKF (Entropy Based) 3,000-Second Time Step Position Error Plot	40
Figure 24. Overall Position RMSE For Every Filter vs. Time Step Magnitude	42
Figure 25. Mean RMSE (except 30 second time step) Against Time For Overall Comparative Results Between Filters	43
Figure 26. Comparative RMSE Throughout Orbit (30-second Time Steps Only).....	46
Figure 27. Computational Time Comparison Graph	47
Figure 28. Normal Distribution For Filters, Position RMSE (left), Computational Time (right)	49
Figure 29. EKF 30-Second Time Step Error Plots, Position.....	56
Figure 30. CKF 30-Second Time Step Error Plots, Position	56

S. Edmondston – Optimal Filtering Algorithms For Orbit Determination In Near Rectilinear Halo Orbit Applications

Figure 31. UKF 30-Second Time Step Error Plots, Position	57
Figure 32. EKF-CKF (time-based) 30-Second Time Step Error Plots, Position.....	57
Figure 33. EKF-CKF (entropy-based) 30-Second Time Step Error Plots, Position.....	58
Figure 34. EKF-UKF (time-based) 30-Second Time Step Error Plots, Position	58
Figure 35. EKF-UKF (entropy-based) 30-Second Time Step Error Plots, Position.....	59
Figure 36. EKF 300-Second Time Step Error Plots, Position.....	59
Figure 37. CKF 300-Second Time Step Error Plots, Position	60
Figure 38. UKF 300-Second Time Step Error Plots, Position	60
Figure 39. EKF-CKF (time-based) 300-Second Time Step Error Plots, Position.....	61
Figure 40. EKF-CKF (entropy-based) 300-Second Time Step Error Plots, Position.....	61
Figure 41. EKF-UKF (time-based) 300-Second Time Step Error Plots, Position.....	62
Figure 42. EKF-UKF (entropy-based) 300-Second Time Step Error Plots, Position.....	62
Figure 43. EKF 1,500-Second Time Step Error Plots, Position.....	63
Figure 44. CKF 1,500-Second Time Step Error Plots, Position.....	63
Figure 45. UKF 1,500-Second Time Step Error Plots, Position	64
Figure 46. EKF-CKF (time-based) 1,500-Second Time Step Error Plots, Position.....	64
Figure 47. EKF-CKF (entropy-based) 1,500-Second Time Step Error Plots, Position	65
Figure 48. EKF-UKF (time-based) 1,500-Second Time Step Error Plots, Position.....	65
Figure 49. EKF-UKF (entropy-based) 1,500-Second Time Step Error Plots, Position.....	66
Figure 50. EKF 3,000-Second Time Step Error Plots, Position.....	66
Figure 51. CKF 3,000-Second Time Step Error Plots, Position.....	67
Figure 52. UKF 3,000-Second Time Step Error Plots, Position	67
Figure 53. EKF-CKF (time-based) 3,000-Second Time Step Error Plots, Position.....	67
Figure 54. EKF-CKF (entropy-based) 3,000-Second Time Step Error Plots, Position	68
Figure 55. EKF-UKF (time-based) 3,000-Second Time Step Error Plots, Position.....	69
Figure 56. EKF-UKF (entropy-based) 3,000-Second Time Step Error Plots, Position.....	69

List of tables

Table 1. Key Differences Between Selected Filters	17
Table 2. UT Parameter Definitions & Study Settings	20
Table 3. DSN Simulation Random Error (3-standard deviations) [38]	25
Table 4. Standard Deviation (STD) Initialisation Settings For Rk, Qd, P0	25
Table 5. Unstable Orbit Phases & Corresponding Time Stamps	28
Table 6. Entropy Threshold Settings	33
Table 7. Computational Times, 30-Second Time Steps	33
Table 8. Overall RMSEs, 30-Second Time Steps	33
Table 9. Changing Position RMSE Throughout Mission, 30-Second Time Steps.....	34
Table 10. Computational Times, 300-Second Time Steps.....	34
Table 11. Overall RMSEs, 300-Second Time Steps	35
Table 12. Changing Position RMSE Throughout Mission, 300-Second Time Steps.....	35
Table 13. Computational Times, 1,500-Second Time Steps.....	36
Table 14. Overall RMSEs, 1,500-Second Time Steps.....	36
Table 15. Changing Position RMSE Throughout Mission, 1,500-Second Time Steps.....	36
Table 16. Computational Times, 3,000-Second Time Steps.....	37
Table 17. Overall RMSEs, 3,000-Second Time Steps.....	37
Table 18. Changing Position RMSE Throughout Mission, 3,000-Second Time Steps.....	38
Table 19. Monte Carlo Simulation Results, Position RMSE	41
Table 20. Monte Carlo Simulation Results, Computational Time	41

Nomenclature

D	= diagonal matrix
\overrightarrow{GB}	= ground station-to-barycentre vector
H	= entropy
H_k	= measurement mapping matrix
k	= newest state, where $t = k$
$k - 1$	= previous state, $t = k - 1$
K_k	= Kalman gain
L	= state dimension
P	= covariance matrix
P_{k-1}	= previous covariance, at $t = k - 1$
P_k	= navigation covariance estimate after measurement update, at $t = k$
\bar{P}_k	= reference covariance estimate (before measurement update), at $t = k$
P_{xy}	= cross-correlation
P_{yy}	= innovation covariance
Q_d	= process noise matrix
R_k	= measurement noise matrix
r	= range
\dot{r}	= range rate
\overrightarrow{SB}	= spacecraft-to-barycentre vector
\overrightarrow{SG}	= spacecraft-to-ground station vector
t	= time
t_{k-1}	= previous time, $t = k - 1$
t_k	= most recent time, $t = k$
U	= upper triangular matrix
W^c	= weight matrix for covariance corresponding to sigma points
W^m	= weight matrix for sigma points
X	= state vector, six components, position and velocity
X_k	= navigation state estimate after measurement update, at $t = k$
\bar{X}_k	= reference state estimate (before measurement update), at $t = k$
X_{k-1}	= previous state, at $t = k - 1$
x_k	= state difference between \bar{X}_k and X_k

S. Edmondston – Optimal Filtering Algorithms For Orbit Determination In Near Rectilinear Halo Orbit Applications

Y_k	= received measurement, at $t = k$
\bar{Y}_k	= reference measurement estimate, at $t = k$
y_k	= measurement innovation
α	= parameter determining spread of sigma points
β	= parameter to set up PDF assumptions
γ	= scaling factor for sigma points
κ	= secondary scaling parameter for UT
λ	= parameter for determining γ
ξ	= spread parameter for CKF
Υ_k	= measurements estimated for each sigma point
Φ	= state transition matrix
χ_k	= propagated sigma points to $t = k$
χ_{k-1}	= sigma points generated around the last state estimate, at $t = k - 1$
BCI	= Earth-Moon barycentre-centred inertial frame
BCR	= Earth-Moon barycentre-centred rotating frame
CKF	= cubature Kalman filter
CR3BP	= circular restricted 3-body problem
DMC	= dynamic model compensation
DRO	= distant retrograde orbit
DSN	= Deep Space Network
ECI	= Earth-centred inertial frame
ECEF	= Earth-centred Earth-fixed inertial frame
EKF	= extended Kalman filter
GMM	= Gaussian mixture model
HALO	= High-fidelity Analyser of Lunar Orbits
LiAISON	= Linked Autonomous Interplanetary Satellite Orbit Navigation
LOS	= line of sight
L1	= Lagrange point 1
L2	= Lagrange point 2
ML	= machine learning
NRHO	= near rectilinear halo orbit
OD	= orbit determination
OMM	= orbit maintenance manoeuvre

S. Edmondston – Optimal Filtering Algorithms For Orbit Determination In Near Rectilinear Halo Orbit Applications

PDF	= probability density function
PP	= perilune passage
RMSE	= root mean square error
SNC	= static noise compensation
SRCKF	= square root cubature Kalman filter
SRCKF – KF	= square root cubature Kalman filter-Kalman filter
SRIF	= square root information filter
SRP	= solar radiation perturbations
SST	= satellite-to-satellite tracking
STD	= standard deviation
STM	= state transition matrix
UDKF	= U-D factorisation Kalman filter
UKF	= unscented Kalman filter
USKF	= unscented Schmidt-Kalman filter
3BP	= 3 body problem

I. Introduction

Orbit determination (OD) refers to the estimation of the position and velocity (or ‘state’) of a spacecraft in orbit. This process takes a ‘reference’ trajectory, the path the spacecraft is supposed to follow, and compares this to the measurements it receives to form a ‘navigation’ trajectory, and arrive at the best approximation of the truth state. For this study, OD for near-rectilinear halo orbits (NRHOs) are the focus, in particular, a 9:2 lunar synodic resonance orbit about the Earth-Moon Lagrange point 2 (L2). This orbit has become of significant interest to the aerospace community in recent years with the renewed focus on cislunar missions, such as for Artemis and the Gateway project. This is due to several key advantages, but most importantly the near-continuous line-of-sight (LOS) with Earth and the close lunar passage (perilune). Continuous LOS would allow for satellite-to-satellite tracking networks, as increased cislunar traffic would need to be monitored to prevent collisions and manage when orbit maintenance manoeuvres should be conducted. The close perilune passages produce good visibility of the lunar surface and high velocities that provide increased manoeuvrability options, enabling for more ambitious transfers to other orbits, including interplanetary travel, at a fraction of the propulsive cost.

The key focus for this project is to determine the most suitable OD solutions at various time-intervals between measurements. In this case three filters were implemented; extended Kalman filter (EKF), cubature Kalman filter (CKF) and the unscented Kalman filter (UKF), to compare their relative performance on two metrics, accuracy and efficiency, measured via root-mean square error (RMSE) and computational time. Additionally, hybridised filters were created, combining the EKF (for the more stable parts of the orbit) with the CKF/UKF for perilune passage. This is due to the unique structure of NRHOs, as the orbit family is largely stable throughout the orbit, with the linearisation assumptions the EKF utilises being sufficient until perilune passage, where the volatile speeds result in highly non-linear dynamics, thus the need for more complex filters. Thus, the need for a comparative study arises, as classical methods of OD fall short in accuracy and efficiency. Determining a suitable balance is crucial for future cislunar navigation success, as failure to do so could cause catastrophe for these cislunar missions, with ill-informed manoeuvres resulting in mission path divergence.

This study is organised into three sections, a literature review, methodology, and results and discussion. The literature review will provide greater detail about OD and the challenges regarding NRHO, as well as current solutions and research gaps that may be filled with newer technologies and approaches, including the filter hybridisation. The methodology will deconstruct the filter implementation program and the generation of results, whereas the results and discussion section will present these results and their

subsequent findings, also discussing which of the filters tested performed the best and why. The conclusion will summarise these findings and provide a recommendation on future work.

II. Literature review

NRHOs have, in recent years, received a renewed interest from the aerospace community, with the increased development of lunar and interplanetary missions, namely Artemis and Gateway. This is because NRHOs exhibit highly favourable characteristics, that enhance mission capabilities. These orbits, exhibiting a ‘near-halo’ shape when viewed in an Earth-Moon barycentric rotating (BRF) coordinate system (as displayed in *Figure 1* below), have two key attributes that make them so advantageous. The first is close passage to the lunar surface, allowing for increased observability, with the second, also due to this close passage, being extreme dynamics near perilune. These dynamics allow for increased possibilities for the manoeuvrability of the spacecraft with a fraction of the propulsive cost, enabling manoeuvres as extreme as orbit transfers for interplanetary travel. This attribute is particularly beneficial, as a reduction in propulsive costs can lead to significant weight savings, and hence considerable decreases in operational costs.

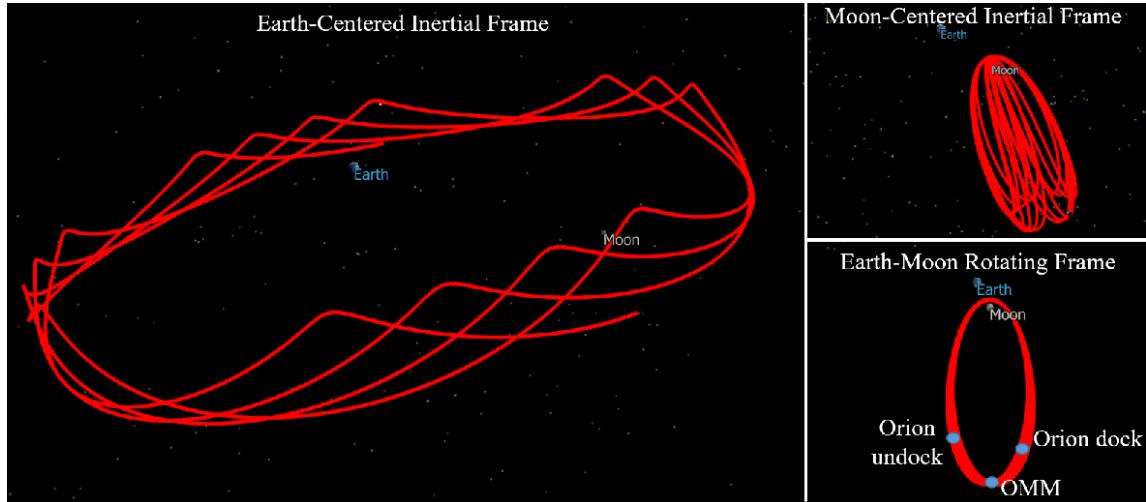


Figure 1. Lunar Synodic Resonant NRHO in Three Coordinate Frames [1]

Of the NRHO family, there is one in particular that provides even greater benefits. This is the NRHO around the Earth-Moon Lagrange point 2 (L2) [2] in the southern family, with a 9:2 lunar resonance. This lunar resonance is particularly important, as by aligning the period of the orbit with the orbital period of the Moon around Earth, lunar eclipses between the spacecraft and the Earth are nearly entirely avoided [3], greatly

improving observability. This is highly important for NRHOs in particular, as OD accuracy greatly relies on accessibility to frequent measurements and data-rich environments to maintain accurate results.

Although the majority of this NRHO is stable, when it reaches its most beneficial characteristic, the perilune (i.e. closest to the Moon) the extreme dynamics caused by such a close perilune passage cause the orbit to become unstable, with the velocity of the spacecraft rapidly increasing. These dynamics are further complicated by environmental perturbations that are extremely difficult to model for, including solar radiation pressure (SRP), variable gravity fields (significant variations in the Moon's case, known as mass concentrations or ‘mascons’ [4], general relativity, three-body problem (3BP) perturbations, Earth albedo and propulsive manoeuvres. This means that classical OD solutions, which rely on linearisation assumptions to simplify the state estimation process, and hence produce timely results, are no longer adequate because the assumptions are no longer valid. Additionally, this failure is not limited to the perilune passage, as the classical OD filters struggle to re-converge with the true trajectory, resulting in significant state accuracy degradation with each passage, as demonstrated below in *Figure 2*.

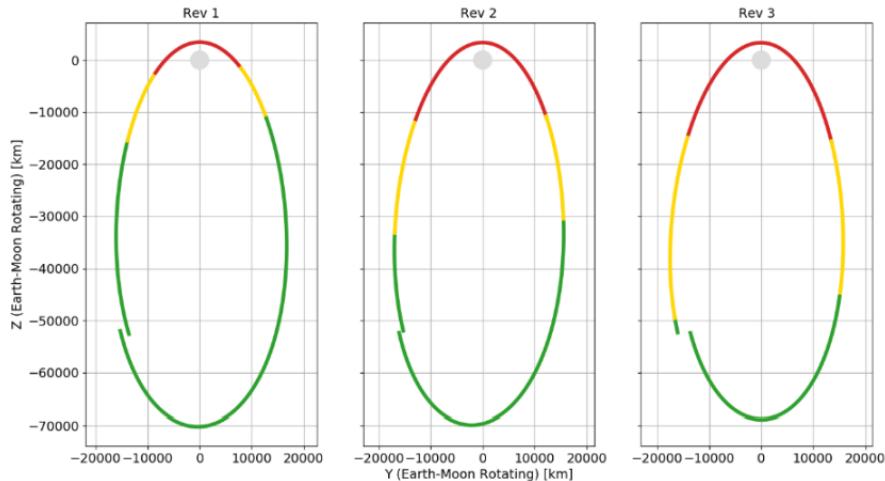


Figure 2. Example Accuracy Degradation With Multiple Perilune Passages (UKF Performance, Mahalanobis Distance Metric) [5]

Another significant contributor to poor OD performance with this NRHO is regarding the motion of the spacecraft. Throughout the orbit, the spacecraft travels perpendicular to the LOS along the Earth-spacecraft line. This creates uncertainty with the noisiness of the measurements (range for position, range rate for velocity) determined by the ground-stations on Earth. As the magnitude of noise attached to these measurements cannot be determined, the OD filter has no capability of differentiating between clean and

accurate measurements and noisy measurements, leading to inaccurate weighting between the two and thus overall decreased filter accuracy. Thus, the need for more complex OD solutions arises.

A. Orbit Determination Fundamentals

The main purpose of OD is to estimate the state (i.e. position, velocity) of the spacecraft as it progresses throughout its orbit. Successful OD means that the state of the spacecraft is known to a high degree, and is highly important in missions especially regarding NRHOs, as due to their volatile dynamics. What makes these orbits so beneficial, i.e. increased manoeuvrability capability at a decreased propulsive cost, also makes them highly risky. An ill-informed manoeuvre at perilune could cause the satellite to completely diverge from the desired path and jeopardise the mission. Such failings are even harder to overcome due to the inherent risks of operating in space, as spacecraft are often designed to carry exactly what they need with only a small margin of error, not recoverable from any significant divergence.

When discussing OD, there are three trajectories. These trajectories and the relationships between them form the backbone of OD theory. These are:

- The reference trajectory which is the trajectory that the spacecraft follows as dictated by the mission and theoretical state propagation models.
- The navigation trajectory which is the trajectory that the spacecraft believes it is on as informed by measurements and the OD filter.
- The truth trajectory, i.e. the trajectory the satellite is truly following.

These relationships are visualised in *Figure 3* below. The aim of OD is to approximate the truth trajectory as closely as possible.

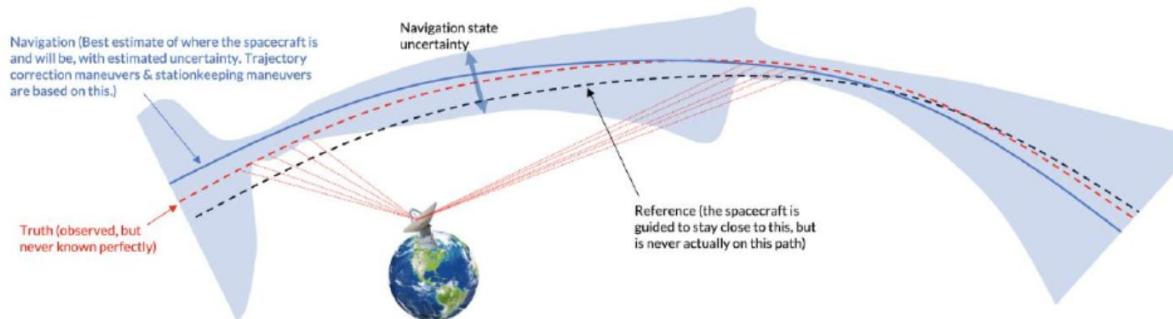


Figure 3. Relationship Between Navigation, Reference & Truth Trajectories [6]

Essentially, the OD filters use orbit models to anticipate where the spacecraft will be at the next time step i.e. ‘in x amount of time’. This is known as the predictor stage, and it forms the reference trajectory, with the corresponding state estimate defined as \bar{X}_k . The results of this estimation are then compared to the measurements received from ground stations (where their true state is known), and the difference between these measurements and the anticipated measurements (known as the innovation, x_k) is used to update the filter and inform the algorithm by how much it can trust its orbit dynamics modelling to anticipate the next state. This is the corrector step. Together, they determine the full estimated state X_k as presented below [7].

$$X_k = \bar{X}_k + x_k$$

There are several different types of OD filters. These can generally be separated into filters that rely on linearisation assumptions and those that do not to drive the estimation process. Classical OD for most orbits generally rely on linearisation assumptions, and include conventional Kalman filters, most frequently utilising the EKF. These Kalman filters rely on linearisation to simplify the orbit calculations to create the reference trajectory. If this reference trajectory still remains close to the truth trajectory the filter will be successful, and both perform accurately and efficiently [7]. Achieving both of these aspects is crucial for mission operation, as anticipated states inform manoeuvres, and the execution of these manoeuvres is often highly time-sensitive.

However, these orbits, especially with unstable orbits such as NRHO, require complex modelling for the 3BP dynamics, as well as all the aforementioned environmental perturbations that further complicate the estimation process. This is where simple Kalman filters often fall short, as the linearisation assumptions are simply invalidated when the spacecraft reaches perilune and the filter can not account for the complexity of the dynamics.

For NRHO, Bayesian filters must be considered. These filters do not rely on linearisation, and generally greatly improve the accuracy of the filters as linearisation does not simply important state information. Filter uncertainty is updated via the formation of the spacecraft’s state probability density function (PDF), which with every measurement is updated, and then the covariance matrix (to directly determine the state uncertainty) is computed from the PDF. The PDF computation relies on Bayes’ theorem of conditional probability [7].

The extent to which the Bayesian filter is accurate is determined by the kind of PDF that is formed, with the most accurate forming non-Gaussian PDFs to match the non-linear dynamics, however at great

computational cost. These include particle filters and Gaussian mixture models (GMMs), however, those that simply rely on Gaussian PDFs such as the CKF and UKF are most often used for aerospace applications as they capture the probability without being extremely computationally intensive [7]. The differences between these filters will be discussed in C. Filtering Approaches For NRHO OD.

B. Challenges Regarding NRHO Applications

Non-Linear Dynamics at Perilune

The non-linear dynamics encountered at perilune in the NRHO is the most significant challenge to accurate OD. For the majority of the orbit, the NRHO is highly stable, with linearisation assumptions being adequate for OD until the perilune is reached. However, due to the proximity to the lunar surface during the perilune passage, the spacecraft velocity rapidly increases. This is in accordance with Kepler's second law [8], the law of motion, with the equal areas of the orbit being covered in equal times, thus, with increasingly closer perilune passages, the velocities increase significantly. This is visualised below in *Figure 4*. Due to this rapid change in velocity, the linearisation assumptions are invalidated, and are no longer adequate in anticipating the spacecraft's state.

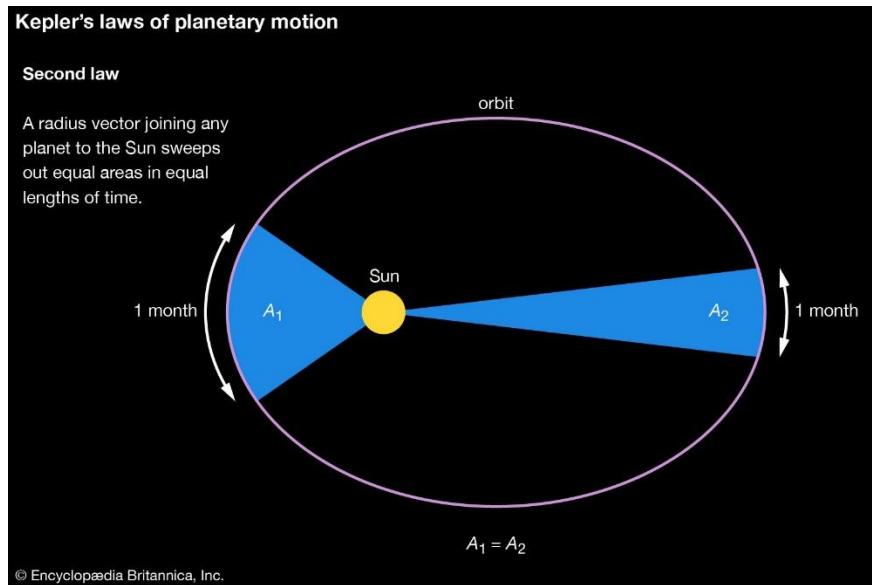


Figure 4. Kepler's Second Law Visualised [9]

Measurement & Range Bias Issues

Another significant issue regards how the spacecraft receives measurements and how systematic biases are estimated to anticipate interference with the accuracy of the OD filter. For the NRHO, and at the time of

writing, it is assumed that Gateway will rely on communication with Deep Space Network (DSN) ground stations. As aforementioned, this means that spacecraft's motion will often be perpendicular to the LOS along the Earth-spacecraft line, meaning that any systematic bias from the range measurements received are impossible to determine. This prevents the filter from being tuned to anticipate the error this bias introduces. Thus, uncertainty models cannot be accurately updated [6], leading to convergence on incorrect trajectories as the filter trusts the measurements more than it should. Additionally, to achieve accurate results and mitigate these issues [6], the filter requires data-rich environments, with frequent measurements to update the filter with. This, for the NRHO of focus, is not an issue, with the 9:2 synodic resonance allows for nearly constant contact, however, LOS disruptions may still occur, with the occasional lunar blockage or even blockages from other spacecraft, specifically in the case proposed Gateway missions likely performing docking manoeuvres with other spacecraft such as Orion. Thus, finding a filter that can successfully estimate the state despite the range bias difficulties and can simultaneously recover quickly from potential data outages is crucial to finding an OD solution for the NRHO.

Environmental & Mission Perturbations

The final challenge to NRHO OD is universal to OD, as no orbit is executed in a vacuum. As such, all spacecraft are subject to additional forces that are not possible to truly model due to their random nature. With the NRHO being classified as a 3BP orbit, with the two primary celestial bodies being the Earth and the Moon, as these bodies exist within the Solar System, several other celestial bodies exert additional gravitational forces upon the spacecraft. The most significant, due to either their mass and/or proximity to Earth, are the Sun, and the planets Venus and Jupiter. As their proximity is not great enough to the system to have extreme effects on the orbital dynamics, it is often sufficient to model their perturbations as point masses in the system [10]. However, this is not the case for the two main bodies, specifically the Moon, as the lunar gravitational field is not uniform, and in fact has significant variations due to the briefly aforementioned ‘mascons’. These are resulting from the Moon’s lack of atmosphere, and therefore lack of protection against incoming meteors, which upon impact increase the density or concentrate parts of the lunar surface. As gravitational force is directly proportional to the density of an object, the gravity towards these mascons increases [4]. The modelling of these mascons and the resultant gravitational field as reliant on field data collected, and as such are often modelled via complex harmonics [10].

Besides gravitational forces, the other main environmental perturbations come from the Sun in the form of SRP, i.e. electromagnetic radiation produced by the Sun. This may act directly on the spacecraft or additionally in the form of Earth albedo, which is SRP reflected by the clouds and the Earth’s atmosphere, as well as snow and ice on the Earth’s surface [11].

Unfortunately, there are more significant issues regarding perturbations introduced by the spacecraft's operation itself. These include navigation errors, orbit insertion errors and orbit maintenance manoeuvres (OMMs) and their associated execution errors. For a basic NRHO these are often minimal, with the occasional OMM to ensure the NRHO is maintained and the near-constant LOS is not lost. However, for more complicated missions, requiring significant manoeuvres or even crewed operation there are a significant number of increased perturbations to take into consideration. These include docking with other spacecraft (likely in the case for Orion) and attitude control system desaturations during docking, reaction control system slewing, and venting activities including wastewater dumps and carbon dioxide puffs. These perturbations in particular, due to their sudden nature, lead to large spikes in error for classical OD filters, leading to increased overall position error as these perturbations cause divergence, and filters may struggle to recover from these perturbations [1]. The difference is demonstrated below in *Figure 5*, showing the significant difference between crewed and uncrewed mission error.

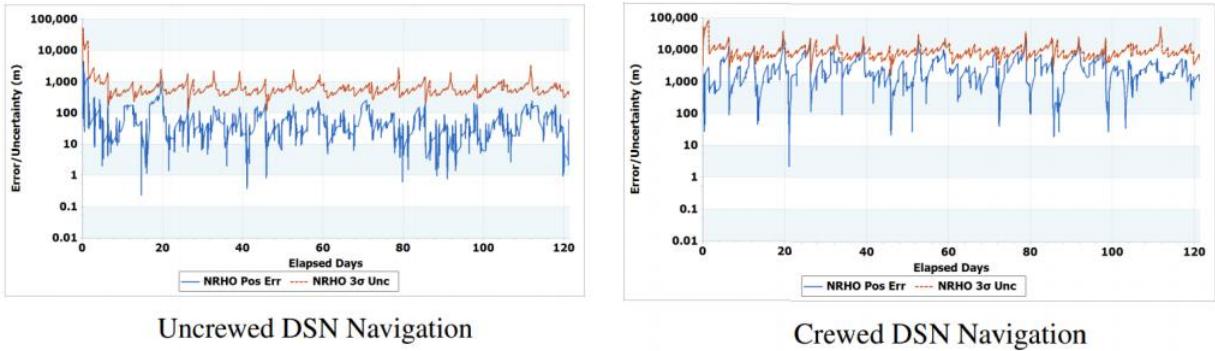


Figure 5. Uncrewed & Crewed Mission Position Error & Uncertainty Using Simulated DSN Measurements [12]

C. Filtering Approaches For NRHO OD

Linearisation-Based Filters

Linearisation-based filters are the most commonly used filters for OD, being fast, efficient, and for classical OD problems highly effective due to their high accuracy-to-efficiency ratio. Conventional Kalman filters generally fail in all OD applications due to their tendency to stop accepting new measurement contributions to the filter due to their insignificant weighting, relying on the linearised dynamics of the whole trajectory and resulting in complete filter divergence.

EKFs, however, mitigate this issue by recalibrating the resultant reference trajectory continuously. This essentially means that the system is linearised with every timestep via Jacobian matrices rather than

linearising the non-linear equations of state via Taylor series. This allows for the state transition matrix (STM) to adapt to the changes in non-linear dynamics more accurately whilst still maintaining the linearisation assumptions and thus the increased efficiency that such simplification allows [13]. EKFs are the most commonly used filters in OD, as they have demonstrated good performance in real-world applications [14]. However, the applicability of an EKF is largely dependent on the mission case, as the validity of the linearisation, although localised and not of the entire trajectory, is subject to how great the magnitude of uncertainty is and how severe the changes in dynamics are throughout the orbit [15]. Hence, it has not been shown to be a viable solution to the NRHO problem due to the extreme dynamics at perilune, with results showing that the linearisation assumptions, due to their invalidity, cannot satisfy the nature of the orbit during this passage and thus the EKF can no longer mitigate the trajectory divergence. These filters also fail to accommodate non-Gaussian perturbations as previously outlined, and thus cannot update the state uncertainty accurately.

Other alternatives to the EKF include the square root information filter (SRIF) and the U-D factorisation Kalman filter (UDKF). These filters are also both common solution in classical OD, with both similar to the EKF, however, altering the handling of the covariance matrices. The key difference between the EKF and the SRIF is that the SRIF propagates the inverse of the covariance/uncertainty matrix ‘P’ (i.e. the square root of the information matrix) rather than the full covariance matrix to predict the state [16]. This key difference greatly increases the stability of the filter, and hence efficiency, making them highly reliable. This reliability is highly favourable in OD, as successful OD is crucial for mission success. Similarly, the UDKF differs in the propagation of the covariance matrix, however rather than directly updating the full covariance or inverse covariance, it operates on upper triangular matrix ‘U’ and the diagonal matrix ‘D’, factors of the covariance matrix. This factorisation is presented below.

$$P = UDU^T$$

Similar to the SRIF, the UDKF offers efficient and numerically stable results, with the factorisation guaranteeing symmetrical and positively definite covariance matrices, despite filter size. Additionally, triangular and diagonal matrices are significantly less computationally demanding, further increasing filter efficiency [17]. Both of these filters however, due to their similar nature to the EKF, still fail due to similar shortcomings, i.e. the invalidation of the linearisation assumptions made close to perilune.

Non-Linear/Bayesian Filters

Bayesian filters are the direct solution to the breakdown in linearisation assumptions, completely erasing them altogether. This results in increased accuracy, as the non-linear dynamics are modelled as non-linear, reducing the amount of information lost, especially at perilune passage in NRHO. This however results in drastic increases in computational time, hence why they are not often used in classical OD. As previously mentioned, these are statistical filters that rely on laws of conditional probability, accommodating for changing uncertainty as time progresses and accounting for unmodelled noise and perturbations in the covariance calculation.

The most advanced of these filters are particle filters. These filters capture the full non-Gaussian PDF by sampling the current state of the system and propagating a large number of samples/particles to the next time step with full non-linear modelling [18]. Each result is initially weighted evenly, and as reweighted in accordance with Bayes' theorem, correcting and constructing the PDF such that the state at any time throughout the propagation can be estimated. These PDFs are highly accurate; however, to produce such a PDF requires millions of samples to be propagated. This is far too computationally costly and hence time consuming, and thus is not viable in OD applications.

GMMs closely follow the particle filter methodology, however, rather than propagating an extreme number of samples, they approximate an arbitrary non-Gaussian uncertainty PDF and update with each propagation. With each propagation the PDF is split into Gaussian components to better reflect a non-Gaussian distribution, hence the term ‘Gaussian mixture’. The occurrence of these splits are determined by entropy criteria, a metric by which the covariance can be tested for non-linearity. The entropy criteria is defined by the user, with lower thresholds producing higher accuracy but more computationally expensive modelling [7]. Thus the overall efficiency is set by the user. This process is visualised below in *Figure 6*.

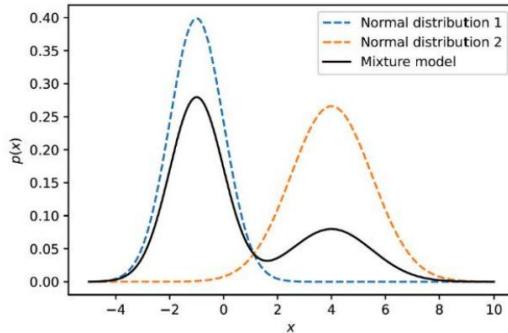


Figure 6. Fitting Gaussian PDFs To Approximate A Non-Gaussian PDF [19]

CKFs and UKFs are also similar to particle filters, however, like GMMs, have a significant advantage over them by requiring a vastly reduced number of points to generate a Gaussian PDF, with the statistical moments of the PDF used to ensure that the one generated still relatively accurately represents the true non-Gaussian PDF. These points are referred to as sigma points in the case of UKF, and cubature points for CKF, with the difference being how these points are selected. Like the particle filter, each point is propagated via non-linear models, constructing the PDF via an unscented transform (UT). This UT is used for both the predictor and corrector steps of the UKF [20], with the mean state being estimated for the predictor and then the mean expected measurement along with its error-covariance. Thus, with this PDF, the mean state and covariance is estimated. CKFs possess extremely similar properties to UKFs, with the key difference being how the samples are selected, with the CKF relying on the cubature rule, using Gaussian quadrature to define the points [21]. This removes the need for parameter tuning, with the points being equally weighted, thus resulting in a more efficient filter. However, similar to most filters, as this efficiency is increased, more information is lost and the filter is comparatively less accurate.

Hybrid Filters & Other Advanced Approaches

Many have explored additional approaches to OD beyond applying a single filter. In [22], a study was performed comparing three filters: EKF, UKF and a hybrid filter, specifically ‘square-root CKF-Kalman filter’ (SRCKF-KF). The SRCKF combines the SRIF with the CKF to update error covariance and prevent filter divergence. This study was performed for general vehicle positioning applications, however, as non-linear dynamics are the primary concern of this study, like NRHO, the results are in theory translatable. Essentially, the SRCKF-KF was developed such that the conventional Kalman filter would be used for periods of operation where linearisation assumptions would maintain validity, and the SRCKF for when they were invalidated. The crucial findings for this study are presented below, with the overall RMSE decreasing significantly with the use of the hybrid filter relative the both the EKF and the CKF, demonstrating an overall higher accuracy. These results are presented below in *Figure 7*.

Method	RMSE	Time(s)
EKF	2.8501	0.0057
UKF	2.2405	0.0253
SRCKF-KF	1.9350	0.0152

Figure 7. Filter RMSE For EKF, UKF, SRCKF-KF

Another approach is to try to account for the uncertainty by modelling it. This approach has been executed via the unscented Schmidt-Kalman filter (USKF), a variant of UKF that incorporates ‘consider’ parameters,

parameters not directly measured but likely to influence uncertainty calculations such as the environmental perturbations previously outlined [23]. Consider covariance analysis is incorporated into the corrector step before the final state estimate, and is calculated completely separately to the rest of the covariance. As it estimates the covariance of what may not be directly measureable, it may provide a solution to the unknown range bias problem regarding NRHO from the LOS issues.

Finally, of the unconventional approaches, is machine learning (ML). This approach simulates the dynamics from real-life data to propagate the state. This has been explored in OD in [24] for the purpose of debris tracking, with the purpose to reduce computational demands while improving accuracy simultaneously. ML was found to succeed in achieving this, especially in real-time scenarios, as ML continuously improves upon its modelling with new data. The key reason this is not currently viable for NRHO in particular is due to the lack of high quality datasets currently available, as the machines need to be trained. However, this is an avenue worth exploring at a later time.

D. Filter Performance Comparison

Of the filters outlined above, their benefits and limitations are largely compared via two metrics: accuracy (“how close to the truth is the estimate”) and efficiency (“how quickly can this result be arrived at”). These metrics are key in selecting which filters are adequate and which are unsuitable for any OD application, including in an NRHO context. For example, clearly, the particle filter provides the best results in terms of accuracy; however, by the time the result is obtained, the state of the spacecraft has already changed. On the other hand, a conventional Kalman filter would provide extremely fast, but extremely incorrect results. Hence, the filters in between are often looked at. These are ranked on a scale based on the metrics below in *Figure 8*.



Figure 8. Filter Ranking In Terms of Accuracy & Efficiency

Several of these filters were compared in [5] against an NRHO via Monte Carlo simulation to estimate a truth trajectory. These included the conventional Kalman filter, EKF, UKF and GMM. As expected, the conventional Kalman filter performed considerably worse than the rest of the filters, the significantly higher position and velocity RMSE throughout every stage of the orbit. The EKF significantly improved upon this,

and UKF improving further still, although overall only by a factor of two, not as significant as the EKF upon the conventional Kalman filter. These results are presented below in *Figure 9*.

Table 2. Filter Root-Mean-Square Errors

		Perilune	Apolune	Steady-State	Full Run
mCKF	Position [km]	4.11	12.58	36.24	23,117*
	Velocity [cm/s]	89.56	16.47	69.84	29,709*
EKF	Position [km]	0.104	0.047	0.261	0.331
	Velocity [cm/s]	2.162	0.029	0.423	2.414
UKF	Position [km]	0.088	0.042	0.056	0.149
	Velocity [cm/s]	1.828	0.025	0.319	1.593

* CKF Divergence after 7th perilune passage included.

Figure 9. Filter RMSE for Conventional Kalman Filter, EKF & UKF

The GMM, although for this study was relatively simple, with one univariate split and no resplitting with later propagation, still achieved significantly better results further still, with the non-Gaussian uncertainty allowing for an unchanged level of accuracy pre- and post- perilune passage for the first orbital period. However, it should be noted that all filters tested had their overall accuracy diminish with every passage, albeit including the GMM to a much lower extent.

As previously discussed, the SRCKF-KF hybrid seemed to also outperform both UKF and EKF, and although unlikely to outperform the GMM or a particle filter, it is due to these results that it is worth a large amount of attention for testing hybridised filters in an NRHO context. From the performance of the GMM, it is highly likely that hybrid approaches will also experience accuracy degradation with every perilune passage.

E. Measurement/Navigation-Targeted Solutions

It is the general assumption that most future missions will rely on DSN ground stations for state measurements, in order to update the OD filters. As previously discussed, this causes issues with uncertainty and measurement bias, as regarding the L2 southern NRHO of interest, the spacecraft's motion is for the majority of the orbit perpendicular to the Earth-spacecraft LOS. As this bias is difficult to estimate, the filter's may overestimate or underestimate its trust in the estimates, thus leading to filter convergence on simply wrong trajectories. Additionally, data-rich environments are crucial for successful OD, and although the 9:2 lunar synodic resonance NRHO generally maintains LOS with DSN ground stations, there may be

occasional blockages, especially on manned missions such as Gateway. Thus, a filter's ability to recover or have alternative data sources is crucial for filter stability maintenance.

There are a variety of solutions that have been discussed in recent years to target these problems. In [6] it was discovered that tracking pass placement could play a crucial role in mitigating inaccuracies caused by the unknown range bias, with range data taken within two hours of perilune passage being found to accurately estimate the range bias. This is especially advantageous, as the perilune passage is the most crucial and volatile part of the orbit, thus successful range bias estimates are highly important in ensuring the filter performs to the best of its ability.

However, the other approach, which solves both the range bias inaccuracies across the whole orbit and the potential for data outages, is utilising a different measurement source, i.e. another spacecraft. This is known as satellite-to-satellite tracking (SST). However, the issue with using other satellites to determine position is that the accuracy of these measurements is highly dependent on the accuracy and confidence in the other satellites' states relative to Earth ground stations. This means that these satellites are also limited by their own OD filters, despite likely utilising more stable orbits. Additionally, these satellites may not constantly maintain LOS with the spacecraft or Earth ground stations, thus reliance on this solution is unfeasible.

This issue, however, could be mitigated via Linked Autonomous Interplanetary Satellite Orbit Navigation (LiAISON) [25]. LiAISON essentially creates a network of SST between two or more spacecraft, generating unique trajectories and thus the states for all the spacecraft involved. This approach takes advantage of the mascons in the lunar gravitational field to generate these trajectories, given that the orbits take place on different planes on the Moon's orbit. Orbits around the L1 and L2 libration points are highly successful at maintaining unique and asymmetrical orbit trajectories due to 3BP perturbations, thus making the NRHO of interest a prime candidate to take advantage of this technology. LiAISON application for the NRHO was recently explored in [12], with a relationship being simulated between the NRHO and a distant retrograde orbit (DRO) not in the same lunar synodic resonance. As the NRHO-DRO pairing very rarely encountered lunar eclipses (with a total of 14 hours per year) this study did not account for their occurrence. Utilising a SRIF as a comparative filter for results, it showed that LiAISON with the DRO would be viable, with only a marginal decrease in accuracy when compared to results relying on the DSN. However, with noisy crewed missions, the difference in accuracy greatly increased, indicating that purely DRO-based tracking passes would not be a feasible option for these missions, likely requiring an additional satellite. However, this additional satellite would be a viable option regarding the augmentation of the DSN data, such that this range bias uncertainty could be mitigated.

The final and most common approach to mitigating the effects of the unknown range bias is to augment the data via filter tuning and augmentation via process noise [26]. This approach attempts to prevent filter divergence as time goes on, ensuring that the filter continues to account for unmodelled perturbations and uncertainty. The process noise compensated for can be one of two types: static (SNC) or dynamic (DMC). SNC simply increases the filter covariance arbitrarily, i.e. decreases how much the filter trusts its own estimate, ensuring that later measurements continue to be weighted accurately against the estimate [27]. Range rate intervention is a type of SNC that inflates position covariance as the orbit continues, ensuring that range measurements are not disregarded against the estimate. DMC on the other hand is considerably more complex, attempting to prevent OD errors from increasing indefinitely as the orbit progresses [28]. This includes increasing the weighting of the most recent measurements to reduce the influence of the past ones.

F. Research Gaps & Future Directions

NRHO OD has been of great interest to the aerospace community for two reasons: the renewed interest in interplanetary travel and space projects and the NRHO's unique and advantageous characteristics that lend itself to greatly enhancing the capabilities of these missions. Although great progress has been made to the field of OD since the days of the Apollo program, the nature of the NRHO, an orbit that has not been greatly explored and can be highly volatile, that has only had one successful mission (CAPSTONE, [29]), the importance of developing and exploring filters that can handle the unstable dynamics is still a topic of great interest, with the delivery of robust, timely and accurate results not yet being achieved.

As aforementioned, the main issues regarding adequate OD for NRHO are regarding the inherently unstable dynamics at perilune, the significant perturbations from 3BP dynamics, mascons in the lunar gravitational field and the space environment, and the inability to accurately know the bias of the measurements being provided to the filters.

Highly unstable dynamics introduce the most significant difficulties associated with NRHO, as the considerable trade-offs between accurately modelling these dynamics and computational efficiency greatly complicate the selection of adequate filters. For classical OD, EKFs are the most common solution but their linearisation assumptions are quickly invalidated. Filters that can greatly capture the filter dynamics, such as particle filters and GMMs, have a significantly increased computational time, and are thus not adequate for NRHO missions, as timely state estimates are crucial for mission success, especially since manoeuvres are likely to be executed at highly volatile stages in the orbit. A solution outside the Kalman filter realm was ML, and was shown to excel in performance in other non-linear cases. However, this solution relies

heavily on real-life training data, currently unavailable. Another approach is the hybridisation of filters, only employing complex and time-consuming filters at highly unstable parts of the orbit. This would reduce computational time wasted on the stable parts of the orbit, whilst reducing accumulative error at perilune passage.

Measurement bias and perturbation uncertainty limitations also pose significant obstacles to the NRHO, with noise compensation and filter tuning being the most common solution for accommodating these errors. Alternative measurement sources besides DSN such as SST and LiAISON have been found to reduce the measurement bias issues, and although have been found to be currently insufficient for NRHO to solely rely on these technologies, the potential use of these technologies for measurement data augmentation could significantly mitigate these measurement bias issues.

III. Methodology

A. Methodological Approach & Filter Selection

From the research gaps discussed in the previous section, it was decided that a further filter comparison study would be performed. This is largely due to the filter selection being the main obstacle in adequate NRHO OD, with uncertainty issues fairly easily compensated for via noise compensation as previously outlined. These filters will be tested against a ‘truth’ trajectory, based off a trajectory similar to the CAPSTONE mission. The filters will receive an initial state, propagate this state to the next timestep, then receive an updated measurement of this state, and will adjust its next state estimate accordingly. States are six dimensional, including position (x, y, z) and velocity (u, v, w) in Earth-Moon barycentre-centred inertial (BCI) cartesian coordinates as presented below.

$$X = [x \ y \ z \ u \ v \ w]^T$$

The difference between the measurement data and the estimate will determine how much the filter trusts its orbit propagation model. The two primary metrics for performance evaluation will be the position RMSE between the state estimate and the truth trajectory, and the total computational time. This process is outlined in *Figure 10* below.

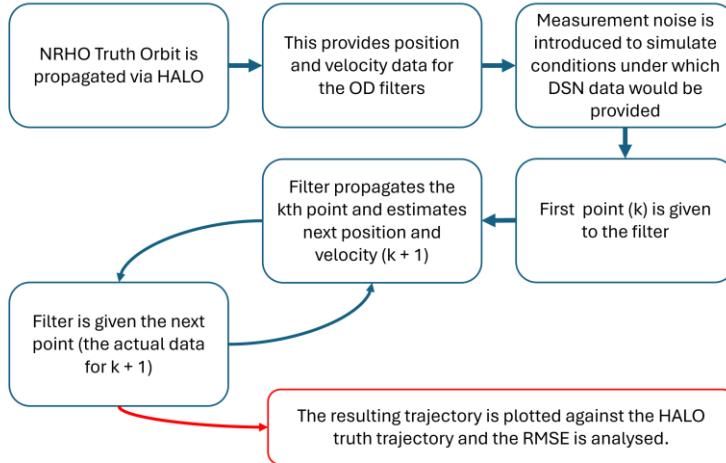


Figure 10. Filtering Process

Three filters would be selected for this study: EKF, CKF and UKF. These filters are selected for two main reasons: firstly, as the EKF and UKF have been compared in past, it provides a benchmark for the results. Secondly, the EKF and UKF are significantly different in computational time and accuracy, whereas the CKF potentially allows for an improvement upon computational time with minimal sacrifice to accuracy. Additionally, these filters are prime candidates for use in a hybridisation study, with the EKF, as it is usually adequate for stable orbits, being ideal for use with either the CKF or the UKF. The key differences between these three filters are presented in *Table 1* below.

Table 1. Key Differences Between Selected Filters

Filter	EKF	CKF	UKF
Key Differences	Linearises between states via state transformation matrix.	Propagates 12 potential points selected via cubature rule, equal weighting.	Propagates 13 potential points selected via the unscented transform, weighting varies.
Efficiency	High	Lower	Low
Precision	Low	Higher	High

With hybridisation being the most crucial contribution to this study, two different methods will be investigated. This method aims to leverage the benefits of the EKF and the CKF/UKF simultaneously, successfully balancing the trade-off between filters and hopefully, as previously discussed and demonstrated by [22], improve upon all three stand-alone filters.

The first is a novel approach, utilising the knowledge of the NRHO orbital dynamics, specifically its unique stability characteristics. Therefore, it can be assumed that the linearisation-reliant EKF may be sufficient during the stable durations of the orbit, and towards the volatile perilune passage filters may be switched to one of the two higher-fidelity filters. The second method involves a more adaptive approach, using the state covariance matrix as a metric to determine when to switch. As this matrix indicates how much the filter trusts its own estimate versus the measurements, it is an indicator of orbit stability, and as such can be used to switch to a higher-fidelity filter when a set threshold is met. These changes are determined by calculating the state entropy, i.e. the degree of uncertainty, which is determined via the formula below [30], where P represents the state covariance matrix.

$$H = \frac{1}{2} \log |2\pi e P|$$

All filters will be compared via the same truth trajectory, using identical measurement noise, process noise and initial state uncertainty to ensure fair comparison. Finally, the filter implementation will be checked and validated in two stages. Firstly, they will be tested directly with state data with minimal noise to ensure the algorithms themselves correctly execute. Then, with the simulated measurement data the filters will be run and tuned such that the magnitudes of RMSE match those in similar studies, such as [22]. A Monte Carlo analysis will also be performed to test the consistency of the filters.

B. Filter Algorithm Implementation

Kalman Filter Algorithm Foundation

Kalman filters all fundamentally follow a five step process. The first step is to have state information at the current time, conventionally denoted as t_{k-1} . Crucial state information is the state itself X_{k-1} and the filter's uncertainty/covariance when it settled on this state estimate P_{k-1} [7].

Then the filter, knowing the expected timestep to t_k uses equations of motion or, depending on the application (and in this case), high-fidelity propagators to estimate the next state \bar{X}_k . This is known as the prediction or time update. In the case of this study both were used, with circular restricted 3BP (CR3BP) equations of motions first used to directly test the algorithms, as the model was simpler and faster for debugging purposes, and later implementing the high-fidelity propagator ‘HALO’, discussed in greater detail below in C. Simulation Framework & Data Generation. The predictor step is where the user can set the measurement and process noise to tune how conservative the filter is with estimating its own uncertainty. Via the process noise, an expected covariance \bar{P}_k is estimated. Using this value, and the set

expected measurement noise, the expected uncertainty for the measurement residual/‘innovation’ is estimated, and thus the Kalman gain K_k is determined.

Finally a measurement update/correction step is performed. This is when the measured state is received and compared against the predicted state. The innovation, y_k , is computed between these two states and is multiplied by the Kalman gain to determine how to correct (x_k) the initial state estimate, and thus determine what the filter deems as the closest approximation to the truth (X_k).

$$x_k = K_k \cdot y_k$$

$$X_k = \bar{X}_k + x_k$$

The uncertainty P_k is updated accordingly, and these two values are used to initialise for the next time step.

Extended Kalman Filter

The EKF was the only selected linearisation-based filter for this comparative study. As previously described, the EKF utilises linearisation assumption to ensure timely OD, however, falling short when these linearisation assumptions fail: i.e. when the orbits become unstable and no longer possess Gaussian PDF qualities. These linearisation assumptions are applied at the time update. After X_{k-1} is propagated to \bar{X}_k , the state transition matrix (STM, Φ) is used to map the transition of state between t_{k-1} and t_k , linearising the relationship between them [7]. This linearised relationship is what is used to estimate the covariance, thus assuming that the uncertainty possesses a Gaussian PDF. For this implementation, the expected covariance \bar{P}_k is also calculated according to the STM as presented below, where Q_d is the process noise [7].

$$\bar{P}_k = \Phi P_{k-1} \Phi^T + Q_d$$

A measurement mapping matrix H_k is also determined to map the relationship for use in determining the estimated innovation covariance by defining the relationship between the states and the expected/received measurements. For direct state measurements, this is simply an identity matrix, however, for this study, as range and range rate measurements are being simulated, H_k is a Jacobian matrix mapping between the state and the measurement, where r is range, \dot{r} is range rate, x is position and $v = \dot{x}$ is velocity.

$$H_k = \begin{bmatrix} \frac{\delta r}{\delta x} & \frac{\delta \dot{r}}{\delta x} \\ \frac{\delta r}{\delta v} & \frac{\delta \dot{r}}{\delta v} \end{bmatrix}$$

These two key simplifications in the dynamics, specifically the STM, lead to tremendous time savings and generally adequate OD accuracy, and although it is known that it by itself is unsuitable for NRHO applications, its implementation in this study provides both a good benchmark for comparison against classical OD solutions and an advantageous opportunity for utilisation in the hybrid filters.

For this implementation of EKF, the final covariance is updated via Joseph form for additional numerical stability [7]:

$$K_k = \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + R_k)^{-1}$$

$$\begin{aligned} P_k &= [I - K_k H_k] \bar{P}_k \\ &= (I - K_k H_k) \bar{P}_k (I - K_k H_k)^T - K_k R_k K_k^T \end{aligned}$$

Unscented Kalman Filter

The UKF, like the EKF follows the same basic Kalman filter structure, however, the key difference once again occurs in the time update step of the filter. Rather than a single point being propagated to the next timestep, several ‘sigma’ points are propagated, and the results are weighed against each other and averaged to arrive at a state and corresponding covariance [31]. This is achieved via the UT, which essentially dictates the spread of the sigma points, their comparative weighting against each other, and the number of sigma points to propagate. These parameters, and their values in this study are presented below in *Table 2*.

Table 2. UT Parameter Definitions & Study Settings

Notation	Definition	Set Value
L	State dimension, number of sigma points dictated as $2L + 1$	6
α	Determines the spread of sigma points, usually between 0.001 and 0.1	0.01
β	Sets up PDF assumptions, set as 2.0 to prime assumptions to be Gaussian	2.0

κ	Secondary scaling parameter, defined as $3 - L$, and used to determine lambda and gamma	-3
λ	Parameter for determining gamma, defined as $\alpha^2(L + \kappa) - L$	-5.999997
γ	Scaling factor for sigma points, defined as $\sqrt{L + \lambda}$	0.001732

The values above are then used to determine the spread and weighting of each propagated point, in order to best approximate the true, non-Gaussian nature of the uncertainty. X_{k-1} is used as the mean to generate the spread of points χ_{k-1} , calculated using the square root of the covariance via the following formula:

$$\chi_{k-1} = [X_{k-1} \quad X_{REF,k-1} + \gamma\sqrt{P_{k-1}} \quad X_{k-1} - \gamma\sqrt{P_{k-1}}]_{L \times 2L+1}$$

After propagating all thirteen of these points, the time update is completed by estimating the expected state and the expected measurements. For the state, this is simply the weighted average of the propagated sigma points. As for the expected measurement, this is achieved by taking the propagated sigma points $\bar{\chi}_{k,i}$ and calculating range and range rate for each point $\bar{Y}_{k,i}$, finally taking the weighted average of these to get an expected value \bar{Y}_k :

$$\bar{X}_k = \sum_{i=0}^{2L} W_i^m \chi_{k,i}$$

$$\bar{Y}_k = \begin{bmatrix} \bar{r} \\ \dot{\bar{r}} \end{bmatrix}_k = \sum_{i=0}^{2L} W_i^m Y_{k,i}$$

The predicted covariance is calculated via the equation below, taking the weighted average of the covariance relationship between the overall estimated state and each sigma point. Note that this is where the process noise is added (Q_d).

$$\bar{P}_k = \sum_{i=0}^{2L} W_i^c [\chi_{k,i} - \bar{X}_k] \cdot [\chi_{k,i} - \bar{X}_k]^T + Q_d$$

Next the innovation covariance P_{yy} and cross-correlation P_{xy} are calculated. These values are used to calculate the Kalman gain and update the final covariance. These equations are presented below. Note that this is where the measurement noise is applied (R_k) [7].

$$P_{yy} = R_k + \sum_{i=0}^{2L} W_i^c [Y_{k,i} - \bar{Y}_k] \cdot [Y_{k,i} - \bar{Y}_k]^T$$

$$P_{xy} = \sum_{i=0}^{2L} W_i^c [\chi_{k,i} - \bar{X}_k] \cdot [Y_{k,i} - \bar{Y}_k]^T$$

$$K_k = P_{xy}[P_{yy}]^{-1}$$

Finally the measurement update is carried out to arrive at best determination of where the spacecraft is by updating the covariance and the state with the new measurement. The measurement residual/innovation y_k is determined, and the following operations provide the final state and covariance. Note that the final state equations are the same for the EKF also.

$$x_k = K_k [Y_k - \bar{Y}_k]$$

$$X_k = \bar{X}_k + x_k$$

$$P_k = \bar{P}_k - K_k P_{yy} K_k^T$$

Cubature Kalman Filter

The CKF is highly similar to the UKF, with the key difference being the sigma point selection process. Rather than using the UT, the CKF relies on the cubature rule, a much simpler process that evenly weighs and spreads the sigma, or in this case, cubature points. The number of points is simply double the state dimension (in this case 12 points, rather than 13 for the CKF). Each propagated point and corresponding covariance are evenly weighed, and the spread parameter, known as ξ , is determined as the square root of the state dimension [32]. Due to the simplicity of the cubature rule, the CKF does not require tuning, and additionally should be computationally more efficient as the mathematics are simpler and it propagates fewer points.

C. Simulation Framework & Data Generation

Truth Trajectory Generation

The truth trajectory upon which the filters were tested with was generated by the ‘High precision Analyser of Lunar Orbits’ (HALO) [33], a high-fidelity propagator, that can generate various lunar orbit trajectories with full six-dimensional state data. This propagator was designed to utilise high-fidelity

ephemeris models to capture the perturbations associated with cis-lunar orbits. These include 3BP, general relativity corrections, SRP, Earth albedo, additional point masses (the Sun and Jupiter in particular), and the complexities of the Earth and lunar gravitational fields, modelled via 100x100 and 350x350 harmonics respectively [34].

Of these orbits included a NRHO ‘close to’ the CAPSTONE orbit, the L2 southern, 9:2 lunar synodic resonance orbit of interest. This orbit extends over two orbital periods, with three perilune passages and includes the spacecraft divergence from the NRHO, over the timespan of 26.6 days, with a data point generated every thirty seconds, and a total of 76,593 data points, however, this mission includes an orbit departure. As the aim of this project is to evaluate the filters’ performance under NRHO, the departure is removed from the trajectory, leaving an NRHO with three perilune passages. This reduced the data points down to 47,500 points, over 16.5 days. This truth trajectory provides a trajectory to directly compare the filter performance against, whilst also allowing for the generation of DSN measurements for the filters to use. This is discussed in the next section.

DSN Measurement Simulation

The DSN measurement simulation was generated via the HALO generated truth data and the addition of standard assumed error models, typical for range and range rate data. Three ground stations were configured for: Canberra (Australia), Madrid (Spain) and Goldstone (USA). All coordinates had to be in the same frame as the points generated by the HALO propagator, an Earth-Moon barycentre-centred inertial frame (BCI). This was achieved by first converting the geodetic coordinates (latitude, longitude, altitude) to Earth-centred Earth-fixed inertial frame (ECEF). The ECEF coordinates would then be converted to Earth-centred inertial coordinates (ECI), and then finally these coordinates would be converted to BCI. These transformations were performed via the SpiceyPy library, loading leap second and Earth-Moon ephemeris models as used by HALO and generating Earth and Moon states in order to transform the coordinates at each time stamp relative to the corresponding Earth, Moon and Earth-Moon barycentre states. The equations for these transformations are included in Appendix A: Coordinate Transformations.

Using the BCI coordinates, the position and velocity relative to each ground station could be calculated. To ensure LOS between the spacecraft and the ground station, there was an elevation criteria. The angle between the Earth horizon and the ground station-satellite vector was calculated to determine the elevation. This is visualised below in *Figure 11*.

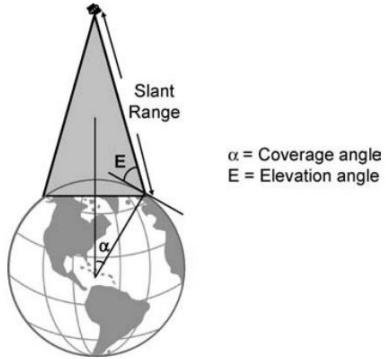


Figure 11. Elevation Angle Diagram [35]

To ensure there was no Earth blockage, the elevation angle had to be greater than 10 degrees from the horizon and less than 170 degrees, as this is standard for DSN ground stations [36]. This ensured that the minimum 10 degrees off the horizon would be met. It should be noted that because this NRHO is designed for continuous communication with Earth, it was assumed that there would be no lunar blockages.

From the ground stations that achieved this criteria, the one with the shortest range, i.e. closest to the satellite, would be selected. This selected ground station state would be stored for the use of the filter, when estimating the expected range and range rate measurements from the propagated estimate state. Then the relative position and velocity data would be calculated between the satellite and the ground station. This calculation is presented below, where S is the true spacecraft state, B is the Earth-Moon barycentre, G is the visible ground station.

$$\overrightarrow{SB} - \overrightarrow{GB} = \overrightarrow{SG}$$

Next, the range and range rate for this data point would be calculated via the following formula, where range is simply the magnitude of the position vector, and range rate is the projection of the velocity vector onto the position vector [37]. Note that these are determined utilising the NumPy library functions.

$$r = |\overrightarrow{SG}_{xyz}|$$

$$\dot{r} = \frac{\overrightarrow{SG}_{xyz} \cdot \overrightarrow{SG}_{uvw}}{|\overrightarrow{SG}_{xyz}|}$$

Finally, from historical DSN error and bias models, random error was applied to the range and range rate values and stored as a DSN simulated measurement. These values are displayed below in *Table 3*.

Table 3. DSN Simulation Random Error (3-standard deviations) [38]

	Range	Range Rate
Noise	3 m	1 mm/s
Bias	7.5 m	2.5 mm/s

Initial State Generation

The initial state and covariance for the filters are all set up in the interface.py file. This file will be described in greater detail in D. Program Architecture & Implementation. However, it is from this file that the initial covariance, process noise covariance and measurement noise covariance are set. Measurement noise, R_k , is simply set to the same settings as the DSN random noise and bias as to anticipate the error, as a 2 by 2 matrix for the range and range rate. The process noise, Q_d , on the other hand is applied to the state and is constructed as a 6 by 6 matrix. This was progressively tuned to optimise the filter results. Initial covariance is set via a scalar, with the matrix P_0 fully constructed within the filter functions, was also structured and tuned in a similar way. The final values selected are presented below in *Table 4*. The first state is simply set as the truth state.

Table 4. Standard Deviation (STD) Initialisation Settings For R_k , Q_d , P_0

	Position/Range* STD	Velocity/Range Rate* STD
Measurement Noise Matrix R_k	2.67 m	0.9 mm/s
Process Noise Matrix Q_d	100 m	10 cm/s
Initial Covariance Matrix P_0	10 m	1 cm/s

*Only for the case of R_k

D. Program Architecture & Implementation

Software Framework

The filter comparison program is coded in Python, and the filters, propagators and other mathematical functions (such as for the DSN simulation generation, Jacobian matrix generation and STM matrix generation) are coded across a number of Python files, utilising the NumPy and SpiceyPy libraries. These files are all run from the single interface.py file, with the relationships between the key files displayed below in *Figure 12*. This file allows the user to select which filters to run, the initialisation settings for noise

and uncertainty, and whether or not hybridisation is run between any of the filters. This file is also where the plotting of the trajectory and the RMSE is coded, with much of the plot code for the trajectory being borrowed from the HALO interface and the functions included in HALO’s ‘process.py’, maintaining the same ability as HALO to swap coordinate frames between Earth-Moon rotational and inertial frames. Additionally, timing of the filter operation begins when the interface is run, timing the whole program operation.

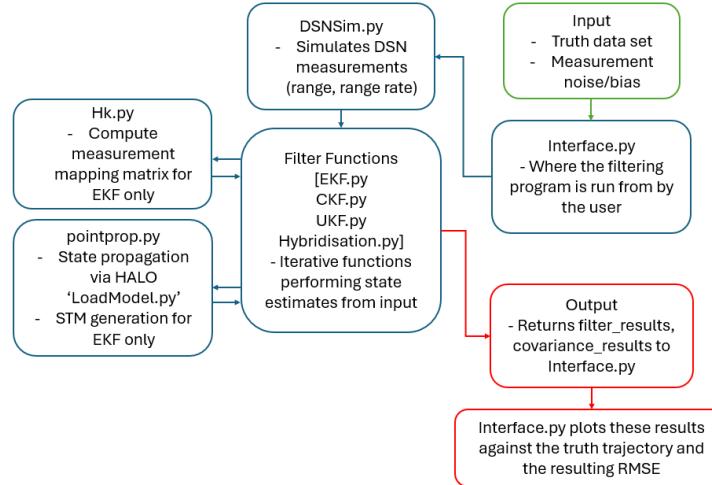


Figure 12. Software Framework

The software can be accessed via GitHub [39], by downloading the ‘main’ folder. The only file needed for the user to use is `interface.py`, as the whole program is tuned and run from this folder. The only exception is in the case of the hybrid filters, where the number of burn-in steps and transition noise must be adjusted within the `hybrid.py` file. These are discussed in E. Filter Hybridisation.

Data Flow & Processing

From `interface.py`, the inputs are loaded. These are the full state vectors generated by the full fidelity HALO propagator, and their corresponding times, in ephemeris times (seconds). This data is processed using the DSN simulation function constructed in `DSNSim.py`, and returns the simulated DSN range and range rate measurements, and the corresponding ground station states, such that the filter functions can use the initial propagated state from the time update to calculate expected range and range rate measurements.

These values and the user set initial covariance, process and measurement noise, are inputs to the filter functions called upon. Within each filter function; `EKF.py`, `CKF.py` and `UKF.py`, the functions utilise the `pointprop.py` file to propagate the previous state for the time update. This `pointprop.py` file has two function:

point propagation and the STM generation (used by EKF.py only). The point propagation function once again utilises the high-fidelity propagator HALO again, however, rather than using the full propagator it uses the “Fast” propagator. This allows for the computational time to remain within reasonable bounds for the experiment, whilst also being used to anticipate the difference between a high-fidelity propagator and real-life environment, as even HALO cannot truly model the environmental perturbations of the space environment. Every ‘closest approximation of the truth’ state that the filter achieves with each time step, the corresponding covariance, measurement residual and covariance entropy are stored for data analysis.

Results & Analysis

Two key metrics for the output are obtained after the running of the filters: the total computational time and the RMSE. The RMSE is calculated for both the position and the velocity by first determining the error between the filter’s final determination of state versus the truth state:

$$\begin{aligned} \text{error} &= |X_{\text{truth}} - X_k| \\ \text{error}_{\text{position}} &= \sqrt{(x_{\text{truth}} - x_k)^2 + (y_{\text{truth}} - y_k)^2 + (z_{\text{truth}} - z_k)^2} \\ \text{error}_{\text{velocity}} &= \sqrt{(u_{\text{truth}} - u_k)^2 + (v_{\text{truth}} - v_k)^2 + (w_{\text{truth}} - w_k)^2} \end{aligned}$$

These errors are plotted with each time, and the overall RMSE is calculated by taking the average of the errors squared and then finally taking the square root:

$$RMSE = \sqrt{\overline{\text{error}^2}}$$

The truth versus filter trajectories are also plotted, with an example shown below in **Error! Reference source not found.**, however, with the magnitude of error being within one kilometre, the difference is too small to be seen from such a large scale. The main purpose of this function was for debugging as it made it easier to identify the nature of any errors, and in case it might be of use for other missions and orbits. As aforementioned, the trajectory can be plotted in either Earth-Moon barycentric inertial or Earth-Moon barycentric rotating frame. The mission trajectory being tested is displayed below.

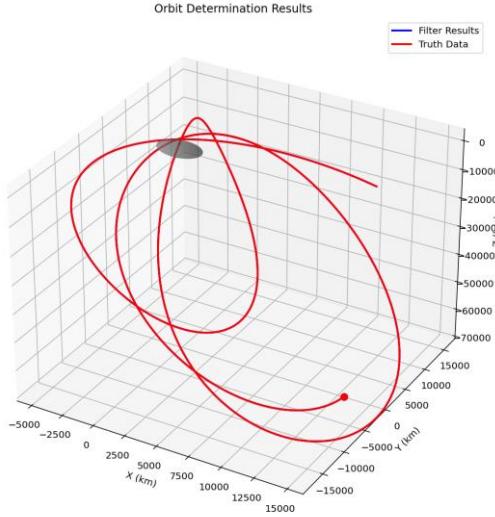


Figure 13. Mission Trajectory, BCI (left), BCR (right)

E. Filter Hybridisation

The filter hybridisation aspect of this experiment is the cornerstone of the project. The hybrid approach has the potential to greatly improve upon not just the computational time of the overall OD process but also considerable accuracy improvements, as demonstrated in [22], where they utilised this method in a traffic context. Thus, there is no reason no to assume that this approach could also be utilised in an aerospace application, potentially helping to solve the severe trade-offs adequate OD for NRHO requires.

Two approaches for the hybridisation approach are to be implemented in this experiment. The first is very simple. As the perilune is known to be the period of instability, when the filter reaches perilune passage the filters swap between the stable and unstable filters selected by the user. These passages are marked by certain time periods in the trajectory, such that the filters swap at these designated stable and unstable times. Increasing/decreasing the defined period of instability is simply done by shifting the time stamps. The time stamps selected indicated periods where the spacecraft was within 50,000 kilometres from the Moon's centre, and are marked below in *Table 5*. These are set in the interface.py file. Note that the truth trajectory ends in the perilune passage as the NRHO departure commences straight afterwards.

Table 5. Unstable Orbit Phases & Corresponding Time Stamps

Orbit Phase	1 st Perilune Passage	Passage 1 End	2 nd Passage	Passage 2 End	3 rd Passage
Time Stamp (start-end)	7.5752778e8	7.5772224e8	7.581112e8	7.583344e8	7.586389e8

The second, and more dynamic approach is regarding the covariance entropy. This is essentially a metric by which the uncertainty is determined, with greater entropy meaning that the filter is more uncertain about its prediction. Generally, uncertainty increases towards perilune because the dynamics do not follow a Gaussian PDF any longer. An entropy threshold can be set by the user, such that when the entropy exceeds this threshold the filter swaps, assuming the spacecraft has entered an unstable section of the orbit. The lower the entropy threshold is, the more time the filter spends using the higher accuracy filter, increasing the overall computational cost. Two entropy thresholds are set, one for each filter, as although relative entropy values increase when instability occurs, these higher values may not be abnormal depending on the filter. These values are shown in the Results, and were delineated based on the entropy values from the full, pure runs of the three filters, watching for how the entropy increased at perilune and other unstable parts of the orbit.

For this experiment, as only three filters are being compared, with two of them being highly similar, the only stable filter will be the EKF, as swapping between the CKF and UKF provides minimal benefits. Thus, the EKF-CKF and EKF-UKF will be the two hybrid filters being compared.

Hybridisation Implementation Structure

Hybridisation is enabled from the interface.py file. Rather than selecting ‘run_filter’, the user selects ‘hybridisation’ and sets the desired stable and unstable filters by setting the ‘unstable’ and ‘stable’ variables as the name of the filter in string format. Then, the hybridisation approach is set by the user. This is done by setting the ‘entropy_hybridisation’ and ‘time_hybridisation’ variables as 1 or 0 to dictate whether they will be run. Thresholds and time stamps for switching are also set from the interface. These are all inputs fed to the run_hybrid (entropy switching function) or run_hybrid_time_based.

These functions’ main use is to enable the switching between filters. Rather than run from the interface, the filter functions are called within these hybrid functions. Both hybrid functions run the stable filter first, swapping when the entropy threshold is met or when the first time stamp is exceeded. The time filter does this by constructing time intervals to run the filters between and splicing the results. The entropy based, on the other hand, iterates through the data points, checking the entropy in ten point segments, and when there is a consecutive number of entropy values exceeding the threshold, the filter swaps. For this study, it was set to five for the largest time step and scaled linearly, but can be changed in the run_hybrid function as the variable ‘consecutive_steps’ to make the filter more or less sensitive. This consecutive step implementation prevented the hybrids from switching too frequently, which would negate the positive benefits they might have in terms of both accuracy and efficiency.

Note that when the filters swap, as the covariance matrices between the EKF and the CKF/UKF are not translatable, the covariance must be reset. To allow the filter to reconverge a certain number of ‘burn-in’ steps are set such that there are not significant spikes in error. Additionally, to quicken the burn-in process and minimise runtime, the covariance is also multiplied by a ‘transition noise factor’. These values vary for each different timestep magnitude to optimise performance (as stated accordingly in the results sections), and are set with in the hybrid functions under ‘BURN_IN_STEPS’ and ‘transition_noise_factor’. These were calibrated via trial-and-error.

This process continues until all points have been iterated through. This hybridisation structure is summarised in *Figure 14* below.

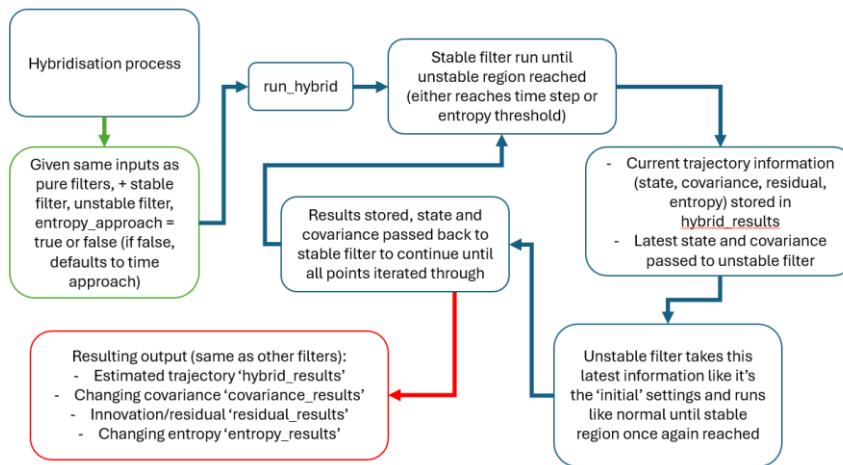


Figure 14. Hybridisation Algorithm

F. Validation & Testing

Simulation Parameters

Time step magnitudes are crucial in determining the effectiveness of each filter. Smaller timesteps may allow the filters to be more accurate as more frequent measurements are coming in, but are computationally very inefficient. Thus, if a filter can maintain accurate performance at greater time steps, this can provide great advantages to OD applications as, combined with the increased efficiency, an accurate state may be determined long in advance from the last measurement. This is also a good indicator of how well the filters can handle data outages and how well they maintain their accuracy or recover and reconverge when measurements are once again available. The smallest time steps for this experiment are 30 seconds, totaling 47,500 states over the whole trajectory. This experiment will run with these time steps, as well as 300 second (5 minute), 1,500 second (25 minute) and 3,000 second (50 minute) time steps.

As it was estimated that shorter time steps would correlate with higher accuracy, measurement and process noise and bias were tuned using longer time steps to allow for quick tuning. These values that were settled on were presented previously in *Table 4.* Standard Deviation (STD) Initialisation Settings For R_k , Q_d , P_0 .

Performance Metrics

As previously stated, the RMSE and computational time will be the two metrics of accuracy and efficiency respectively. Computational time is measured over the entirety of the running of the program, including DSN simulation and graph generation, via the Python ‘time’ tool. Both metrics are sourced with the purpose of comparison, and are not meant to necessarily be indicative of real life performance, as it is likely that higher-fidelity propagators would be used real life applications, increasing both the computational time and accuracy simultaneously. Monte Carlo analysis will be performed on the 3,000 time interval case to evaluate the consistency of the filters.

These metrics will be measured for each time step variation, and the RMSE will be calculated over several regions of the trajectory for further information about how the filters respond to the unstable regions. These are: the overall RMSE of the total trajectory, before first perilune passage, before second passage, and before third passage.

Debugging & Verification

The filter algorithms were debugged and validated by having the input simply be the original truth states. If the filter could deliver RMSE that was extremely close to the truth trajectory, they were working correctly. This worked successfully, with trajectories being within meters apart in RMSE, and fairly smooth RMSE plots with spikes occurring just at the perilunes, as shown in *Figure 15* below.

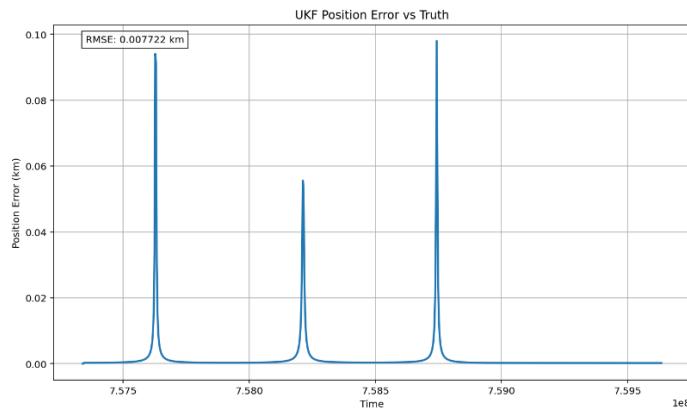


Figure 15. Example UKF RMSE State Only Debugging

The next step was to implement the DSN simulation, and alter the filters such that they would accept the measurements, calculating the expected measurements and the corresponding covariance between those and the simulated actual measurements. This once again consisted of monitoring how the RMSE changed, and if the resulting measurements produced a trajectory that aligned with the truth trajectory to ensure the correct coordinate transformations were being performed. Error patterns were also closely monitored. As strong filter divergence was only expected at the NRHO departure, it was clear when there were significant issues when filter divergence occurred at other stages of the simulation. During simulation, innovations were monitored, and if there were significant increases, the filter (while it was running) would alert the user to the large innovation, indicating that filter divergence was occurring and debugging should commence.

The other key indicator of correct filter operation, especially at the tuning phase of the experiment, was the monitoring of the covariance matrices, specifically the eigenvalues. For a filter to be stable, these matrices must remain positive definite, i.e. eigenvalues must both be positive. If these became negative, an alert was raised in the console indicating instability, allowing the user to stop the simulation and adjust the tuning accordingly.

IV. Results & Discussion

The results section is divided into the results for each varied time step. These include 30 second, 300 second, 1,500 second and 3,000 second time intervals. Each section provides three tables of results: a table to compare computational time, a table to compare overall RMSE, and a table containing various position RMSE values as the mission progressed. These reflect how the filters respond to perilune passages (PPs). Note that only the position RMSE is being taken for the mission progression analysis, however, changes in velocity error are magnitudes smaller than the position error, and seem to correlate with the position error anyway. Finally, the position error plots for the 3,000 second time steps are included. All position error plots can be found in Appendix B: Position Error Plots For All Runs.

A. Results

Entropy Threshold Settings

Below, in

Table 6, are presented the entropy threshold settings used for the entropy-based hybrid filters. The stable thresholds are an upper limit, and as such, when exceeded the filter swaps to the unstable filter. In the same

vein, the unstable thresholds are lower limits, and when the entropy decreases below this limit the filter swaps back to the stable filter. These threshold settings were selected from the entropy data generated from the ‘pure’ filters to reflect how the values changed towards perilune.

Table 6. Entropy Threshold Settings

	Entropy Threshold
Stable	17
Unstable	28.5

30 Second Time Intervals

The following results reflect the filter performances when provided data for every 30 seconds. This amounts to a total of 47,500 data points over the span of the CAPSTONE-like mission. Optimised reconvergence settings for the hybrid filters are as follows. The transition noise factor was set to 10,000 for the time-based and 20,000 for the entropy-based. Burn-in data points were set to 2,000 and 4,000 for time and entropy-based respectively. For the entropy-based filters, 250 points were required to exceed the entropy threshold before the switch would be triggered.

Table 7. Computational Times, 30-Second Time Steps

Filter	Computational Run Time	
EKF	206 minutes 21.37 seconds	
CKF	2301 minutes 18.30 seconds	
UKF	2799 minutes 8.18 seconds	
EKF-CKF	t-Based	1174 minutes 33.61 seconds
	H-Based	357 minutes 14.30 seconds
EKF-UKF	t-Based	1275 minutes 20.55 seconds
	H-Based	357 minutes 24.30 seconds

Table 8. Overall RMSEs, 30-Second Time Steps

Filter	RMSE	
	Position (km)	Velocity (km/s)
EKF	0.158794	0.000017

CKF		0.289947	0.000021
UKF		0.288551	0.000022
EKF-CKF	t-Based	0.186577	0.000018
	H-Based	0.180279	0.000017
EKF-UKF	t-Based	0.185001	0.000018
	H-Based	0.161979	0.000017

Table 9. Changing Position RMSE Throughout Mission, 30-Second Time Steps

Filters		Position RMSE (km)			
		Before 1 st PP	Before 2 nd PP	Before 3 rd PP	After 3 rd PP (Total)
EKF		0.253857	0.184383	0.163380	0.158794
CKF		0.254966	0.289879	0.291917	0.289947
UKF		0.275136	0.289740	0.288369	0.288551
EKF-CKF	t-based	0.216771	0.192655	0.176733	0.186577
	H-based	0.217950	0.205628	0.184413	0.180279
EKF-UKF	t-based	0.183668	0.186579	0.183742	0.185001
	H-based	0.166481	0.188177	0.166468	0.161979

300 Second Time Intervals

These results are determined over a total of 4,750 data points over the span of the CAPSTONE-like mission. Optimised reconvergence settings for the hybrid filters are as follows. The transition noise factor was set to 5,000 for the time-based and 10,000 for the entropy-based. Burn-in data points were set to 200 and 400 for time and entropy-based respectively. For the entropy-based filters, 200 points were required to exceed the entropy threshold before the switch would be triggered.

Table 10. Computational Times, 300-Second Time Steps

Filter	Computational Run Time	
EKF	33 minutes 13.40 seconds	
CKF	248 minutes 58.40 seconds	
UKF	265 minutes 23.83 seconds	
EKF-CKF	t-Based	198 minutes 45.98 seconds

	H-Based	222 minutes 25.29 seconds
EKF-UKF	t-Based	215 minutes 32.69 seconds
	H-Based	236 minutes 21.99 seconds

Table 11. Overall RMSEs, 300-Second Time Steps

Filter	RMSE	
	Position (km)	Velocity (km/s)
EKF	0.198575	0.000018
CKF	0.162630	0.000017
UKF	0.163105	0.000017
EKF-CKF	t-Based	0.188520
	H-Based	0.191625
EKF-UKF	t-Based	0.186909
	H-Based	0.191585

Table 12. Changing Position RMSE Throughout Mission, 300-Second Time Steps

Filters	Position RMSE (km)			
	Before 1 st PP	Before 2 nd PP	Before 3 rd PP	After 3 rd PP (Total)
EKF	0.240626	0.241059	0.203567	0.198575
CKF	0.222488	0.188189	0.167112	0.162630
UKF	0.225550	0.179153	0.166716	0.163105
EKF-	t-based	0.217012	0.234599	0.193689
CKF	H-based	0.240058	0.228906	0.196701
EKF-	t-based	0.238251	0.225748	0.192461
UKF	H-based	0.243526	0.225673	0.197122
				0.191585

1,500 Second Time Intervals

These results are determined over a total of 950 data points over the span of the CAPSTONE-like mission. Optimised reconvergence settings for the hybrid filters are as follows. The transition noise factor was set to 10,000 for all hybrid filters. Burn-in data points were set to 40 and 80 for time and entropy-based respectively. For the entropy-based filters, 20 points were required to exceed the entropy threshold before the switch would be triggered.

Table 13. Computational Times, 1,500-Second Time Steps

Filter	Computational Run Time	
EKF	8 minutes 10.15 seconds	
CKF	55 minutes 41.02 seconds	
UKF	59 minutes 52.81 seconds	
EKF-CKF	t-Based	33 minutes 41.49 seconds
	H-Based	34 minutes 22.00 seconds
EKF-UKF	t-Based	35 minutes 32.96 seconds
	H-Based	37 minutes 2.86 seconds

Table 14. Overall RMSEs, 1,500-Second Time Steps

Filter	RMSE	
	Position (km)	Velocity (km/s)
EKF	0.315101	0.000023
CKF	0.286520	0.000021
UKF	0.289156	0.000022
EKF-CKF	t-Based	0.305021
	H-Based	0.321047
EKF-UKF	t-Based	0.299023
	H-Based	0.313775

Table 15. Changing Position RMSE Throughout Mission, 1,500-Second Time Steps

Filters	Position RMSE (km)			
	Before 1 st PP	Before 2 nd PP	Before 3 rd PP	After 3 rd PP (Total)
EKF	0.299911	0.361369	0.322400	0.315101
CKF	0.344816	0.293320	0.296413	0.286520
UKF	0.364635	0.308947	0.299407	0.289156
EKF-	t-based	0.273436	0.308847	0.330487
CKF	H-based	0.402094	0.383665	0.330052
	t-based	0.259354	0.363371	0.323206
				0.299023

EKF-UKF	H-based	0.399797	0.369811	0.327702	0.313775
---------	---------	----------	----------	----------	----------

3,000 Second Time Intervals

These results are determined over a total of 475 data points over the span of the CAPSTONE-like mission. Optimised reconvergence settings for the hybrid filters are as follows. The transition noise factor was set to 5,000 for the time-based and 10,000 for the entropy-based. Burn-in data points were set to 10 and 40 for time and entropy-based respectively. For the entropy-based filters, 10 points were required to exceed the entropy threshold before the switch would be triggered.

Table 16. Computational Times, 3,000-Second Time Steps

Filter		Computational Run Time
EKF		4 minutes 41.69 seconds
CKF		31 minutes 58.89 seconds
UKF		28 minutes 10.27 seconds
EKF-CKF	t-Based	13 minutes 52.18 seconds
	H-Based	17 minutes 2.33 seconds
EKF-UKF	t-Based	14 minutes 43.91 seconds
	H-Based	24 minutes 42.10 seconds

Table 17. Overall RMSEs, 3,000-Second Time Steps

Filter	RMSE	
	Position (km)	Velocity (km/s)
EKF	0.557353	0.000034
CKF	0.547147	0.000033
UKF	0.545540	0.000033
EKF-CKF	t-Based	0.523341
	H-Based	0.539025
EKF-UKF	t-Based	0.520257
	H-Based	0.522642

Table 18. Changing Position RMSE Throughout Mission, 3,000-Second Time Steps

Filters		Position RMSE (km)			
		Before 1 st PP	Before 2 nd PP	Before 3 rd PP	After 3 rd PP (Total)
EKF		0.438009	0.615168	0.570248	0.557944
CKF		0.581823	0.559877	0.565401	0.547147
UKF		0.603595	0.568411	0.564751	0.545540
EKF-	t-based	0.441273	0.586715	0.543699	0.523341
CKF	H-based	0.446625	0.540396	0.554296	0.539025
EKF-	t-based	0.438096	0.590854	0.540704	0.520257
UKF	H-based	0.573154	0.549878	0.568411	0.522642

Error Plots

Below, the key error plots are shown. These plots are the results of the 3,000 second time steps, however, similar error patterns are reflected across all time intervals (note that the only plots with a significantly different pattern is the 30-second time step plots, however, these will be discussed in the Discussion section). As such, these additional plots are not shown in this section, but can be found in Appendix B: Position Error Plots For All Runs. Note that the 3,000 second time step case was deemed the best representation of the patterns as the points were further spaced apart, and thus they were easier to interpret.

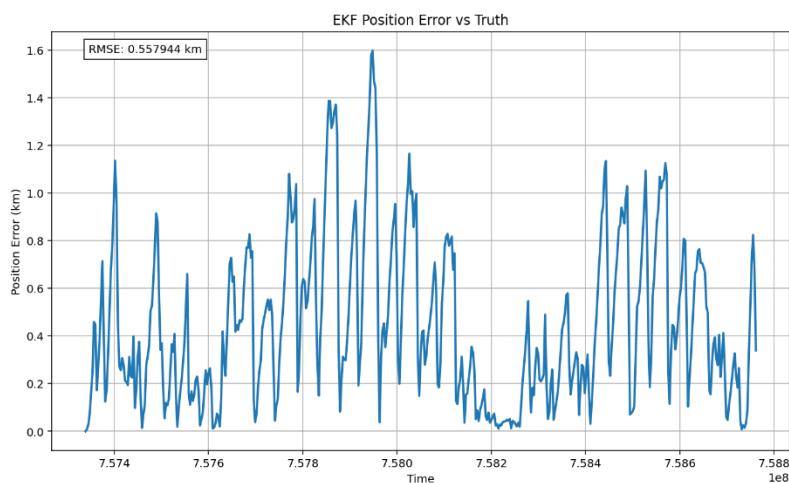


Figure 16. EKF 3,000-Second Time Step Error Plots, Position (left), Velocity (right)

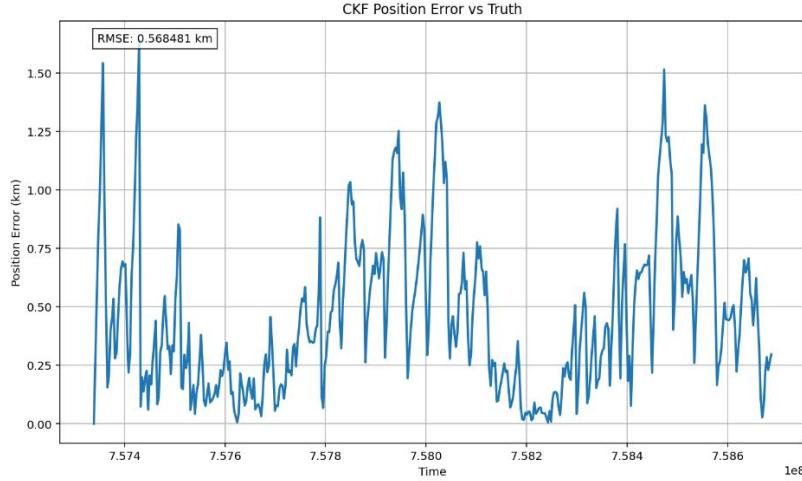


Figure 17. CKF 3,000-Second Time Step Error Plots, Position (left), Velocity (right)

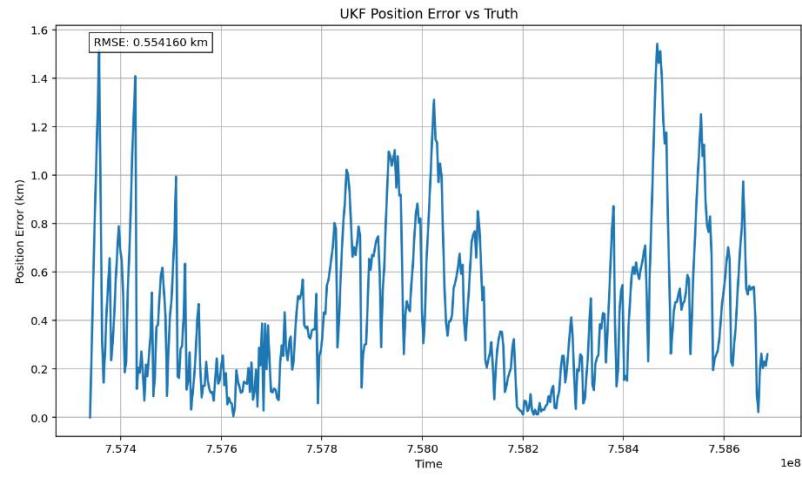


Figure 18. UKF 3,000-Second Time Step Error Plots, Position (left), Velocity (right)

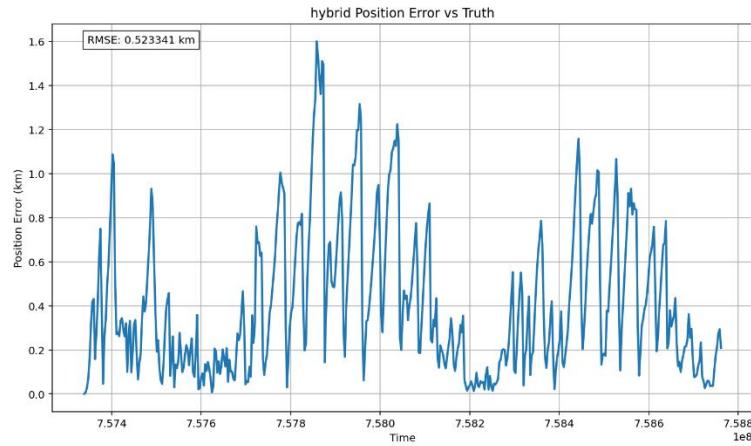


Figure 19. EKF-CKF (Time Based) 3,000-Second Time Step Error Plots, Position (left), Velocity (right)

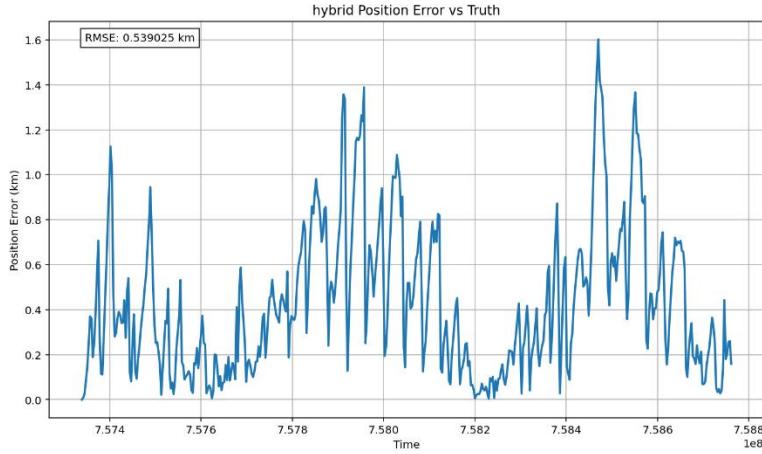


Figure 20. EKF-CKF (Entropy Based) 3,000-Second Time Step Position Error Plot

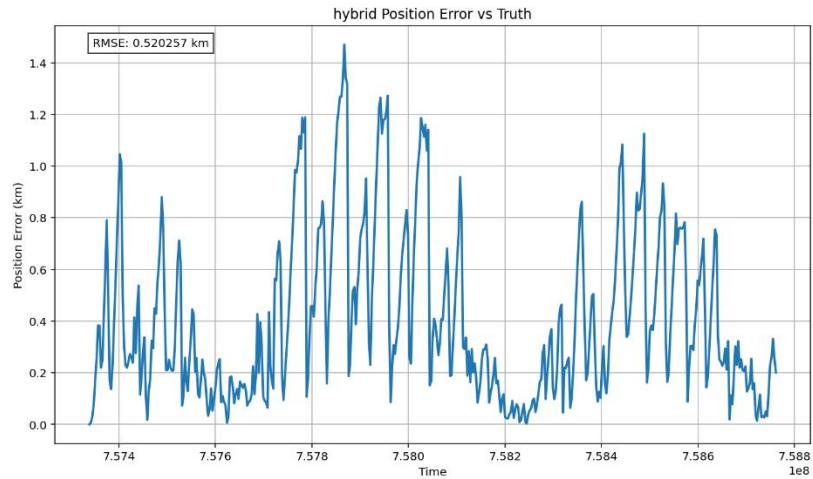


Figure 21. EKF-UKF (Time Based) 3,000-Second Time Step Position Error Plot

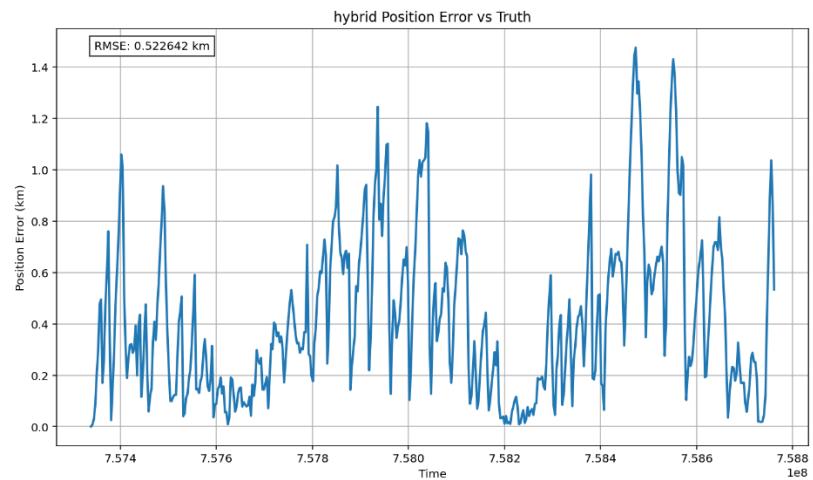


Figure 22. EKF-UKF (Entropy Based) 3,000-Second Time Step Position Error Plot

Monte Carlo Analysis

Monte Carlo analysis is a method in which the consistency of the filters are tested. This is achieved by doing a number of trials with random error applied and taking the resulting mean and standard deviation [40]. The resulting normal distribution plots were created with the aid of Microsoft Excel. This is carried out to confirm that the results of the filter analysis above to be valid, i.e. reproducible, and that these results do not occur by coincidence. As the shorter timesteps take considerable time to run, only the 3,000 time step filters will be tested for consistency. This should be adequate, as although the timesteps differ, the filters themselves are being tested. The number of trials taken for each filter was 5, and these results are presented below.

Table 19. Monte Carlo Simulation Results, Position RMSE

Filters		Position RMSE (km)						
		Trials					Mean	STD
EKF		0.567201	0.557353	0.560056	0.547042	0.566071	0.559545	0.008102
CKF		0.557879	0.547147	0.548917	0.543396	0.554396	0.552769	0.009343
UKF		0.542566	0.545540	0.548336	0.547504	0.550187	0.546827	0.002909
EKF-	t-based	0.523341	0.509702	0.518969	0.518650	0.533847	0.520902	0.008772
	H-based	0.531927	0.539025	0.535448	0.530173	0.544096	0.536134	0.005600
EKF-	t-based	0.520257	0.515840	0.515074	0.517813	0.533524	0.520502	0.007551
	H-based	0.539399	0.540074	0.526277	0.526932	0.522642	0.531065	0.008087

Table 20. Monte Carlo Simulation Results, Computational Time

Filters		Computational Time						
		Trials					Mean	STD
EKF		3:36.26	4:41.69	4:15.85	4:21.79	4:05.40	4:12.19	0:26.41
CKF		32:20.83	31:58.89	32:02.17	31:04.01	30:26.70	31:34.52	0:41.89
UKF		31:08.33	28:10.27	34:54.60	31:54.61	33:21.42	31:29.85	2:23.96
EKF-	t-based	13:52.18	17:57.95	17:49.02	19:48.22	19:24.34	17:46.34	2:20.92
	H-based	24:50.31	17:02.33	28:17.77	28:37.95	28:22.87	25:23.21	4:54.51
EKF-	t-based	14:43.91	20:27.68	19:59.91	20:08.02	19:52.61	19:02.43	2:25.11
	H-based	24:42.10	25:01.27	23:36.68	23:34.38	23:48.54	24:08.59	0:40.28

B. Discussion

The evaluation of a filter's ability is split into three components: precision/accuracy, efficiency, and consistency. Accuracy is measured via the RMSE and error plots, efficiency is measured via the computational time to run entirely, and the consistency is determined via Monte Carlo analysis. These metrics were measured and can be found in A. Results. The discussion section is split into these three components.

Accuracy

Accuracy is highly important for NRHO OD. As aforementioned, it can dictate the outcome of the mission, as ill-informed manoeuvres may cause significant divergence from the desired NRHO path. For the 3,000, 1,500 and 300 second time steps, unsurprisingly, with increasing data gaps between each data point, the RMSE also increased across all filters. Interestingly, however, there was an exception to this trend, with 30 second time steps only continuing to decrease in RMSE for the EKF, and instead increased compared to the 300 second time step. These results are plotted below in *Figure 23*.

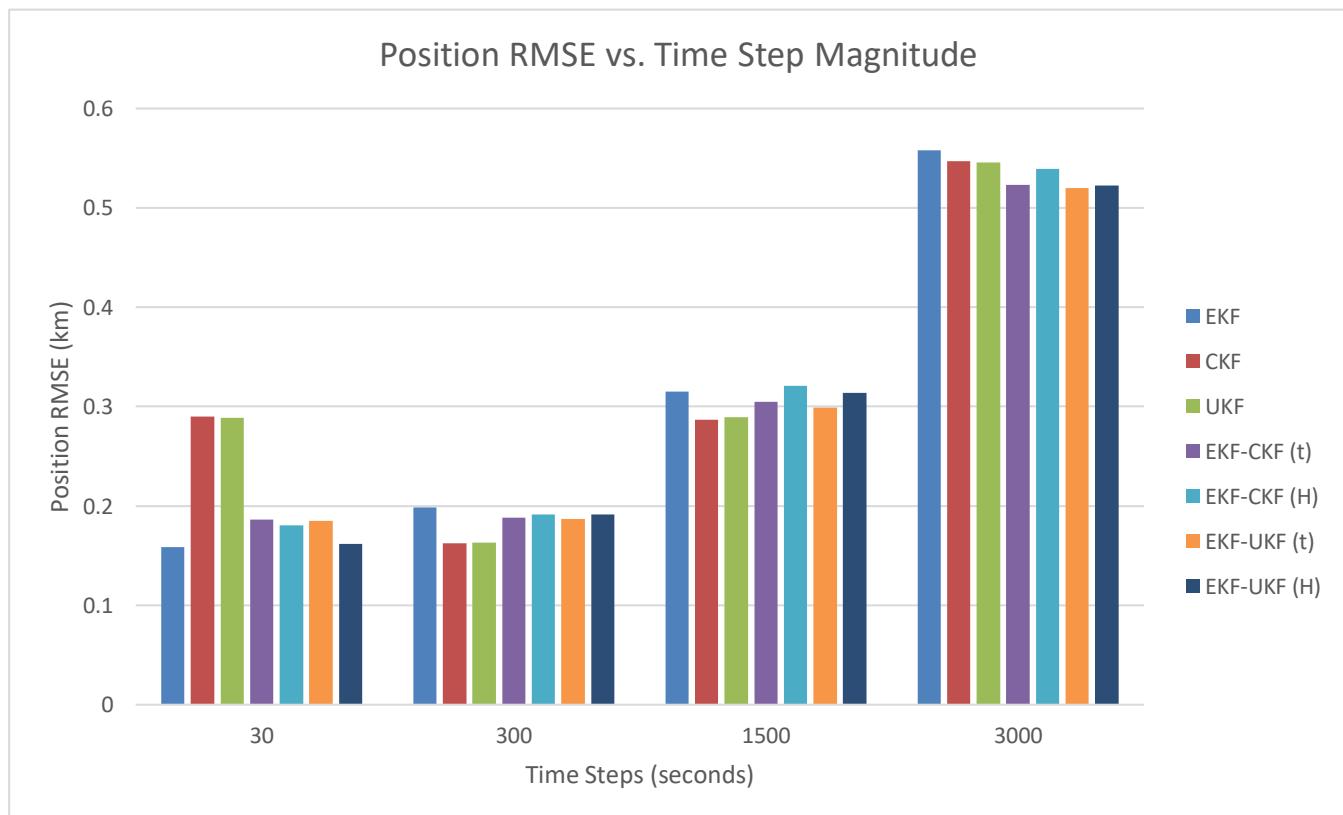


Figure 23. Overall Position RMSE For Every Filter vs. Time Step Magnitude

Below, the results with each perilune passage are presented. This plot takes the mean RMSEs of each varied timestep to plot comparative results across all of the filters. Note that this excludes the 30-second time step exception, which will be analysed separately.

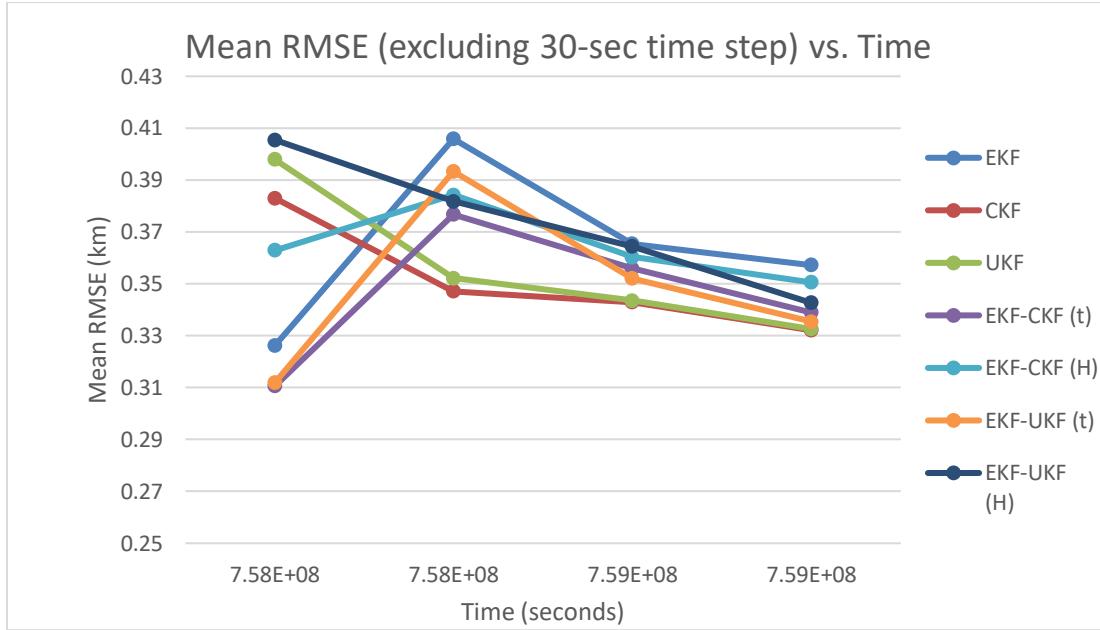


Figure 24. Mean RMSE (except 30 second time step) Against Time For Overall Comparative Results Between Filters

As can be seen, the CKF and UKF perform very similarly to each other across all stages of the mission, with the RMSE decreasing with each perilune passage. This performance was expected, as the differences between the CKF and UKF algorithms are minimal. The most significant difference is that the CKF is not tuned at all, and evenly weighs the 12 propagated points. On the other hand, the UKF propagates one extra point and weighs the points evenly except one, which is weighted higher than the rest as it is the point directly propagated from the last state and is the most likely outcome. As expected the UKF marginally outperformed the CKF in all cases. The most significant decrease is seen between the first and second perilune passage. This is likely due to where the mission trajectory starts. As it starts in a highly linear section of the orbit, generating a spread of points produces less accurate results than simply a direct propagation, confirming that often EKF and its linearisation assumptions are often better suited for stable orbits. The significant decrease in RMSE, compared to all the other filters involving EKF which increase after the first PP, also reinforces the effectiveness of the CKF and UKF, as although the overall position RMSE lands within the magnitudes of tens of metres within each other, having a more consistently accurate filter is far more important than a volatile one that may reconverge after perilune. This is due to

the importance of accurate OD throughout the mission rather than just the overall error at the end, as manoeuvres may need to be performed at an stage in the mission, not just at the final overall state.

Unsurprisingly, the EKF performed worse than all of the filters. Noticeably it possessed the smallest average RMSE out of all the filters before the first perilune passage on average, due to the highly stable dynamics. This likely beat the CKF and UKF because the Gaussian dynamics did not require a non-Gaussian solution, thus the propagation of additional points actually produced a negative impact on this portion of the NRHO orbit. After the first perilune passage, the EKF saw significant spike in error, on average 100 metres. The overall RMSE did decrease, however the significant spike meant that across the mission, the accuracy was very inconsistent despite the similar final result. These results are also reflected in the error plots, with the CKF and UKF seeing overall lower peak error by up to a magnitude of 100 metres. Additionally, the width of these spikes are much narrower for the CKF/UKF compared to the EKF, demonstrating the filters' ability to recover quickly from these passages.

This pattern of a spike in RMSE after one perilune passage and a steady decrease despite additional perilune passages is an interesting outcome, and an unexpected one, with [5] demonstrating filter degradation with every perilune passage. However, this is not entirely unexplainable. This is most likely to do with the stable section orbit duration, with the longer the stable section runs, the more likely the filter is to converge with the truth. The first perilune passage occurs early on in the mission duration, and as such, the filter does not receive a chance to fully converge before reaching the unstable section of the orbit. As such, the overall RMSE between the mission start and the first perilune passage is higher than the subsequent perilune passages as an entire orbit has been carried out, all the way to apolune.

The hybrid filters' relative performance varied across the differing time steps, as well as between the time-based and entropy-based applications. For the time-based hybridisation, the greatest benefit was seen across the largest (3,000 second) time step, outperforming all of the 'pure' filters. Both the EKF-CKF and EKF-UKF followed a similar pattern to the EKF regarding a spike in error after the first perilune passage, as it too utilised the EKF first. However, the overall performance seemed to diminish for the 1,500 and 300 second time steps, despite the same perilune passage time intervals applied for each time step case. This indicates that with greater time steps, the filters require more time to reconverge.

The improved performance for the hybrid filters is indicative of the EKF's superior performance in stable orbits, with linearisation being sufficient to capture the steadily changing dynamics. The CKF and UKF application to the perilune passages also successfully reduced the spike in error at these stages, thereby

improving accuracy. The EKF-UKF performed marginally better than the EKF-CKF, as expected. Interestingly, the relative performance of the hybridised time-based filters seemed to diminish with the more information they received, i.e. smaller time steps. These filters outperformed the CKF and UKF for the 3,000 time step test, and overall on average matching the CKF/UKF performance. The diminishing performance with decreasing time steps indicates that the EKF had greater difficulty at reconverging with the truth trajectory with the increased data points exiting the perilune passage, thus indicating that a larger period of ‘perilune passage’ may be required to ensure that the EKF component of the filter is sufficiently in a stable region of the orbit.

The entropy-based hybrid filters performed similarly to the time-based hybrid filters, with similar overall RMSE; however, all entropy-based results saw a slight increase in RMSE compared to the time-based results. The reason for this increase is due to the tuning of the entropy-based filters. As it was expected that the CKF and UKF would perform better than the EKF, the entropy thresholds were set such that they would switch more liberally, increasing the accuracy of the filters going into the perilune passage. However, the results suggest that increasing the duration of EKF is more beneficial to the RMSE than reducing it, and as such the thresholds should have been increased to switch more conservatively, reducing the period of perceived instability rather than increasing it.

The ‘burn-in’ steps required to produce results that did not spike significantly at filter swaps had to be calibrated differently at different time-step magnitudes, indicating that the more frequent the data, the longer it took for the filter to reconverge with the truth trajectory after perilune passage. This is likely because there are more data steps in these volatile stages, thus the dynamics are unstable for longer. This results in covariance updates from the switch reset take longer to reconverge, especially for the EKF with the burn-in going through unstable dynamics back to stable dynamics. This is likely another factor contributing to why the hybrid filters saw a diminishing advantage with decreasing time step magnitude. Interestingly, increasing the transition noise factor also aided in maintaining optimal performance of the hybridised filters as time step magnitude decreased, despite overall less information being provided to the filters.

Thus, the filter divergence when the algorithms are swapped with smaller timesteps is actually greater compared to when the hybrids run with larger time steps. This indicates that for lesser time-steps, the covariance reset needs to be more conservative to prevent full filter divergence and the number of burn-steps also needs to be larger to allow for the filter to reconverge adequately to ensure sufficient performance. This was reflected in the calibration of the filters.

There was a significant exception to these results in the case of the 30-second time steps. It was expected that these results, due to having the most information, would produce the lowest RMSE; however, this was only the case for the EKF. Below, in *Figure 25*, the comparative results with each perilune passage are presented for the 30-second time steps.

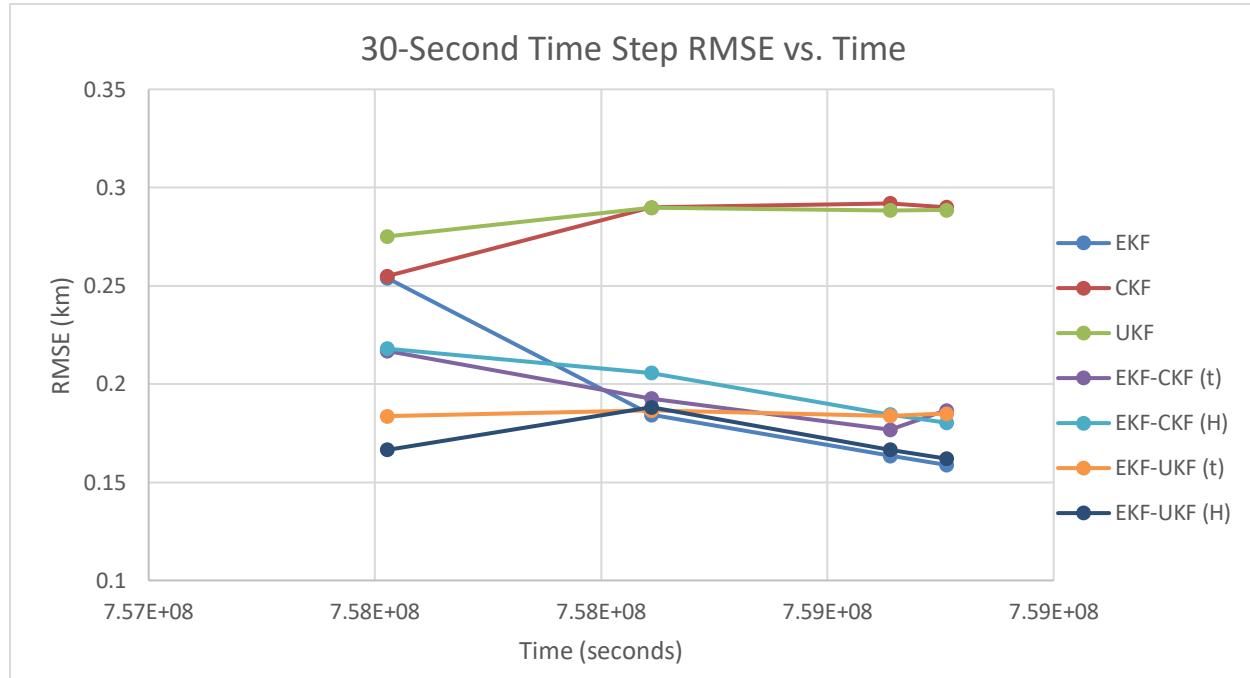


Figure 25. Comparative RMSE Throughout Orbit (30-second Time Steps Only)

Contrary to the other cases, the EKF outperformed the rest, with the overall RMSE of rest of the filters involving the CKF and UKF diminishing significantly compared to the 300-second time steps. Note that the CKF and UKF remained the most consistent throughout the mission. The hybridised filters also followed suit, albeit to a much lesser extent, as for the time-based filters majority of the hybrid runs the EKF is utilised. For the entropy filters, interestingly, no switches occurred, indicating that the uncertainty never significantly increased for the EKF, further supporting that the shortest time intervals could handle the unstable dynamics. The EKF continuing to improve was expected, as the increasingly short time steps reduced the time across the state propagation was being linearised, leading to more accurate results. The CKF and UKF, on the other hand, although showing similar peaks at the perilune passages, had consistently higher error throughout the runs. This indicates that with such small time steps, propagating a spread of points and taking the mean actually increases the error rather than reducing it, as the direct propagation from the last state is significantly more likely to occur than any other possibility. Thus, taking a spread of expected states without significantly reducing the weighting of the subsequent results

results in higher error. As the UKF more heavily weighed the direct propagation over the rest of the spread, rather than the CKF with equal weightings, the UKF was marginally more accurate.

Efficiency

Efficiency is the other determining factor when it comes to selecting an optimal filter. This is measured by the overall computational time it takes to run through the entire program with each filter. The results from this are presented in *Figure 26*.

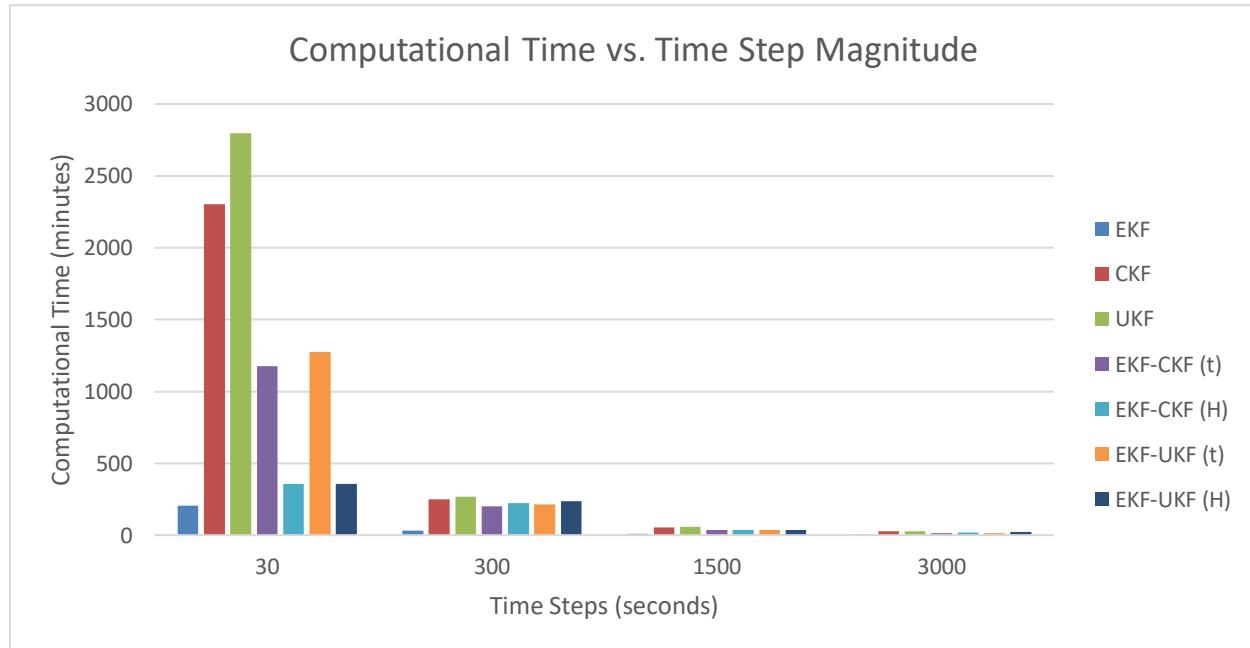


Figure 26. Computational Time Comparison Graph

None of the results presented above are unexpected. As the time step magnitude decreases and the number of data points being given to the filters is increased, the total computational time increases at a rate directly proportional to the number of points being processed.

For the main three pure filters, their comparative computational time is maintained across all varying time steps. The EKF is significantly faster than any of the other filters, being up to ten times faster than the CKF and UKF. This was predicted, as the linearisation process for propagating to the next time step is significantly faster as it only requires one full-fidelity propagation, compared to the CKF and UKF with 12 and 13 propagations per time step respectively. The relative computational times between time steps seemed to decrease, with the 30-second and 300-seconds having the greatest difference between the EKF and the CKF/UKF, at nine times faster. This decreased to approximately five times faster for the 1,500

second timestep and four times faster for the 3,000 second timestep, indicating exponential growth for runtime corresponding to decreasing timesteps.

As for the hybrid filters, they performed as anticipated, being less efficient than the EKF significantly, but also much more efficient than the CKF and UKF. The improvement on efficiency seemed to diminish as time step magnitudes were decreased, particularly for the time-based hybrid filters. This is likely due to how they were constructed and calibrated. When the filter swaps between the selected stable and unstable algorithms, as the two algorithms handle covariance differently, this covariance was not translatable and could not simply be passed over. Instead, as outlined in methodology, the covariance was reset to the initial covariance, multiplied by a standard transition noise factor as determined via trial and error to optimise the filter performance with each varied time step. This means that the filter must reconverge with the truth trajectory, taking a few iterations for the filter to do so successfully. As such, a ‘burn-in’ period is introduced, running some of the last time interval and discarding these results before the next filter results are stored. This is crucial, as the number of burn-in steps had to be increased as the time step magnitude decreased, thus increasing the overall number of time-steps propagated significantly, and increasing the relative time the hybridised filters took to fully run.

The entropy-based hybrid filters were significantly less efficient than the time-based hybrid filters. This is because filter swapping was allowed to occur in accordance with the entropy of the state (i.e. the metric of uncertainty) rather than strict bounds, and the thresholds were set to allow for more CKF/UKF usage. This meant that larger periods of the mission were filtered using the less efficient ‘unstable filters’. Comparatively, the EKF-CKF and EKF-UKF time and entropy variants were similarly efficient, as expected, as the differences between the CKF and UKF were minimal. The EKF-UKF performed overall less efficiently due to the extra point propagated and the more complex weighting processes.

Consistency

Consistency is the final metric to test the filters, with these results needing to be reproducible and not a consequence of lucky random error. As previously stated, it is reasonable to assume that the filter performance of the 3,000 timestep should reflect the consistency of the filter algorithms in general as the algorithms do not change, with decreases in time step magnitude leading to improved performance regardless. The consistency is analysed by taking a sample of five and approximating a normal distribution. The standard deviation indicates the consistency of the filtering algorithms. The normal distribution plots are presented below in *Figure 27*.

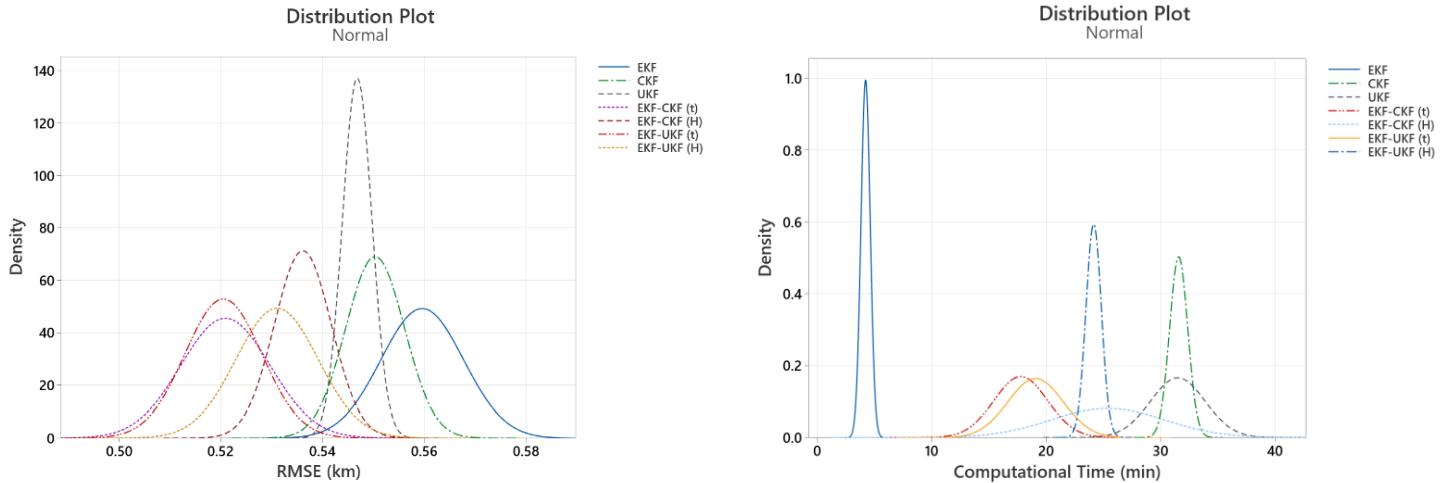


Figure 27. Normal Distribution For Filters, Position RMSE (left), Computational Time (right)

From the normal distribution plots for the RMSE (i.e. filter accuracy), the filters generally had similar standard deviations, except the UKF which seemed to be significantly more consistent than the rest, with the CKF being more consistent than the other filters as well, albeit to a lesser extent. This may indicate that the UKF handles the unstable regions of the mission significantly better than the CKF, EKF and hybrid filters. However, as this analysis was only taken over five trials, it could also be coincidental, and should be further investigated with a greater number of trials. The hybridised filters seemed to generally have similar standard deviations to the EKF, and as the majority of the hybrid filters utilise the EKF, this result seems reasonable. Additionally, from this plot, it seems that there is significant overlap between many of the filters, particularly between the EKF-CKF and EKF-UKF for their time-based and entropy-based approaches, as well as between the CKF and UKF, indicating that the CKF and UKF in the NRHO context have minimal difference in performance, other than the UKF potentially producing more consistent results.

Also displayed are the normal distribution plots for the computational time. These results were significantly more inconsistent compared to the accuracy, with standard deviations for efficiency varying significantly. This could be due to the lack of dedicated control computer, with certain simulations being carried out simultaneously and other programs running in the background, thus affecting the results by a number of seconds to up to several minutes. The consistency of the efficiency also seemed to loosely correlate with the overall runtime for each filter, with the EKF being highly consistent compared to the rest of the filters, likely because each run was finished so quickly, the effects of the additional processes

were minimised. It should be noted, however, that the means for the computational times aligned with what was expected, as discussed in the efficiency section.

V. Conclusions

OD, concerned with accurately estimating a spacecraft's state, is a field long studied; however, unique difficulties arise in the case of NRHOs due to instability at perilune passage. To address this unique challenge three classical solutions were implemented: the EKF, CKF and UKF. Hybrid solutions were created with two methods: time-based switching (filters swap within 50,000 kilometers of the Moon) and entropy-based switching (more sensitive to increases in instability, increasing CKF/UKF use).

The CKF and UKF consistently outperformed other filters in terms of accuracy, except for the 30-second time step case, where they performed significantly worse. This indicates that for such short time intervals the EKF is sufficient, suggesting that an alternative approach to improving accuracy may be to increase communication frequency at perilune passage. The EKF dominated in efficiency, being up to ten times faster than the CKF/UKF. The hybrid filters achieved promising balances, significantly improving efficiency and generally improving upon the EKF results. However, their relative advantages diminished as time step magnitude decreased. Consistency analysis suggested that the UKF was the most reliable, though limited sample size may warrant further investigation.

The time-based hybrids were more efficient and accurate than entropy-based variants, indicating that increasing the CKF/UKF relative use was not necessarily beneficial. Time-based hybrid filters successfully outperformed the CKF and UKF for the 3,000 second time step.

From this study, it was determined that the EKF could be utilised in an NRHO context within hybrid filters to improve efficiency and could be sufficient with continuous communications. Hybridisation offers significant benefits, with CKF and UKF providing options for lengthy data outages and perilune passages, however, their full-use is unnecessary given efficiency trade-offs.

To further improve and develop upon this study, Monte Carlo should be performed with large samples and testing of different entropy and time based criteria should be compared and evaluated. To further reduce the effects of perilune passage on filter divergence, the implementation of adaptive noise for passage and varying filter transition to minimise burn-in steps to further improve upon hybrid accuracy and efficiency. With further investigation, and improvements in accuracy and efficiency, should see the increases in navigation success and mission safety.

References

- [1] D. C. D. R. J. W. J. R. G. a. M. S. R. C. P. Newman, “Stationkeeping, Orbit Determination, and Attitude Control for Spacecraft in Near Rectilinear Halo Orbits,” 19 August 2018. [Online]. Available: <https://ntrs.nasa.gov/citations/20180006800>.
- [2] “Why NRHO: The Artemis Orbit,” 2022. [Online]. Available: https://www.lpi.usra.edu/lunar/artemis/resources/WhitePaper_2023_WhyNRHA-TheArtemisOrbit.pdf.
- [3] T. G. a. A. F. B. Cheetham, “Cislunar Autonomous Positioning System Technology Operations & Navigation Experiment (CAPSTONE),” 3 November 2021. [Online]. Available: <https://arc-aiaa-org.wwwproxy1.library.unsw.edu.au/doi/epdf/10.2514/6.2020-4140>.
- [4] S. A. H. J. A. B. A. J. Dombard, “On the origin of macon basins on the Moon (and beyond),,” 15 January 2013. [Online]. Available: https://artscimedia.case.edu/wp-content/uploads/sites/181/2016/06/14222622/dombard_grl_2013.pdf.
- [5] C. O. E. K. D. C. D. M. R. Thompson, “Comparisons of Filtering Algorithms for Orbit Determination in Near Rectilinear Halo Orbits,” August 2022. [Online]. Available: https://www.researchgate.net/profile/Michael-Thompson-46/publication/362830002_Comparisons_of_Filtering_Algorithms_for_Orbit_Determination_in_Near_Rectilinear_Halo_Orbits/links/6303c747eb7b135a0e53d236/Comparisons-of-Filtering-Algorithms-for-Orbit-Deter.
- [6] M. B. M. T. N. P. R. a. D. C. D. C. Ott, “Range Biases, Measurement Noise, and Perilune Accuracy in Near Rectilinear Halo Orbit Navigation,” January 2022. [Online]. Available: https://www.researchgate.net/publication/357568569_Range_Biases_Measurement_Noise_and_Perilune_Accuracy_in_Near_Rectilinear_Halo_Orbit_Navigation.
- [7] S. Gehly, *Statistical Orbit Determination*, Melbourne: RMIT, 2016.
- [8] A. Barnett, “Orbits and Kepler's Laws,” 21 May 2024. [Online]. Available: <https://science.nasa.gov/resource/orbits-and-keplers-laws#:~:text=Kepler's%20Third%20Law%3A%20the%20squares,the%20radius%20of%20its%20orbit..>
- [9] “Kepler's laws of planetary motion,” [Online]. Available: <https://cdn.britannica.com/42/196742-050-C74B43FA/law-Kepler-motion-radius-vector-planet-lengths.jpg>.
- [10] Y. Y. A. D. Q. Granier, “Comprehensive review of orbit propagation for mission design in the cislunar domain,” UNSW, 2024.
- [11] R. L. P. Przyborski, “Measuring Earth's Albedo,” 31 December 2011. [Online]. Available: <https://earthobservatory.nasa.gov/images/84499/measuring-earths-albedo#:~:text=Ice%20cover%20cloud%20cover%2C%20and,property%20of%20the%20climate%20system.%E2%80%9D&text=Your%20browser%20can't%20play%20this%20video.&ext=An%20error%20occurred.,though%..>
- [12] M. J. V. a. D. C. Davis, “Examining the Feasibility of Relative-Only Navigation for Crewed Missions to Near Rectilinear Halo Orbits,” 25 August 2022. [Online]. Available: <https://ai-solutions.com/wp-content/uploads/2022/08/EXAMINING-THE-FEASIBILITY-OF-RELATIVE-ONLY-NAVIGATION-FOR-CREWED-MISSIONS-TO-NEAR-RECTILINEAR-HALO-ORBITS.pdf>.
- [13] “Extended Kalman Filter,” 2021. [Online]. Available: <https://www.sciencedirect.com/topics/computer-science/extended-kalman->

- filter#:~:text=Kalman%20filter%20equations%20are%20used,filter%20to%20the%20linearized%20equations..
- [14] A. M. T. Henderson, “Application of Kalman Filters In Orbit Determination: A Literature Review,” [Online]. Available: https://indico.esa.int/event/111/contributions/392/attachments/555/600/Application_of_Kalman_Filters_in_Orbit_Determination_A_Literature_Survey.pdf.
 - [15] K. Kitani, “Extended Kalman Filter,” [Online]. Available: https://www.cs.cmu.edu/~16385/s17/Slides/16.4_Extended_Kalman_Filter.pdf.
 - [16] “SquareRootInformationFilterOD Object,” [Online]. Available: https://aisolutions.com/_help_Files/squarerootinformationfilterod_nanosecond.htm.
 - [17] X. J. P. L. J. W. M. G. H. S. X. Zuo, “A square root information filter for multi-GNSS real-time precise clock estimation,” 2021. [Online]. Available: <https://satellite-navigation.springeropen.com/articles/10.1186/s43020-021-00060-0#:~:text=Real%2Dtime%20satellite%20orbit%20and,numerical%20stability%20and%20computational%20efficiency..>
 - [18] J. L. G. J. R. C. A. Mashiku, “Statistical Orbit Determination using the Particle Filter for incorporating Non-Gaussian Uncertainties,” 13-16 August 2012. [Online]. Available: <https://ntrs.nasa.gov/api/citations/20120016941/downloads/20120016941.pdf>.
 - [19] R. Yehoshua, “Gaussian Mixture Models (GMMs): from Theory to Implementation,” 29 November 2023. [Online]. Available: <https://towardsdatascience.com/gaussian-mixture-models-gmms-from-theory-to-implementation-4406c7fe9847..>
 - [20] V. C. J. Woodburn, “Analysis of Relative Merits of Unscented and Extended Kalman Filters In Orbit Determination,” 14 September 2019. [Online]. Available: <https://forum.orekit.org/uploads/short-url/ptF0WcWgPrFEtbA3Dg3lCAxuY0o.pdf..>
 - [21] “Cubature Kalman Filter,” 29 January 2023. [Online]. Available: <https://kalman-filter.com/cubature-kalman-filter/>.
 - [22] H. S. H. Y. J. Duan, “Square Root Cubature Kalman Filter-Kalman Filter Algorithm for Intelligent Vehicle Position Estimate,” 10 February 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S187770581600285X#:~:text=A%20new%20filtering%20algorithm%2C%20adaptive,poor%20convergence%20or%20even%20divergence..>
 - [23] M. J. J. Stauch, “Unscented Schmidt-Kalman Filter Algorithm,” January 2015. [Online]. Available: https://www.researchgate.net/publication/279292888_Unscented_Schmidt-Kalman_Filter_Algorithm..
 - [24] C. S. F. Caldas, “Machine learning in orbit estimation: A survey,” July 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0094576524001917..>
 - [25] G. H. B. M. W. L. K. Hill, “Satellite Orbit Navigation (LiAISON) In Lunar Halo Orbits,” August 2005. [Online]. Available: https://www.academia.edu/24081847/LINKED_AUTONOMOUS_INTERPLANETARY_SATELLITE_ORBIT_NAVIGATION_LiAISON_IN_LUNAR_HALO_ORBITS..
 - [26] J. R. H. D. C. D. C. P. Newman, “Perturbation Modeling And Navigation Tuning For A Crewed Station In Near Rectilinear Halo Orbit,” 7 February 2024. [Online]. Available: <https://ntrs.nasa.gov/citations/20240000552>.
 - [27] C. Rabotin, “Static noise compensation,” 2024. [Online]. Available: https://nyxspace.com/nyxspace/MathSpec/orbit_determination/snc/.
 - [28] S. F. S. A. E. B. N. L. A. H. Jazwinski, “Suboptimal Filtering Part 2: Compensation For Modeling Errors In Orbit Determination Problems,” May 1968. [Online]. Available: <https://ntrs.nasa.gov/api/citations/19680024356/downloads/19680024356.pdf>.

- [29] T. G. A. F. B. Cheetham, “Cislunar Autonomous Positioning System Technology Operations & Navigation Experiment (CAPSTONE),” 3 November 2021. [Online]. Available: <https://ntrs.nasa.gov/citations/20240004138>.
- [30] J. B. C. J. W. M. D. K. H. P. B. C. C. J. F. C. Channing Chow II, “Cislunar Orbit Determination: Improvements in Uncertainty Realism and Data Fusion,” [Online]. Available: <https://amostech.com/TechnicalPapers/2022/Poster/Chow.pdf>.
- [31] M. V. K. G. Y. Kulikov, “Do the Cubature and Unscented Kalman Filtering Methods Outperform Always the Extended Kalman Filter?,” 18 October 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896317308418>.
- [32] “Cubature Kalman Filter,” 29 January 2024. [Online]. Available: <https://kalman-filter.com/cubature-kalman-filter/>.
- [33] Q. Granier, “High-precision-Analyser-of-Lunar-Orbits,” 2024. [Online]. Available: <https://github.com/Quent2G/High-precision-Analyser-of-Lunar-Orbits>.
- [34] Y. Y. A. G. D. Q. Granier, “Comprehensive review of HALO, an high precision orbit propagation tool for mission design in the cis-lunar domain,” 20 September 2024. [Online]. Available: https://github.com/Quent2G/High-precision-Analyser-of-Lunar-Orbits/blob/main/High-precision_Analyser_of_Lunar_Orbit.pdf.
- [35] “Coverage Angle of a satellite,” [Online]. Available: <https://www.satnow.com/calculators/coverage-angle-of-a-satellite>.
- [36] C. Ho, “Interference Estimate from the Deep Space Network High-Power Transmitter at Goldstone with Nearby Third-Generation Mobile Users at 2 Gigahertz,” 15 February 2005. [Online]. Available: https://ipnpr.jpl.nasa.gov/2000-2009/progress_report/42-160/160A.pdf.
- [37] Hassan, “I want to plot graph for my intersatellite range rate. How can i do that,” 12 January 2024. [Online]. Available: <https://au.mathworks.com/matlabcentral/answers/2069206-i-want-to-plot-graph-for-my-intersatellite-range-rate-how-can-i-do-that>.
- [38] E. K. M. B. M. R. T. J. S. P. B. W. C. D. C. D. J. S. N. L. Parrish, “Near Rectilinear Halo Orbit Determination with Simulated DSN Observations,” [Online]. Available: <https://ntrs.nasa.gov/api/citations/20200011550/downloads/20200011550.pdf>.
- [39] S. Edmondston, “OD Filter Testing,” 2025. [Online]. Available: <https://github.com/stellaaaaaaa/OD-filter-testing/tree/main>.
- [40] C. Bernard, “How To Run Monte Carlo Simulation In Excel,” Datamation, 20 May 2024. [Online]. Available: <https://www.datamation.com/big-data/how-to-run-monte-carlo-simulation-in-excel/>.
- [41] J. R. Lynch, “Geodetic Coordinate Conversions,” 2002. [Online]. Available: <https://www.oc.nps.edu/oc2902w/coord/coordcvt.pdf>.
- [42] J. Stryjewski, “Coordinate Transformations,” 19 August 2020. [Online]. Available: https://x-lumin.com/wp-content/uploads/2020/09/Coordinate_Transforms.pdf.
- [43] Y. Yang, *AERO9500 Time and Coordinate Systems*, Sydney: UNSW School of Mechanical and Manufacturing Engineering, 2023.
- [44] V. P. & Engineering, “how to find barycenter (earth and the moon),” 2013. [Online]. Available: <https://www.youtube.com/watch?v=CnLwLF4e6uQ>.
- [45] R. B. B. Semenov, “Planetary Data System Navigation Node,” August 2025. [Online]. Available: https://naif.jpl.nasa.gov/pub/naif/generic_kernels/spk/planets/.

Appendices

A. Coordinate Transformations

Geodetic Coordinates To ECEF

Convert geodetic coordinates (i.e. latitude ϕ , longitude λ , altitude h) to ECEF cartesian coordinates [41]. Ensure geodetic coordinates are converted from degrees to radians. Note that these will only provide position coordinates.

Using the flattening factor of Earth f , calculate Earth's first eccentricity e :

$$f = \frac{1}{298.25}$$

$$e = 2f - f^2$$

Then, with f and the equatorial radius of Earth $a = 6378\text{ km}$, the radius of curvature in the prime vertical R_N is determined:

$$R_N = \frac{a}{\sqrt{1 - e^2(\sin \phi)^2}}$$

Finally, converting to Cartesian:

$$\begin{aligned} x &= (R_N + h) \cos \phi \cos \lambda \\ y &= (R_N + h) \cos \phi \sin \lambda \\ z &= (R_N(1 - e^2) + h) \sin \phi \end{aligned}$$

ECEF to ECI

To convert from ECEF to ECI, the coordinates have to be rotated to match the Earth's rate of rotation via a ‘rotation matrix’ [42]. This matrix is formed by determining the Greenwich Mean Standard Time (GMST) [43] at the time stamp. First the Julian time J_0 is determined, where t is in days:

$$J_0 = 2451545.0 + t$$

Then T_0 is determined for use in the GMST formula:

$$T_0 = \frac{(J_0 - 2451545)}{36525}$$

$$GMST = \theta = 280.46061837 + 360.98564736629(J_0 - 2451545) + 0.000387933T_0^2 - \frac{T_0^3}{38710000}$$

Then GMST is normalised to be within 0 and 360 degrees, then converted to radians. Thus the rotation matrix R can be formed:

$$R = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

To rotate the coordinates, simply multiply the position coordinates by the rotation matrix:

$$X_{position\ ECI} = R^T X_{position\ ECEF}$$

To determine the state velocities, take the cross product of the position states with the vector below, containing the angular rotational velocity of Earth ω in radians per second.

$$X_{velocity\ ECI} = \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \times X_{position\ ECI}$$

ECI to BCI

Finally, to convert ECI to BCI, simply determine the states relative to the Earth-Moon barycentre as so [44]. Note that the Earth (X_{Earth}) and Moon (X_{Moon}) states at each time were determined using the SpiceyPy library, utilising the de430.bsp Earth-Moon ephemeris kernel and naif0012.tls leap seconds kernel [45], the same ones used in HALO to ensure modelling matched.

$$X_{barycentre} = \frac{M_{Earth}X_{Earth} + M_{Moon}X_{Moon}}{M_{Earth} + M_{Moon}}$$

$$X_{Earth\ BCI} = X_{Earth} - X_{barycentre}$$

$$X_{BCI} = X_{ECI} + X_{Earth\ BCI}$$

B. Position Error Plots For All Runs

30-Second Plots

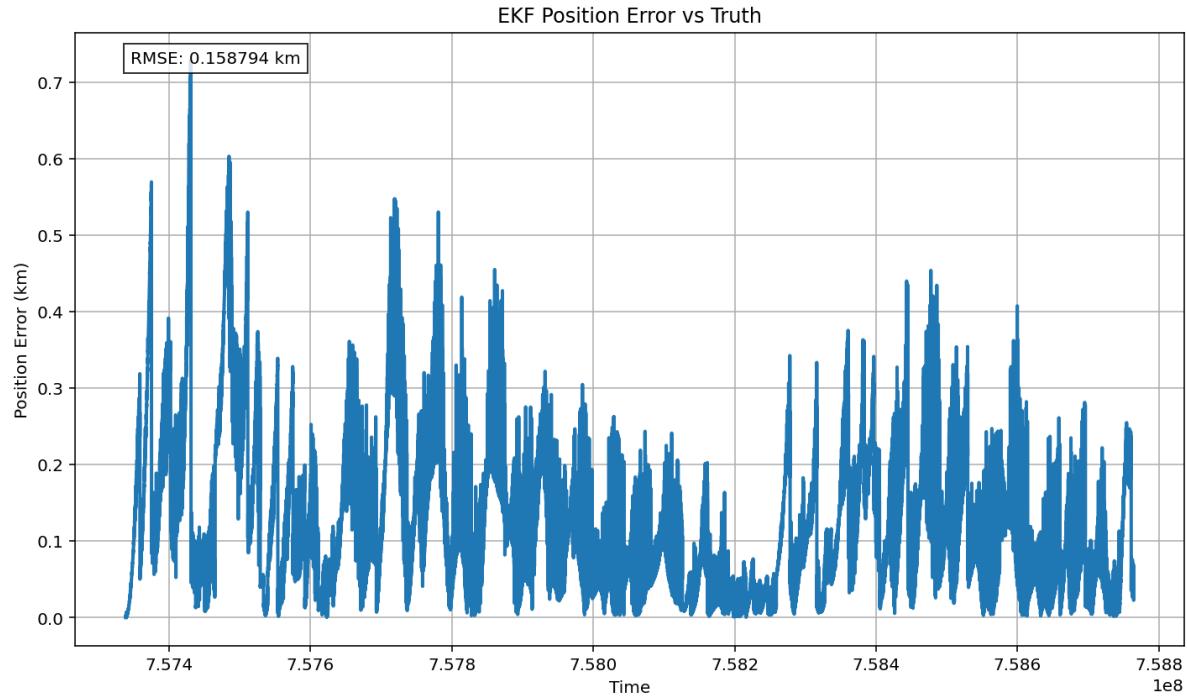


Figure 28. EKF 30-Second Time Step Error Plots, Position

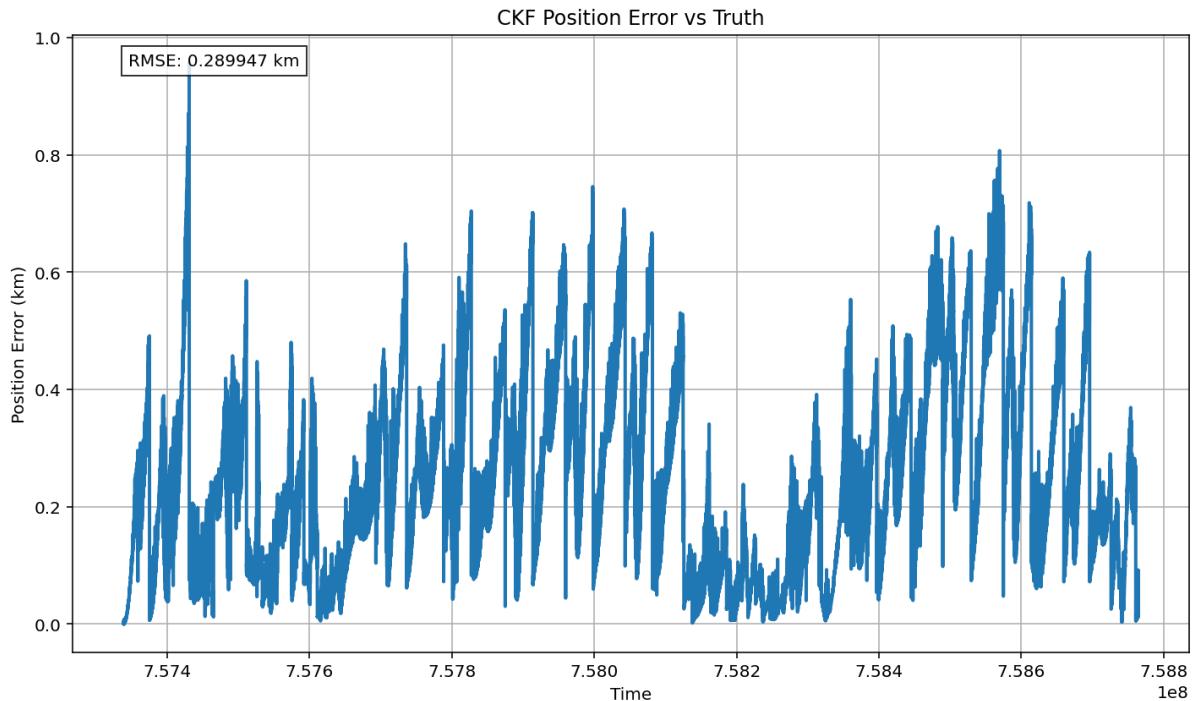


Figure 29. CKF 30-Second Time Step Error Plots, Position

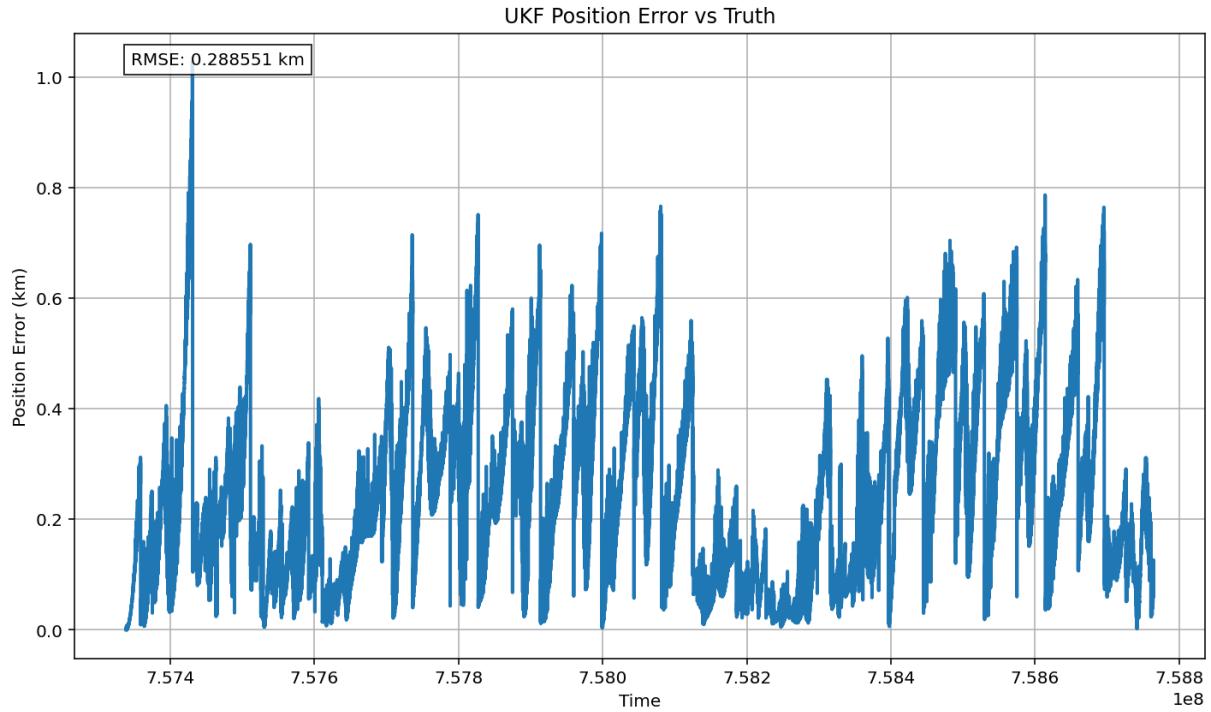


Figure 30. UKF 30-Second Time Step Error Plots, Position

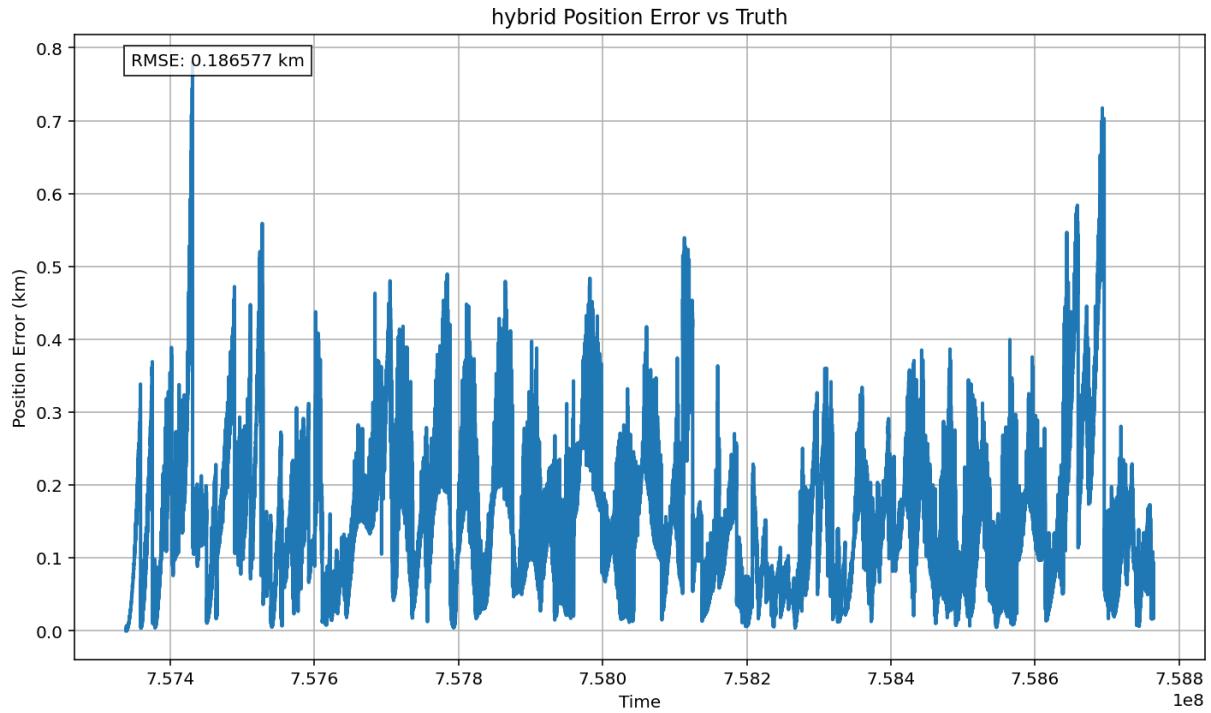


Figure 31. EKF-CKF (time-based) 30-Second Time Step Error Plots, Position

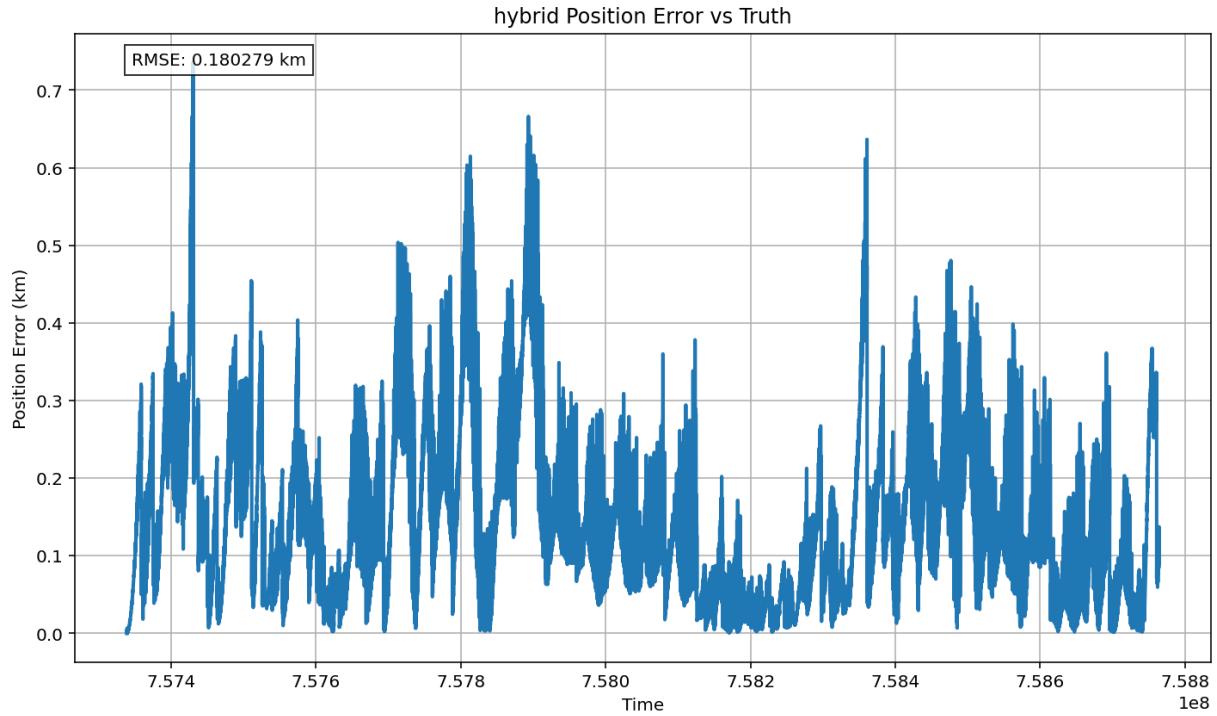


Figure 32. EKF-CKF (entropy-based) 30-Second Time Step Error Plots, Position

Figure 33. EKF-UKF (time-based) 30-Second Time Step Error Plots, Position

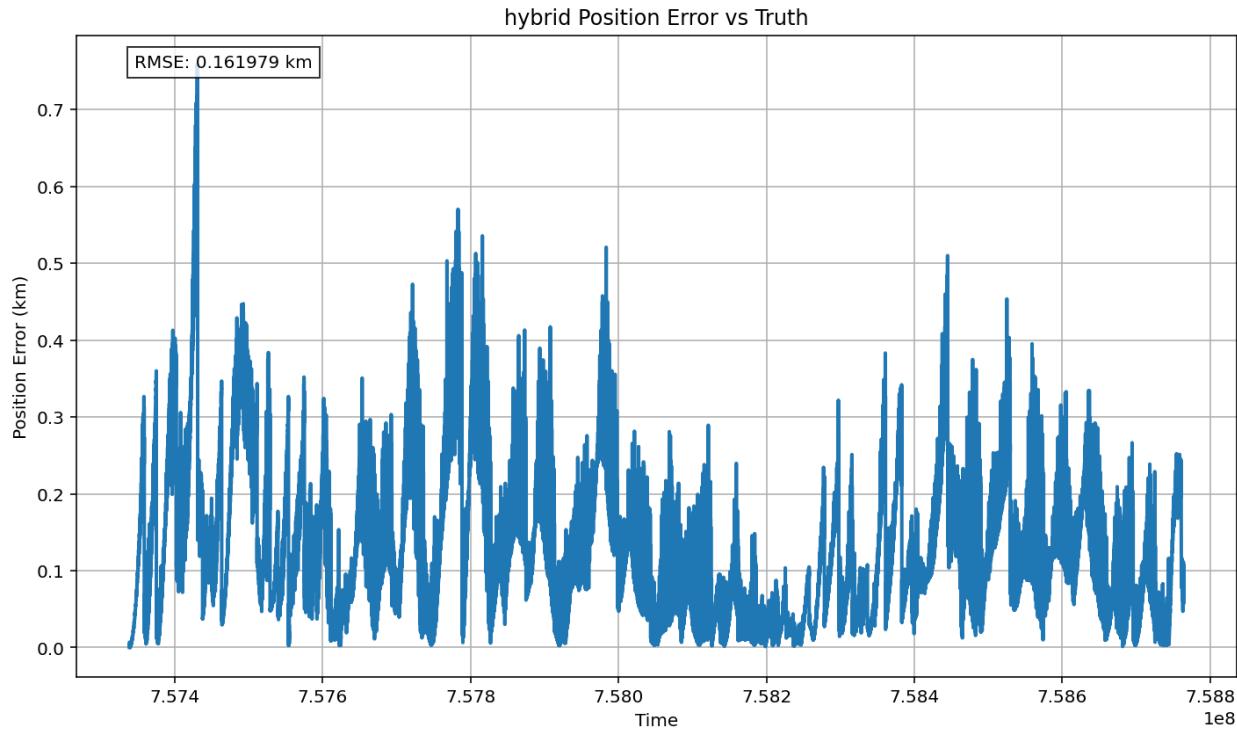


Figure 34. EKF-UKF (entropy-based) 30-Second Time Step Error Plots, Position

300-Second Plots

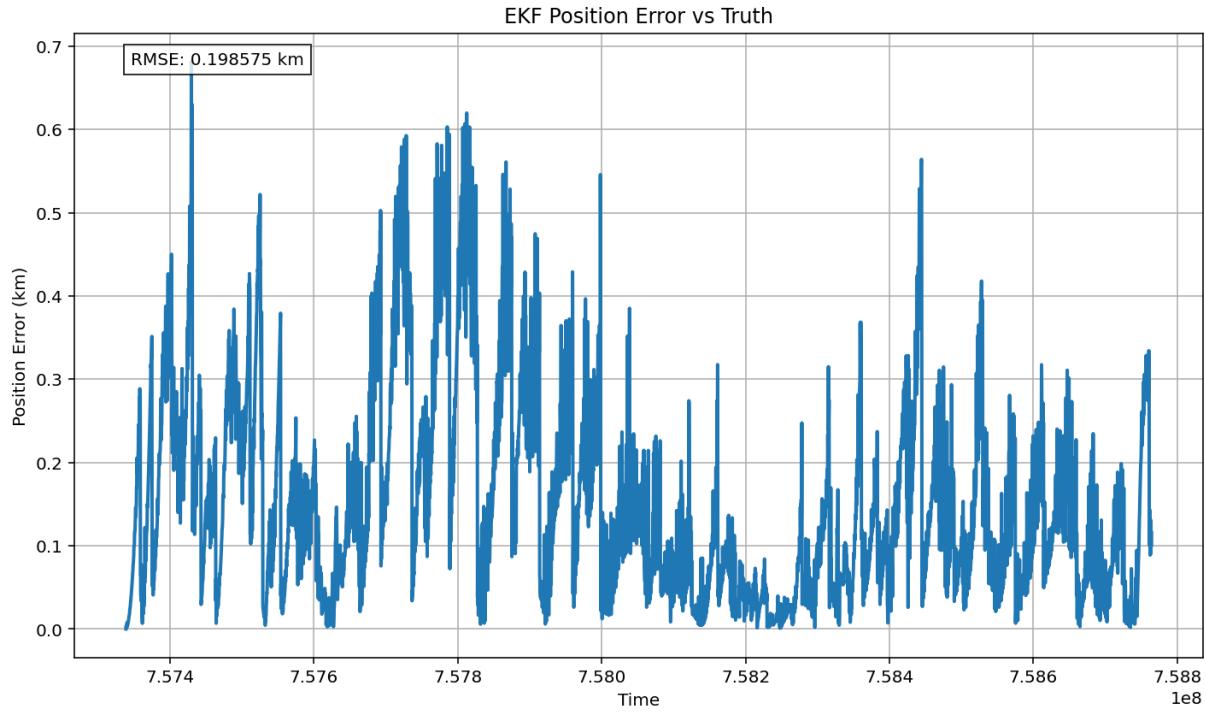


Figure 35. EKF 300-Second Time Step Error Plots, Position

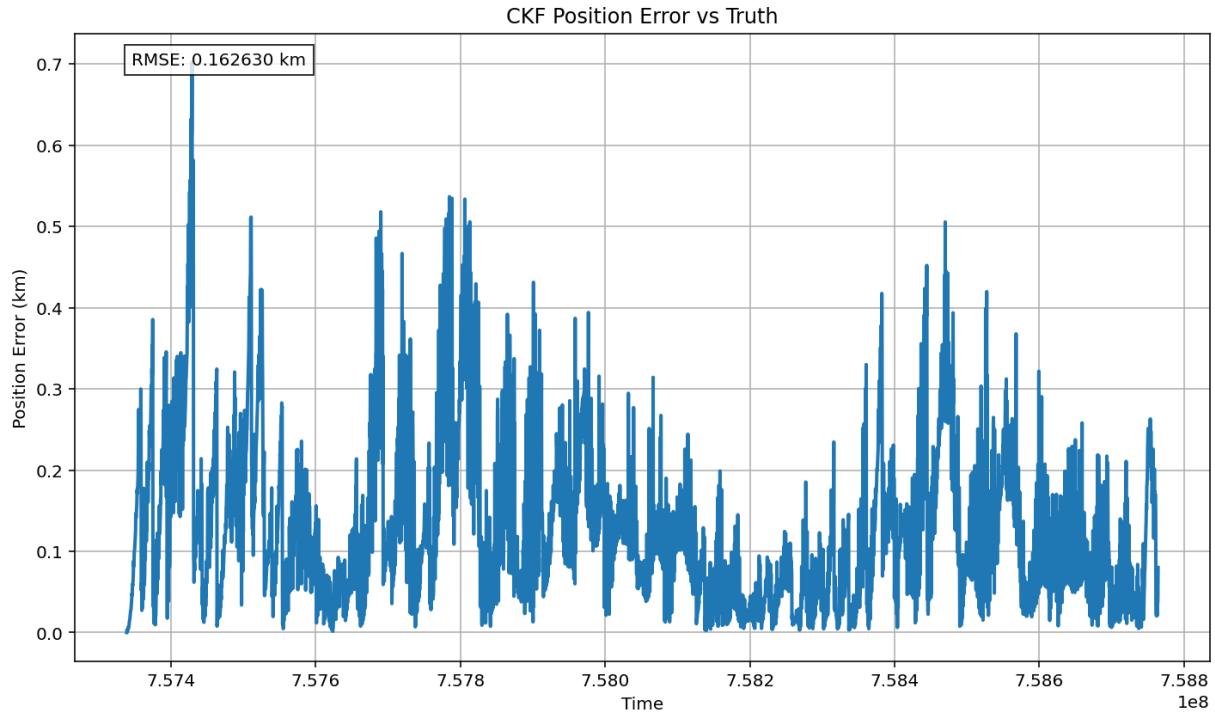


Figure 36. CKF 300-Second Time Step Error Plots, Position

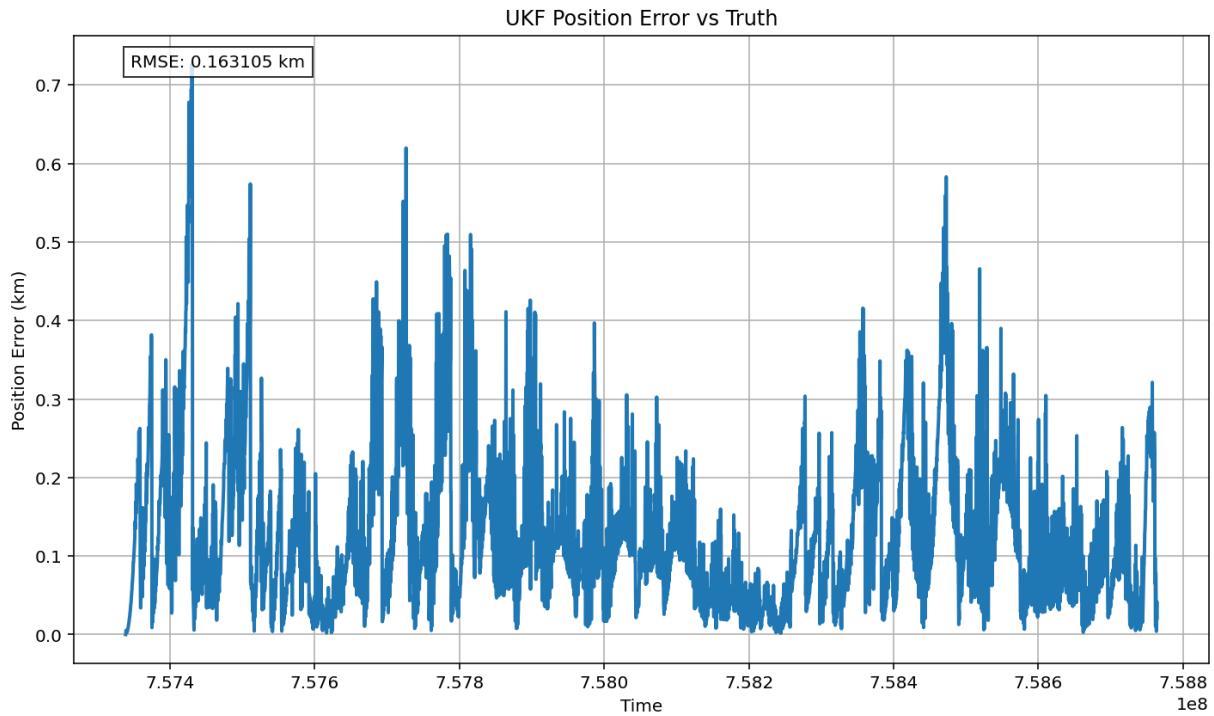


Figure 37. UKF 300-Second Time Step Error Plots, Position

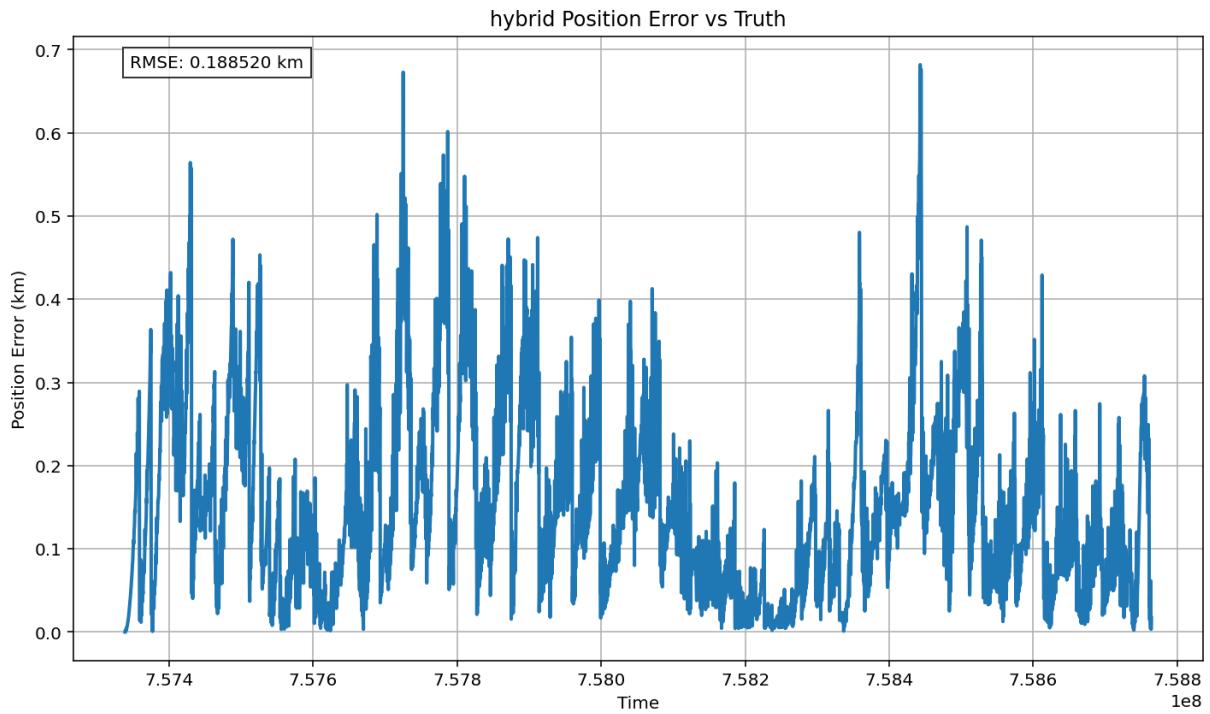


Figure 38. EKF-CKF (time-based) 300-Second Time Step Error Plots, Position

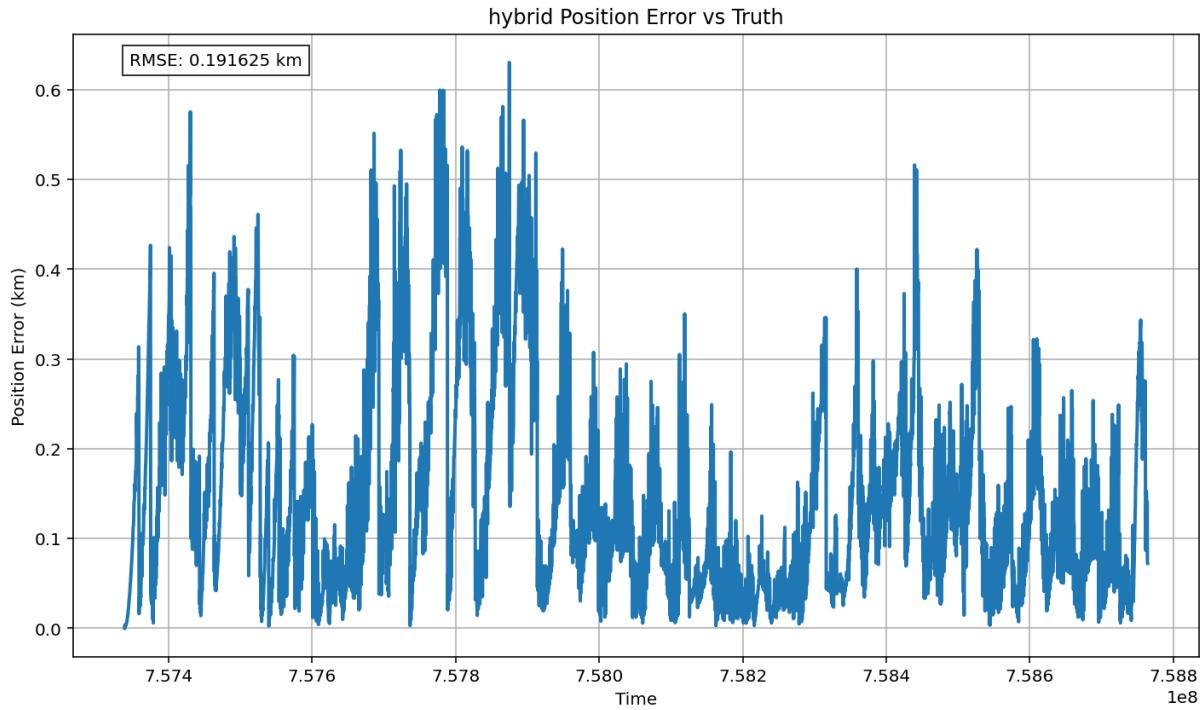


Figure 39. EKF-CKF (entropy-based) 300-Second Time Step Error Plots, Position

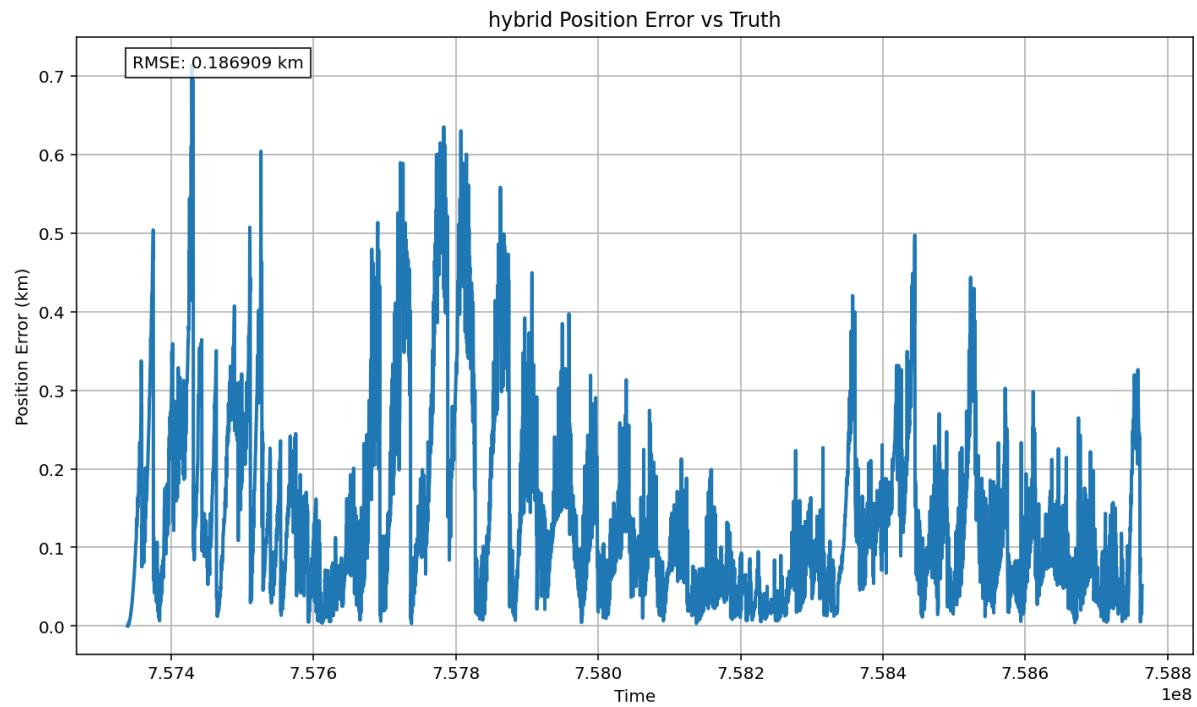


Figure 40. EKF-UKF (time-based) 300-Second Time Step Error Plots, Position

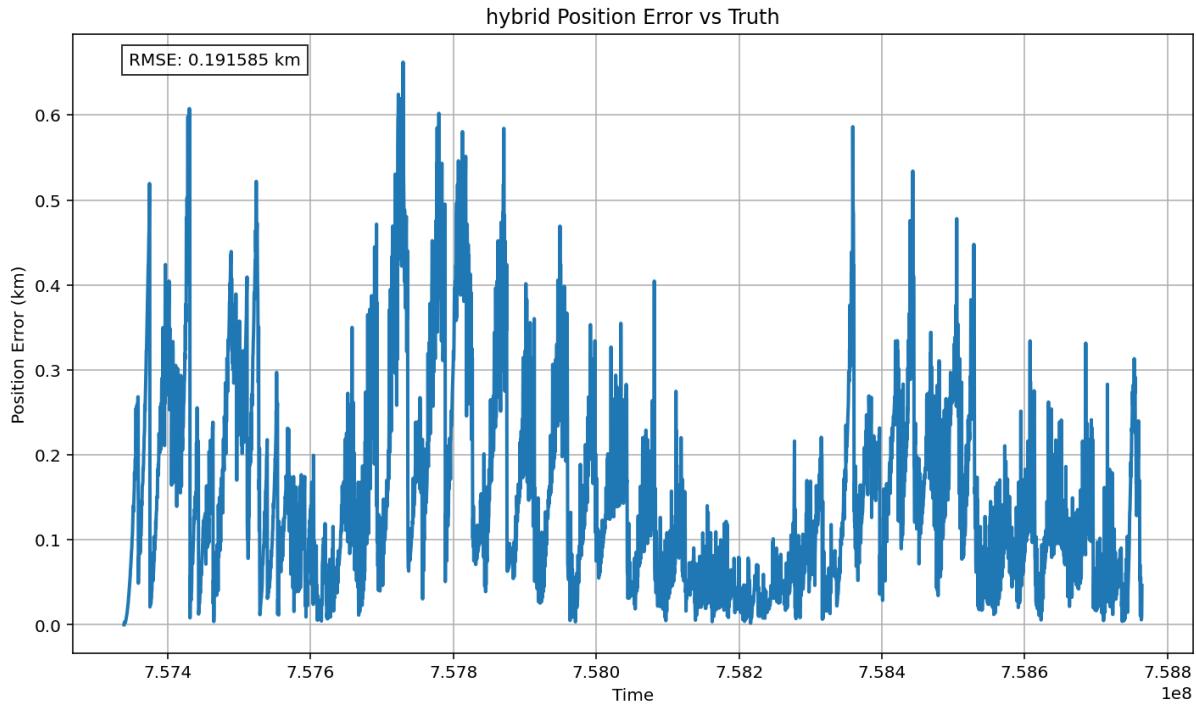


Figure 41. EKF-UKF (entropy-based) 300-Second Time Step Error Plots, Position

1,500-Second Plots

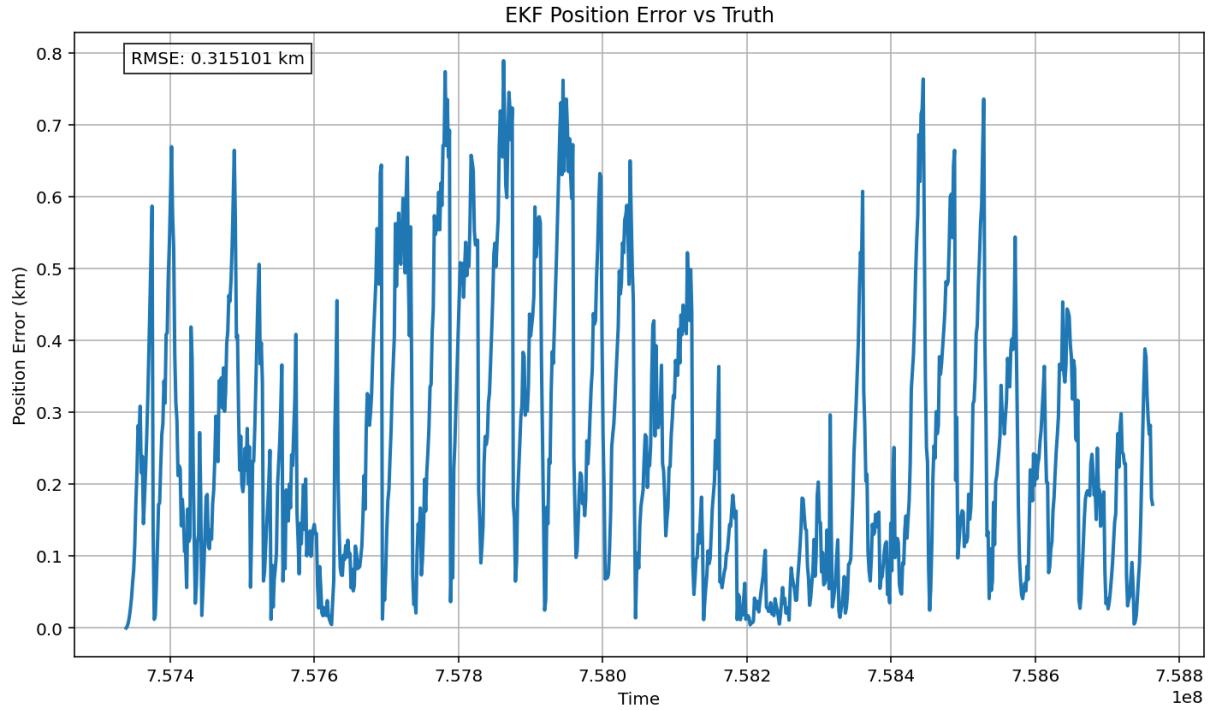


Figure 42. EKF 1,500-Second Time Step Error Plots, Position

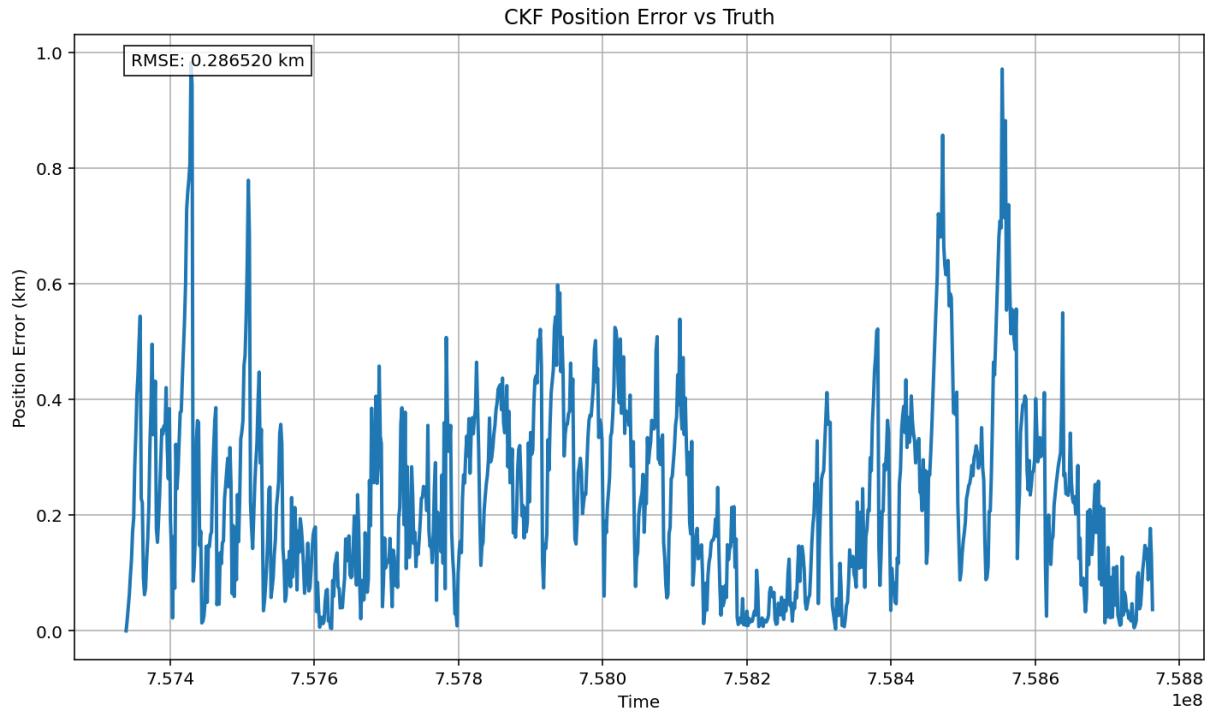


Figure 43. CKF 1,500-Second Time Step Error Plots, Position

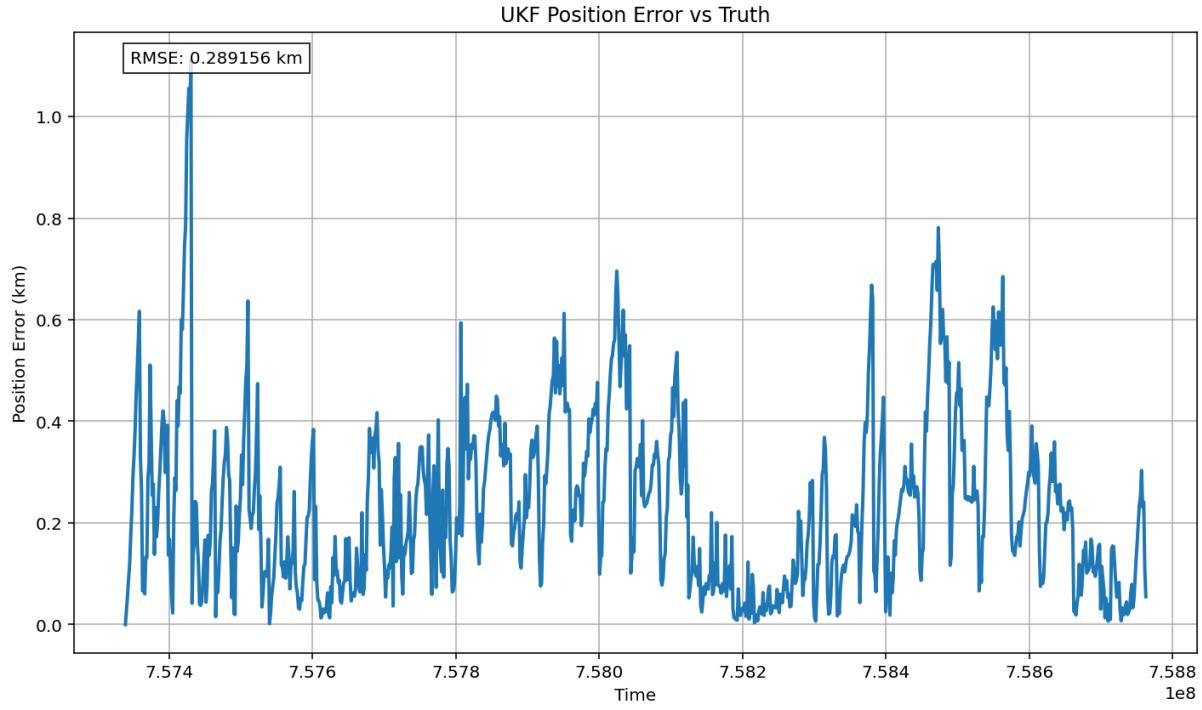


Figure 44. UKF 1,500-Second Time Step Error Plots, Position

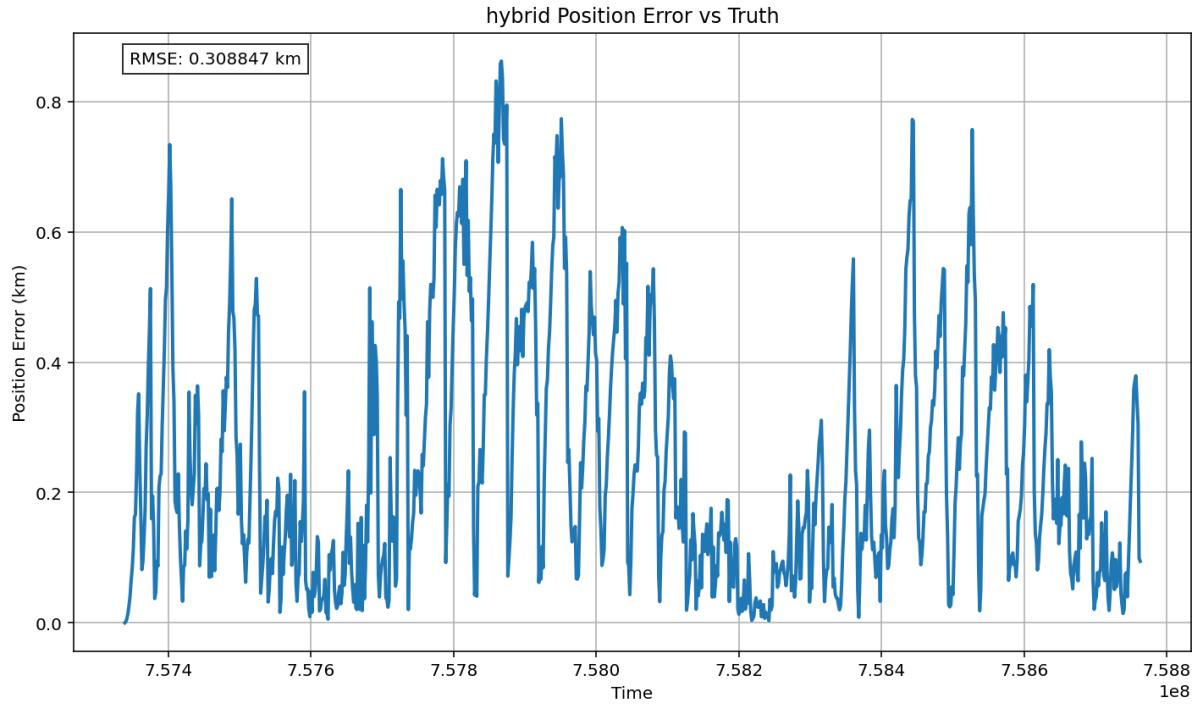


Figure 45. EKF-CKF (time-based) 1,500-Second Time Step Error Plots, Position

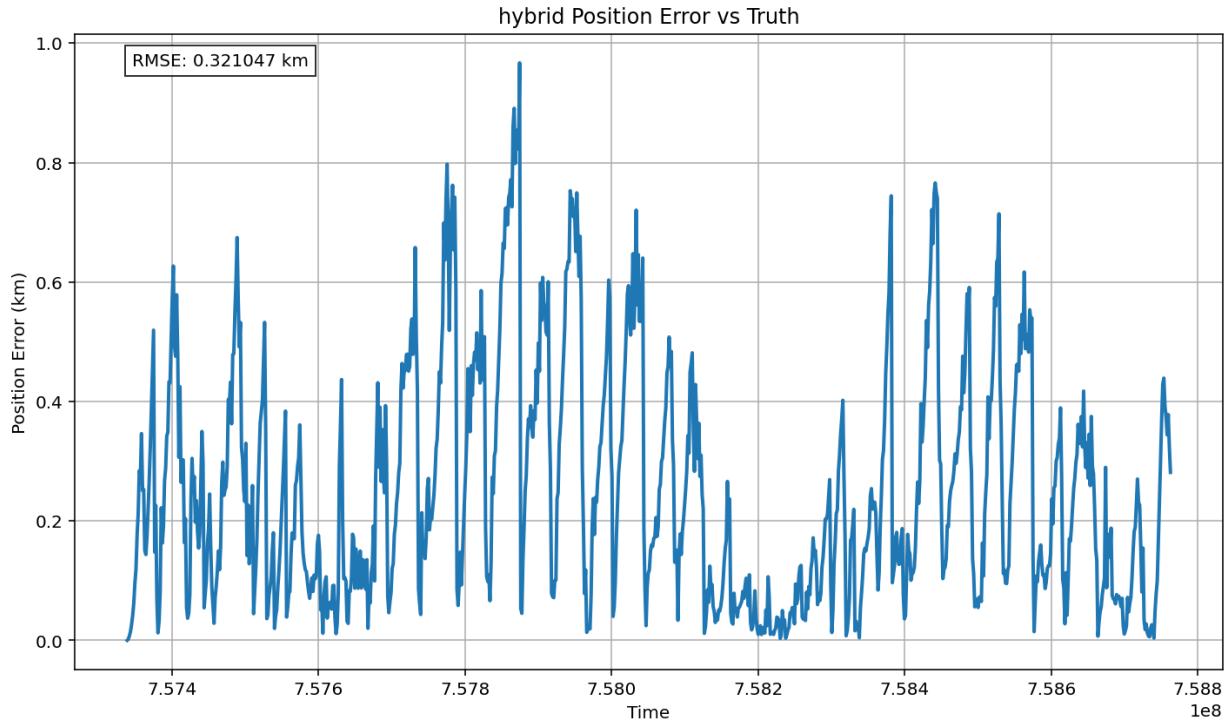


Figure 46. EKF-CKF (entropy-based) 1,500-Second Time Step Error Plots, Position

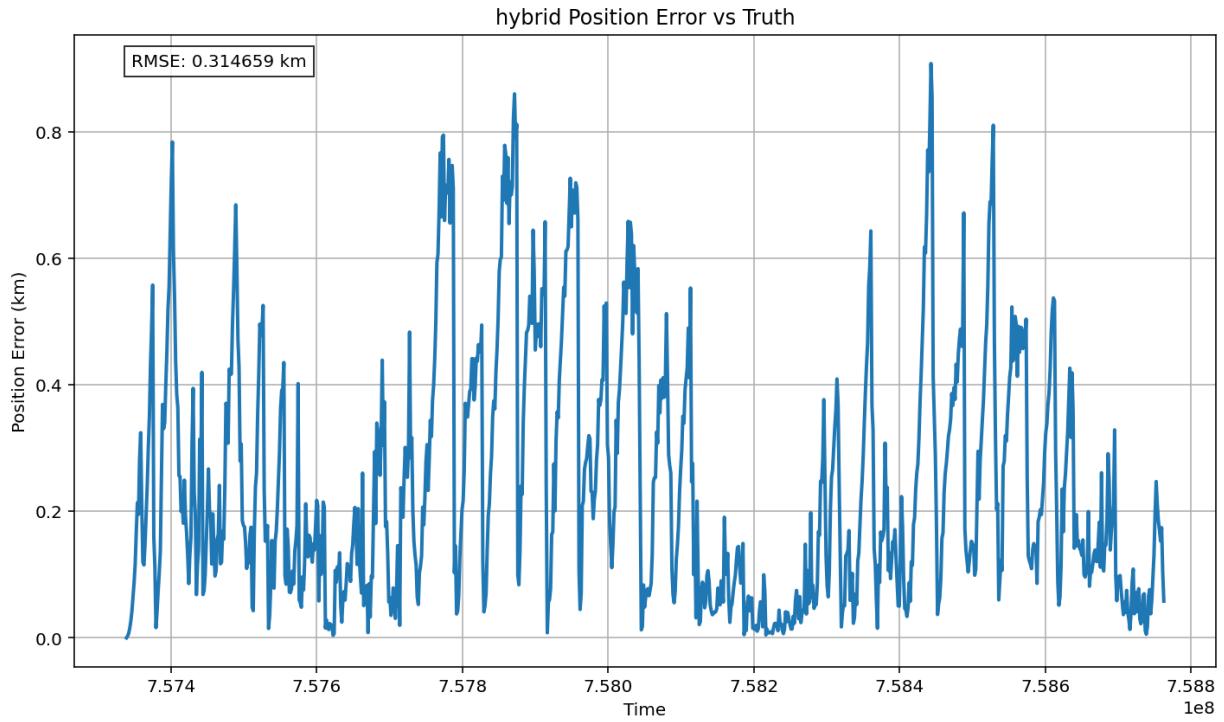


Figure 47. EKF-UKF (time-based) 1,500-Second Time Step Error Plots, Position

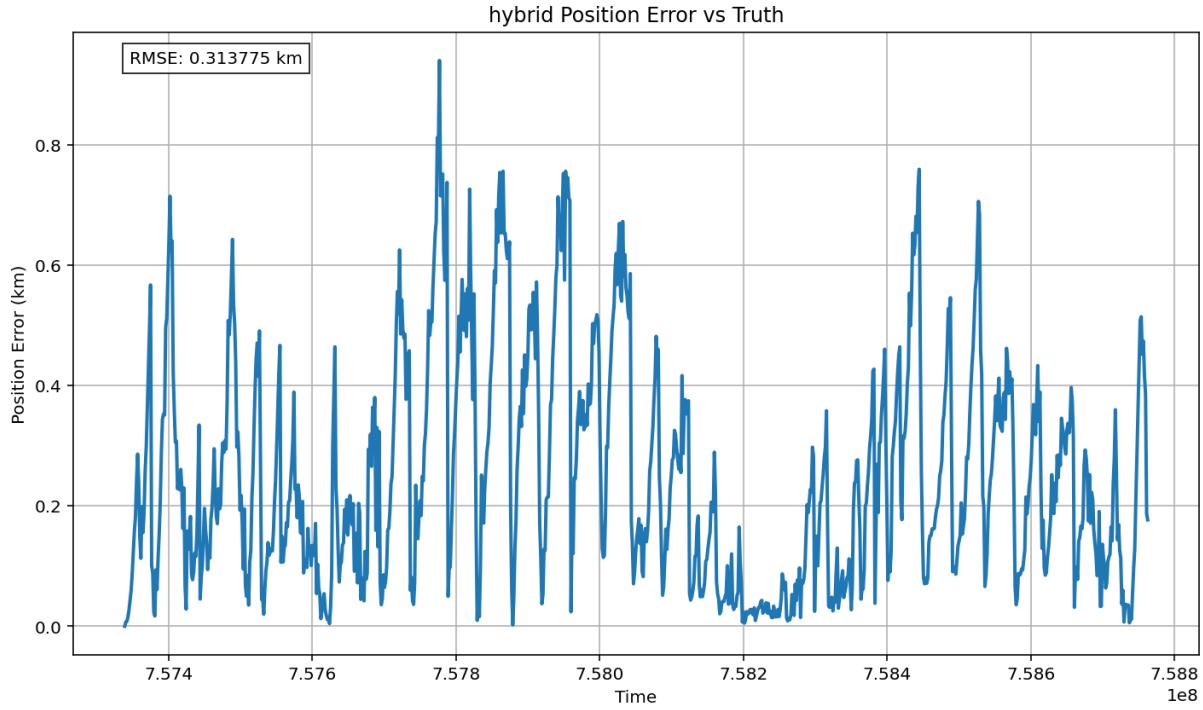


Figure 48. EKF-UKF (entropy-based) 1,500-Second Time Step Error Plots, Position

3,000-Second Plots

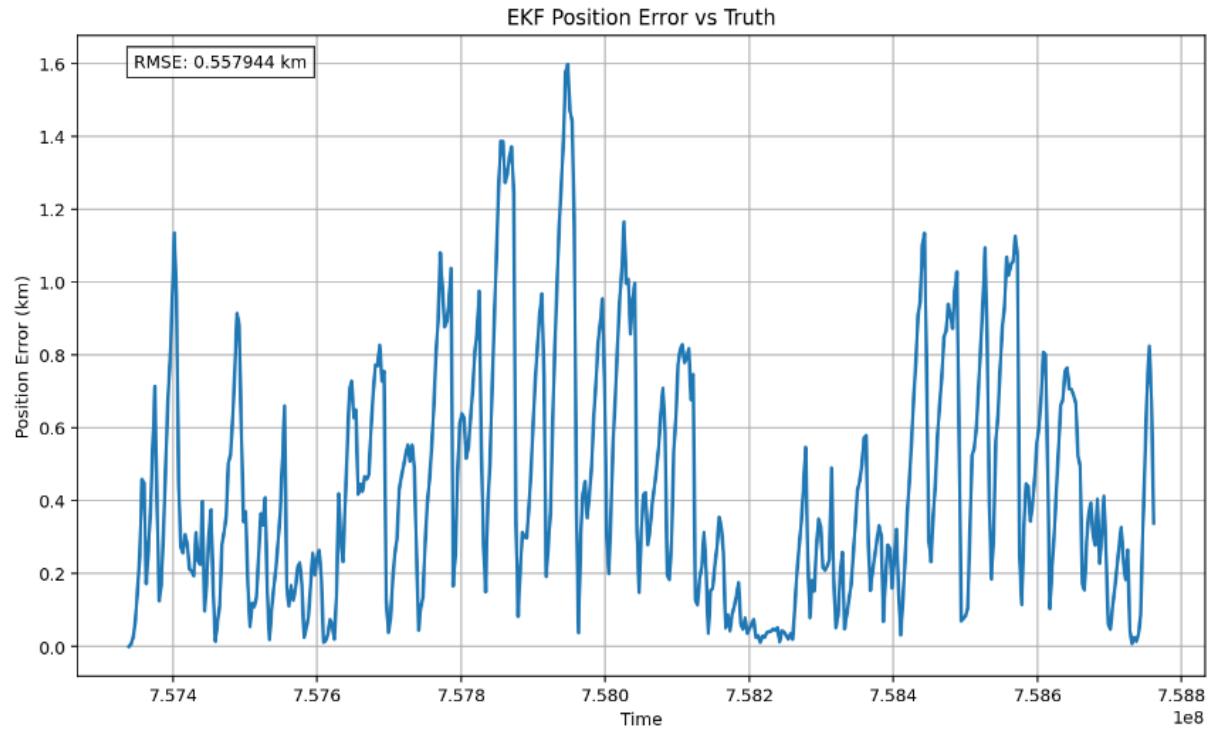


Figure 49. EKF 3,000-Second Time Step Error Plots, Position

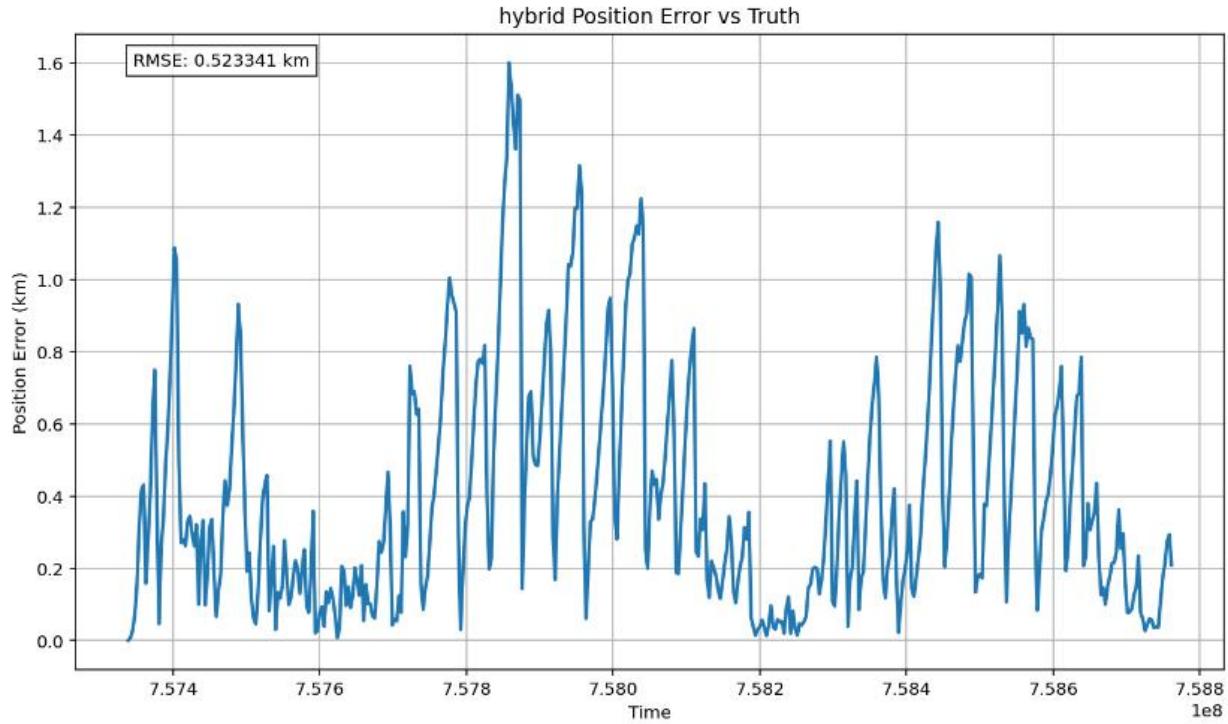


Figure 50. CKF 3,000-Second Time Step Error Plots, Position

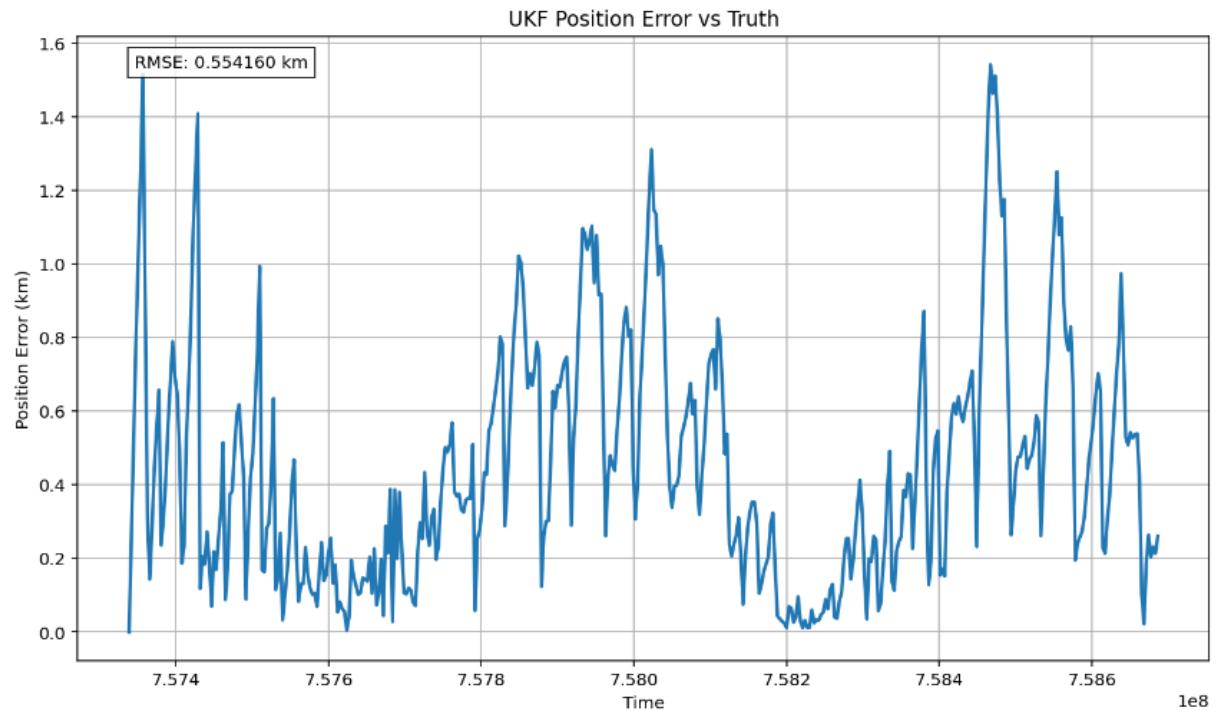


Figure 51. UKF 3,000-Second Time Step Error Plots, Position

Figure 52. EKF-CKF (time-based) 3,000-Second Time Step Error Plots, Position

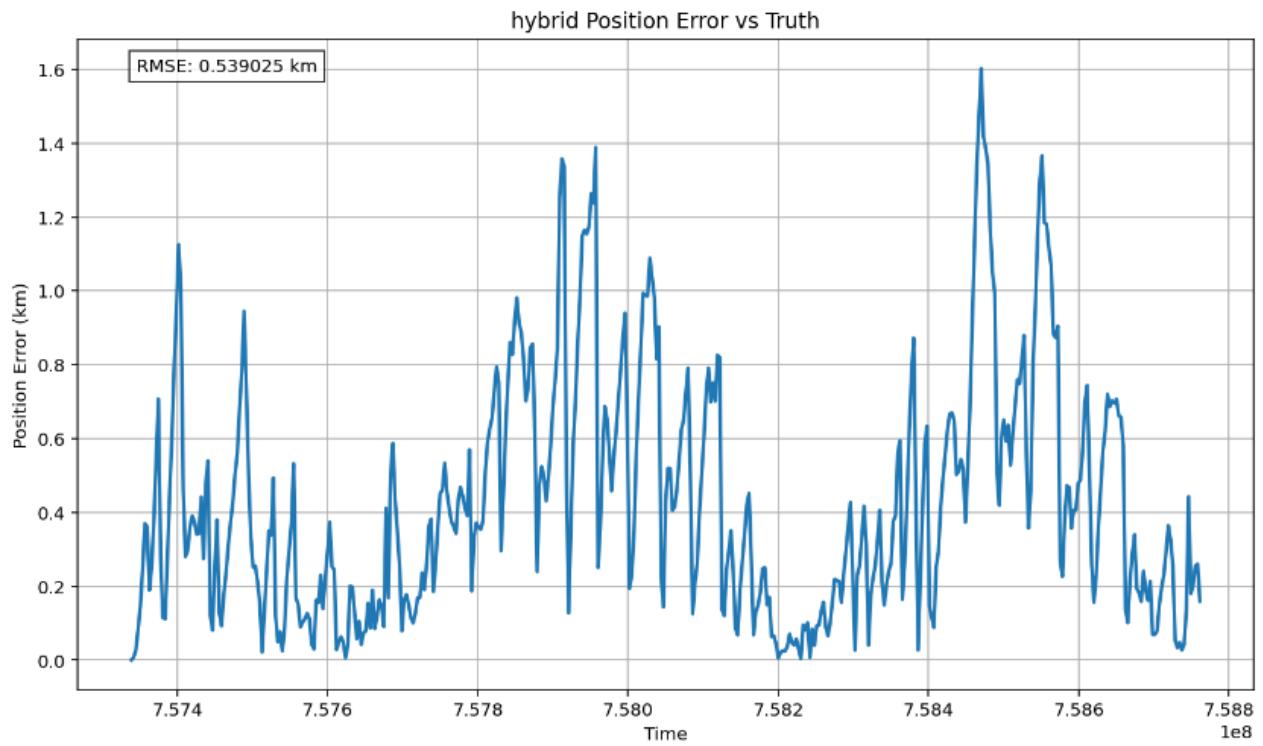


Figure 53. EKF-CKF (entropy-based) 3,000-Second Time Step Error Plots, Position

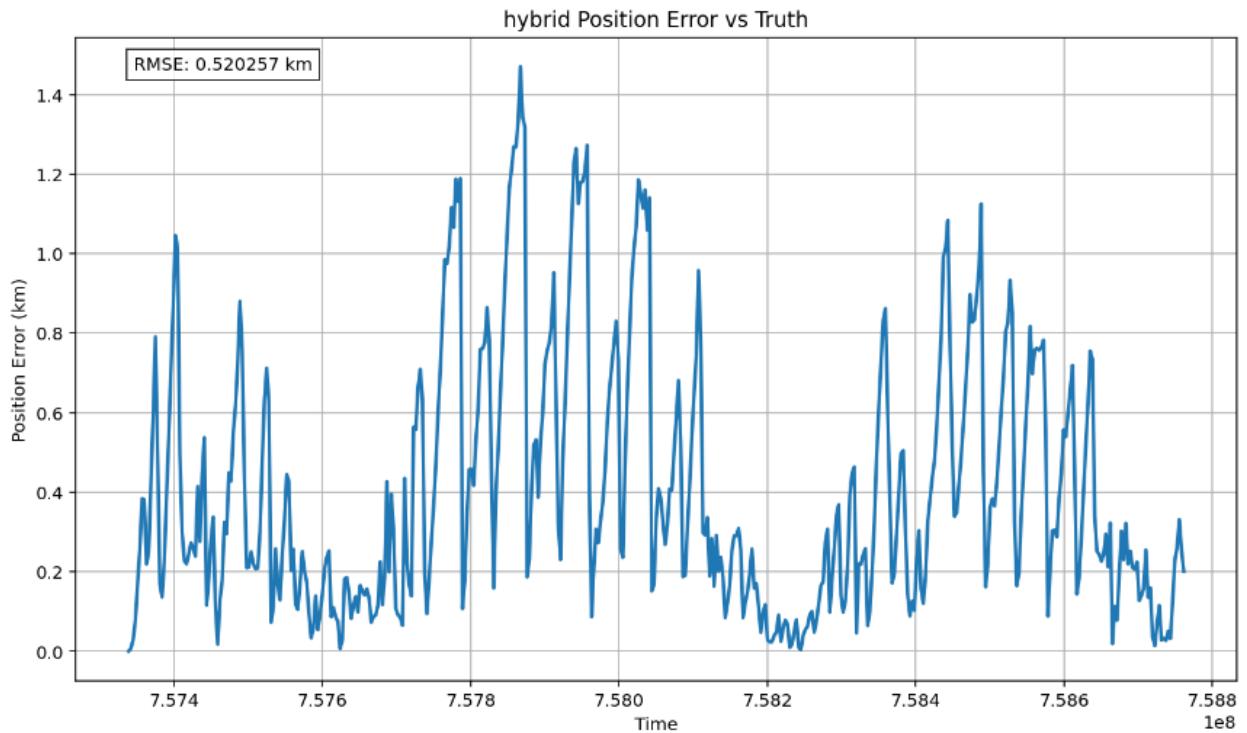


Figure 54. EKF-UKF (time-based) 3,000-Second Time Step Error Plots, Position

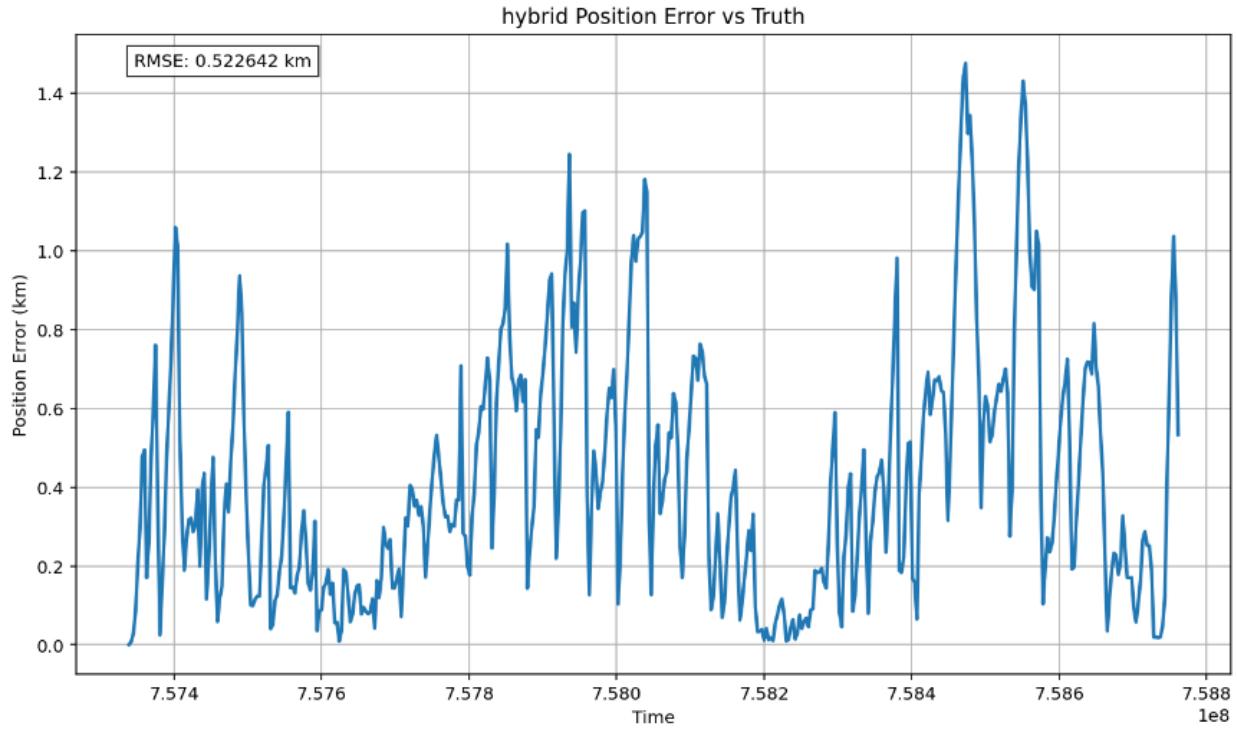


Figure 55. EKF-UKF (entropy-based) 3,000-Second Time Step Error Plots, Position