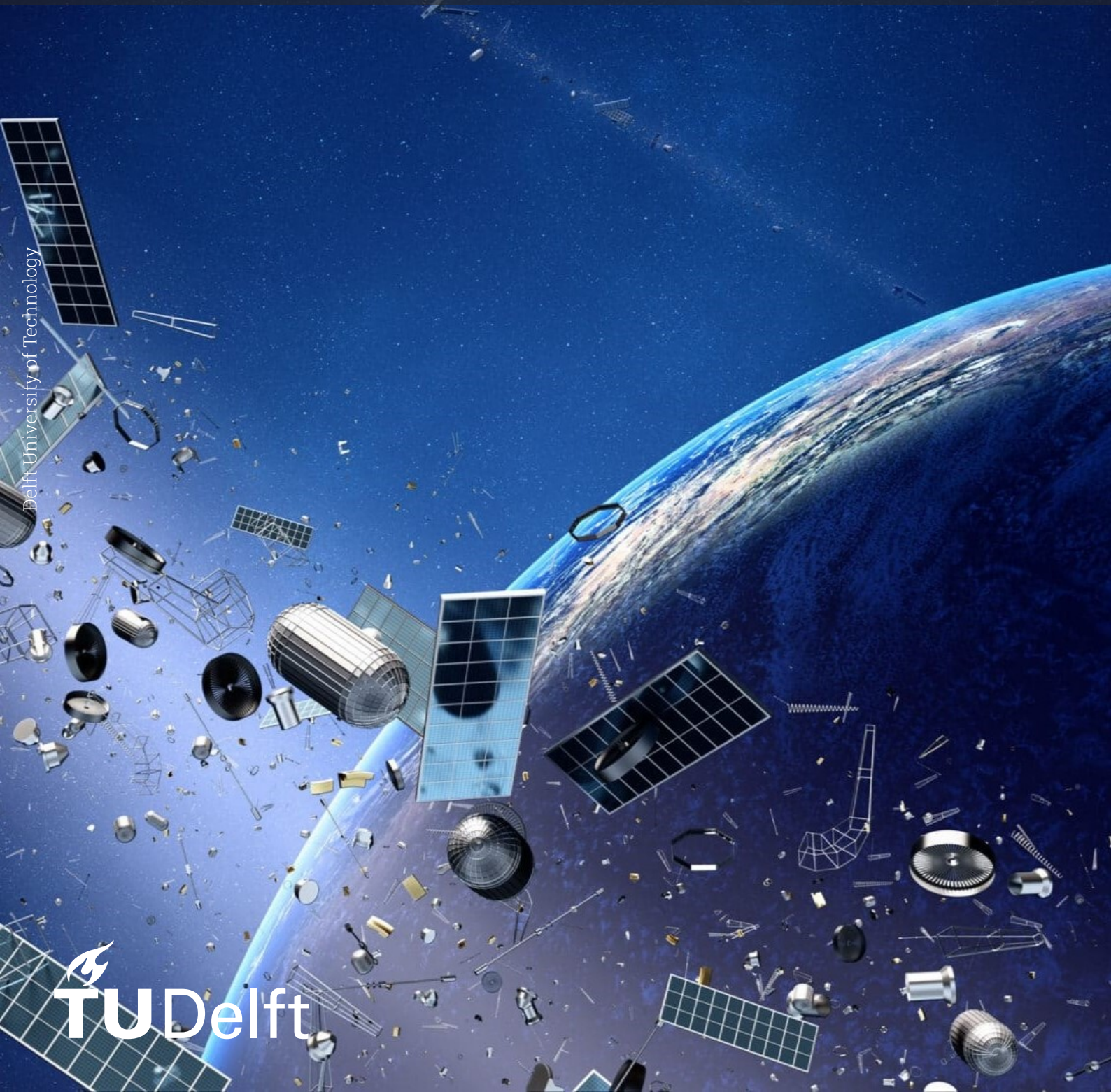
MSc. Thesis

Space Debris Object Characterisation through Light Curve Inversion

Wieger Verbeek



Delft University of Technology

TU Delft

MSc. Thesis

Space Debris Object Characterisation through Light Curve Inversion

by

Wieger Verbeek

Student number:	4675150
Date:	Friday 23rd May, 2025
Thesis supervisors:	Dr. S. Gehly
	Dr. M. Langbroek
Exam committee:	Dr. Ir. M.J. Heiligers (chair)
	Dr. A. Cervone (external)
Code repository:	`https://github.com/WiegerVerbeek/Thesis`

TU Delft

Preface

Dear reader,

What lies before you is the culmination of nine months of hard work, setbacks and lucky breaks. It also marks the final step in my journey to become an aerospace engineer, which started a long time ago at the same place it finishes, at the Aerospace Engineering faculty in Delft. With this thesis I hope to have demonstrated all the skills and knowledge required to carry this title, and I hope to put these skills to good use after graduating.

Before you read on, I would like to take a moment to thank a few people. First and foremost my supervisors for this thesis, Steve Gehly and Marco Langbroek, for helping me throughout the entire process, by listening to my sometimes incoherent thoughts and giving helpful suggestions to put me on the right track. I would also like to thank my examination committee, Jeannette Heiligers and Angelo Cervone, for making time free in their busy schedules to read my thesis and being available for the graduation.

Without my friends and fellow thesis students in 6.19 this thesis would also not have been finished on time, so thank you for all the coffee breaks, post-lunch walks and listening to my rants on algorithms you did not understand. Lastly, I would of course like to thank my family for supporting me throughout my studies, especially since I took my time to enjoy life in Delft before finally graduating.

Before you continue reading the rest of the thesis full of technical details and plots, I would like to share something I stumbled upon towards the end of my thesis period, namely a poem about space debris. We often consider science and art to be complete opposites of each other, but with this poem I would like to remind you that there is art to science, and science to art. Enjoy!

"The Universe is infinite
But space has its limits
Rockets a launching
Sat'lites are orbiting
Explosions in Space
Oh what a waste
Fragments go flying
And we go crying "Space junk we've got" Man-made or not
Then comes Kessler Who knows the better
When things collide
Their debris do multiply
Thanks to partnering
And NASA's gathering
We look for ways
To manage the spray"

– S. Thuy Nguyen-Onstott

Abstract

Space debris is an ever-increasing problem in all space applications. As the number of satellites launched every year grows, so does the number of debris objects orbiting the Earth. Without mitigation this problem increases exponentially, leading to more collisions and subsequently even more debris objects. Mitigation techniques exist, and can be divided into passive and active techniques. Active techniques actively remove debris from orbit, like on-orbit capture and de-orbit or laser debris removal. Passive techniques include collision avoidance of active satellites. All mitigation techniques require knowledge about the states and shapes of debris objects, in order to make future predictions of the object states. Since debris objects do not transmit information about their states, and objects can change shape over many orbits, acquiring this information becomes a difficult task.

One technique to obtain this information is called light curve inversion. Apparent brightness of objects over time is measured from the ground. From the variations in brightness it is possible to perform combined shape characterisation and attitude estimation. In this thesis an established method for this light curve inversion, called Multiple-Model Adaptive Estimation (MMAE), is implemented and tested on simulated measurements of existing LEO objects. Extensions to this method that using a variable model bank, called Variable-Structure Multiple Model (VSMM) estimation methods, are also implemented.

Performance of the methods on LEO test objects was tested for three spin cases with different rotational states, called Nadir pointing, Single-axis spin and Tumbling, to investigate what improvements could be made on the results of the existing MMAE method. An adjusted version of the MMAE method proved to be very successful in identifying the shapes and estimating the attitude of the test objects for all test cases. Sensitivity analyses showed robustness of results to changing simulation parameters differ per spin case examined. Finally, an attempt was made to validate the improved MMAE method with real brightness measurements of two of the LEO test objects. This validation was unsuccessful in both shape characterisation and state estimation, highlighting important differences between simulations and real-life applicability.

Contents

Nomenclature

Abbreviations

3DOF	3 degrees-of-freedom
ADR	Active Debris Removal
BRDF	Bidirectional Reflectance Distribution Function
ECEF	Earth-Centred Earth-Fixed
EGI	Extended Gaussian Image
FSMM	Fixed-Structure Multiple-Model
GRP	Generalised Rodrigues Parameter
LEO	Low Earth Orbit
MGen	Model Generation
MGS	Model-Group Switching
ML	Machine Learning
MMAE	Multiple-Model Adaptive Estimation
NPT	Nadir Pointing
PDF	Probability Density Function
PMG	Point Mass Gravity
PSO	Particle Swarm Optimisation
RMSE	Root Mean Square Error
RQ	Research question
SAS	Single Axis Spin
SHG	Spherical Harmonic Gravity
SRP	Solar Radiation Pressure
SSN	Space Surveillance Network
TBL	Tumbling
UKF	Unscented Kalman Filter
VSMM	Variable-Structure Multiple-Model

1 Introduction

The past decade has seen an ever-increasing number of new satellites being launched into space, as illustrated by Figure 1.1. This increase is caused by two main factors. The first is that satellite technology has advanced significantly, and small satellites like cubesats can be purchased off-the-shelf for a relatively cheap price, making satellites much more commercially available. The second factor is that is has become much cheaper to launch a satellite. The average launch cost per kg for commercial satellites has decreased by a factor of 16.5 between 2000 and 2020 [1]. A main contributor to this is SpaceX, which have drastically reduced the cost of launching their Falcon 9 by reusing the lower stage of the rocket. Additionally, the company has made it possible to launch a large number of small satellites in a single launch through ridesharing.

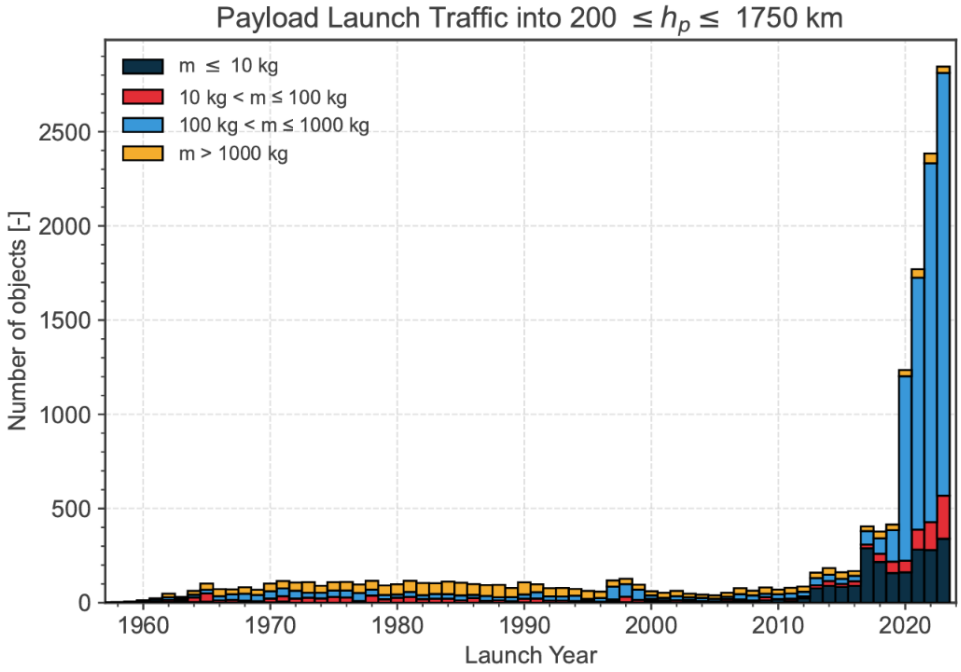


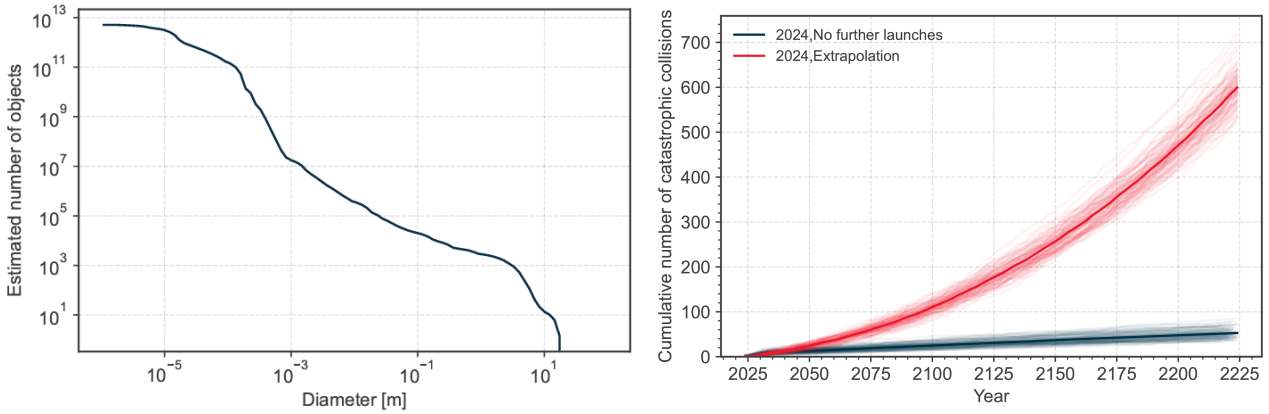
Figure 1.1: Number of objects launched per year into Low Earth Orbit since 1958 [2].

These factors combined have lead to a large amount of small satellites being launched in record time. Companies like Airbus and Starlink are launching large constellations of small satellites for telecommunication purposes. This trend of launching a large number of small satellites is reflected in Figure 1.1, with the majority of objects being launched having a mass between 100 and 1000 kg. The growth in number of launches is matched by a growth in economic worth of the space industry. The global space economy was worth 570 billion USD in 2023[1], which is nearly double what it was a decade ago. This number is expected to rise even further to 1.8 trillion USD by 2035[2].

However, this growth does not only have upsides. In the same time period the amount of space debris objects orbiting the Earth has also drastically increased. It is estimated that there are now over 30.000 debris objects larger than 10 cm, 900.000 objects with size between 1 and 10 cm, and around 128 million objects between 1 cm and 1 mm in geocentric orbit, as per the 2024 edition of the Annual Space Environment Report by ESA [2]. Figure 1.2a shows the distribution of the amount of objects in orbit per size. All of these objects pose a threat to existing and future space missions. Even small objects could inflict catastrophic damage on a satellite, leaving the satellite unable to function.

[1] `https://www.spacefoundation.org/2024/07/18/the-space-report-2024-q2/`. Accessed on 30-12-2024.
[2] https://www.mckinsey.com/industries/aerospace-and-defense/our-insights/space-the-1-point-8-trillion-dollar-opportunity-for-global-economic-growth. Accessed on 30-12-2024.

The amount of debris is only expected to rise, as debris objects stay in orbit for a long time before naturally decaying and re-entering the atmosphere, and with more satellites orbiting the same regions as the debris collisions become increasingly more likely. If nothing is done to mitigate the amount of debris, while satellites continue to be launched at the current rate, the number of catastrophic collisions will steadily increase, as shown in Figure 1.2b. Even if no new satellites would be launched the number of catastrophic collisions would slowly increase, because any catastrophic collision creates more debris objects, which in turn could potentially create more catastrophic collisions. It is clear mitigation is required to ensure a viable space environment and continuous access to space.



(a) Distribution of amount of objects in Earth orbit per size.

(b) Expected number of catastrophic collisions with and without new launches.

Figure 1.2: Figures highlighting the magnitude of the space debris problem [2].

There are two ways of dealing with space debris in order to reduce the risk of collision with an active satellite: collision avoidance and active debris removal (ADR). Collision avoidance is what most active missions employ. For a certain time in the future the likelihoods of a collision with a number of known objects is estimated and if a certain collision risk level is reached the satellite performs an avoidance maneuver. With ADR the debris problem is dealt with directly, as objects are actively removed from orbit. For large debris like inactive satellites or large debris fragments this can be done by launching a new satellite to capture it and collectively de-orbit the pair. For smaller debris a promising method is hitting debris objects with high energy laser burst so that objects are slowed down, such that they enter a lower orbit and decay into the atmosphere faster [3]. The ADR field is in early stages of development, and as of writing no space debris has been removed through ADR.

What these methods have in common is that they require knowledge about the debris objects, either to avoid it or to capture it. For this objects are modelled and simulated to make predictions of their states in the future. However, in order to make accurate predictions information of the objects is necessary. This information can be obtained through estimation of the states and characterisation of the shape of the debris objects. One method that is used in object characterisation is light curve inversion. This method uses the sunlight reflected by debris objects, measured from the ground, in order to obtain information about the states and shape of an object. In this MSc. thesis research this method is investigated further.

The thesis is structured as follows. First, in Chapter 2 the research that is performed in the thesis is described in more detail, providing more background information, highlighting the knowledge gap and presenting the research questions. In Chapter 3 the simulation environment in which the thesis research is performed is established. Then in Chapter 4 the methodology behind the thesis research is described in detail. Chapter 5 defines the test cases and testing strategy with which the implemented methods will be analysed and evaluated. In Chapter 6 the results of these analyses and evaluations are presented. Finally, in Chapter 7 the conclusions for the thesis research are drafted, with an answer to the research questions established in the next chapter. Additionally, a discussion of the results is presented, and recommendations for future work are made.

2 Research description

This chapter gives an overview of the aspects the research performed in this MSc. thesis is based on. In Section 2.1 first the relevant background information is highlighted. Then in Section 2.2 the knowledge gap that was identified in the literature review in the space debris object characterisation through light curve inversion field is presented. Finally, Section 2.3 presents the research questions that drive the thesis research.

2.1. Background

This section gives a summary of background information on which the research questions are based. More details on all the topics covered can be found in the literature review that was performed as part of the MSc. thesis process, which is included in this report as Appendix C. In this section it is first explained what light curves are, and how they can be used for object characterisation. Then a few different methods that have been developed for object characterisation of space objects are discussed.

Light curves

A light curve is the time history of the apparent brightness of an object measured by an observer [4], for example an optical telescope on the ground. The act of extracting information from such a light curve is called light curve inversion. Light curves were already used in 1958 to estimate the rotation period of asteroids [5], and were later also used to estimate asteroid shapes [6]. At the start of the 21st century interest began to gather around using light curve inversion on man-made or artificial objects. Figure 2.1.1 shows the light curve of such an object, a rocket body that was left in orbit after payload separation. A periodic variation in apparent brightness is clearly visible in the graph, indicating that the object is spinning.

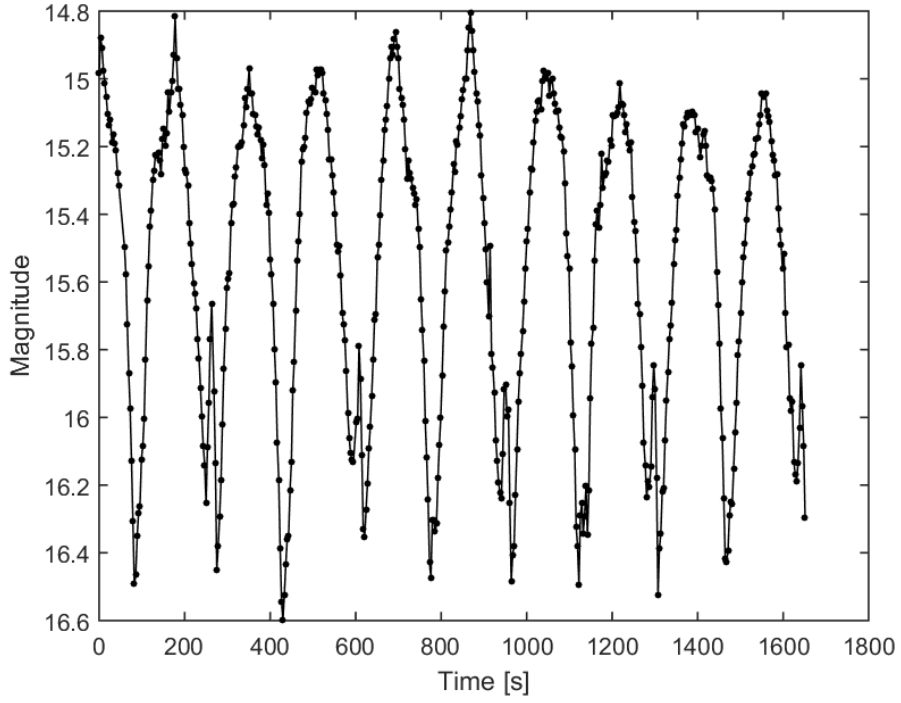


Figure 2.1.1: Example of a light curve from a rocket body left in orbit [7].

It is however not straightforward to use the same characterisation techniques on artificial objects as developed for natural objects like asteroids. Asteroids generally have a relatively smooth surface, and have mostly uniform reflection properties across their surface [8]. In contrast, artificial objects like satellites are made from very different materials, reflecting light with varying intensity. Satellite surfaces are generally also much less smooth, with many protruding components like solar panels, antennae and other instruments. Lastly, attitude states of satellites are usually more complex than those of asteroids, where the latter can be assumed to be spinning stably.

Object characterisation

In the literature review it was found that many different methods have been developed to characterise different parameters of artificial space objects. It was determined that the parameter most of interest to characterise is object shape, combined with estimation of orbit and attitude states. This is because these aspects have the largest influence on the magnitude of non-conservative forces like aerodynamic drag and solar radiation pressure, which over time have a large impact on the orbits of the objects. The main difficulty with extracting these parameters from a light curve is that there are a lot of different combinations of shapes and attitudes that can produce similar light curves [9]. To produce good estimates additional information is often required beforehand or *a-priori* [10].

Most often some sort of filtering is included in the characterisation method. The Unscented Kalman Filter (UKF) is commonly employed to estimate orbit and attitude states, possibly with additional parameters, based on astrometric and photometric measurements, where the first are observations of the orbital positions and velocities through range and angles measurements, and the latter are the light curves. Unfortunately these filter methods are very sensitive to the initial guess, especially for the attitude states, and if the initial guess is too far from the truth the filters do not converge [11].

Another difficulty with using filter methods is that they rely on a constant physical model. This makes it difficult to estimate object shapes, as the shape is part of the physical model included in the UKF. One method that attempts to overcome this downside is called Multiple-Model Adaptive Estimation (MMAE). As the name implies, this method uses multiple shape models in a model bank. Each model is input into a separate UKF. In the MMAE algorithm these different UKFs are run in parallel, and for each measurement it is compared which model fits the observations the best. Based on how well each shape model fits the measurements, each shape is assigned a weight between 0 and 1, with high weights assigned to shapes that fit the measurements better. These weights are also used to output a fused state estimation, which are weighted averages of the state outputs of each individual UKF. The goal is to find the shape model or models that best fit the measurements, and provide a fused state estimation [12]. A flowchart of this method is given in Figure 2.1.2

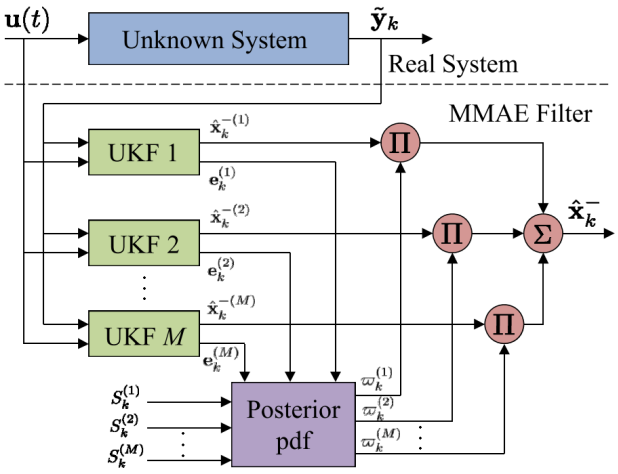


Figure 2.1.2: Flowchart of MMAE algorithm [12].

This method is further explained in more detail in Section 4.3. A downside to this method is that it is computationally expensive, with multiple UKFs being run at the same time. There is also a delicate balance in the number of models included in the model bank, as too few models might lead to poor convergence, but too many can also degrade performance [13].

As is the case for many fields in the present-day, Machine Learning (ML) has been applied to the object characterisation problem [14]. The problem with ML is that it requires a lot of data to properly train a ML algorithm, which in this case means light curves and the true states and shapes that created each curve. There is not enough real data available to do this, so ML training relies mainly on simulated data. The performance of these algorithms on real data is very poor, as real data is subject to noise, gaps in data and more parameters that cannot all be simulated, meaning that these algorithms are not trained on representative data.

A different approach that was developed to estimate object shapes uses the so-called the Extended Gaussian Image (EGI) of objects [15]. An EGI is a representation of a 3D object on a unit sphere based on surface normal vectors and surface areas. The method works well for convex objects, but returns much more ambiguous results for non-convex objects. This makes the method not very suitable for space object shape characterisation, as satellites are very often non-convexly shaped due to extended solar panels and instruments.

The last method discussed in the literature review is the Particle Swarm Optimisation (PSO) employed by Burton [16]. This method addresses the reliance of filtering methods on a-priori information and close initial guess to achieve convergence. The PSO method tries to find, for a given shape, all the possible attitude states that correspond to a certain light curve measurement, usually the first measurement. These initial attitude states are then paired with possible rotational velocity states, and the optimiser attempts to find the initial attitude and rotational velocity pair so the light curve produced by these states best fits the observations. While the method produces very good results, it requires the shape of the object for which the states are estimated to be known beforehand. The method is thus not directly applicable for simultaneous attitude and shape estimation.

2.2. Knowledge gap

Based on the background information and literature review a knowledge gap was identified on three different areas of the light curve inversion problem. These areas are: a-priori information, measurement data and fixed model bank.

A-priori information

The first area where a knowledge gap is present is the fact that in most cases a-priori information is required to generate good estimations. This could mean that a filtering method requires an initial guess that is close enough to the truth, or that the true shape of an object is known beforehand. Ideally it would be possible to estimate both the attitude states and object shape from observations without any prior knowledge. While this might be unfeasible at the moment, there is a lot of potential to improve on this aspect of light curve inversion.

Measurement data

There are two aspects on which a knowledge gap was identified related to measurement data. The first is that in a lot of papers the data that was simulated in order to develop and test the method was made by modelling satellites in near-GEO orbits. This ensured that there were continuous measurements available to allow good convergence of estimations. However, the majority of space debris and satellites are present in LEO. This means that measurements cannot be made continuously, only for short observational arcs lasting several minutes. Often sun-illuminated passes of LEO satellites last only between 1 to 2 minutes, and sometimes even shorter. This is a big limitation that needs to be addressed in order to make the methods relevant for real life applications.

The second aspect is the fact that most methods are only developed and tested using simulated data. There is a significant lack of validation of these methods using real observation data. This is to be expected, as for proper validation the true attitude states of objects needs to be known, which for most objects are not readily available. It is however important to test these methods on real data, as this is in the end what these methods are developed for, and without this validation it is unknown what limitations need to be overcome in order to deploy the methods for real object characterisation.

Fixed model bank

Finally, a big limitation encountered in several methods, especially MMAE, is that the algorithms use a fixed model bank. This requires all the possible models that are used in the estimations to be determined beforehand. However, as mentioned previously, using more models does not necessarily lead to better performance. Another issue with having a fixed model bank is that it is not guaranteed that the true model is included. This limitation could lead to estimations not converging, or converging on an incorrect model.

2.3. Research questions

Based on the knowledge gap identified in the literature review and discussed in the previous section the following main research question was drafted for the MSc. thesis:

What improvements can be made on existing techniques for space debris object characterisation of LEO objects through light curve inversion to estimate attitudes and characterise shapes?

As this question is quite broad, it is broken down into three separate research questions, one for each area of the knowledge gap as discussed in the previous section. The research questions, abbreviated by RQ, are the following:

RQ-1: How can the limitations of using a fixed model bank in attitude and shape estimation be overcome?

RQ-2: What is required of the initial guess to ensure good convergence of estimation?

RQ-3: What are requirements on measurement data (e.g. noise, rate, variability) to ensure accurate estimates?

All activities performed over the duration of the MSc. thesis period were done in order to answer these three questions, with the end goal to give one or multiple answers to the main research question. This work is presented in the next chapters of this report.

3 Simulation environment

In this thesis research many simulations will be run that model the dynamics of different satellites. These simulations need to be performed in a simulation environment that includes certain forces and accelerations. In order to ensure the relevance of the research it is desired that this simulation environment models the real environment to a certain degree of accuracy. Accurate simulation models are however more complex, and come with a higher computational cost. Since the time in which the research needs to be completed is limited, there is a limit on how long a single simulation can run. The accuracy and run time are thus conflicting requirements. In order to make a trade-off between the two, and define the settings of the simulation environment, some analysis needs to be performed. This will be discussed in this chapter.

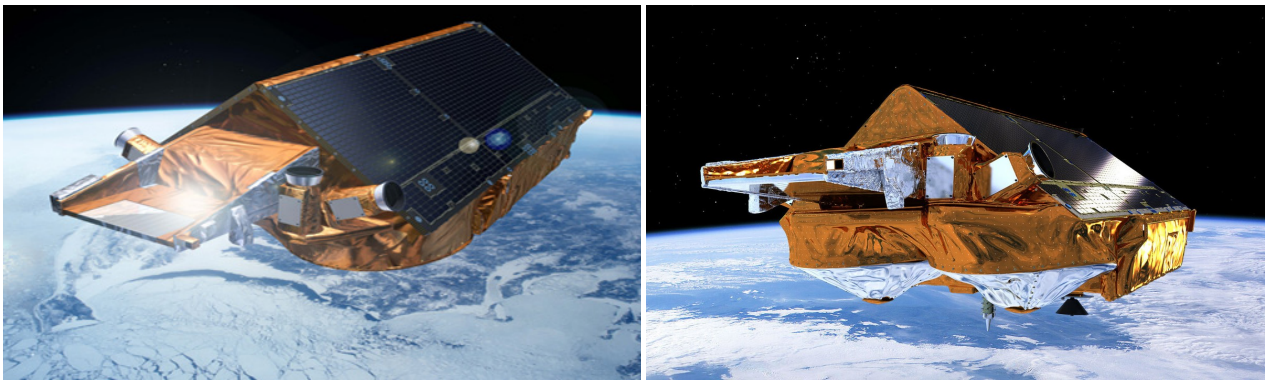
First, in Section 3.1 the objects that were selected to be used in the simulations are highlighted, as well as their orbits. In Section 3.2 the dynamics that are included in the simulation environment analysis are discussed. Then in Section 3.3 the analysis that was performed to compare the accuracy and run times of different simulation environment settings is shown, concluding with the final environment settings that will be used in the simulations throughout the rest of the thesis.

3.1. Test objects

The first step is to define the objects and orbits for which estimations will be performed. It is of course possible to define an arbitrarily shaped satellite in any desired orbit and simulate these. However, one of the research objectives is to validate the results from the estimations with real data. Therefore it is more convenient to select satellites for which real data is available, as this makes validation possible. The satellites discussed in this section all have a known shape, so for all satellites the shape estimations can be validated. For two satellites the attitude data is also available online, which makes it possible to validate both the attitude and shape estimation. The satellites that were selected are CryoSat 2, the Swarm constellation, BlueWalker 3 and ACS3. Their details are highlighted next.

3.1.1. CryoSat 2

CryoSat 2 is a LEO satellite from ESA, launched in 2010. It is not shaped like a common box or box wing satellite. Instead, it has a rectangular underside, and a triangular upper section like a roof, on top of which lie its solar panels. The satellite is shown in Figure 3.1.1. On the underside there are also two circular dishes placed side by side, as can be seen in Figure 3.1.1b.



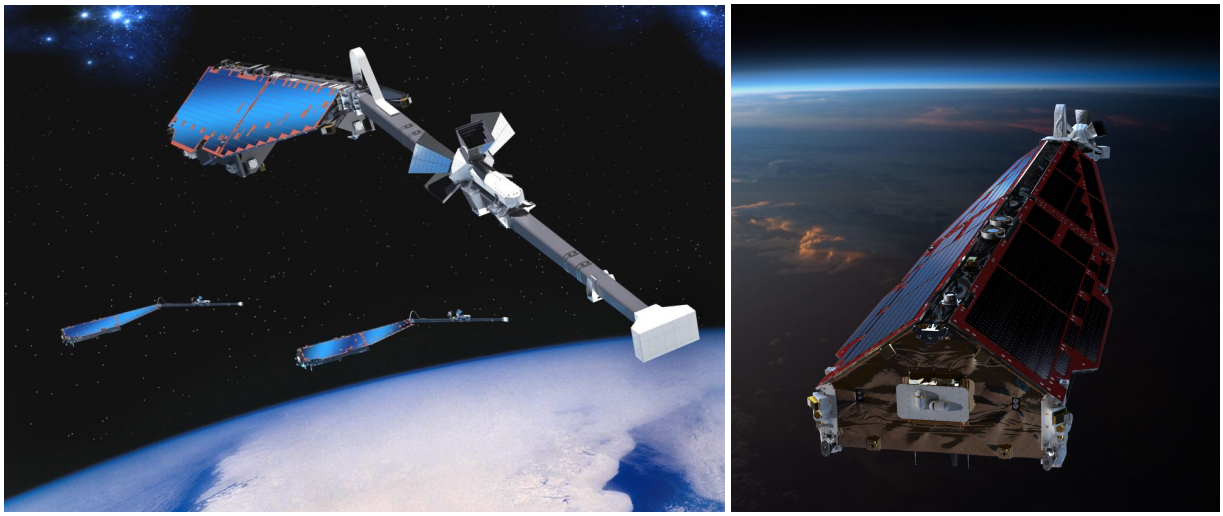
(a) (b)

Figure 3.1.1: Two views of CryoSat 2.

The reason this satellite was selected to be observed is that ESA has made the orbit and attitude data publicly available. This means that the attitude estimation through light curves can be validated with real light curve measurements. It also has a shape that is different from the more common box or box-wing satellite design, which makes it an interesting test case for shape characterisation.

3.1.2. Swarm A/B/C

The Swarm satellites are an ESA LEO constellation consisting of three satellites, Swarm Alpha, Bravo and Charlie or A, B and C. The goal of the mission was to create a highly detailed survey of the Earth's geomagnetic field.[1] Swarm A and C were placed in an almost identical orbit so they would orbit side-by-side, while Swarm B was placed in a higher orbit. A view of the three satellites in their respective orbits is shown in Figure 3.1.2a. It can be seen that these satellites also have a shape different from the common satellite shape. A front view of one of the satellites is shown in Figure 3.1.2b. The reason these satellites have been selected as test objects is that light curve observations for these objects have been made in the past by one of the thesis supervisors, for multiple different dates. Since their orbit and attitude data is publicly available through ESA, these objects serve as the ideal cases to perform validations of the algorithms implemented for this thesis with the real measurements.



(a) Swarm A, B and C visualised in their respective orbits.

(b) Front view of a Swarm satellite.

Figure 3.1.2: Swarm A, B and C.

3.1.3. BlueWalker 3

The next object is BlueWalker 3. This satellite has the largest commercial communication array ever deployed, namely 64 m^2. The satellite is shown in Figure 3.1.3. The deployment of this satellite has led to dissatisfaction amongst astronomers, as the large array reflects a lot of sunlight, which means the satellite is one of the brightest objects visible in the night sky, which interferes with astronomical observations. There is a lot of observational data from this satellite because astronomers wanted to voice their concerns about the threat these types of satellites pose to astronomy [17].[2] The attitude data for this satellite is not available, but the object is included because it will be interesting to see if it is possible to perform shape characterisation for such a bright object.

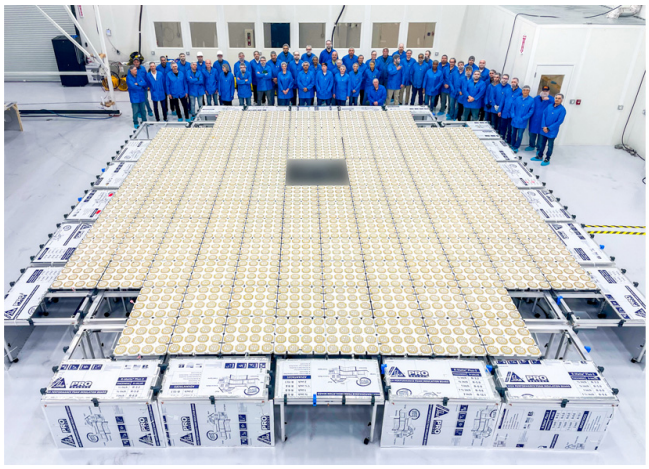


Figure 3.1.3: BlueWalker 3 satellite in a clean room.

[1] `https://earth.esa.int/eogateway/missions/swarm`. Accessed on 14-11-2024.

[2] `https://skyandtelescope.org/astronomy-news/the-bright-bluewalker-3-satellite-threatens-astronomy/`. Accessed on 14-11-2024.

3.1.4. ACS3

The last satellite that will be used as a test object is the Advanced Composite Solar Sail System or ACS3 satellite. This satellite was launched in April 2024 by NASA and is a technology demonstrator for solar sailing satellites. The deployed composite sail has a size of 9 by 9 meters, and has an area of 80 m^2. This size and the fact that the sail is made from highly reflective material means that this satellite is also bright and clearly visible in the sky. It has been observed since before and after the deployment of the sail. It is included as a test object because of similar reasons that BlueWalker 3 is included. Additionally, ACS3 has a very low weight of only 15 kg in order to be able to use solar sailing as a form of propulsion, which also means it has a low mass moment of inertia. Because of this, combined with its large surface area, it is expected that its attitude has a higher variability, as torques caused by low magnitude forces like Solar Radiation Pressure have more effect than on a heavier satellite with more inertia. This makes it a good case to investigate for the combined attitude estimation and shape characterisation problem.

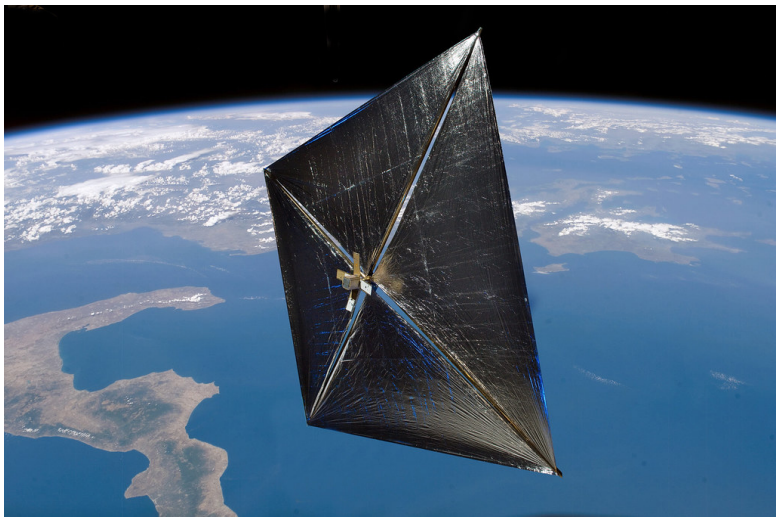


Figure 3.1.4: ACS3 satellite.

3.1.5. Orbit data

The orbital parameter data for the different satellites were retrieved in Two-Line Elements (TLEs) form from the *space-track.org* website, for the date 14-11-2024. They were then converted to Keplerian parameters using the conversion functionality of TudatPy, the astrodynamics software used for this thesis, which will be discussed in more detail in next section. The raw TLE data for each satellite is given in Appendix A. The Keplerian orbital parameters of the test objects are given in Table 3.1.1.

Table 3.1.1: Orbit parameters for objects to be simulated.

Satellite	Norad ID	e	i	h_p	h_a	Ω	ω
CryoSat 2	36508	0.00031934	92.0163°	704.97 km	709.49 km	137.2817°	289.0605°
Swarm A	39452	0.00175787	87.3370°	441.14 km	465.15 km	268.9912°	63.1381°
Swarm B	39451	0.00182824	87.7499°	481.55 km	506.68 km	157.7462°	117.8716°
Swarm C	39453	0.00178117	87.3370°	440.82 km	465.15 km	267.5801°	63.0753°
Bluewalker 3	53807	0.00144233	53.2482°	469.87 km	489.65 km	221.4666°	104.4297°
ACS3	59588	0.00332531	97.5035°	969.01 km	1018.04 km	352.9487°	178.0922°

3.2. Dynamics

The propagations of the orbit and attitude states will be done using TudatPy[3], the astrodynamics software developed by the Faculty of Aerospace Engineering, which is readily available to use in Python. The main reason this software is used instead of writing a new propagation module is that parameters for celestial bodies like ephemerides and properties like mass can easily be loaded in. This greatly reduces the effort required to set up simulations. The software has a high flexibility in which bodies and forces can be added to a simulation, with different levels of complexity and accuracy, allowing simulations with different levels of fidelity to be made relatively easily.

In order to set up a simulation in Tudat, a selection of bodies and forces to be included needs to be made. For this an analysis will be performed in the next section. Based on this analysis it will be decided which bodies and dynamics are included in the final simulation environment used in the rest of the thesis. In this section it will be highlighted which bodies and dynamics are included in the analysis.

3.2.1. Forces

Earth gravity

The main force acting on the satellite is the gravity exerted by the Earth. However, the Earth is not a perfect sphere, and its mass is distributed unevenly over the planet. At some points on the planet more mass is concentrated, so a satellite flying over these parts experiences slightly more gravitational pull, while at other points less mass is concentrated, so less pull is exerted on the satellite. The distribution of the mass can be modelled with spherical harmonics, and the effect on gravity is modelled using a spherical harmonic representation of the gravity field potential. The general equation for the spherical harmonic gravity potential is given in Equation 3.2.1.

$$U = \frac{\mu}{r} \sum_{\ell=0}^{\infty} \left(\frac{a_e}{r}\right)^{\ell} \sum_{m=0}^{\ell} P_{\ell,m}(\sin\phi)\left[C_{\ell,m}\cos m\lambda + S_{\ell,m}\sin m\lambda\right] \tag{3.2.1}$$

Here μ is the standard gravitational parameter of the body in question; r, ϕ and λ are the distance, latitude and longitude of the satellite in the body-fixed coordinate frame, respectively; a_e is the semi-major axis of the reference ellipsoid of the body in question; $C_{\ell m}$ and $S_{\ell m}$ are the spherical harmonic coefficients of degree ℓ and order m; $P_{\ell m}$ are the associated Legendre functions of degree ℓ and order m.

For the Earth the coefficients for over degree and order 2000 have been determined. This level of fidelity is not required for most applications. In Tudat the coefficients are available up to degree and order 200, which is also too high for most applications. For the simulation setup a benchmark propagation will be created, which will act as the truth to which errors are compared. For this benchmark the degree and orders up to and including 100 will be used in the gravity model for which the gravitational acceleration of the Earth on the satellite will be calculated. For the test cases either degree and order 100, 10 or 2 will be used. The latter are commonly known to be the most important, as they model the flattening of the Earth, and have the largest effect on perturbing the orbit. In the test cases it will be investigated which degree and order are required in order to achieve certain requirements, which will be set in the next section.

Aerodynamic drag

As can be seen in Table 3.1.1 all satellites that are included as test objects are in LEO, with semi-major axis values ranging between 6826 and 7404 km. In this orbit regime there is still thin layer of atmosphere, which means objects in orbit experience aerodynamic drag to a certain extent. However, it can be seen in Figure 3.2.1 that the magnitude of this acceleration is between the order of 10^{-5} and 10^{-7} m/s^2 for the altitudes considered. This is much lower than the other accelerations acting on the objects. Over long periods of time, meaning multiple hundreds of orbits, this acceleration would have an effect on the orbit. For this thesis it is however not expected that such a large number of orbits will be continuously propagated.

In addition to this, the magnitude of the aerodynamic acceleration is dependent on aerodynamic coefficients, which are in turn dependent on the shape of the object experiencing drag. For this thesis many different objects with different shape will be considered. Spending a lot of time on determining the right coefficients would take away from the important research tasks that need to be performed, which would result in unsatisfactory results. These reasons combined have led to the decision to not include the aerodynamic drag in the forces considered for the simulations in this thesis.

[3] https://docs.tudat.space/

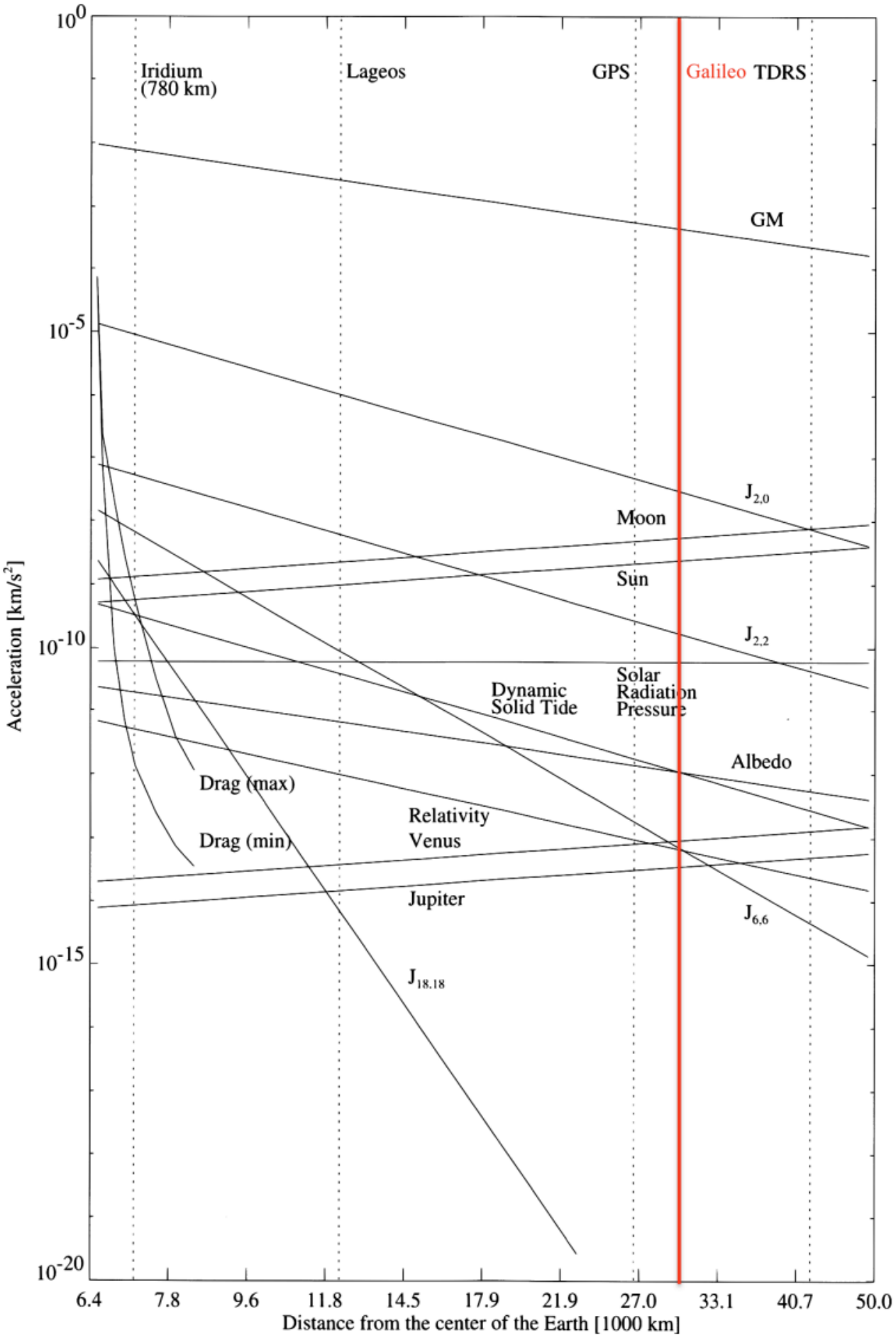


Figure 3.2.1: Magnitudes of perturbing accelerations at different Earth orbit altitudes [18].

Solar Radiation Pressure

Photons emitted from the Sun carry momentum. When these photons impact an object in orbit they are either reflected or absorbed. In both cases the photons exchange momentum with the object. The momentum imparted by a single photon is extremely small, but since there are many photons hitting a surface every second, the sum of the individual impacts has a cumulative effect. This effect is called Solar Radiation Pressure (SRP). Multiplying this pressure with the surface area impacted by the photons gives another perturbing force. It can be seen that the magnitude of the acceleration caused by this force is on the order of magnitude of 10^{-7} m/s^2. It is therefore again expected that the effect of this force on the small amount of orbits propagated is negligible. However, since a main part of this thesis research is the determination of attitude from light curves, which are made by modelling the reflection of light from the object surfaces, it was decided to still include SRP in the simulation model for the test cases, to investigate whether it needs to be included in the final environment.

Modelling the SRP accurately is a complex task however, since the surface areas impacted by photons change with changing attitude. With many different shapes considered this again becomes an arduous task to implement correctly. Instead the the SRP is modelled using the so-called cannonball model. This is a simplified way of modelling the SRP, where the spacecraft is considered to be a sphere. The acceleration is then calculated using the cross-sectional area of the sphere given by $A = \pi r^2$, where r will be taken as the largest of the actual satellite dimensions. This area will generally be larger than the actual areas of the spacecraft that the SRP acts on, so this model will overestimate the effect of SRP in the model. The main advantage of using the cannonball model is that it does not require the attitude of the satellite and its surfaces, reducing model complexity.

The acceleration due to SRP in the cannonball model is given by Equation 3.2.2.

$$\mathbf{a_{srp}} = -\frac{P}{4\pi c}\frac{C_R A}{m}\hat{\mathbf{r}}_{Sun} \tag{3.2.2}$$

Here P is the total power output of the Sun in W, c the speed of light, C_R the reflectivity coefficient, A the surface area in m^2, m the mass of the satellite in kg, and $\hat{\mathbf{r}}_{Sun}$ the Sun position vector measured from the satellite in m.

Third-Body Accelerations

The Earth is not the only celestial body exerting gravitational pull on objects in Earth orbit. Both the Moon and the Sun have a noticeable effect, the Moon due to its close proximity to the Earth and the Sun, while being much further away, because it has a very high mass. The third-body gravitational force exerted by these bodies will be included in the simulation environment for the test cases, to see if they have a significant effect on the model accuracy.

3.2.2. Torques

In the dynamics of the simulations not only forces need to be considered, since the attitude of the objects is one of the key aspects of this thesis. Therefore it also needs to be determined if and which torques should be included in the simulation environment. It was decided that the translational and rotational dynamics will be simulated independent from each other, in two separate 3 degrees-of-freedom (3DOF) propagations.

Solar Radiation Pressure

The first torque that is considered is caused by SRP. Due to their changing orientation objects will experience a difference in SRP force across different axes. This induces a rotational acceleration around these axes. However, since the force exerted by the SRP is very small, the torque caused by SRP will also be very small. This means it would again only have a noticeable effect on the estimations for very long propagations. Additionally, implementing this torque would also require complex dependencies on attitudes and surface areas, which was already determined to be too time consuming for the scope of this thesis.

Aerodynamic

Similar to the SRP, the changing attitudes of the objects also result in a difference of aerodynamic drag force exerted on different parts of the objects. Implementing this torque would come with the same complexity issue as the SRP torque, while also only having a noticeable effect over long propagation times, so this torque is also not included in the simulation environment.

Gravitational

Gravitational torque is caused by mass being separated from the center of mass of the objects, which then experience a difference in gravitational acceleration, leading to a torque. However, this torque only applies for large mass differences or very large separations, which will not be the case in this thesis, so this torque is also not included in the simulation envirnoment.

3.3. Analysis

In order to make a selection of the combination of simulation settings that will be used in the estimations, an analysis of different cases with varying simulation environment parameters was performed. This section covers the different aspects with which this analysis was set up, from the requirements to the orbit and the set up of the different cases. At the end the simulation environment parameters that are used in the rest of the thesis are decided.

3.3.1. Requirements

As mentioned at the start of this chapter, the analysis of different simulation environment parameters is done to trade-off two conflicting requirements: simulation accuracy and runtime. To make this trade-off possible, specific values for the requirements need to be set. The simulation accuracy requirement is split into two parts, one for the orbital position and one for the attitude.

For the orbit a requirement is set on the maximum three-dimensional position error, as this parameter indicates the overall quality of the orbit propagation of a simulation. For the attitude a requirement is set on the maximum error of all three Euler angle attitude components. This is done because the attitude propagation is one of the main focuses of the thesis, and it is therefore important that a simulation is able accurately propagate the attitude in all components. Finally, because simulation runtime is difficult to measure exactly, a requirement is set for the average runtime of a simulation with the same environment parameters. This means all the different cases with varying simulation environment parameters will be run multiple times to get an accurate estimation of the average runtime. The values for each of the requirements are given in Table 3.3.1.

Table 3.3.1: Simulation requirements

Requirement	Value	Unit
Maximum 3D orbital position error	1.0	km
Maximum attitude component error	0.5	°
Average simulation runtime	2.0	s

The average runtime requirement is set to 2.0 seconds because it is expected that a lot of simulations will have to be run during throughout the thesis research, so it is necessary that these simulations run quickly while still adhering to the required accuracy.

3.3.2. Orbit and attitude

For the analysis the CryoSat 2 orbit will be simulated for 1000 minutes for each case, which comes down to approximately 10 full orbits. The starting time was set to 2024-09-18 04:33:42 TDB. The initial Cartesian orbital position and velocity states are given in Table 3.3.2a. The quaternion attitude and body-axis rotational velocities are given in Table 3.3.2b.

Table 3.3.2: Initial states for CryoSat 2 propagation used in analysis.

(a) Initial Cartesian orbit states.

r_x [km]	r_y [km]	r_z [km]	v_x [km/s]	v_y [km/s]	v_z [km/s]
-3.8555×10^3	4.0479×10^3	-4.3494×10^3	-2.9348	3.5392	5.9156

(b) Initial attitude and rotational velocity states.

q_1 [-]	q_2 [-]	q_3 [-]	q_4 [-]	ω_1 [°/s]	ω_2 [°/s]	ω_3 [°/s]
0.860254	0.1	0.1	0.3	1.0	1.0	1.0

The satellite is given high rotational velocities of 1 °/s on each axis, which is higher than is expected to be encountered in real life. This is done in order to more rigorously test the different simulation environment parameters on their accuracy.

3.3.3. Cases

To make a selection of the combination of simulation parameters to be used, 9 cases with different simulation parameters were set up. Each subsequent case is lower in complexity than the previous model in one parameter. This way the performance of the different cases on the requirements can be compared to each other. The goal of this analysis is to determine the best parameter combination that balance the runtime and accuracy requirements. The cases and their respective simulation parameters are given in Table 3.3.3.

Table 3.3.3: Simulation environment parameters of analysis cases.

Case	Integrator	Time step [s]	Earth SHG coefficients	SRP	Sun PMG	Moon PMG
Benchmark	RKF1412	0.1	(100, 100)	x	x	x
0	RKF1412	0.1	(100, 100)	x		
1	RKF1412	1.0	(100, 100)	x		
2	RKF56	1.0	(100, 100)	x		
3	RKF56	5.0	(100, 100)	x		
4	RKF56	5.0	(10, 10)	x		
5	RK4	5.0	(10, 10)	x		
6	RK4	5.0	(10, 10)			
7	RKF56	5.0	(2, 2)	x		
8	RK4	5.0	(2, 2)	x		
9	RKF56	5.0	(2, 2)			

The position and attitude errors for each case were calculated with respect to a benchmark propagation. This benchmark was set up as the most accurate propagation by using a high order, fixed-step propagator, namely RKF1412 of order 12, a small time step of 0.1 s, Earth Spherical Harmonic Gravity (SHG) coefficients of (100, 100), cannonball SRP and both Sun and Moon Point Mass Gravity (PMG). Setting the results of this propagation as the base makes it possible to compare the errors of the other cases to each other.

3.3.4. Results

The results for the all the test case propagations are shown in Table 3.3.4. It can be seen that as the complexity of the propagations decrease the average runtimes decrease and the errors increase. Only case 5, 6 and 8 have average runtimes that fall below the runtime required of 2.0 s. However, case 8 has an orbital position error higher than 1.0 km. Only case 5 and 6 fall within all the requirements for accuracy and runtime.

Table 3.3.4: Results for simulation environment analysis cases.

Case	Avg. runtime [s]	Max e_r [km]	Max $e_{\theta_1}[^\circ]$	Max $e_{\theta_2}[^\circ]$	Max $e_{\theta_3}[^\circ]$
0	5009	0.068	0.0	0.0	0.0
1	563	0.068	$1.0265 * 10^{-10}$	$1.0764 * 10^{-11}$	$1.0328 * 10^{-10}$
2	141	0.068	$7.0791 * 10^{-11}$	$1.2323 * 10^{-11}$	$7.8554 * 10^{-11}$
3	28	0.068	$2.8888 * 10^{-6}$	$1.9954 * 10^{-7}$	$3.0412 * 10^{-6}$
4	3.01	0.139	$2.8888 * 10^{-6}$	$1.9954 * 10^{-7}$	$3.0412 * 10^{-6}$
5	1.82	0.139	0.3158	0.0022	0.3325
6	1.76	0.147	0.3158	0.0022	0.3325
7	2.71	7.461	$2.8888 * 10^{-6}$	$1.9954 * 10^{-7}$	$3.0412 * 10^{-6}$
8	1.60	7.461	0.3158	0.0022	0.3325
9	2.59	7.447	$2.8888 * 10^{-6}$	$1.9954 * 10^{-7}$	$3.0412 * 10^{-6}$

Between case 5 and 6 the only difference is that in case 5 cannonball SRP is included, and in case 6 it is not. The only difference this makes is that the orbital position error for case 6 is slightly higher than that of case 5, and the runtime for case 6 is slightly lower.

3.3.5. Final environment settings

It is decided to use the simulation environment parameters of case 6 for the environment settings used in the rest of this thesis. This means that the cannonball SRP is not included in the dynamics of the simulations. The reason for this is that the cannonball SRP model is an oversimplification of the real life dynamics that overestimates the effects of SRP on the propagation. Since the inclusion of the model in the simulation only leads to a negligible decrease in position error, it was determined that including it in the simulations only adds unnecessary complexity to the simulations.

The analysis for the simulation environment was only performed for the CryoSat 2 orbit. This is seen as a representative orbit, having an altitude that is in-between the lowest and highest orbit altitudes in Table 3.1.1. It can be seen in Figure 3.2.1 that the magnitudes of the forces and torques not included in the final environment are also very low for the different altitudes. Analyses for the other orbits would therefore produce very similar results, and lead to the same conclusion for the final environment settings.

4 Methodology

In this chapter the methods that have been used and developed for the thesis are explained. In Section 4.1 the method used to simulate light curves for the estimations are discussed. In Section 4.2 the attitude estimation algorithm is explained. In Section 4.3 the process of Multiple-Model Adaptive Estimation is highlighted. Next a method not yet applied to the light curve inversion problem called Variable-Structure Multiple-Model estimation is detailed in Section 4.4.

4.1. Light curve simulation

In this thesis the attitude and shape estimation will be done using light curves. A light curve is the variation in apparent brightness over time measured from an observer on the ground. It is possible to measure this light curve for many different objects. However, for most of these satellites the attitude is either not known, or not publicly available. This makes it impossible to verify whether the attitude estimated by the model is close to the real attitude, or completely wrong. Therefore it is necessary to produce simulated light curves from known propagated attitudes. In order to do this use is made of so-called Bidirectional Reflectance Distribution Functions (BRDFs). These functions model the way light is reflected diffusely and specularly off a surface. In this section the equations for one such BRDF model is highlighted.

4.1.1. Bidirectional Reflectance Distribution Functions

The BRDFs model the way light is scattered from a surface due to incident light. At any point on the surface a BRDF is a function of two directions, the direction from which the light is originating and the direction from which the scattered light is observed. The BRDF is usually decomposed into a specular and a diffuse component.

The diffuse component represent light that is scattered uniformly in all directions, while the specular component represents light that is concentrated in a certain direction, like a mirror. A visualisation of the difference between these reflections is given in Figure 4.1.1.

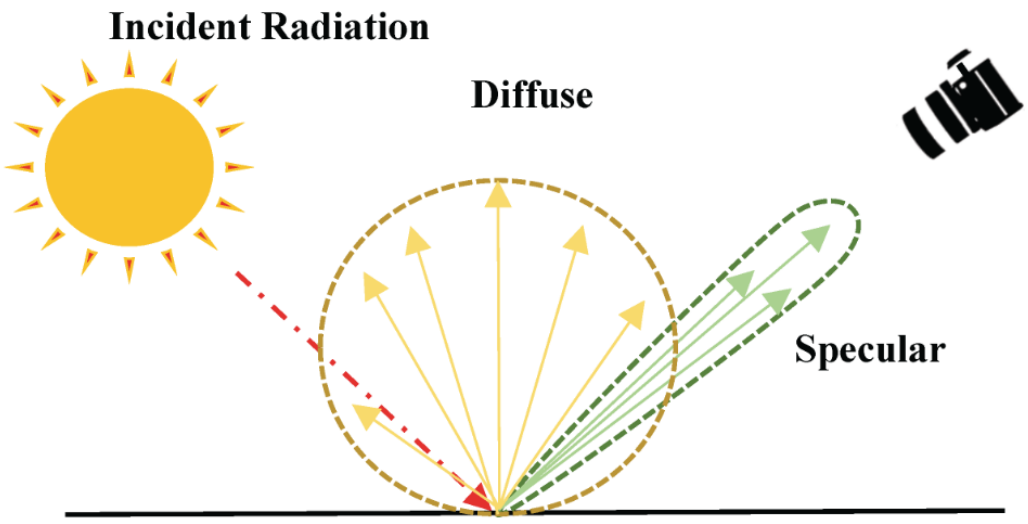


Figure 4.1.1: Specular and diffuse reflections off a surface towards an observer [19].

There exist models that can calculate the BRDF for any continuous arbitrary surface. In practice it is more convenient to use a simplified model that works for flat surfaces, and represent an object with a finite number of flat surfaces or facets. The total observed brightness of an object is then calculated to be the sum of the individual brightnesses of each facet. For a curved surface more facets would be required to accurately model the brightness. The observation geometry of a single facet is shown in Figure 4.1.2.

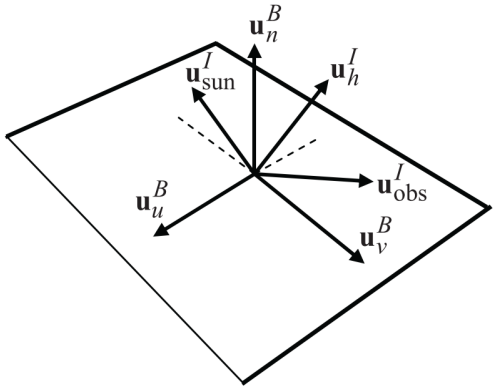


Figure 4.1.2: Facet reflection geometry for BRDF calculations [12].

The basis vectors of the facet are the unit vectors $\mathbf{u}_n^B$, $\mathbf{u}_u^B$ and $\mathbf{u}_v^B$, where the superscript B denotes that the vector is expressed in the body frame of the object. The vector $\mathbf{u}_n^B$ is the surface normal vector of the facet. The vectors $\mathbf{u}_u^B$ and $\mathbf{u}_v^B$ are in the plane of the facet. The objects that are modelled are assumed to be rigid bodies, so these three vectors do not change over time in the body frame. For the brightness calculations in the BRDFs the vectors need to be expressed in the inertial coordinate frame. Since the body is rotating, the vectors will change in this coordinate frame. The body vectors can be rotated to the inertial frame via quaternion rotation, using the quaternion that defines the rotation from the body frame to the inertial frame at the corresponding time, given by Equation 4.1.1. In this thesis the scalar part of the quaternion is defined as q_1, and the vector part ϱ consists of q_2, q_3 and q_4. The complete definition of the quaternion attitude representation is given in Subsection 4.2.2.

$$\mathbf{u}_i^B = A(\mathbf{q}_B^I)\mathbf{u}_i^I \tag{4.1.1}$$

$$A(\mathbf{q}) = \mathbf{\Xi}^T(\mathbf{q})\Psi(\mathbf{q}) \tag{4.1.2}$$

$$\Xi(\mathbf{q}) \equiv \begin{bmatrix} -\varrho^T \\ q_1 I_{3x3} + [\varrho\times] \end{bmatrix} \tag{4.1.3a}$$

$$\Psi(\boldsymbol{q}) \equiv \begin{bmatrix} -\varrho^T \\ q_1 I_{3x3} - [\varrho\times] \end{bmatrix} \tag{4.1.3b}$$

$$[\varrho\times] \equiv \begin{bmatrix} 0 & -q_4 & q_3 \\ q_4 & 0 & -q_2 \\ -q_3 & q_2 & 0 \end{bmatrix} \tag{4.1.4}$$

The additional vectors $\mathbf{u}_{\text{sun}}^{\mathbf{I}}$ and $\mathbf{u}_{\text{obs}}^{\mathbf{I}}$ are the unit vectors pointing from the facet in the direction of the sun and the observer, respectively. Lastly, the vector $\mathbf{u}_{\mathbf{h}}^{\mathbf{I}}$ is the normalised half vector or bisector between $\mathbf{u}_{\text{sun}}^{\mathbf{I}}$ and $\mathbf{u}_{\text{obs}}^{\mathbf{I}}$.

For this thesis the so-called Ashikmin-Shirley model is implemented. It is based on a complex model that works for any arbitrary continuous surface, and was simplified by Wetterer et al. [20] to work only for flat surfaces. The defintion of this model is given in the next subsection.

4.1.2. Ashikmin-Shirley model

This BRDF model is based on an anisotropic Phong light diffusion model defined by Ashikmin and Shirley [21]. In general the BRDF is calculated as a function of the diffuse bidirectional reflectance R_{diff} and the specular bidirectional reflectance R_{spec}, and the fraction of each to the total reflectance, which are d and s, respectively, with $d + s = 1$. For the Ashikmin-Shirley model these reflectances are constant, and are given by Equation 4.1.5. Here ρ is the diffuse reflectance and F_0 is the specular reflectance of the surface at normal incidence. Both parameters have a value between 0 and 1. The second part of the equation defines that energy has to be conserved in the system.

$$R_{\text{diff}} = d * \rho, \; R_{\text{spec}} = s * F_0 \tag{4.1.5a}$$

$$R_{\text{diff}} + R_{\text{spec}} \leq 1 \tag{4.1.5b}$$

The equations to calculate specular and diffuse terms of the BRDF ρ_{spec} and ρ_{diff} are adapted from Linares et al. [12]. The specular term is calculated using Equation 4.1.6. The parameters n_u and n_v define the anisotropic reflectance properties of a surface.

$$\rho_{\text{spec}} = \frac{\sqrt{(n_u+1)(n_v+1)}}{8\pi} \frac{\left(\mathbf{u}_n^I \cdot \mathbf{u}_h^I\right)^z}{\left(\mathbf{u}_h^I \cdot \mathbf{u}_{\text{sun}}^I\right) \max\left(\mathbf{u}_n^I \cdot \mathbf{u}_{\text{sun}}^I, \mathbf{u}_n^I \cdot \mathbf{u}_{\text{obs}}^I\right)} F_{\text{reflect}} \tag{4.1.6a}$$

$$z = \frac{n_u \left(\mathbf{u}_h^I \cdot \mathbf{u}_u^I\right)^2 + n_v \left(\mathbf{u}_h^I \cdot \mathbf{u}_v^I\right)^2}{1 - \left(\mathbf{u}_n^I \cdot \mathbf{u}_{\text{sun}}^I\right)^2} \tag{4.1.6b}$$

$$F_{\text{reflect}} = R_{\text{spec}} + (1 - R_{\text{spec}})(1 - \mathbf{u}_h^I \cdot \mathbf{u}_{\text{sun}}^I)^5 \tag{4.1.6c}$$

The dot products in the equation are functions of the reflection geometry described by Figure 4.1.2, with all vectors being expressed in the inertial coordinate frame. The diffuse term of the BRDF is given by Equation 4.1.7.

$$\rho_{\text{diff}} = \left(\frac{28 R_{\text{diff}}}{23\pi}\right)(1 - R_{\text{spec}}) \left[1 - \left(1 - \frac{\mathbf{u}_n^I \cdot \mathbf{u}_{\text{sun}}^I}{2}\right)^5\right] \left[1 - \left(1 - \frac{\mathbf{u}_n^I \cdot \mathbf{u}_{\text{obs}}^I}{2}\right)^5\right] \tag{4.1.7}$$

The total BRDF is then given by the sum of the specular and diffuse terms, as given by Equation 4.1.8.

$$\rho_{\text{total}} = \rho_{\text{spec}} + \rho_{\text{diff}} \tag{4.1.8}$$

The fraction of visible sunlight that hits a facet and is not absorbed is calculated by Equation 4.1.9. Here $C_{\text{sun,vis}}$ is the power per square meter on a facet due to visible light hitting the surface, and is equal to 455 W/m^2 at 1AU.

$$F_{\text{sun}} = C_{\text{sun,vis}} \rho_{\text{total}} (\mathbf{u}_n^I \cdot \mathbf{u}_{\text{sun}}^I) \tag{4.1.9}$$

If the angle between the surface normal and the direction of the observer or the angle between the surface normal and the direction of the sun is larger than 90° or $\pi/2$ rad then no light is reflected from that facet to the observer. The value of F_{sun} is then set to 0.

The fraction of sunlight that hits a facet and is reflected is calculated by Equation 4.1.10. Here $A_{\textbf{facet}}$ is the area of the facet in m^2, and $||\mathbf{r}_{\text{obs}}||^2$ is the range from the observer to the object in m.

$$F_{\text{obs}} = \frac{F_{\text{sun}} A_{\text{facet}} (\mathbf{u}_n^I \cdot \mathbf{u}_{\text{obs}}^I)}{||\mathbf{r}_{\text{obs}}||^2} \tag{4.1.10}$$

The F_{obs} value is calculated for each facet seperately, and all the values are summed to one value of $F_{\text{obs,total}}$ for the entire object. The apparent brightness magnitude of the full object can then be calculated with Equation 4.1.11. The -26.7 value is the apparent brightness of the sun.

$$m_{\text{app}} = -26.7 - 2.5 \log_{10} \left| \frac{F_{\text{obs,total}}}{C_{\text{sun,vis}}} \right| \tag{4.1.11}$$

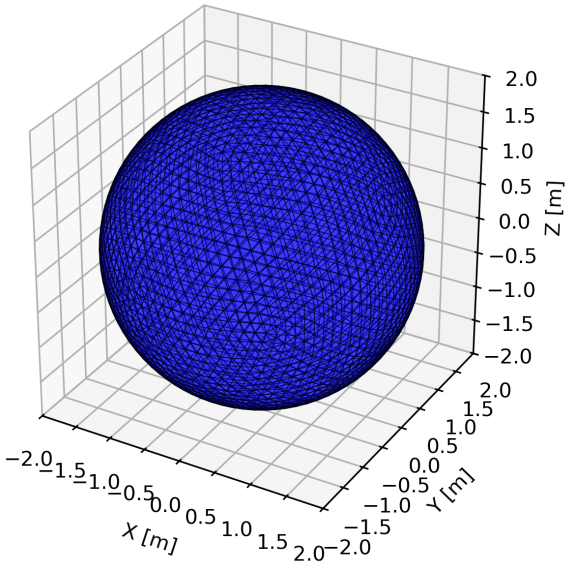
4.1.3. Verification

The implementation of the Ashikmin-Shirley BRDF model can be verified by simulating different cases and checking if the light curves produced by the model conform to expectations based on the physics and geometry of the specific cases. The aspects that will be looked at with these cases are the influence of observation geometry, object rotation and reflectivity coefficients.

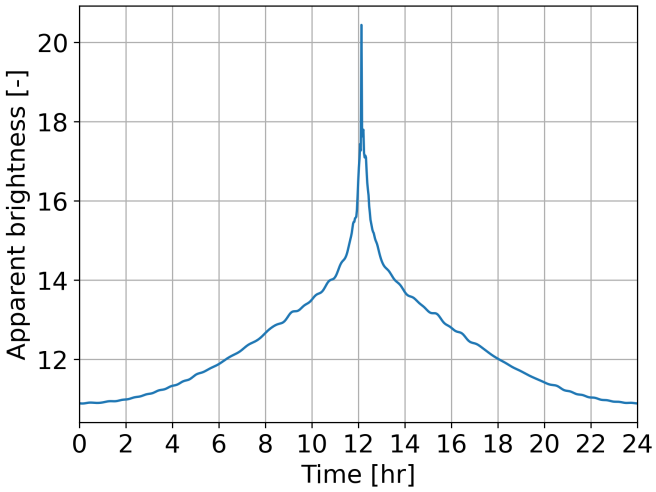
Geostationary sphere

The first case that will be analysed is a spherical object with a radius of 2.0 m in geostationary orbit. The sphere is given 0 body-axis rotational velocity, and is observed from a station at the equator at 0°N latitude and 0°W longitude and 0 m altitude. At the start of propagation the sphere is placed directly above the observer. This orientation will stay constant during the propagation because of the geostationary orbit. In setting up the case this way the effects of the object shape and attitude on the light curve are removed, leaving only the influence of the geometry of the observation.

The sphere is modelled using 5120 triangular facets, in order to make the shape very smooth while still running in reasonable time. The model is shown in Figure 4.1.3a. The BRDF constants of all the facets are: $n_u = 1000$, $n_v = 1000$, $R_{\text{spec}} = 0.5$, $R_{\text{diff}} = 0.5$. The light curve is simulated for an orbit duration of 24 hours, with a timestep of 100 s. Shadowing of the Earth is not taken into account. The resulting light curve is shown in Figure 4.1.3b.



(a) Sphere model consisting of 5120 triangular facets.



(b) Simulated light curve of non-spinning sphere of 2.0 m radius in geostationary orbit, observed from equator.

Figure 4.1.3: Spherical model and simulated light curve.

As mentioned the only aspect influencing the shape of the light curve is the observation geometry, meaning the relative positions of the observer, satellite and the sun. Since the sphere is in geostationary orbit, the relative position between the sphere and the observer does not change during the orbit. The only change is thus the position of the sun with respect to the observer and the sphere. This is reflected very clearly in Figure 4.1.3b.

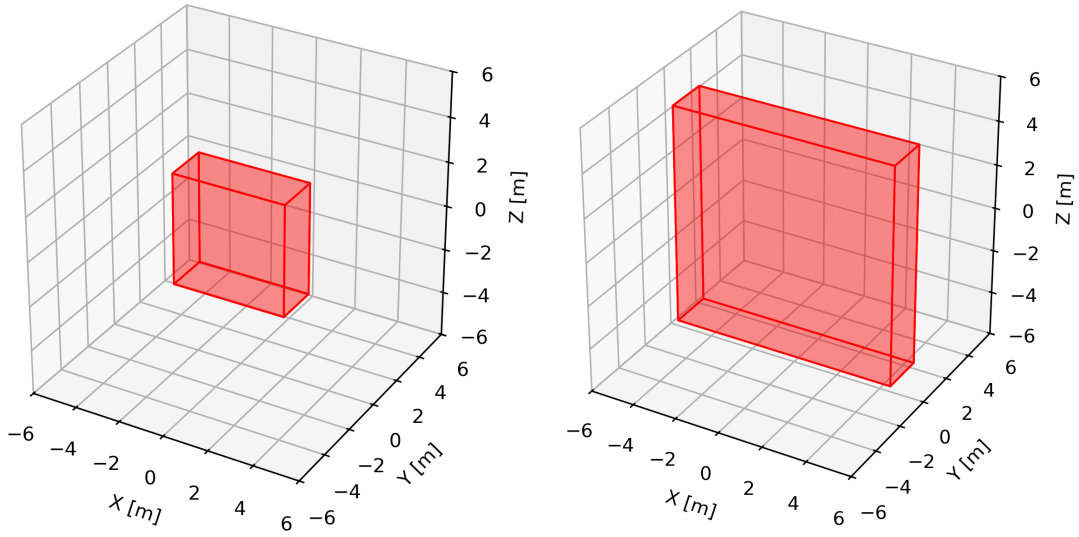
At local midnight, both at 0 and 24 hours propagation time, the apparent brightness of the sphere is at a maximum. This is in line with expectations, as the satellite is in full view of the sun, and thus the largest part of the sphere is illuminated and reflects light towards the observer. Conversely, at local noon the brightness is at a minimum, highlighted by the sharp peak in the figure. At this point in time the satellite is positioned between the sun and the Earth. This means only a small part of the sphere is visible to the observer, as mostly the back of the satellite will be illuminated, which does not reflect light towards the observer. In between local midnight and noon the visibility gradually changes from more to less visible, and vice versa. This behaviour is again what is expected from the brightness of a spherical satellite.

It can be concluded that the implemented Ashikmin-Shirley BRDF model works correctly in the case of a geostationary sphere. However, this is a very simple case, and not representative of the type of objects and orbits that will be encountered during the thesis. The verification of the BRDF model is therefore continued with more realistic test objects next.

Spinning cuboids

It has been established that the Ashikmin-Shirley algorithm works as expected for the simple case of a non-spinning sphere in geostationary orbit. This is of course not the type of object that is gonna be simulated during the thesis. To examine the results of the BRDF functions the next cases that are going to be looked contain two cuboids of different sizes. In the first case both cuboids are given 0 rotational velocity. In the second case they are given a slow rotational velocity around one body axis. The two cuboids that are chosen are similarly shaped, but one of them has the x- and z-dimensions doubled compared to the other. This is done to assess whether the effect of facets having larger cross-sectional areas is translated correctly in the light curves.

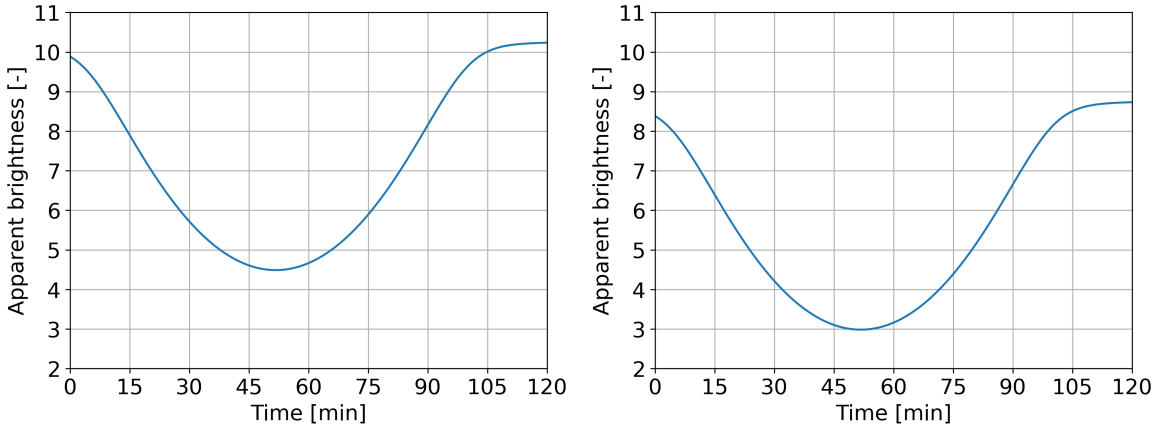
The cuboids are shown in Figure 4.1.4. The first has dimensions (x=5.0 m, y=2.0 m, z=5.0 m), while the second has the x- and z-dimensions doubled to 10.0 m. Both cuboids have an initial quaternion attitude equal to the identity quaternion $\mathbf{q}_I = [1, 0, 0, 0]$. The cuboids are propagated in the same geostationary orbit as the sphere, but are only propagated for 120 minutes, with a time step of 1 s.



(a) Cuboid with dimensions (x=5.0 m, y=2.0 m, z=5.0 m). **(b)** Cuboid with dimensions (x=10.0 m, y=2.0 m, z=10.0 m).

Figure 4.1.4: Cuboids used in verification case.

The light curves for both cuboids propagated without rotational velocities are shown in Figure 4.1.5. Here the effect of the larger surface areas of the larger cuboid can be seen clearly. The light curve is shifted down by about 1.5 brightness values for the larger cuboid compared to the small one, while the shape of the curve remains the same. As is expected, the large cuboid thus appears brighter than the smaller cuboid, because the larger surface areas of the large cuboid reflect more light. What is interesting is that the maximum brightness in both curves does not occur at 0 minutes propagation time, but around 50 minutes. This is the case because at the start of the propagation the sun, Earth and satellites are not exactly in line. At 50 minutes they are aligned, meaning the satellites are maximally illuminated by the sun and reflect the most light to the observer.

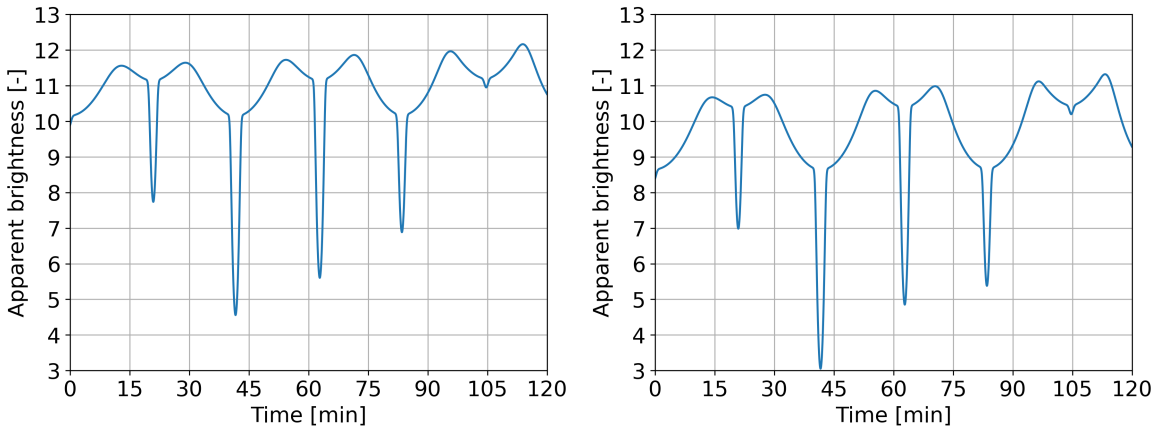


(a) Light curve of smaller cuboid, non-spinning. **(b)** Light curve of larger cuboid, non-spinning.

Figure 4.1.5: Light curves of non-spinning cuboids.

Next the spinning cuboids are examined. Both cuboids are given the same body-axis rotational velocity $\boldsymbol{\omega} = [0, 0.0025, 0]$ rad/s, meaning a rotational velocity of 0.14 °/s around the y-axis. The corresponding light curves are shown in Figure 4.1.6.

The effect of the rotations is clearly visible in both curves. Initially they start at the same brightness magnitudes as the non-spinning case, but these quickly decrease as the facet with the largest surface sectional area spins away. After this initial decrease the brightness only slowly decreases, and after a peak it slowly increases again until a sharp increase around 20 minutes. This peak corresponds exactly with the time it takes to rotate 180° with a rotational velocity of 0.14°/s, namely 1286 seconds or 21.4 minutes, and therefore corresponds to the other large area facet rotating into full view.



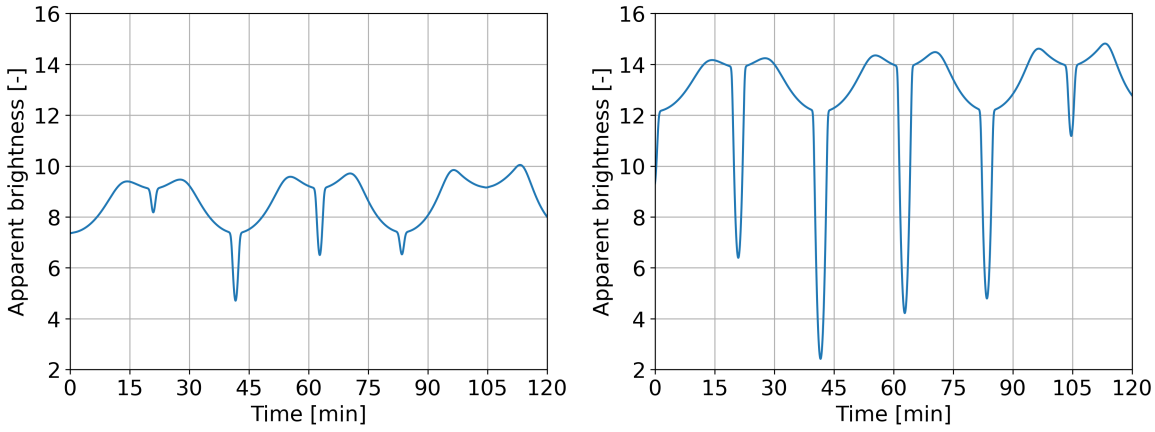
(a) Light curve of smaller cuboid, spinning.

(b) Light curve of larger cuboid, spinning.

Figure 4.1.6: Light curves of non-spinning cuboids.

Both light curves show these sharp peaks every 21 minutes, which is as expected as they have the same rotational velocity. Similar to the non-spinning case, the curve for the larger cuboid is shifted down compared to the curve of the small cuboid, again matching the fact that the larger cuboid appears brighter. The effect of the changing observation geometry is also still visible, as the values of the increased brightness peaks are the largest around 50 minutes, which is when the satellites are positioned to receive and reflect the most sunlight.

The last aspect that is studied is the effect of the reflectivity coefficients R_{spec} and R_{diff}. These determine the way light is reflected off the facet surfaces. If a surface has a higher diffuse coefficient, light is spread more in all different directions, while a higher specular coefficient means light is reflected more mirror-like. To study this effect light curves were simulated for the large cuboid, in one case where all the surfaces have coefficients $R_{\text{spec}} = 0.1$ and $R_{\text{diff}} = 0.9$ (Figure 4.1.7a), and the other $R_{\text{spec}} = 0.9$ and $R_{\text{diff}} = 0.1$ (Figure 4.1.7b).



(a) Light curve for large cuboid with $R_{\text{spec}} = 0.1$ and $R_{\text{diff}} = 0.9$

(b) Light curve for large cuboid with $R_{\text{spec}} = 0.9$ and $R_{\text{diff}} = 0.1$

Figure 4.1.7: Light curves for large cuboid with different reflectivity coefficients R_{spec} and R_{diff}.

It is clear that for the diffuse surfaces in Figure 4.1.7a the light curve is much more constrained, with only small peaks for the large area facets. This is in line with expectations, as in this case the light is spread in a lot of different directions, so an observer receives less direct light and the objects appears less bright. However, overall the objects appears more bright for a longer time period, as throughout the entire observation more light is reflected towards the observer. In contrast, the peaks in Figure 4.1.7b are much larger, as at the right time a lot more light is reflected to the observer. However, this also means less light is directly reflected to the observer in between these peaks, causing the object to appear less bright most of the time.

Conclusion

The light curves simulated by the Ashikmin-Shirley algorithm match the results expected by looking at the physics and geometries of both cases. The case with the non-rotating sphere clearly demonstrates the capability of the algorithm to simulate the effects caused by the orbit and changing observational geometry. Then the cases with the spinning cuboids showed that the algorithm models both the effect of larger facet surface areas and rotations correctly. The effect of different reflectivity coefficients is also clearly visible in the curves. The algorithm is thus ready to be used for the estimation of orbit and attitude states and the characterisation of object shape, the methodology for which is explained in the next sections.

4.2. Attitude estimation through light curve inversion

The method that was implemented for attitude and angular velocity estimation was first introduced by Crassidis and Markley [22], and was later adapted by Wetterer and Jah [4] and Linares et al. [12][23].

4.2.1. Unscented Kalman Filter

The method uses an Unscented Kalman Filter (UKF) to estimate attitudes and angular velocities, next to orbital positions and velocities. The UKF works by generating so-called sigma points. These sigma points are sampled from a Probability Density Function (PDF) around the mean state prediction. The reasoning behind this is that it is easier to propagate samples from a PDF trough a non-linear function, like attitude propagation, than to propagate the PDF itself. The idea is that Gaussian distributions can be represented by finite set of deterministically selected samples, which are the sigma points. Given a state vector $\mathbf{x}_k$ with mean $\boldsymbol{w}_k$ of size n and a state error-covariance $\mathbf{P}_k$ of size n x n, a sigma point matrix $\boldsymbol{\chi}$ with 2n + 1 sigma points is constructed using Equation 4.2.1.

$$\boldsymbol{\sigma}_k \leftarrow 2n \text{ columns from } \pm\sqrt{(n+\lambda)\mathbf{P}_k} \tag{4.2.1a}$$

$$\chi_k(0) = \boldsymbol{w}_k \tag{4.2.1b}$$

$$\chi_k(i) = \boldsymbol{w}_k + \sigma_k(i), \quad i = 1, 2, \ldots, 2n \tag{4.2.1c}$$

Here $\sqrt{M}$ is shorthand notation for a matrix Z such that $M = ZZ^T$. The scalar $\lambda = \alpha^2(n+\kappa) - n$ is a composite scaling parameter. The constant α controls the spread of the sigma point distribution and should be a number between 0 and 1, with a typical lower bound being 10^{-4}. Constant κ provides an extra degree of freedom that can be used to fine-tune the higher order moments. As the sigma points are selected to represent the distribution of the state vector, each point is given a weight that preserves the information contained in the initial distribution, which are given by Equation 4.2.2.

$$W_0^{\text{mean}} = \frac{\lambda}{n+\lambda} \tag{4.2.2a}$$

$$W_0^{\text{cov}} = \frac{\lambda}{n+\lambda} + \left(1 - \alpha^2 + \beta\right) \tag{4.2.2b}$$

$$W_i^{\text{mean}} = W_i^{\text{cov}} = \frac{1}{2(n+\lambda)}, \quad i = 1, 2, \ldots, 2n \tag{4.2.2c}$$

The parameter β is used to incorporate prior knowledge of the distribution by weighting the mean sigma point in the covariance calculation step. Typically $\kappa = 3 - n$ and $\beta = 2$ are used as good starting guesses to tune the filter.

4.2.2. State representation

Because Euler angle attitude representations have the shortcoming of singularities and gimbal lock, the attitude is represented with body-frame quaternions. Propagating and estimating quaternions also come with an important limitation, namely that they need to abide a normalisation constraint. The quaternion is based on the Euler angle/axis parametrisation, and is defined by Equation 4.2.3. Here $\hat{\mathbf{e}}$ and ν are the Euler axis of rotation and rotation angle, respectively. The quaternion has to satisfy the unit norm constraint $\mathbf{q}^T\mathbf{q} = 1$.

$$\mathbf{q} \equiv \begin{bmatrix} q_1 \\ \boldsymbol{\varrho} \end{bmatrix} \tag{4.2.3a}$$

$$q_1 = \cos(\nu/2) \tag{4.2.3b}$$

$$\boldsymbol{\varrho} = \hat{\mathbf{e}}\sin(\nu/2) \tag{4.2.3c}$$

In the UKF the quaternions are not directly propagated themselves. Instead, a 3-dimensional attitude error represented by Generalised Rodrigues Parameters (GRPs) is used in the state vector to propagate the attitude. This is done in order to better utilise the capabilities of the UKF. This representation does come with a singularity, either at 180° or 360°, but according to Crassidis et al. [22] this is not encountered in practice. The update steps in the UKF are performed using quaternion multiplication, which leads to a natural way of maintaining the normalization constraint.

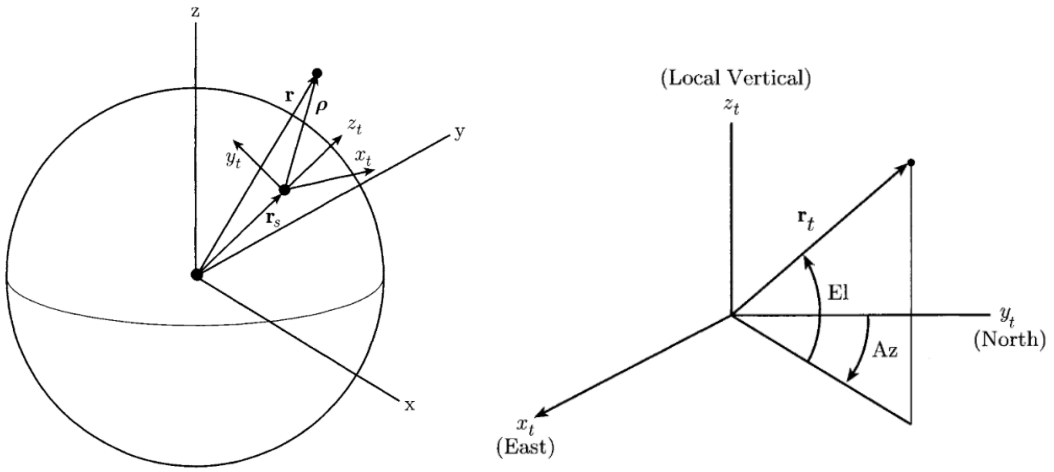
The full state vector that is used in the joint attitude and position estimation is given by Equation 4.2.4. Here $\delta\hat{\mathbf{p}}$ is the error GRP associated with the quaternion $\hat{\mathbf{q}}_I^B$ and $\hat{\cdot}$ is used to denote the parameter in question is an estimate. The vector $\boldsymbol{\omega}$ is the angular velocity of the body with respect to the inertial frame, expressed in body frame coordinates. The full notation $\boldsymbol{\omega}_{B/I}^B$ is omitted for clarity. Both $\mathbf{r}$ and $\mathbf{v}$ are expressed in the inertial frame.

$$\hat{\mathbf{x}}_k^+ = \left[\begin{array}{c} \hat{\mathbf{r}}^{I+} \\ \hat{\mathbf{v}}^{I+} \\ \delta\hat{\mathbf{p}}^+ \\ \hat{\boldsymbol{\omega}}^+ \end{array} \right]\Bigg|_{t_k} \tag{4.2.4}$$

4.2.3. Observation models

Measurements are given as a combination of astrometric and photometric observations. The astrometric observations are used for the orbit estimation in the UKF, and the photometric observations are used for the attitude estimation. Both types of observations are made from an optical ground station. The photometric observations are the apparent brightness measurements that form a light curve as discussed in Section 4.1, using the Ashikmin-Shirley BRDF model to simulate the brightness measurements.

The astrometric observations are the azimuth, elevation and range to the object that is measured, as seen from the ground station. These parameters are defined in the topocentric reference frame, as shown in Figure 4.2.1a. The origin of this frame is the position of the observer, located at $\mathbf{r}_s$ from the center of the Earth. The topocentric coordinates consist of x_t pointing east, y_t pointing north and z_t pointing toward the local vertical direction.



(a) Topocentric reference frame on Earth.

(b) Local view of topocentric reference frame.

Figure 4.2.1: Visualisations of topocentric frame [24].

In order to calculate the azimuth and elevation angles of a satellite passing over an observer the position of the satellite in the Earth-Centred Earth-Fixed (ECEF) coordinate frame $\mathbf{r}$ needs to be known. With this the line of sight vector $\boldsymbol{\rho}$ can be calculated, which is given by $\mathbf{r} - \mathbf{r_s}$. Knowing the longitude λ and latitude ϕ of the observer position on the Earth, the position of the satellite in the topocentric frame can be calculated using Equation 4.2.5.

$$\mathbf{r_t} = \begin{bmatrix} r_e \\ r_n \\ r_u \end{bmatrix} = T_t(\mathbf{r} - \mathbf{r_s}) = T_t\boldsymbol{\rho} \tag{4.2.5a}$$

$$T_t = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\sin\phi\cos\lambda & -\sin\phi\sin\lambda & \cos\phi \\ \cos\phi\cos\lambda & \cos\phi\sin\lambda & \sin\phi \end{bmatrix} \tag{4.2.5b}$$

With r_e, r_n and r_u being the east, north and up coordinates of the satellite in the topocentric frame, as shown in Figure 4.2.1b, the azimuth angle Θ and elevation angle Φ can be calculated with Equation 4.2.6.

$$\Theta = \tan^{-1}\frac{r_e}{r_n} \tag{4.2.6a}$$

$$\Phi = \sin^{-1}\frac{r_u}{||\boldsymbol{\rho}||} \tag{4.2.6b}$$

4.2.4. Prediction step

At the start of the prediction the error GRPs $\delta\hat{\mathbf{p}}^+$ in the initial state vector are set to 0. Sigma points are then calculated for the error GRPs, with the 0 GRP vector as mean. These are then first converted into error quaternions so that the quaternion sigma points can be computed. The error quaternion $\delta\mathbf{q}_k^+(i)$ corresponding with the ith error GRP sigma point is calculated with Equation 4.2.7. In this equation a is a parameter with a value between 0 and 1, and f is a scaling factor, which is often set to $f = 2(a+1)$.

$$\delta\varrho_k^+(i) = f^{-1}\left[a + \delta q_{1_k}^+(i)\right]\boldsymbol{\chi}_k^{\delta p}(i) \tag{4.2.7a}$$

$$\delta q_{1_k}^+(i) = \frac{-a\left\|\boldsymbol{\chi}_k^{\delta p}(i)\right\|^2 + f\sqrt{f^2 + (1-a^2)\left\|\boldsymbol{\chi}_k^{\delta p}(i)\right\|^2}}{f^2 + \left\|\boldsymbol{\chi}_k^{\delta p}(i)\right\|^2} \tag{4.2.7b}$$

$$\delta\mathbf{q}_k^+(i) = \begin{bmatrix}\delta q_{1_k}^+(i)\\ \delta\varrho_k^+(i)\end{bmatrix} \tag{4.2.7c}$$

The ith quaternion sigma point is given by a rotation of $\delta\mathbf{q}_k^+(i)$ about the initial quaternion estimate $\hat{\mathbf{q}}_{k_0}^+$ using Equation 4.2.8.

$$\hat{\mathbf{q}}_k^+(i) = \delta\mathbf{q}_k^+(i) \otimes \hat{\mathbf{q}}_{k_0}^+ \tag{4.2.8}$$

The symbol $\otimes$ represent a quaternion multiplication. Given two quaternions $\mathbf{p} = [p_1, p_2, p_3, p_4]^T$ and $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$ where p_1 and q_1 are the scalar part of the quaternions and $\mathbf{p}_v = [p_2, p_3, p_4]^T$ and $\mathbf{q}_v = [q_2, q_3, q_4]^T$ the vector parts, quaternion multiplication is defined by Equation 4.2.9.

$$\mathbf{p}\otimes\mathbf{q} = \begin{bmatrix} p_1q_1 - \mathbf{p}_v\cdot\mathbf{q}_v \\ p_1\mathbf{q}_v + q_1\mathbf{p}_v + \mathbf{p}_v\times\mathbf{q}_v\end{bmatrix} = \begin{bmatrix} p_1q_1 - p_2q_2 - p_3q_3 - p_4q_4\\ p_1q_2 + q_1p_2 + p_3q_4 - p_4q_3\\ p_1q_3 + q_1p_3 + p_4q_2 - p_2q_4\\ p_1q_4 + q_1p_4 + p_2q_3 - p_3q_2\end{bmatrix} \tag{4.2.9}$$

The quaternion sigma points are then propagated through the system dynamics given in Equation 4.2.10. For this thesis $\mathbf{f}(\boldsymbol{\chi}, \hat{\mathbf{q}})$ is the attitude propagation using Tudat, which is defined in Chapter 3.

$$\dot{\boldsymbol{\chi}}(i) = \mathbf{f}(\boldsymbol{\chi}(i), \hat{\mathbf{q}}(i)) \tag{4.2.10}$$

The estimated mean propagated sigma point $\hat{\mathbf{q}}_{k+1}^-(0)$ is stored, and the error quaternions corresponding to each propagated quaternion sigma point are calculated with Equation 4.2.11, where the conjugate quaternion $\mathbf{q}^{-1}$ is given by Equation 4.2.12.

$$\delta\hat{\mathbf{q}}_{k+1}^-(i) = \hat{\mathbf{q}}_{k+1}^-(i) \otimes \left[\hat{\mathbf{q}}_{k+1}^-(0)\right]^{-1} \tag{4.2.11}$$

$$\mathbf{q}^{-1} \equiv \begin{bmatrix} q_1 \\ -\varrho\end{bmatrix} \tag{4.2.12}$$

The error GRP sigma points are then calculated by Equation 4.2.13. The error GRPs for the mean sigma point are again set to 0.

$$\delta\mathbf{p}_{k+1}^-(i) = f\frac{\delta\hat{\varrho}_{k+1}^-(i)}{a + \delta\hat{q}_{1_{k+1}}^-(i)} \tag{4.2.13}$$

The propagated mean and covariance predictions are calculated as the weighted sum of the sigma points using Equation 4.2.14 and Equation 4.2.15 respectively, with the weights calculated in Equation 4.2.2. In the covariance calculation $\mathbf{Q}_{k+1}$ is the discrete-time process noise covariance.

$$\hat{\mathbf{x}}_{k+1}^{-} = \sum_{i=0}^{2n} W_i^{\text{mean}} \, \boldsymbol{\chi}_{k+1}^{-}(i) \tag{4.2.14}$$

$$\mathbf{P}_{k+1}^{-} = \sum_{i=0}^{2n} W_i^{\text{cov}} \left[\boldsymbol{\chi}_{k+1}^{-}(i) - \hat{\mathbf{x}}_{k+1}^{-}\right] \left[\boldsymbol{\chi}_{k+1}^{-}(i) - \hat{\mathbf{x}}_{k+1}^{-}\right]^T + \mathbf{Q}_{k+1} \tag{4.2.15}$$

4.2.5. Correction step

Measurements are given in the form described in Subsection 4.2.3 describing the observation model. The estimated observations are computed for each sigma point using Equation 4.2.16. Because the state vector contains error GRPs for the attitude, the quaternion $\hat{\mathbf{q}}_k^{-}(i)$ is added separately into the observation model.

$$\boldsymbol{\gamma}_k(i) = \mathbf{h}\left[\boldsymbol{\chi}_k^{-}(i), \hat{\mathbf{q}}_k^{-}(i)\right] \tag{4.2.16}$$

The mean estimated output is calculated using Equation 4.2.17.

$$\hat{\mathbf{y}}_k^{-} = \sum_{i=0}^{2n} W_i^{\text{mean}} \, \boldsymbol{\gamma}_k(i) \tag{4.2.17}$$

The output, innovation and cross-correlation covariances P_k^{yy}, P_k^{vv} and P_k^{xy} are calculated with Equation 4.2.18.

$$P_k^{yy} = \sum_{i=0}^{2n} W_i^{\text{cov}} \left[\boldsymbol{\gamma}_k(i) - \hat{\mathbf{y}}_k^{-}\right] \left[\boldsymbol{\gamma}_k(i) - \hat{\mathbf{y}}_k^{-}\right]^T \tag{4.2.18a}$$

$$P_k^{vv} = P_k^{yy} + R_k \tag{4.2.18b}$$

$$P_k^{xy} = \sum_{i=0}^{2n} W_i^{\text{cov}} \left[\boldsymbol{\chi}_k^{-}(i) - \hat{\mathbf{x}}_k^{-}\right] \left[\boldsymbol{\gamma}_k(i) - \hat{\mathbf{y}}_k^{-}\right]^T \tag{4.2.18c}$$

The Kalman gain is given by Equation 4.2.19.

$$K_k = P_k^{xy} \left(P_k^{vv}\right)^{-1} \tag{4.2.19}$$

The updated predictions for the mean state and covariance are given by Equation 4.2.20.

$$\hat{\mathbf{x}}_k^{+} = \hat{\mathbf{x}}_k^{-} + K_k \left[\tilde{\mathbf{y}}_k - \hat{\mathbf{y}}_k^{-}\right] \tag{4.2.20a}$$

$$P_k^{+} = P_k^{-} - K_k P_k^{vv} K_k^T \tag{4.2.20b}$$

Additionally, the updated error quaternion $\boldsymbol{\delta}\hat{\mathbf{q}}_k^{+}$ is calculated using Equation 4.2.21.

$$\delta q_1 = \frac{-a\|\boldsymbol{\delta p}\|^2 + f\sqrt{f^2 + (1-a^2)\|\boldsymbol{\delta p}\|^2}}{f^2 + \|\boldsymbol{\delta p}\|^2} \tag{4.2.21a}$$

$$\boldsymbol{\delta\varrho} = f^{-1}\left(a + \delta q_1\right)\boldsymbol{\delta p} \tag{4.2.21b}$$

$$\boldsymbol{\delta}\mathbf{q}_k^{+} = \begin{bmatrix} \delta q_1 \\ \boldsymbol{\delta\varrho} \end{bmatrix} \tag{4.2.21c}$$

The updated quaternion estimate is then calculated using Equation 4.2.22. After this the error GRPs are set to 0 in the state vector, and the filter process repeats for the next time step.

$$\hat{\mathbf{q}}_k^{+} = \boldsymbol{\delta}\mathbf{q}_k^{+} \otimes \hat{\mathbf{q}}_k^{-}(0) \tag{4.2.22}$$

4.2.6. Algorithm summary

The full algorithm of the attitude and orbit estimation using light curve inversion is quite complex with a lot of equations. To explain the algorithm more clearly and in one overview the steps of the algorithm are summarized in Table 4.2.1 below.

Table 4.2.1: Summary of steps in attitude and orbit estimation through light curve inversion algorithm.

Step	Description
0	Initialize $\mathbf{dp}_0 = [0, 0, 0]$.
Prediction	
1	Input initial state $\mathbf{x}_0 = [\mathbf{r}_0, \mathbf{v}_0, \mathbf{dp}_0, \boldsymbol{\omega}_0]$, quaternion $\mathbf{q}_0$, covariance $\mathbf{P}_0$, and process noise covariance $\mathbf{Q}$.
2	Compute sigma points $\boldsymbol{\chi}$ from $\mathbf{x}$ and $\mathbf{P}$.
3	Compute error quaternion sigma points $\boldsymbol{\chi}_{\mathbf{dq}}$ from $\mathbf{dp}$ sigma points, then compute quaternion sigma points $\boldsymbol{\chi}_{\mathbf{q}}$.
4	Construct sigma points state vector for propagation: $\boldsymbol{\chi}^{\text{prop}} = [\boldsymbol{\chi}_{\mathbf{r}}, \boldsymbol{\chi}_{\mathbf{v}}, \boldsymbol{\chi}_{\mathbf{q}}, \boldsymbol{\chi}_{\boldsymbol{\omega}}]$.
5	Propagate sigma points.
6	Compute propagated error quaternions. Set the first propagated error quaternion to $[1, 0, 0, 0]$.
7	Compute propagated error GRP sigma points $\boldsymbol{\chi}_{\mathbf{dp}}$ from propagated error quaternions. Construct the full propagated sigma points: $\boldsymbol{\chi}^{\text{prop}} = [\boldsymbol{\chi}_{\mathbf{r}}^{\text{prop}}, \boldsymbol{\chi}_{\mathbf{v}}^{\text{prop}}, \boldsymbol{\chi}_{\mathbf{dp}}^{\text{prop}}, \boldsymbol{\chi}_{\boldsymbol{\omega}}^{\text{prop}}]$.
8	Compute predicted mean $\mathbf{x}_{\text{pred}}$ and error covariance $\mathbf{P}_{\text{pred}}$. Set the mean propagated quaternion as $\mathbf{q}_{\text{pred}}$.
9	**If no measurement is available:** Reset error GRPs $\mathbf{dp}$ in $\mathbf{x}_{\text{pred}}$ to $[0, 0, 0]$. Set inputs: $\mathbf{x} = \mathbf{x}_{\text{pred}}$, $\mathbf{P} = \mathbf{P}_{\text{pred}}$, $\mathbf{q} = \mathbf{q}_{\text{pred}}$. Repeat from step 2.
Correction	
10	**If a measurement is available:** Compute predicted measurement for all propagated sigma points: $\boldsymbol{\chi}_{\mathbf{z}}^{\text{pred}} = \mathbf{h}[\boldsymbol{\chi}^{\text{prop}}, \boldsymbol{\chi}_{\mathbf{q}}^{\text{prop}}]$.
11	Compute weighted predicted measurement $\mathbf{z}_{\text{pred}}$.
12	Compute innovation, covariance, and gain matrices: $\mathbf{S}$, $\mathbf{P}_{xz}$, and $\mathbf{K}$.
13	Compute corrected state mean $\mathbf{x}_{\text{cor}}$ and error covariance $\mathbf{P}_{\text{cor}}$.
14	Convert $\mathbf{dp}$ values in $\mathbf{x}_{\text{cor}}$ to $\mathbf{q}_{\text{cor}}$ using the mean propagated quaternion.
15	Reset $\mathbf{dp}$ values in $\mathbf{x}_{\text{cor}}$ to $[0, 0, 0]$.
16	Set inputs: $\mathbf{x} = \mathbf{x}_{\text{cor}}$, $\mathbf{P} = \mathbf{P}_{\text{cor}}$, $\mathbf{q} = \mathbf{q}_{\text{cor}}$.
17	Repeat from step 2.

4.2.7. Verification

In order to verify the implementation of the attitude estimation through light curve inversion works correctly some verification is required. The verification is based on the paper by Wetterer et al. [4], where they attempt to estimate the attitude of an Atlas-Centaur II upper-stage rocket body. The verification is divided in three steps. In the first step the implementation of the UKF is verified. Then an attempt is made to replicate the results from Wetterer et al., and lastly the simulation time is extended to investigate long term behaviour of the algorithm.

Initialisation

The Keplerian orbit element of the true rocket body orbit and the initial guess used in the estimation are given in Table 4.2.2. The elements were converted from TLEs retrieved from *space-track.org* for March 24$^{\text{th}}$ 2003 14:04:00 UTC, which is the start epoch of the estimation. The orbit is a geosynchronous transfer orbit. The initial guess orbit is generated by adding 1 km to the true orbit positions and [5, -5, 2.5] m/s to the orbital velocities. The initial attitudes in Euler angles and rotational velocities for the truth and initial guess are given in Table 4.2.3.

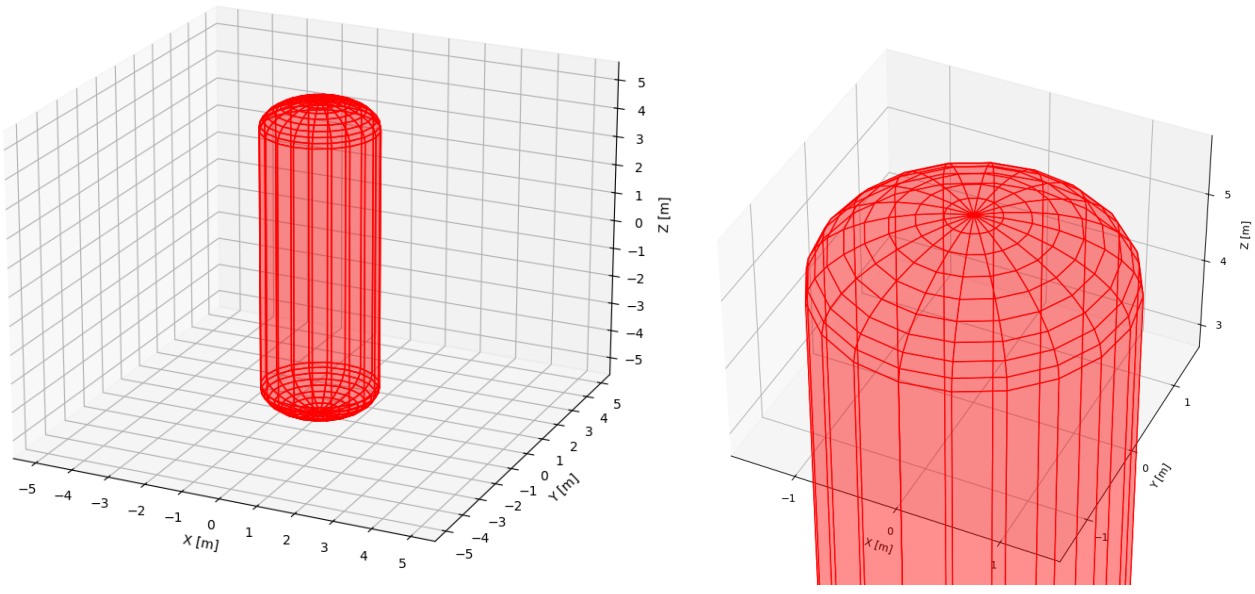
Table 4.2.2: Keplerian orbit elements for Atlas-Centaur II rocket body.

	a [km]	**e [-]**	**i [°]**	**ω [°]**	**Ω [°]**	**ν [°]**
Truth	7,305.607	0.0635484	30.38	328.94	73.97	68.23
Guess	7,304.597	0.0627248	30.42	329.04	74.00	68.10

Table 4.2.3: Initial attitude and rotational velocities for truth and guess of Atlas-Centaur II rocket body.

	θ_1 [rad]	**θ_2 [rad]**	**θ_3 [rad]**	**ω_1 [rad/s]**	**ω_2 [rad/s]**	**ω_3 [rad/s]**
Truth	0.2	-0.3	1.4	0.0	0.0	0.252
Guess	0.6	-0.9	1.0	0.001	0.001	0.253

The rocket body is modelled as a cylinder with two flattened hemispherical end caps. The cylinder has a height and radius of 9 m and 1.5 m respectively, and the end caps have a height of 1.0 m and the same radius as the cylinder, giving the rocket body a total height of 11 m. The shape is created by dividing the body in facets, with 20 facets for the cylinder around its diameter and 180 for each of the end caps. A 3D visualisation of the rocket body shape model is given in Figure 4.2.2.



(a) Full 3D shape model of the rocket body.

(b) Zoomed in rocket body end cap. Each cap consists of 180 facets.

Figure 4.2.2: Shape model of the Atlas-Centaur II upper-stage rocket body. Full model made up of 380 facets.

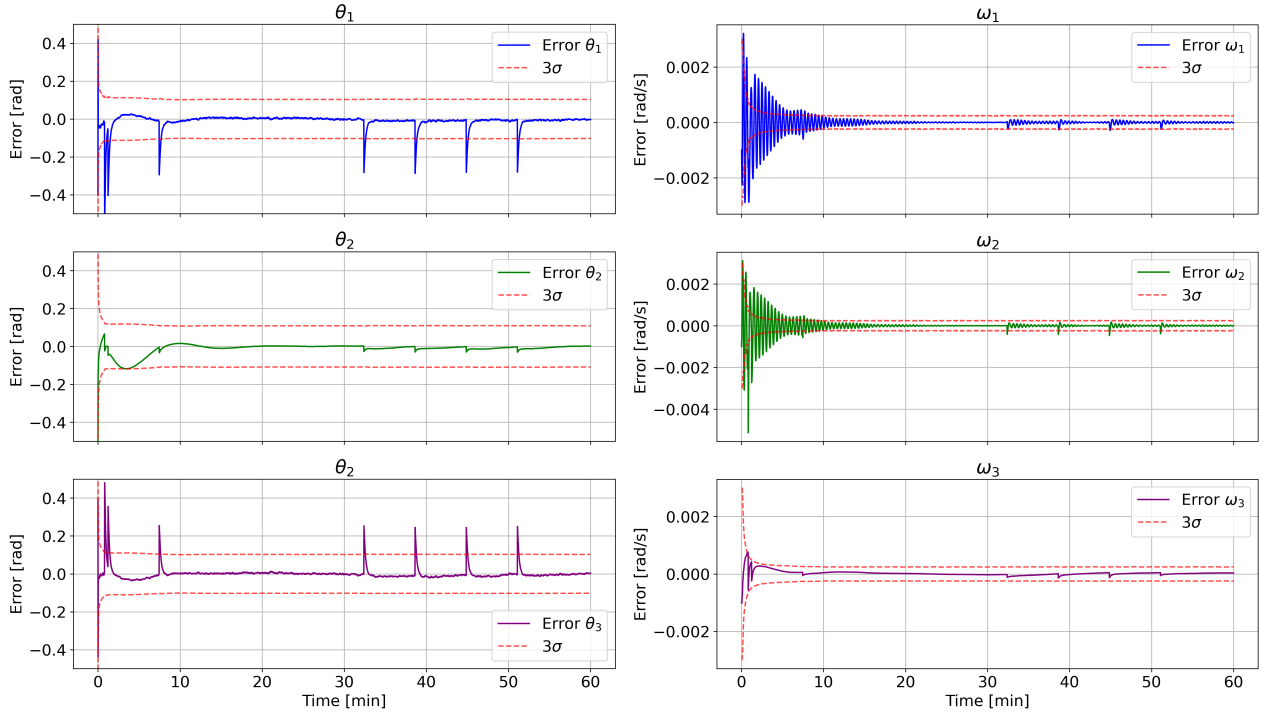
The initial covariance matrix $\boldsymbol{P}_0$ and the process noise matrix $\boldsymbol{Q}$ for the estimation are given in Equation 4.2.23.

$$\boldsymbol{P}_0 = \text{diag}\left[(1000\text{ m})^2 * I_3 \quad (10\text{ m/s})^2 * I_3 \quad 0.2^2 * I_3 \quad (10^{-3}\text{ rad/s})^2 * I_3\right] \tag{4.2.23a}$$

$$\boldsymbol{Q} = \text{diag}\left[(100\text{ m})^2 * I_3 \quad (0.1\text{ m/s})^2 * I_3 \quad (2 * 10^{-4})^2 * I_3 \quad (10^{-12}\text{ rad/s})^2 * I_2 \quad (10^{-5}\text{ rad/s})^2 * I_1\right] \tag{4.2.23b}$$

UKF implementation

To verify the implementation of the UKF an estimation is run where the filter is fed direct measurements of Euler angle attitude and orbital position, instead of light curve and angles measurements. This way the behaviour of the UKF can be checked independently from the combined implementation with the light curve inversion. The measurements are generated by taking the state values of the truth at each time step and adding zero-mean Gaussian noise, for the orbital position with a standard deviation of 1 km, and for the Euler angle attitude with a standard deviation of 1°. The estimation is run for 60 minutes, with a time step of 1 s, with measurements available at each time step. In Figure 4.2.3 the error plots for the attitude and angular velocities are shown for the verification of the UKF.



(a) Euler angle errors with 3σ uncertainty bounds.

(b) Angular velocity errors with 3σ uncertainty bounds.

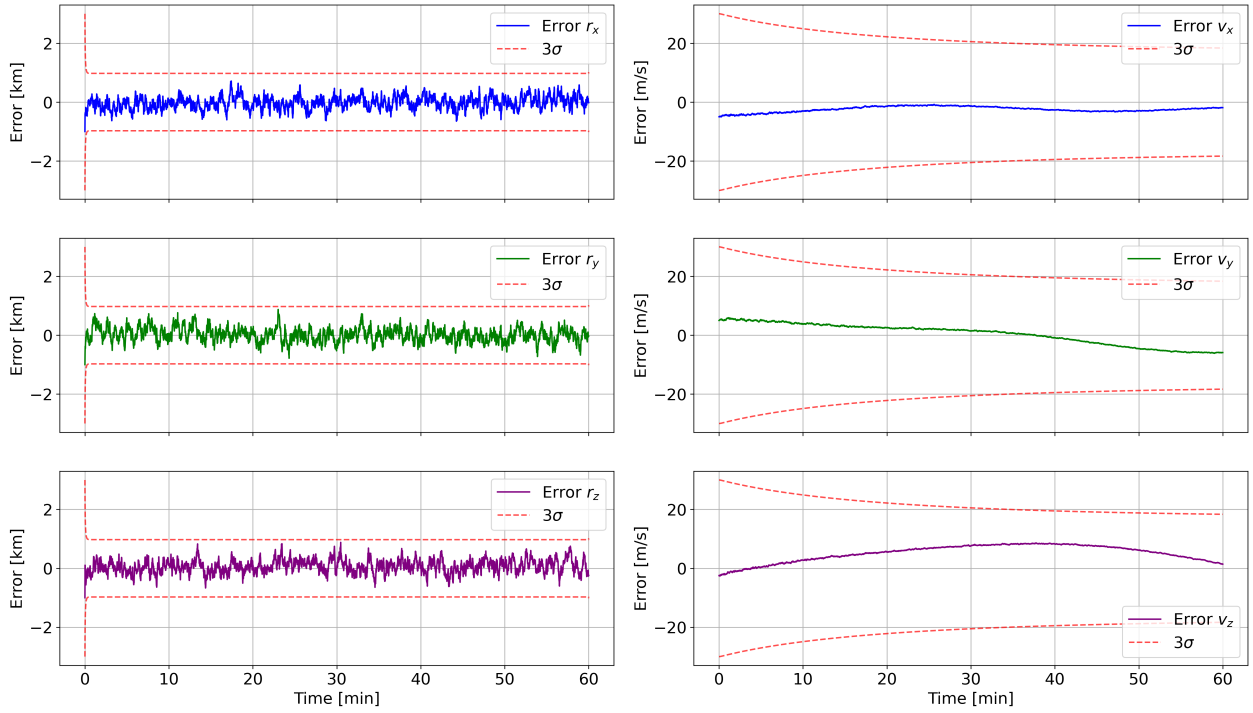
Figure 4.2.3: Attitude and omega error for verification of UKF with direct position and Euler angle measurements.

An important note that should be mentioned concerns the way the uncertainty bounds for the Euler angle attitude error plots are obtained. In the UKF the attitude states that are propagated are the error GRPs. This means the attitude covariance values that are calculated and output by the UKF are also those of the error GRPs. However, Crassidis and Markley state [22]: "for small errors, the attitude part of the covariance is closely related to the attitude-estimation errors." In their paper Wetterer and Jah state [4]: "The Euler angles at each step are derived from the quaternion with the uncertainty derived by equating the error GRP portion of the state covariance to the square of the uncertainties in the Euler angles." This means the covariance bounds for the Euler angle errors can be converted from the covariance values of the error GRPs that are output by the UKF, using Equation 4.2.24.

$$\sigma_{\boldsymbol{\theta}}^2 = \sigma_{\mathbf{dp}} \tag{4.2.24}$$

It can be seen in Figure 4.2.3 that for both the attitude and angular velocities the the UKF converges quickly to low errors. The uncertainty bounds for both parameters also quickly decrease. Both are expected behaviour for the UKF. After 30 minutes there are some error spikes that can be seen in the attitude plots, and for the first and third Euler angle θ_1 and θ_3 these spikes exceed the uncertainty bounds. These spikes are caused by the errors in the angular velocity increasing slightly at the same times, as can be seen in Figure 4.2.3b, which directly leads to larger attitude errors. These slight increases in rotational velocity errors are most likely caused by the process noise that is added during the estimation. Process noise is added to ensure the estimation does not get stuck once it has converged, which can cause slight changes in the estimation values throughout the estimation, which is seen here in the rotational velocity errors increasing slightly. These errors quickly converge back towards zero however, conforming to expected behaviour of a filtering algorithm like the implemented UKF.

Figure 4.2.4 shows the error plots for the orbital positions and velocities. It can be seen that both errors are nicely constrained between the uncertainty bounds. The errors for the orbital positions are noisily spread around zero error. This matches the expected results of the UKF perfectly, as these states are directly measured with random noise, and the error is affected by this noise. In the orbital velocity errors the effect of noise can also be seen, but it is much less prominent compared to the overall behaviour of the error, which acts on a larger scale than the noise.



(a) Orbital position errors with 3σ uncertainty bounds.

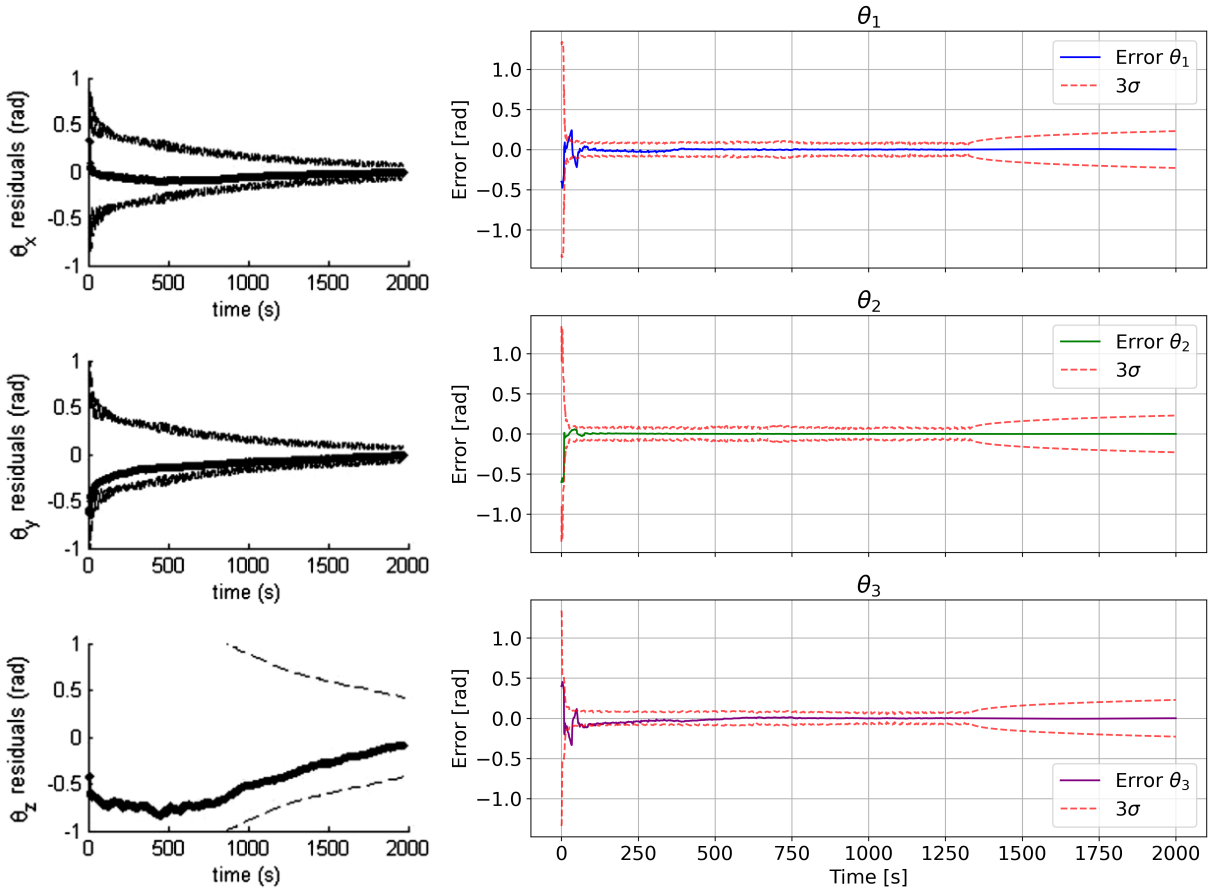
(b) Orbital velocity errors with 3σ uncertainty bounds.

Figure 4.2.4: Position and velocity error for verification of UKF with direct position and Euler angle measurements.

Overall the results obtained from the UKF where the filter uses direct measurements of the Euler angle attitudes and orbital positions match the behaviour that is expected of a filter implementation like the UKF. It is therefore concluded that the implemented UKF for attitude and orbit estimation works as expected. Next it will be investigated whether the algorithm works well in combination with light curve inversion.

Paper result replication

Next an attempt is made to replicate the results obtained by Wetterer et al. [4] for the attitude estimation of the Atlas-Centaur II rocket body through light curve inversion. For this case angles and range measurements of the orbital positions and apparent brightness measurements are generated by simulating observations from a ground station at the Maui Space Surveillance Complex, located at 20.708°N latitude, 156.257°W longitude and 3.075 km altitude. Measurements are made at the same time step as the estimation of 1 s. Zero-mean Gaussian noise is added to each measurement, with a standard deviation of $1*10^{-5}$ rad for the azimuth and elevation observations, 1 km for the range and 0.01 for the brightness observations. In the paper the estimation is run for 2000 s, producing the Euler angle and rotational velocity error plots in Figure 4.2.5a and 4.2.6a, respectively. Running the implemented attitude estimation algorithm results in the error plots in Figure 4.2.5b and 4.2.6.



(a) Euler angle errors from Wetterer et al. [4].

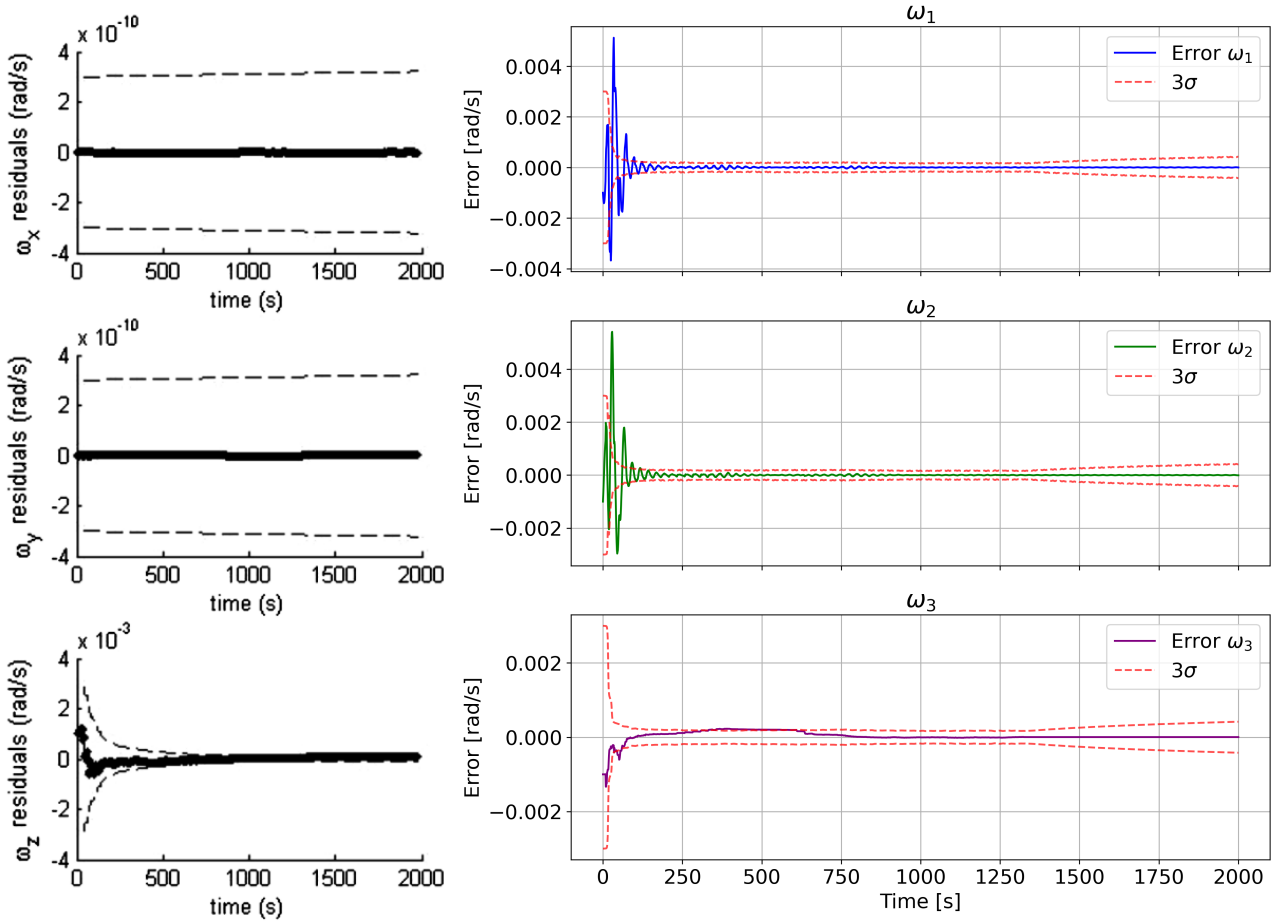
(b) Euler angle error from result replication.

Figure 4.2.5: Euler angle errors from paper and result replication.

Looking at the Euler angle error plots, it is clear the plots are not identical. The errors for the replication attempt converge much quicker than those in the paper. The other difference is that at the end of the replication plot the covariance bounds increase again. This happens because the visible pass of the simulated rocket body does not last the full 2000 seconds. For the paper result the covariance bounds do decrease for the full estimation, indicating that there the measurements lasted until the end of the propagation. This difference is caused by the fact that the orbital state parameters in the paper were not described in full detail. For the replication the orbit data was therefore retrieved from TLE data. This means the orbits used in the paper and the replication are most likely not the same, leading to the difference in results.

However, it can be seen that the implemented algorithm is still able to estimate the attitude with low errors, with the errors remaining low after the visible pass of the rocket body is over.

The rotational velocity error plots in Figure 4.2.6b also do not look the same as the results from the paper. Initially the errors for the replication are much larger than those in the paper. However, after about 250s the errors converge to low errors, remaining like this for the rest of the estimation. This shows that the algorithm is also able to estimate the rotational velocity states of the rocket body correctly.



(a) Rotational velocity errors from Wetterer et al. [4].

(b) Rotational velocity errors from result replication.

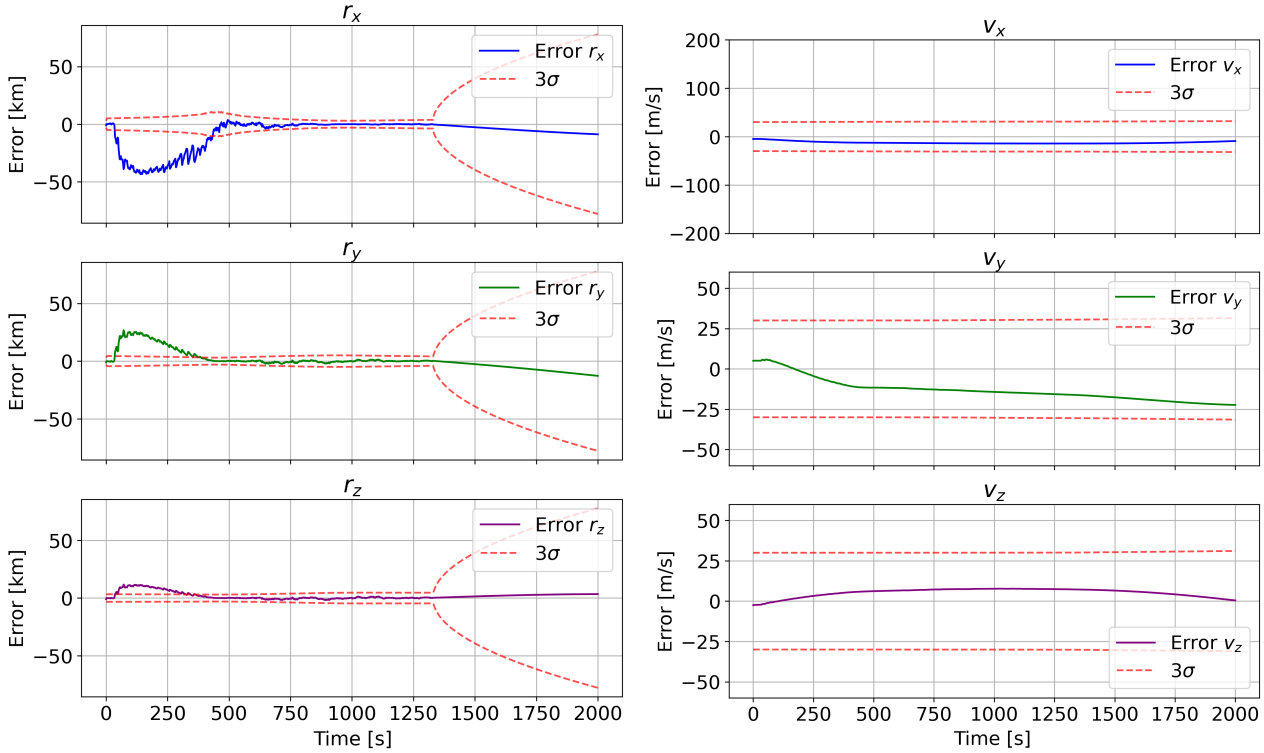
Figure 4.2.6: Rotational velocity errors from paper and result replication.

To conclude this part of the verification, while the results do not look identical to the results from Wetterer, overall they show that the implemented version of the attitude estimation from light curve algorithm works as expected, and is able to estimate the attitude states with low error values.

In the paper the only states that are estimated are the attitude and rotational velocities. However, in this thesis the orbit states are also included in the estimation. The orbit estimation results for this case will therefore still be examined, to see if the implementation with the azimuth, elevation and range angles work correctly.

Orbit estimation

The orbital position and velocity errors for the verification case are shown in Figure 4.2.7. Initially the orbit errors are quite large, and stray outside of the covariance bounds. However, around 500s the estimation converges to low position errors inside the covariance bounds. When the visible pass of the rocket is over the covariance bounds increase again, just like was seen in the attitude error plots. This is exactly what should happen, because there are no measurements to correct the predictions in the UKF, so the uncertainties increase. Likewise, the errors also start to increase again, as the predictions are not corrected with measurements, so the orbit estimation starts to drift from the true orbit.



(a) Orbital position errors.

(b) Orbital velocity errors.

Figure 4.2.7: Orbital state errors for replication case.

The orbital velocity error plots looks different. The covariance bounds appear to say constant throughout the entire estimation. The errors also show different behaviour from the position errors. These differences are caused by the fact that the velocity errors are relatively a lot smaller than the position errors. The position errors are on the order of magnitude of km, and initially even tens of km, while the velocity errors stay between -25 and 25 m/s. Because the UKF is able to estimate the velocity to such accuracy, the uncertainties only change in such small ways that they appear constant.

Conclusion

Overall it can be concluded that the implemented algorithm for attitude and orbit estimation through light curve inversion works as expected, producing orbit and attitude state estimates with low errors. Even though the results do not look exactly like those produced by Wetterer et al. [4] with their implementation of the algorithm, the implemented algorithm produces similar results, finding the correct attitude and rotational velocities for the rocket body through light curve measurements. Additionally, it was shown that the implemented algorithm is also able to correctly estimate the orbital states using azimuth, elevation and range measurements. The implement algorithm forms the basis for the methods that are described in the next sections of the methodology chapter, starting with a well-developed method called Multiple-Model Adaptive Estimation.

4.3. Multiple-Model Adaptive Estimation

Estimation is relatively straightforward when the physical model underlying the observations is well known. In the case of space debris it is however not always known what shape an objects has, next to its orbit and attitude. This makes it very difficult to estimate especially the latter, as brightness observations are highly dependent on shape and attitude. One approach to tackle this issue is to run multiple UKFs in parallel, each with a different underlying physical model like object shape, and compare which model outputs correspond the best with the observations. This is the basis for the Multiple-Model Adaptive Estimation (MMAE) method.

4.3.1. Estimation weights

The method of MMAE has been applied to object characterisation through light curve inversion by Linares et al. [12][23] in multiple contexts. A visualisation of the MMAE process is given in Figure 4.3.1. In essence there is a model bank with M different models that could be the correct physical model driving the real system. Each of these models has a different hypothesis, giving the finite set of hypotheses $\{\mathbf{p}^{(\ell)};\ \ell = 1, ..., M\}$.

The goal of the estimation process is to determine the conditional PDF of each hypothesis $\mathbf{p}^{(\ell)}$ given all the measurements $\tilde{\mathbf{Y}}_k = \{\tilde{\mathbf{y}}_0, \tilde{\mathbf{y}}_1, ..., \tilde{\mathbf{y}}_k\}$. Bayes' Theorem can be applied to find this conditional probability $p\left(\mathbf{p}^{(\ell)} \mid \tilde{\mathbf{Y}}_k\right)$ as given by Equation 4.3.1.

$$p\left(\mathbf{p}^{(\ell)} \mid \tilde{\mathbf{Y}}_k\right) = \frac{p\left(\tilde{\mathbf{Y}}_k \mid \mathbf{p}^{(\ell)}\right) p\left(\mathbf{p}^{(\ell)}\right)}{\sum_{j=1}^{M} p\left(\tilde{\mathbf{Y}}_k \mid \mathbf{p}^{(j)}\right) p\left(\mathbf{p}^{(j)}\right)} \tag{4.3.1}$$

The posterior probabilities can be calculated using Equation 4.3.2. Here $p\left(\tilde{\mathbf{y}}_k, \mathbf{p}^{(\ell)} \mid \tilde{\mathbf{Y}}_{k-1}\right)$ is the joint probability of observing the current measurement $\tilde{\mathbf{y}}_k$ and that hypothesis $\mathbf{p}^{(\ell)}$ is true, given all previous measurements $\tilde{\mathbf{Y}}_{k-1}$.

$$p\left(\mathbf{p}^{(\ell)} \mid \tilde{\mathbf{Y}}_k\right) = \frac{p\left(\tilde{\mathbf{y}}_k, \mathbf{p}^{(\ell)} \mid \tilde{\mathbf{Y}}_{k-1}\right)}{p\left(\tilde{\mathbf{y}}_k \mid \tilde{\mathbf{Y}}_{k-1}\right)} = \frac{p\left(\tilde{\mathbf{y}}_k \mid \hat{\mathbf{x}}_k^{-(\ell)}\right) p\left(\mathbf{p}^{(\ell)} \mid \tilde{\mathbf{Y}}_{k-1}\right)}{\sum_{j=1}^{M}\left[p\left(\tilde{\mathbf{y}}_k \mid \hat{\mathbf{x}}_k^{-(j)}\right) p\left(\mathbf{p}^{(j)} \mid \tilde{\mathbf{Y}}_{k-1}\right)\right]} \tag{4.3.2}$$

The probabilities of observing $\tilde{\mathbf{y}}_k$ given the state estimate for a certain hypothesis $\hat{\mathbf{x}}_k^{(\ell)}$ are given by Equation 4.3.3.

$$p\left(\tilde{\mathbf{y}}_k \mid \hat{\mathbf{x}}_k^{-(\ell)}\right) = \frac{1}{\det\left(2\pi S_k^{(\ell)}\right)^{1/2}} \exp\left\{-\frac{1}{2}\mathbf{e}_k^{(\ell)T} S_k^{(\ell)-1} \mathbf{e}_k^{(\ell)}\right\} \tag{4.3.3}$$

Here the measurement residual for the ℓth hypothesis $\mathbf{e}_k^{(\ell)}$ is given by Equation 4.3.4 and the corresponding residual covariance matrix $S_k^{(\ell)}$ is equal to the innovation covariance matrix P_k^{vv} given in Equation 4.2.18b.

$$\mathbf{e}_k^{(\ell)} = \tilde{\mathbf{y}}_k - \hat{\mathbf{y}}_k^{-(\ell)} \tag{4.3.4}$$

Equation 4.3.2 can be rewritten as a recursion formula to define MMAE weights $w_k^{(\ell)}$ to give Equation 4.3.5, where $w_k^{(\ell)} \equiv p\left(\mathbf{p}^{(\ell)} \mid \tilde{\mathbf{Y}}_k\right)$. Only the current time likelihood function is needed to update the weights. The second part of the equation is necessary to normalise the weights.

$$w_k^{(\ell)} = w_{k-1}^{(\ell)} p\left(\tilde{\mathbf{y}}_k \mid \hat{\mathbf{x}}_k^{(\ell)}\right), \quad w_k^{(\ell)} \leftarrow \frac{w_k^{(\ell)}}{\sum_{j=1}^{M} w_k^{(j)}} \tag{4.3.5}$$

The weights correspond to how well each hypothesis fits the observations, relative to the others. It can be seen in Equation 4.3.3 that a hypothesis with lower residuals $\mathbf{e}_k^{(\ell)}$ will have probabilities that increase with time. This means that hypotheses that fit the observations better are favoured in the MMAE algorithm. In this equation it can also be seen that hypotheses with small values for the determinant of $S_K^{(\ell)}$ will have increasing probabilities. This means hypotheses with smaller variances are also favoured, assuming all filters have the same measurement noise covariance R_k. The MMAE algorithm will thus tend to select the minimum variance hypothesis from the full hypotheses set.

4.3.2. Fusion of estimates

The MMAE method has an additional capability next to assigning conditional probabilities to hypotheses. Depending on the type of parameters being estimated the MMAE algorithm can output a fused state estimate and error covariance, usually being a weighted average of the different filters post-update state estimates. This so-called conditional mean typically is calculated as a weighted sum of all of the filters estimates and is given by Equation 4.3.6.

$$\hat{\mathbf{x}}_k^+ = \sum_{\ell=1}^{M} w_k^{(\ell)} \hat{\mathbf{x}}_k^{+(\ell)} \tag{4.3.6}$$

However, since in this case quaternions are part of the state vector being estimated, it is not possible to sum up the different quaternion estimates, as this would give a quaternion not abiding the unit norm constraint. A different approach is required to determine the conditional mean and covariance for the quaternion estimates. Luckily, Linares et al.[12][23] have already implemented this.

For their approach a reference quaternion attitude $\overline{\mathbf{q}}_k$ needs to be defined, from which all the different quaternion estimates are referenced. In this case this reference quaternion is the weighted average quaternion from the set of all UKF quaternion estimates. This average quaternion is given by the normalized eigenvector corresponding to the maximum eigenvalue of matrix $\mathcal{M}$, where $\mathcal{M}$ is calculated using Equation 4.3.7.

$$\mathcal{M} = -\sum_{\ell=1}^{M} w_k^{(\ell)} \Xi\left(\hat{\mathbf{q}}_k^{+(\ell)}\right) P_{\alpha\alpha}^{+(\ell)-1} \Xi^T\left(\hat{\mathbf{q}}_k^{+(\ell)}\right) \tag{4.3.7}$$

Here $\Xi(\mathbf{q})$ is the function given by Equation 4.1.3b. The matrix $P_{\alpha\alpha}^{+(\ell)}$ is the 3 x 3 part of the covariance matrix corresponding to the attitude estimates for each hypothesis.

With the reference attitude calculated, the attitude correction corresponding to the estimate of the ℓth filter is given by Equation 4.3.8. This quaternion correction is then converted to a GRP correction $\boldsymbol{\delta}\mathbf{p}_k^{+(\ell)}$ using Equation 4.2.13.

$$\boldsymbol{\delta}\hat{\mathbf{q}}_k^{+(\ell)} = \hat{\mathbf{q}}_k^{-(\ell)} \otimes \hat{\mathbf{q}}_k^{-1} \tag{4.3.8}$$

With $\overline{\mathbf{x}}_k^{+(\ell)} \equiv \begin{bmatrix} \hat{\mathbf{r}}_k^{I+(\ell)T} & \hat{\mathbf{v}}_k^{I+(\ell)T} & \hat{\boldsymbol{\delta p}}_k^{I+(\ell)T} & \hat{\boldsymbol{\omega}}_k^{I+(\ell)T} \end{bmatrix}$ the mean estimate is calculated using Equation 4.3.9.

$$\overline{\mathbf{x}}_k^+ = \sum_{\ell=1}^{M} w_k^{(\ell)} \overline{\mathbf{x}}_k^{+(\ell)} \tag{4.3.9}$$

Finally, the conditional mean is given by Equation 4.3.10. Here the mean quaternion is calculated as $\hat{\mathbf{q}}_k^+ = \boldsymbol{\delta}\hat{\mathbf{q}}_k^+ \otimes \overline{\mathbf{q}}_k^+$ and $\boldsymbol{\delta}\hat{\mathbf{q}}_k^+$ is the quaternion formed from $\boldsymbol{\delta}\hat{\mathbf{p}}_k^+$ from $\overline{\mathbf{x}}_k^+$

$$\hat{\mathbf{x}}_k^+ = \begin{bmatrix} \hat{\mathbf{r}}_k^{I+} \\ \hat{\mathbf{v}}_k^{I+} \\ \hat{\mathbf{q}}_k^+ \\ \hat{\boldsymbol{\omega}}_k^+ \end{bmatrix} \tag{4.3.10}$$

The conditional covariance can then be calculated using Equation 4.3.11.

$$P_k^+ = \sum_{\ell=1}^{M} w_k^{(\ell)} \left(P_k^{+(\ell)} + \left[\overline{\mathbf{x}}_k^{+(\ell)} - \overline{\mathbf{x}}_k^+\right] \left[\overline{\mathbf{x}}_k^{+(\ell)} - \overline{\mathbf{x}}_k^+\right]^T \right) \tag{4.3.11}$$

4.3.3. Flowchart

For better visualisation and understanding of the MMAE process the flowchart in Figure 4.3.1 was created.

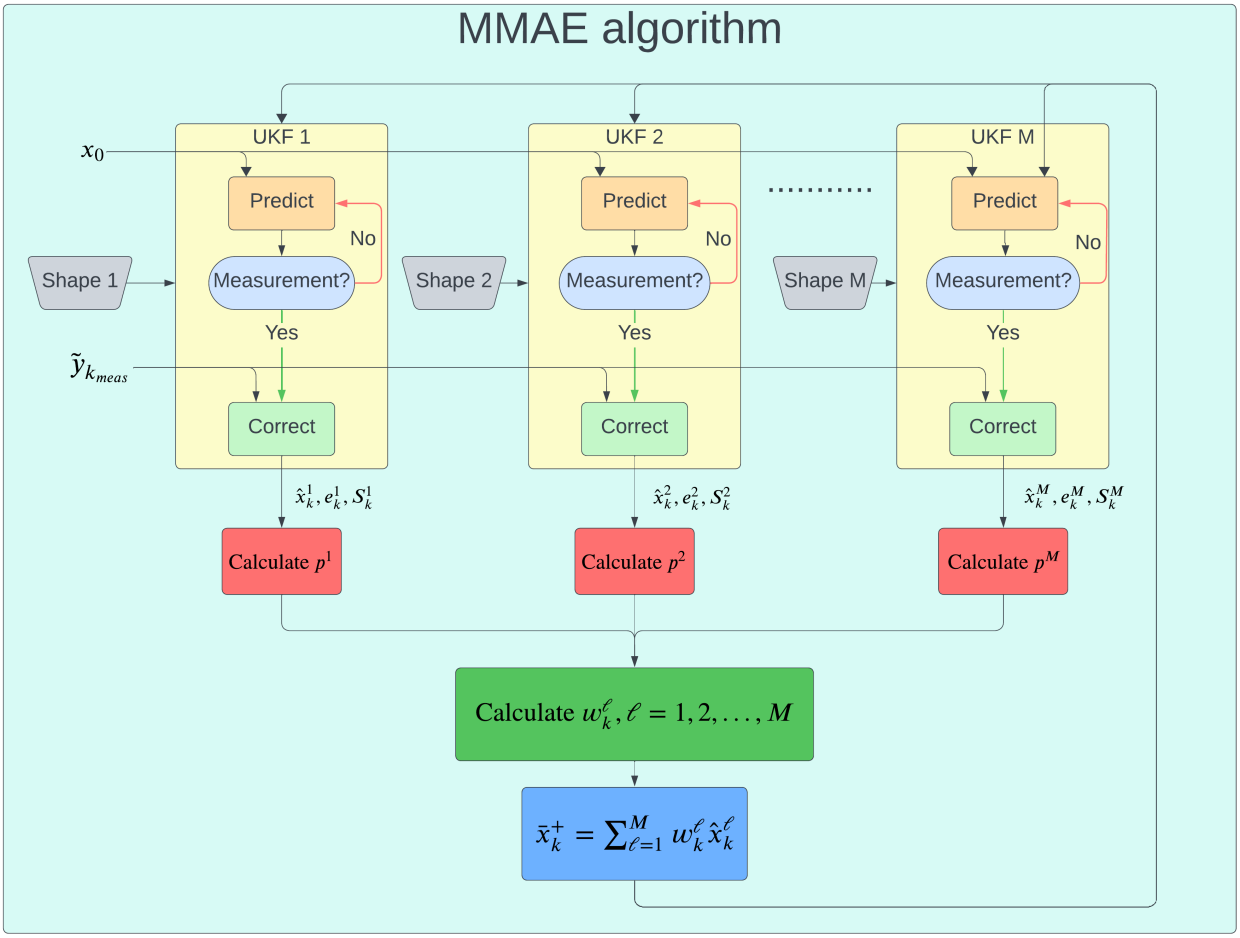


Figure 4.3.1: Flowchart of MMAE algorithm for M shape models.

For M different shape models there are M UKFs that are run in parallel. Given an initial state $\mathbf{x}_0$, the UKFs perform a prediction step at time k. If there is no measurement at this time, the next prediction at time $k+1$ is made, with the prediction from t_k as starting value. When there is a measurement at time $t_{k_{meas}}$ the UKF correction step is executed. With the corrected estimates, measurement residuals and S matrices the probabilities $p^{(\ell)}$ are calculated. With these the weights are updated and the conditional mean is calculated. This is then input into each UKF as the new initial state, and the process repeats.

4.3.4. Verification

The verification of the MMAE algorithm is done in two parts. For the first part only a few different shape models are included in the model bank in order to test the capabilities of the algorithm. In the second part it is attempted to replicate the results obtained by Linares et al. [23] with their implementation of the MMAE algorithm.

Initialisation

For both cases the initial states and BRDF parameters are taken from the paper by Linares et al. [23]. The simulation epoch is set to March 15$^{\text{th}}$ 2010 at 04:00:00 UTC. The Keplerian orbital elements for the true orbit and the orbit guess that is input in each UKF are given in Table 4.3.1. The initial attitude and rotational velocities for the true case and the guess for each UKF are given in Table 4.3.2.

Table 4.3.1: Keplerian orbit elements for true orbit and orbit guess.

	a [km]	**e [-]**	**i [°]**	ω [°]	Ω [°]	ν [°]
True orbit	42,364.17	$2.429 * 10^{-4}$	30	0.0	0.0	91.121
Orbit guess	42,364.148255	$2.429 * 10^{-4}$	30.0083	-1.172	0.0	92.165

Table 4.3.2: Initial attitude and rotational velocities for truth and guess.

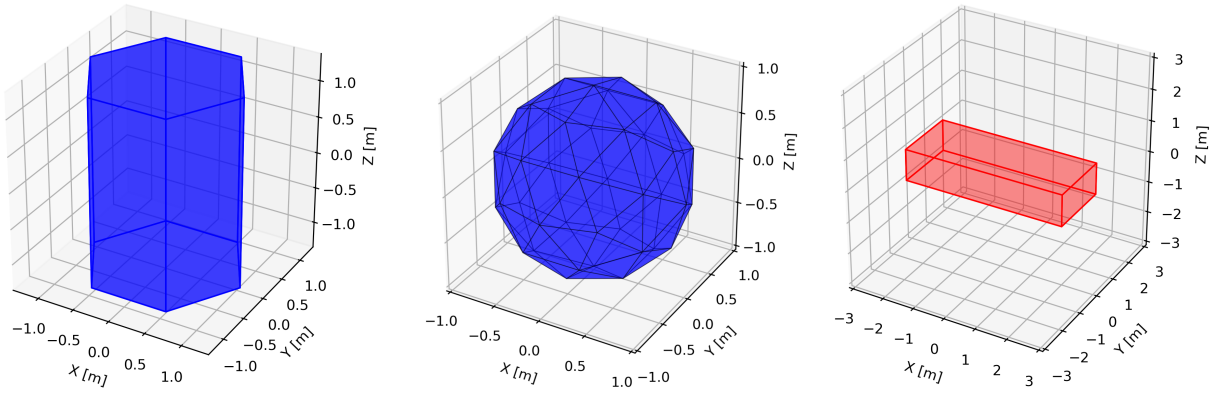
	q_0 [-]	q_1 [-]	q_2 [-]	q_3[-]	ω_1 [°/hr]	ω_2 [°/hr]	ω_3 [°/hr]
Truth	0.7041	0.0199	0.0896	0.7041	206.26	103.13	540.41
Guess	0.7500	0.0712	0.0947	0.6508	220.26	117.13	554.41

All shape models are made up from facets. For all the facets the apparent brightness is calculated using the Ashikmin-Shirley BRDF model discussed in Subsection 4.1.2. Each facet is given the same reflectivity coefficients $R_{\text{spec}} = 0.7$, $R_{\text{diff}} = 0.3$. The parameters n_u and n_v are set to be equal to 1000 for all facets. The brightness and orbit measurements are simulated with a ground station located at 20.71°N latitude, 156.26°W longitude and 3058.6 m altitude. Zero-mean Gaussian noise is added to the measurements, with a standard deviation of 1 arcsecond on the azimuth and elevation observations, 100 m on the range and 0.1 on the brightness observations. Observations are made every 5 s, while the estimation runs at a time step of 1 s.

The initial covariances in $\mathbf{P}$ are set to 300 km and 3 km/s for the orbital position and velocity respectively, 0.2 for the error GRPs and 72 °/hr for the rotational velocities. The process noise covariances in $\mathbf{Q}$ are all set to zero.

Simple test case

For the simple case three shape models are included: a hexagonal prism, a sphere, and rectangular cuboid. The hexagonal prism is defined by two dimensions, the length of the hexagon sides s and height h. The sphere is defined by the radius r and number of facets n_f. The rectangular cuboid is defined by the dimensions of the sides x, y and z. The shapes are visualised in Figure 4.3.2. These shapes were selected because they are very distinct from each other, so it is expected that the MMAE algorithm is able to correctly identify the true model.



(a) Hexagonal prism. $s = 1.0466$ m, $h = 2.6199$ m

(b) Sphere. $r = 1.0$ m, $n_f = 80$

(c) Rectangular cuboid. $x = 5$ m, $y = 2$ m, $z = 1$ m

Figure 4.3.2: Shape models included in simple test case.

The MMAE algorithm is run three times, each time one of the three models is selected as the true model from which measurements are generated. The MMAE weights over time for all three cases are shown in Figure 4.3.3. For the cases where the hexagonal prism (Figure 4.3.3a) and the sphere (Figure 4.3.3b) are the true shape model the MMAE algorithm is able to quickly find the correct true model. However, in the case where the cuboid is the true model (Figure 4.3.3c) the algorithm incorrectly determines that the hexagonal prism is the true model.

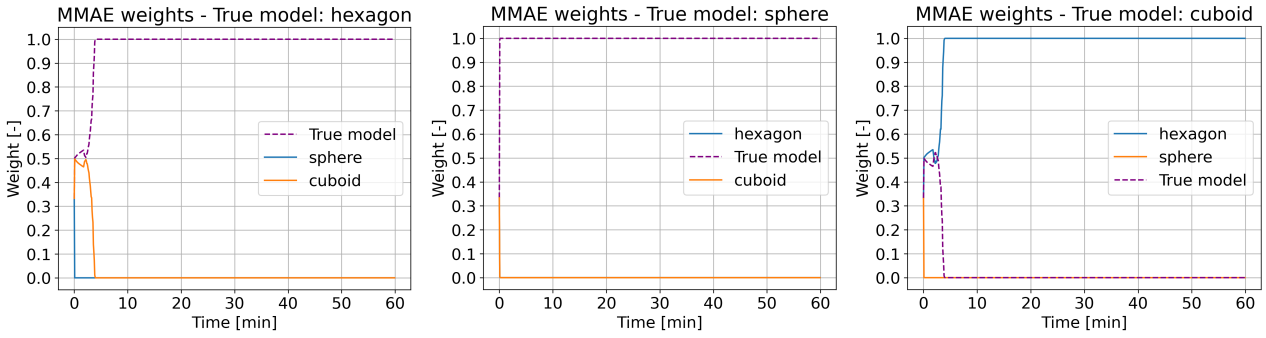


Figure 4.3.3: MMAE weights for simple test cases with different true shape model.

The reason the MMAE algorithm incorrectly determines the hexagonal prism to be the true model is explained by the brightness measurements for each shape model in Figure 4.3.4. The brightness measurements for the sphere are clearly very distinct from the other two, so it is expected that the MMAE algorithm can correctly identify it. The brightness measurements for the other two are nearly identical. This makes it almost impossible for the MMAE algorithm to distinguish between the two models, and it happens to favour the hexagonal prism.

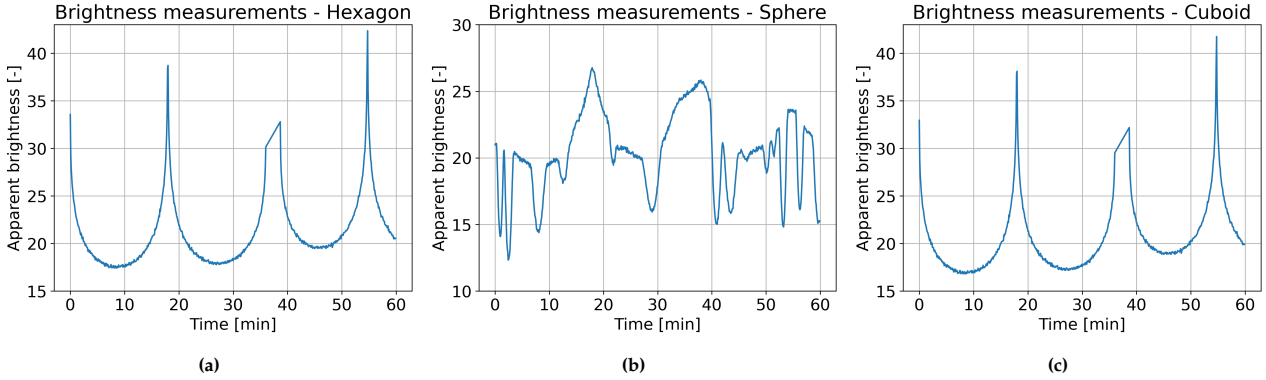


Figure 4.3.4: Brightness measurements for the three shape models.

It is interesting that the measurements for the hexagonal prism and the cuboid are so similar, even though the objects have a distinctly different shape. A clue behind this is given by the values of the apparent brightnesses. For both shapes the values are relatively high, which means they have a low brightness in the sky. This is caused by the specific geometry of the positions of the sun, observer and satellite during the time period of the estimation. This is confirmed by shifting the simulation epoch back by 4 hours, giving the brightness measurements in Figure 4.3.5.

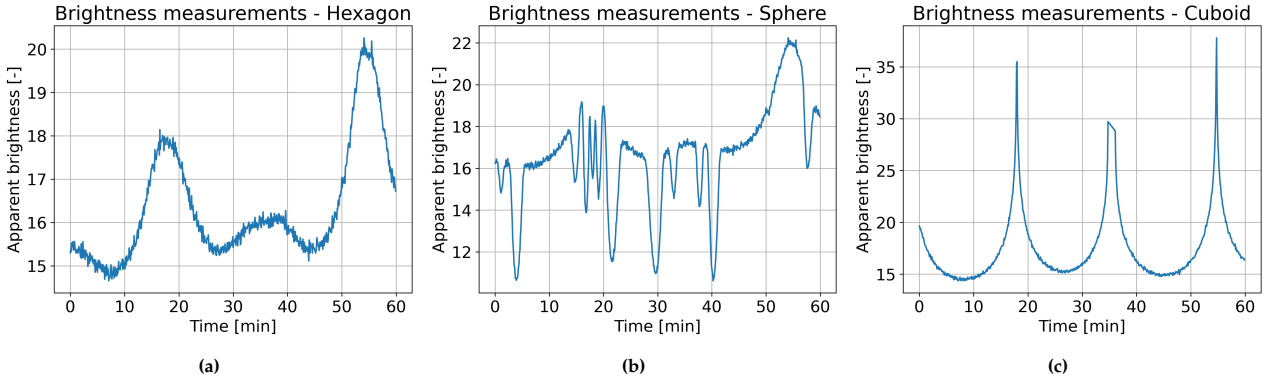


Figure 4.3.5: Brightness measurements for the three shape models, simulation epoch 4 hours earlier.

The observation geometry is now different, leading to different brightness measurements for all three shapes. Now the measurements for the hexagonal prism and the cuboid are very different from each other. The MMAE weights for the three cases for this simulation epoch are shown in Figure 4.3.6. The MMAE algorithm is as expected now able to correctly identify the correct true model in each case.

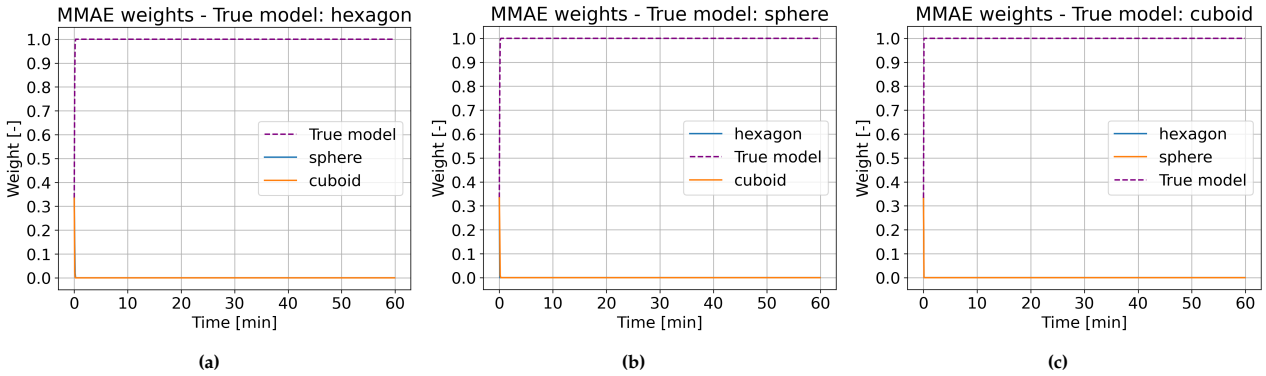


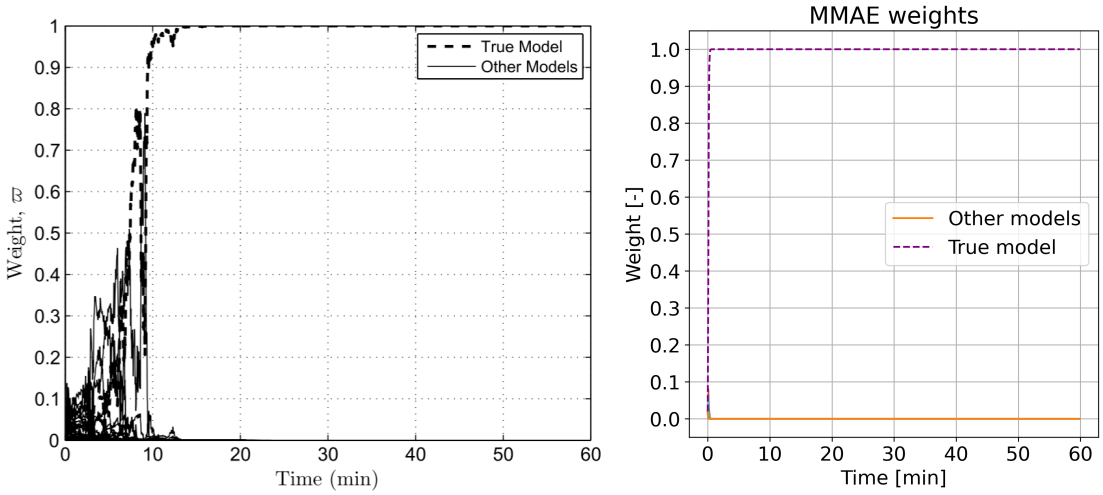
Figure 4.3.6: MMAE weights for simple test cases with different true shape model, simulation epoch shifted 4 hours back.

Paper result replication

With the basic functionality of the MMAE algorithm verified, the next step is to try and replicate the results obtained by Linares et al. [23], as the MMAE algorithm implemented is mostly based on their paper.

In the paper 50 different shape models are used, 25 regular polygon prisms and 25 rectangular cuboids. The regular polygon prisms are either triangular, square or hexagonal. The dimensions of the polygon prisms are defined by the s of the sides of either the triangle, square or hexagon and height h. The rectangular cuboids are again defined by length, width and height, but for consistency with the paper they are now denoted by s_1, s_2 and h, respectively. To create 50 shapes they randomly generated the dimensions from a uniform distribution on the interval [0.01, 5] m. The number of sides for the polygon prisms were randomly selected on the interval [3, 6], with any instance of five sides being set to four, as pentagonal prisms are not included. The dimensions of the 50 models used can be found in the paper [23].

The true model used in the paper is one of the 25 polygon prisms, namely a hexagonal prism with the dimensions $s = 1.0466$ m and $h = 2.6199$ m, which is why these dimensions were also used in the simple case. The MMAE weights obtained in the paper are shown in Figure 4.3.7a. In their case the MMAE algorithm was able to correctly identify the true hexagonal prism from the 49 other shape models. Figure 4.3.7b shows the weights for the case where these results were attempted to be replicated. It can be seen that while the implemented MMAE algorithm is also able to identify the correct shape model, the weight plots do not look identical. The implemented MMAE algorithm finds the correct shape model much quicker than the algorithm in the paper does.



(a) MMAE weights from paper.

(b) MMAE weights for replication attempt.

Figure 4.3.7: MMAE weights for 25 regular polygon prism and 25 rectangular cuboid shape models. Left plot obtained from Linares et al. [23], right plot attempted replication of results.

There is no clear reason as to why these results differ from each other. Most likely it is caused by the differences in the exact implementations of the algorithms. For example, in the implemented version for this thesis the orbit and attitude states are propagated using Tudat, while in the paper this is not the case. These and other differences have an effect on the overall estimation results. Still, it can be seen that the implemented MMAE algorithm also finds the correct shape out of the model bank of 50 models, showcasing that it works as expected.

While the results look good, it might be the case that the shapes used in the paper were tuned such that the algorithm exactly found the correct shape model. Therefore, to further verify the capabilities of the MMAE algorithm, another case was run with the same number of shape models, but with different randomly generated dimensions. The dimensions are generated from the same intervals as in the paper. The dimensions of the new polygon prisms and cuboids are given in Table 4.3.3. The same hexagonal prism was added as one of the polygon prism, to again act as truth model for the estimation. In Table 4.3.3a it has replaced Prism 10.

Table 4.3.3: Dimensions of regular polygon prisms and rectangular cuboids used in successful result replication.

(a) Regular polygon prism number of sides and dimensions.

Prism	# of Sides	s [m]	h [m]
1	6	4.9945	3.6032
2	6	4.7125	2.2873
3	6	3.3695	3.3989
4	6	2.9267	4.424
5	6	2.7632	4.777
6	6	2.6546	1.3485
7	6	2.4244	4.3458
8	6	1.4163	1.7505
9	6	1.3488	1.4997
10	6	1.0466	2.6199
11	4	3.2091	2.5259
12	4	2.8915	2.6896
13	4	2.09	0.1038
14	4	1.3626	4.6914
15	4	1.2478	0.9977
16	4	1.2076	2.9334
17	4	0.7221	0.1233
18	3	4.6387	2.8739
19	3	3.4742	3.2473
20	3	3.4196	2.5113
21	3	2.6869	0.6329
22	3	2.2566	4.933
23	3	2.2333	2.0019
24	3	1.919	0.653
25	3	1.8313	0.632

(b) Rectangular cuboid dimensions.

Cuboid	s_1 [m]	s_2 [m]	h [m]
1	4.976	1.7099	2.6031
2	4.7939	2.3149	0.9233
3	4.7263	3.0607	2.5697
4	4.4159	1.983	3.4651
5	4.3825	3.2072	0.7348
6	3.7736	4.7121	2.0078
7	3.0962	2.1312	0.9224
8	2.9998	0.4546	1.8168
9	2.7188	3.5834	2.5
10	2.6196	2.5301	1.1105
11	2.1998	4.0259	0.7255
12	2.146	1.3382	1.2344
13	2.1407	4.7461	1.0894
14	2.1277	3.1094	1.767
15	1.7383	4.8529	2.9993
16	1.4658	0.6511	3.0736
17	1.021	4.5468	0.0216
18	0.876	1.7406	3.9317
19	0.6713	0.383	2.0789
20	0.6457	2.6036	0.8397
21	0.5733	3.6726	3.7491
22	0.539	4.2862	2.8323
23	0.4139	4.8817	2.3854
24	0.0917	1.5792	1.5757
25	0.0732	2.6599	2.0288

With these shape models a new estimation was performed. The weights for this estimation are shown in Figure 4.3.8. For this case the algorithm was also able to identify the hexagonal prism as the true shape model, showcasing that the MMAE algorithm also works for a model bank with different shapes than in the paper.

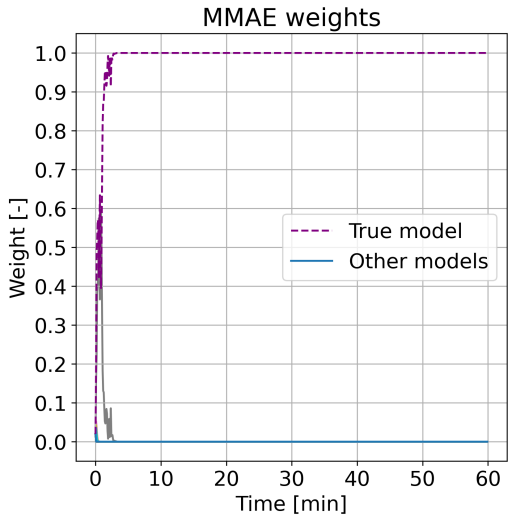
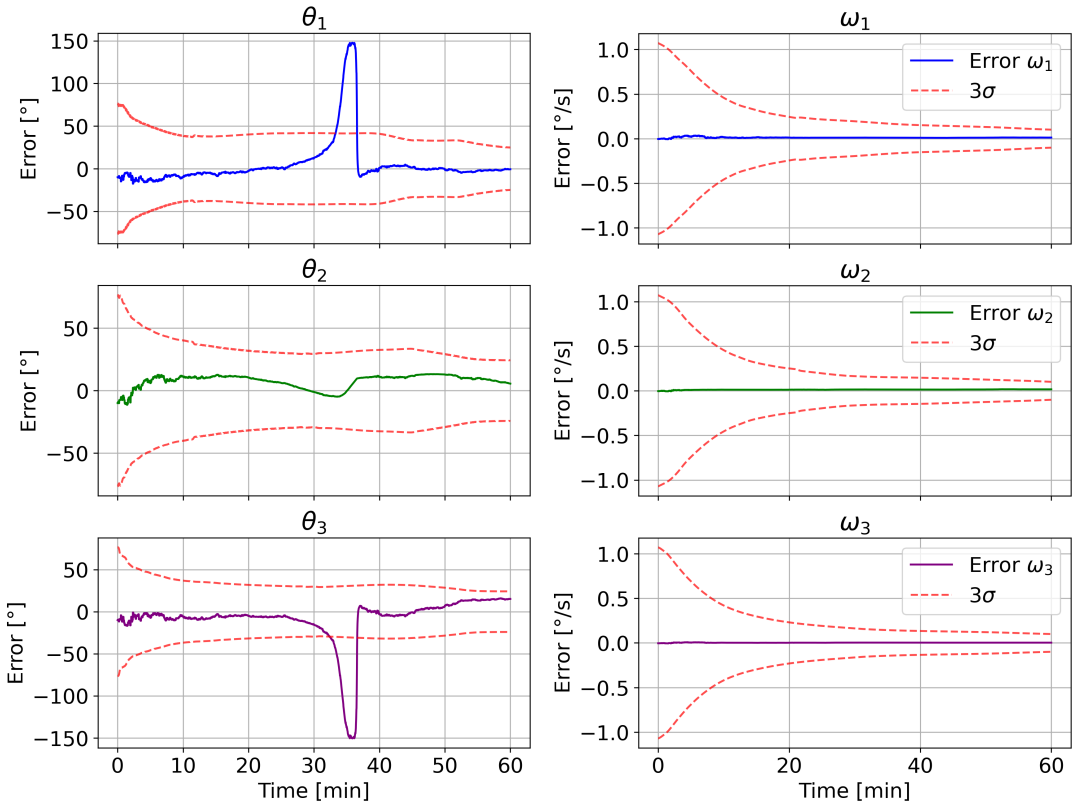


Figure 4.3.8: MMAE weights for case with new random generated shapes. The same hexagonal prism was used as truth model.

The identification of the shape is not the only task of the MMAE algorithm. At the same time the attitude and orbit states are estimated. To asses the capabilities of the algorithm on the state estimation, the estimation errors for the case with the newly generated shapes are plotted. In Figure 4.3.9 the Euler angle attitude and rotational velocity errors are shown, together with the 3σ covariance bounds. Overall the MMAE algorithm is able to estimate both the attitude and rotational velocities with low errors, bounded by the covariance bounds. However, in the attitude plots for θ_1 and θ_3 between 30 and 40 minutes large error increases can be seen. While the algorithm is able to converge to low errors again after these peaks, this result does require further investigation.



(a) Euler angle attitude error.

(b) Rotational velocity error.

Figure 4.3.9: Attitude state errors for MMAE estimation with new shapes.

To explain the large error peaks the true and estimated attitude states are plotted in Figure 4.3.10.

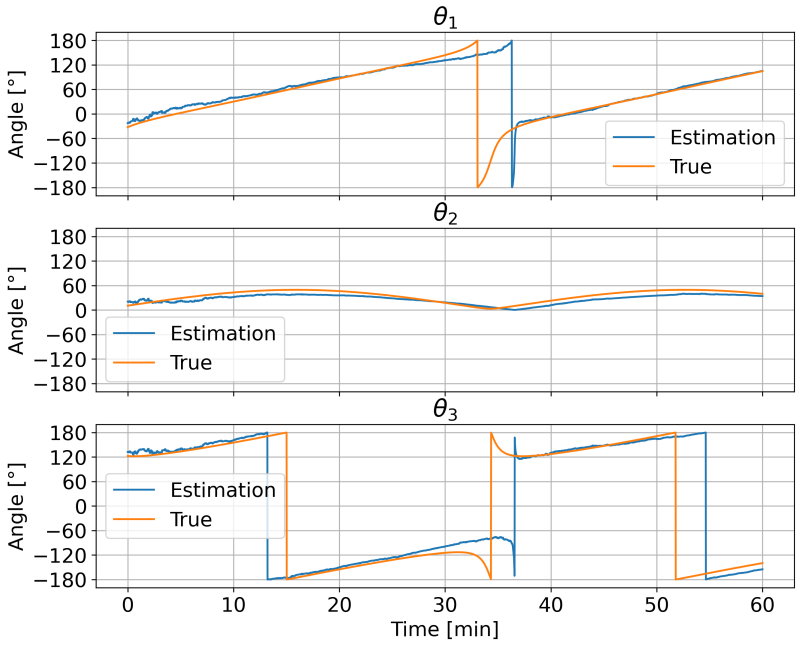


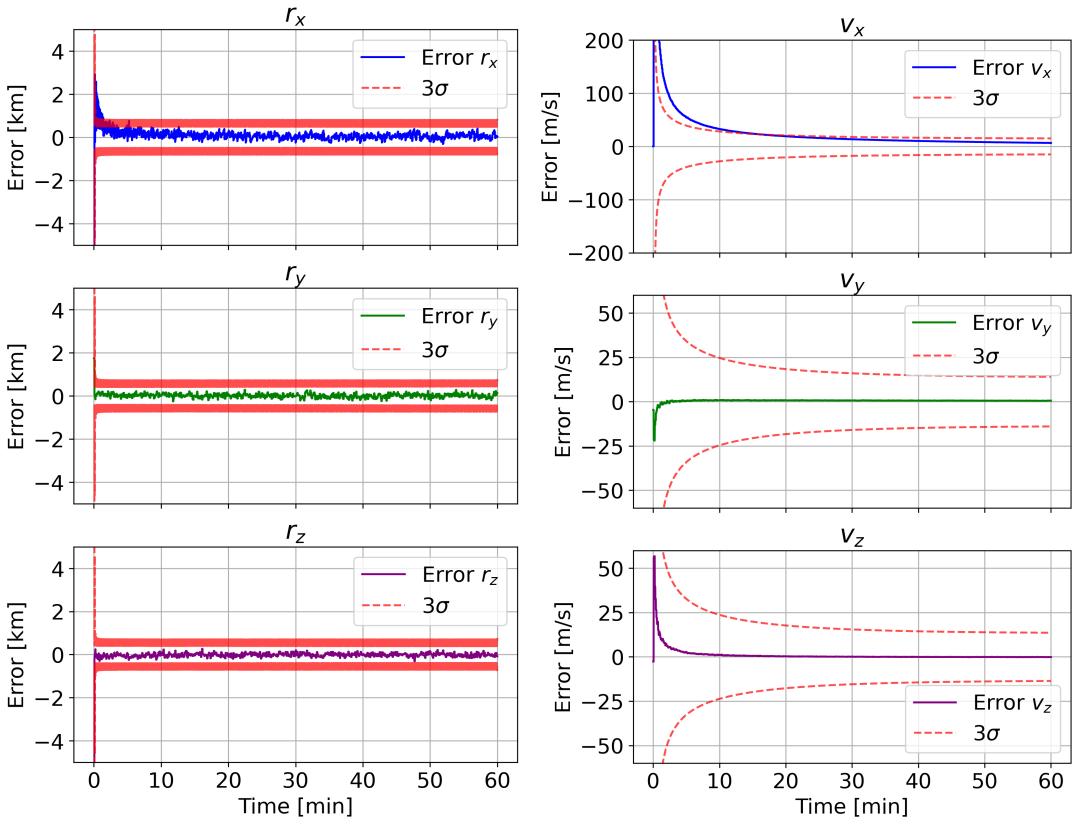
Figure 4.3.10: True and estimated Euler angle attitude for MMAE estimation with new shapes.

At the time of the large error peaks, between 30 and 40 minutes, there are indeed large differences between the estimated and true attitudes for θ_1 and θ_3. While both attitude lines have an angle roll-over to the opposite sign, the roll-over for the estimated states is delayed, leading to a divergence between the truth and estimation. After a few minutes the algorithm is able to correct the attitudes and converge closely on the true attitudes again. For θ_3 it can be seen that a delay in angle roll-over compared to the truth happens again around 55 minutes, but this causes lower errors than before. The reason for this is that the error values in Figure 4.3.9 are adjusted such that they also fall within the [-180°, 180°] range.

There are multiple aspects that could contribute to the estimations having this delay in angle roll-over. The first is that the states in the MMAE algorithm are fused from the estimations from all the shapes included in the model bank. Even though the weights of the other models are very low at this point in the estimation, this could have had an effect early on in the estimation, leading to the offset between the estimation and the truth.

Another aspect that could result in the differences is that shape used to simulate the measurements used in the estimation, the hexagonal prism, is a very symmetric shape. This means that different attitude states can produce very similar brightness measurements. Thus when the estimated attitude is close to the truth, but not exactly equal, the predicted measurements would result in very similar values as the true measurements. Because the measurement residuals are in such a case very low, the covariance values of the estimation are also not increased by much, which is exactly what is seen Figure 4.3.9, where the covariance bounds only slightly increase during the error peaks, while the error lines do cross these bounds. In the case of Figure 4.3.10 it can be seen that for θ_1 the estimated attitude around 30 minutes is lower than the true attitude. The estimation is then continued, causing the angle roll-over to occur later than in the true attitude. After this delayed roll-over the algorithm is quickly able to correct the estimated attitude, most likely due to the predicted measurements differing enough for the algorithm to register the estimated attitude is incorrect.

Even with this behaviour of the estimation it is clear that the MMAE algorithm is able to correctly estimate the attitude states of the true shape. It also needs to be seen whether this is the case for the orbital states estimations. The orbital position and velocity errors of the estimation are shown in Figure 4.3.11.



(a) Position errors.

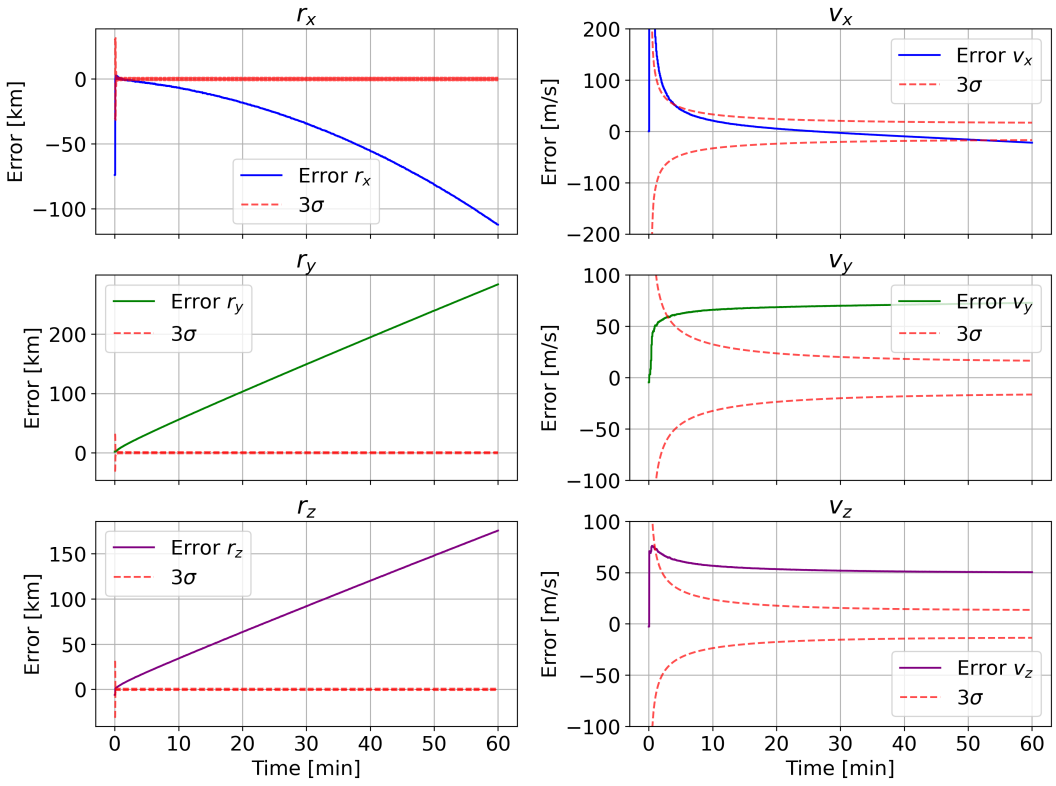
(b) Velocity errors.

Figure 4.3.11: Orbital state errors for MMAE estimation with new shapes.

It can be seen that while initially some of the errors are very large and outside of the covariance bounds, like for r_x and v_x, the algorithm is able to converge on the correct position and velocity states with lower errors. In the position error plot it can be seen that the errors are essentially 0, and are only offset from the truth by the random noise that is added to the angles and range measurements.

Overall it has thus been shown that the implemented MMAE algorithm works as intended and expected, with it being able to identify the correct shape model out of a model bank of 50 shape models of similar shape but with different dimensions, and at the same time estimate the rotational and orbital states with low errors. There is however one final aspect that needs to be discussed.

Up until now all the estimation performed in the thesis have included simulated range measurements for the orbit estimation, next to angles measurements in the form of azimuth and elevation angles. However, in real life apparent brightness measurements are most often made using passive optical sensors. These sensors only measure the apparent brightness and the azimuth and elevation angles from the sensor. Therefore it needs to be investigated if the MMAE algorithm is still able to perform without simulated range measurements. For this the same estimation with the newly generated shapes was performed, but with only brightness, azimuth and elevation measurements simulated. It was found that the algorithm was still able to find the correct shape model, but the estimation results were different from those in Figure 4.3.9 and Figure 4.3.11. To show this only the orbital position and velocity states are shown, given by Figure 4.3.12.



(a) Position errors.

(b) Velocity errors.

Figure 4.3.12: Orbital state errors for MMAE estimation with new shapes, without range measurements.

The position errors continuously increase throughout the estimation, completely diverging from the true orbital states by hundreds of km in all directions. The orbital velocity errors are also far larger than before.

While attempts were made to figure out why the orbit estimation results are so much worse without range measurements, due to the limited time available for the thesis it was decided to continue the research by keeping the range measurements in the simulations. This means all the other estimations performed in this thesis include simulated measurements for the apparent brightness, azimuth, elevation and range.

This approach is justified by the fact that the files with real measurements of real satellites that were used in the validation in Section 6.4 also contain range data for each measurement. This range data was added after the measurements were made using TLE data. It can therefore be assumed that for estimations of real satellites this range data can be acquired through TLEs. Since the objects for which the methods implemented in this thesis are real satellites, the inclusion of simulated range measurements in the estimations is thus a valid approach to the problem of shape characterisation and state estimation through light curves.

Conclusion

Overall it can be concluded that the implemented MMAE algorithm works as expected. It is able to identify the true shape out of a model bank of 50 shapes, and simultaneously estimate attitude and orbit states with lower errors. Next new methods for shape characterisation and state estimation are explored.

4.4. Variable-Structure Multiple-Model estimation

The previous section covered a method where multiple models were run simultaneously in order to characterize object shape and estimate attitude and orbit states. In this method the model bank used stays the same throughout the estimation. This means the method is a fixed-structure multiple-model (FSMM) estimation method. While the method proved capable of correctly identifying model shapes and providing accurate state estimates, such an FSMM method has a few downsides.

The first is that in an FSMM method like MMAE all models in the model bank are run at every time step, including models that after a certain time have low weights and are thus unlikely to be the true model. Running all these models is computationally expensive, and not efficient. Secondly, determining which models need to be included in the model bank is a non-trivial task. If too few models are used the true model might not be in the model bank, but if too many models are included the algorithm can become overwhelmed by the amount of options, leading to deteriorated results, for example because the algorithm is not able to converge on any one model, or because the estimations of the different models are so different that the fused state estimation diverges from the true states. The final downside is that the model bank needs to be determined before running the estimation, which restricts the search for the true model to only models included in the model bank.

Variable-Structure Multiple-Model (VSMM) estimation is a method of multiple-model estimation that has the potential to overcome these downsides. The word variable means that the model bank is not fixed during the estimation, but can vary based on different rules and circumstances. Two types of VSMM method are explored in this section, called Model-Group Switching and Model Generation.

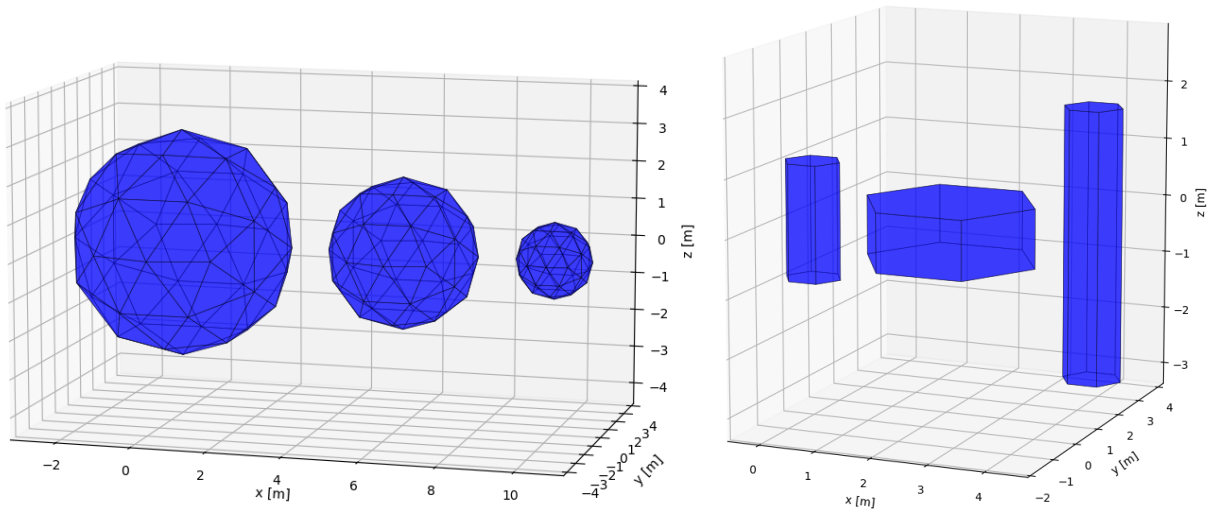
4.4.1. Model-Group Switching

As the name suggests, in the Model-Group Switching (MGS) method of VSMM estimation there are different predetermined groups of models. The method is considered variable because during the estimation the algorithm switches between model groups, depending on the measurements and certain rules.

There are different options for the way an MGS implementation can choose to switch between groups. The two main options are so-called 'soft switching' and 'hard switching'. In soft switching the assumption is that at any time each group has a certain probability of having one of its models being correct, and the overall estimate is a probabilistically weighted sum of all model-group estimates [13]. With hard switching only one model-group is run at a time. The MGS method implemented for this thesis uses hard switching. This makes the method applicable to the shape characterisation problem, as the objects which are characterised do not change shape throughout the estimation, and only one shape will be the true shape per estimation.

Basics

The implementation of the MGS algorithm for the object characterisation and attitude estimation problem is based on the chapter on VSMM in the book by Li, Bar-Shalom and Blair [13], and was adapted to work for the shape characterisation problem. The estimation starts with one central group, called M0, consisting of models of different distinct shape types, for example a sphere and a hexagonal prism. Each shape type in the central group has a corresponding separate model group, consisting of models of the same shape type with different dimensions, an example for which is shown in Figure 4.4.1 below.



(a) Group of sphere models.

(b) Group of hexagonal prism models.

Figure 4.4.1: Example of different model groups in MGS method.

The models in the central group are input into an MMAE algorithm, run for a single time step at a time. When measurements are received, the MGS algorithm decides if and which one of the specific shape type groups needs to be activated, based on the switching rules, which will be discussed shortly. If a new group is activated, the models of this group are run in the MMAE algorithm. It might be that the wrong group is activated, and that the true shape is in another group. The algorithm can switch to the correct model group, but only by first switching back to the central model. Direct switching between other model groups is not possible. The key aspect that makes this switching work is that all the models in the central group are also included in their shape type model group.

The orbit and attitude states are continuously estimated by the MMAE runs, outputting the weighted state estimation for the models active at each time step. In the end the MGS algorithm outputs the weights of the models in each model group that was activated, as well as the state estimations with corresponding covariance matrix. The goal for the algorithm is to activate the model group of the right shape type, and identify the model with the dimensions matching that of the true shape model.

Switching rules

The switching rules are based on the likelihoods L_k and weights w_k of the models active at time k, which are output by the MMAE algorithm. The likelihoods L are calculated with Equation 4.3.3 and the weights w with Equation 4.3.5. With the central group being M_0, the other model groups are called M_j, with j going from 1 until the number of model groups n. The models in the central group are called m_j, with each model m_j being the model in M_j linking M_j to M_0. This means model m_1 is present in both group M_0 as group M_1. Lastly, there is a set of thresholds, T_0 and T_1, that determine whether a group is activated or not.

The following sets of logic are implemented to determine which groups are activated at which times:

- Activate group M_j while M_0 is active, if *both* of the following conditions for the likelihood and weights are satisfied:

$$L_k^{(j)} = \max_{m_l \in M_1} L_k^{(l)}, \quad w_k^{(j)} > T_0 \tag{4.4.1}$$

  For example, if model m_3 has the highest likelihood of group M_0 and its weight is higher than threshold T_0, then group M_3 is activated.
- Activate group M_0 while M_j is active, if *both* of the following conditions for the likelihood and weights are satisfied:

$$L_k^{(1)} = \max_{m_l \in M_j} L_k^{(l)}, \quad w_k^{(1)} > T_1 \tag{4.4.2}$$

  For example, if model m_2 has the highest likelihood of group M_2 and its weight is higher than threshold T_1, then group M_0 is activated.

The mangitude of thresholds T_0 and T_1 thus determines how difficult it is to switch from group M_0 to a group M_j, or switch back from a group M_j to M_0, respectively. For the initial implementation of the MGS algorithm both thresholds are assigned a value of 0.5.

Additional rules

There is a final set of rules in the algorithm that further regulate which model groups are activated and fed into the MMAE algorithm for the next time step, $k+1$. Let the model group active at time k be M_k. When the switching rules Equation 4.4.1 or Equation 4.4.2 determine a new model group is activated, the models in this group are first put into an MMAE run for time k. Here the linking model that is in both model groups is not input in the MMAE algorithm again, so this model is only run once. Defining the initial active model group to be $M_o = M_k$, and the newly activated group M_a, weights and marginal likelihoods of model groups $M_l = M_o, M_a$ can be calculated using Equation 4.4.3.

$$w_k^{M_l} = \sum_{m_i \in M_l} w_k^{(i)} \tag{4.4.3a}$$

$$L_k^{M_l} = \sum_{m_i \in M_l} L_k^{(i)} \hat{w}_{k|k-1}^{(i)} / \sum_{m_i \in M_l} \hat{w}_{k|k-1}^{(i)} \tag{4.4.3b}$$

The final set of activation rules determines which models group will be active in the next time step, defined to be group M_{k+1}. These rules use an additional set of thresholds, namely T_1^w, T_2^w, T_1^L and T_2^L. Again, for the initialisation of the MGS algorithm these thresholds are all assigned a value of 0.5. The final activation rules are as follows:

- If
$$\frac{w_k^{M_a}}{w_k^{M_o}} < T_1^w \quad \text{or} \quad \frac{L_\kappa^{M_a}}{L_\kappa^{M_o}} < T_1^L \tag{4.4.4}$$
is true, terminate group M_a. Set $M_{k+1} = M_o$.
- If
$$\frac{w_k^{M_a}}{w_k^{M_o}} > T_2^w \quad \text{and} \quad \frac{L_\kappa^{M_a}}{L_\kappa^{M_o}} > T_2^L \tag{4.4.5}$$
is true, terminate group M_o. Set $M_{k+1} = M_a$
- If neither are true, fuse the state estimates, covariances and weights of both model groups. Set M_{k+1} to the union of both model groups, $M_{k+1} = M_o \cup M_a$.

The last rule means that it is possible to run central group M_0 and a group M_j at the same time. This is however undesirable behaviour, as this increases the amount of models run at the same time. The rule is left to catch the cases where neither Equation 4.4.4 nor 4.4.5 are true.

Algorithm summary

For clarity all the steps in the MGS algorithm are summarized in Table 4.4.1 below.

Table 4.4.1: Summary of steps in MGS algorithm.

Step	Description
0	Set $M_k = M_0$
1	Run MMAE with group M_k
2	Check Equation 4.4.1 and 4.4.2 if new group is activated. If so, set new group to be M_a. If not, output state estimate, covariance and weights and return to Step 1.
3	If group M_a is activated, run MMAE with M_a, excluding the model that is in both M_k and M_a.
4	Set $M_o = M_k$. Calculate $w_k^{M_l}$ and $L_k^{M_l}$ for groups $M_l = M_o, M_a$ using Equation 4.4.3.
5	If Equation 4.4.4 is true, terminate group M_a. Set $M_{k+1} = M_o$.
6	If Equation 4.4.5 is true, terminate group M_o. Set $M_{k+1} = M_a$.
7	If neither are true, fuse state estimates, covariance and weights of both model groups. Set $M_{k+1} = M_o \cup M_a$
8	Set $M_k = M_{k+1}$ and return to step 1.

The MGS algorithm can be considered a VSMM method because the active model bank changes throughout the estimation. However, in essence the method still works with a fixed model bank, as the different models in the model groups need to be determined before the start of an estimation. The next method, Model Generation, can be considered truly variable in this sense.

4.4.2. Model Generation

The Model Generation method, shortened to MGen, that is implemented for this thesis also starts with a central group of models of different shape type, similar to the MGS method. The difference is that the MGen algorithm does not switch to a model group of a certain shape type, but generates new models of that shape type from scratch. This makes this method truly variable, as the total model bank used in an estimation can not be determined beforehand, and will be different for each estimation.

Generation algorithm

There are different ways a MGen algorithm could be implemented. For this thesis a simple algorithm was developed, mainly based on the model likelihoods previously encountered in the MGS algorithm. First a certain likelihood threshold t^L is set. The MGen algorithm then starts by running an MMAE algorithm with the central group M_0, consisting of models m_j, with j ranging from 1 to the number of models in the central group n. When measurements are received, the algorithm checks whether any of the model likelihoods exceed t^L. If multiple models exceed the threshold, the model with the maximum likelihood is selected. For the selected model new models are generated by increasing and decreasing the dimensions of the model by fixed increments. This means if a model is defined by 2 dimensions, like the hexagonal prisms with side length s and height h, four new models are generated, and are designated m_{j0} up to and including m_{j3}. This process is visualised in Figure 4.4.2.

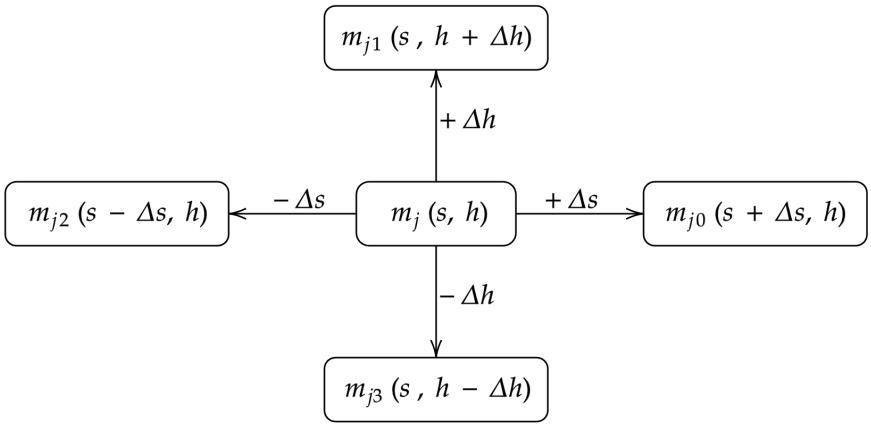


Figure 4.4.2: Diagram of how MGen algorithm generates new models starting from model m_j with dimensions s and h.

These new models are added to the model bank, and are run in the MMAE algorithm for the next time step. The process then starts again, with the model that exceeds the likelihood threshold becoming the next center model in the model generation as shown in Figure 4.4.2.

Model removal

If this process is left unchecked it is possible that new models are continuously added to the model bank, causing it to grow very large. Additionally, there is a chance that duplicate models are added to the model bank, like models m_{j01} and m_{j10}, as first adding Δs and then Δh to model m_j is the same as first adding Δh and then Δs. This is of course undesired behaviour. To avoid duplicate models being added to the model bank the dimensions of each newly generated model are checked against all the models that are in the current model bank. If a newly generated model has the same dimensions, it is not added to the model bank.

To avoid the continuous growing of the model bank the following system was decided upon. Once a model has exceeded the threshold t^L and new models are added to the model bank, the new model bank is run in the MMAE algorithm. Any models that have likelihoods below the threshold after this are removed from the model bank. Because the goal is to generate models that fit the measurements better, it is likely that after a while the model bank consists only of models exceeding the threshold, but still grows too large. Therefore an additional rule that was added is that if the number of models is larger than 10, the threshold t^L attains the value of the largest model likelihood at that time. If however the number of models then drops below 5, the threshold value is divided by 10.

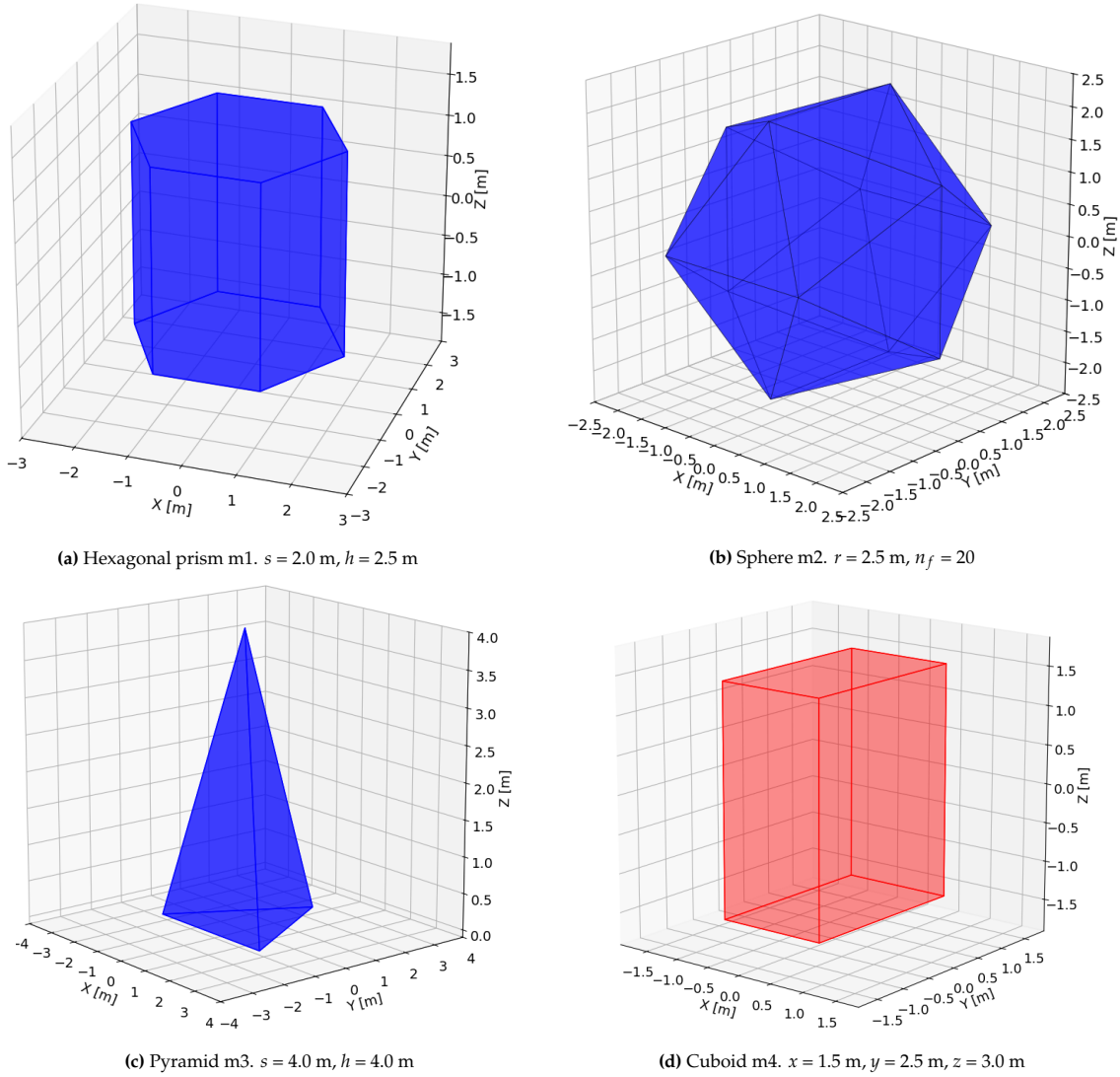
In theory this should only leave models in the model bank that fit better to the measurements, and thus have high likelihoods. This method of model removal is quite simple, and it will be investigated during the verification of the MGen algorithm whether this works sufficiently, or if changes to the algorithm are required.

4.4.3. Verification

To verify that the MGS and MGen algorithms work as intended, and are able to correctly identify object shape and estimate attitude and orbit states, both models are tested on the same test case as the MMAE algorithm was verified with in Subsection 4.3.4. The same orbit and attitude states, initial covariance matrices, BRDF parameters and time steps are used as this test case. One constraint that was added was a lower limit on the apparent brightness magnitude of 20 on the measurements simulated, as this is a realistic limiting magnitude for ground-based telescopes [25].

MGS

For the MGS algorithm a central group M0 is created, consisting of four differently shaped models. The model shapes are hexagonal prism (m1), sphere (m2), triangular pyramid (m3) and cuboid (m4). Like the prisms the triangular pyramid is defined by the length of the base sides s and height h. The central models and their dimensions are given in Figure 4.4.3.



(a) Hexagonal prism m1. $s = 2.0$ m, $h = 2.5$ m

(b) Sphere m2. $r = 2.5$ m, $n_f = 20$

(c) Pyramid m3. $s = 4.0$ m, $h = 4.0$ m

(d) Cuboid m4. $x = 1.5$ m, $y = 2.5$ m, $z = 3.0$ m

Figure 4.4.3: Shape models in central group of MGS verification.

Each model in the central group has a corresponding model group consisting of 4 additional models of the same shape with different dimensions. These shape model groups also contain the central group model of that shape. Group M1 thus contains 5 hexagonal prism models with different dimensions, with m1 being the prism that is in the central group. The dimensions of the 4 model groups are given in Table 4.4.2.

Table 4.4.2: Model groups used in MGS verification.

(a) Group M1 - Hexagonal prisms

Model	s [m]	h [m]
m1	2.0	2.5
m11	0.5	2.0
m12	2.0	0.5
m13	1.5	1.0
m14	0.5	4.5

(b) Group M2 - Spheres

Model	n_f	r [m]
m2	20	2.5
m21	20	1.5
m22	20	2.0
m23	20	3.0
m24	20	4.0

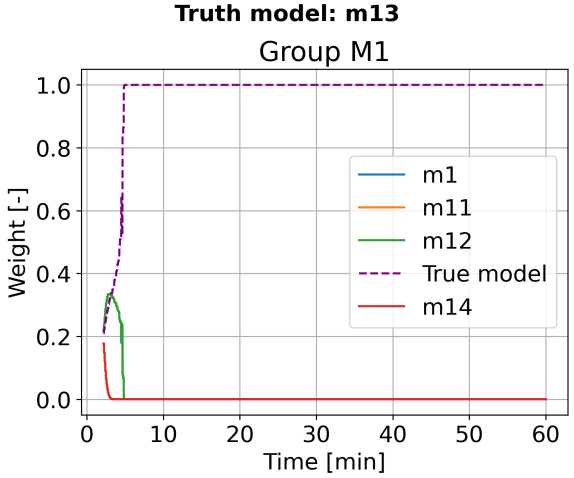
(c) Group M3 - Triangular pyramids

Model	s [m]	h [m]
m3	4.0	4.0
m31	4.0	2.0
m32	4.0	8.0
m33	1.0	4.0
m34	1.0	1.0

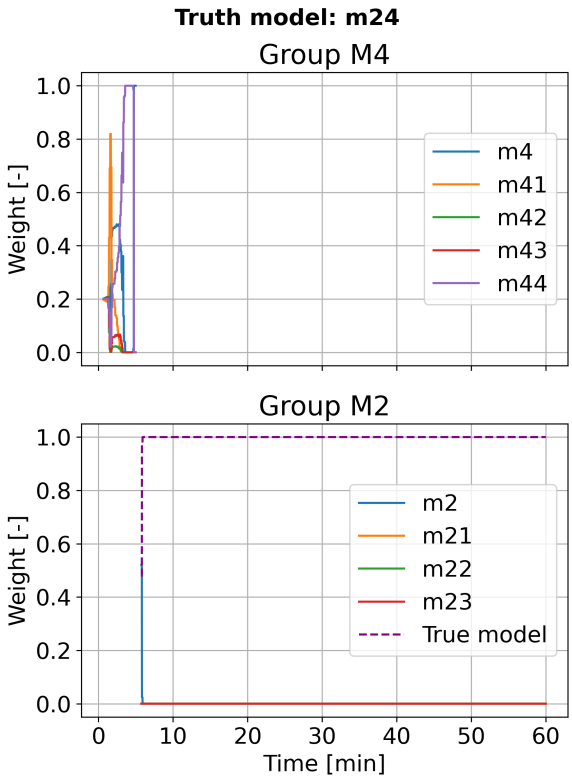
(d) Group M4 - Cuboids

Model	x [m]	y [m]	z [m]
m4	1.5	2.5	3.0
m41	2.5	2.5	3.0
m42	2.5	0.5	3.0
m43	1.0	1.5	4.0
m44	1.5	1.5	1.5

For the verification one model from each group is selected as the truth model as a test case. The MGS algorithm then has to activate the right model group and identify the model with the correct dimensions, as well as estimate the attitude and orbit. The weights for the cases where models m13, m24, m32 and m43 are the true shape model are shown in Figure 4.4.4a, 4.4.4b, 4.4.5a and 4.4.5b, respectively. For all cases initially group M0 is active. As soon as measurements are available the algorithm switches to a different model group. Because of this all plots of M0 weights are left out, as these do not provide any insight.

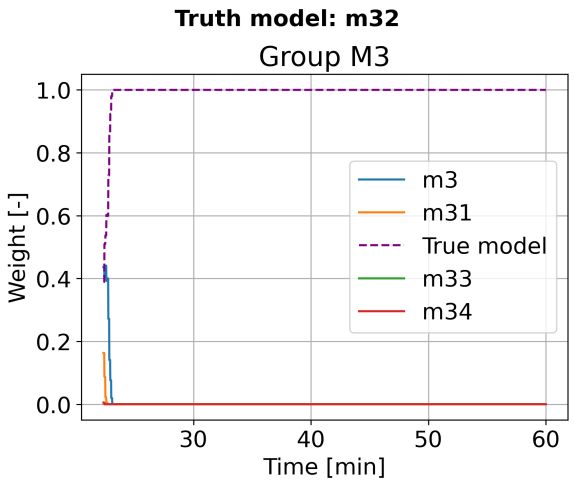


(a) Weights of group M1 for case with m13 as truth.

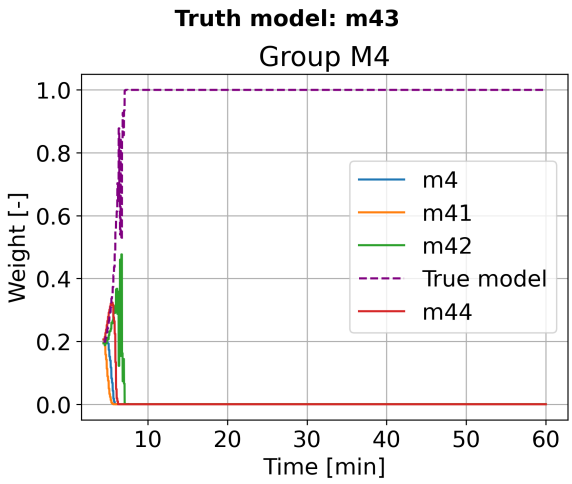


(b) Weights of group M4 and M2 for case with m24 as truth.

It can be seen that in all cases the MGS algorithm activates the correct model group and identifies the true shape model. The outlier in this is the case where spherical model m24 is used as the truth. Here the algorithm first incorrectly activates group M4. After a short while the model corrects this and activates group M2, and immediately finds the true shape model.

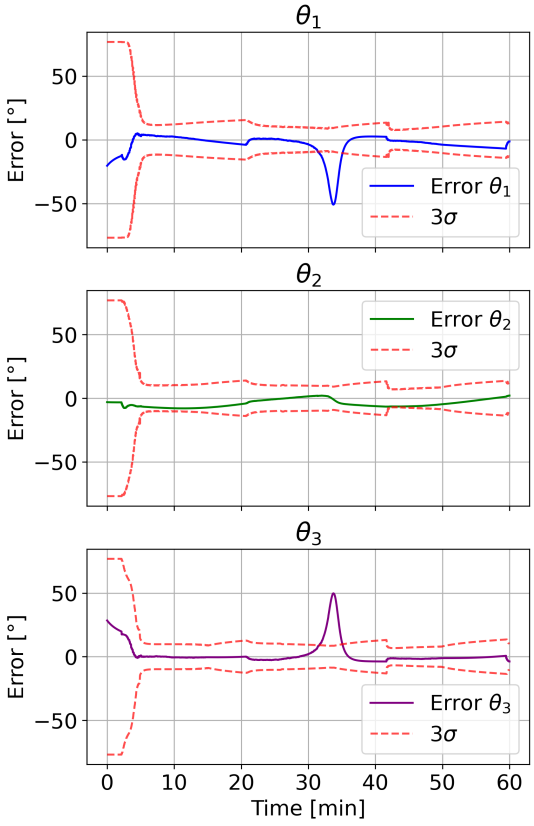


(a) Weights of group M3 for case with m32 as truth.

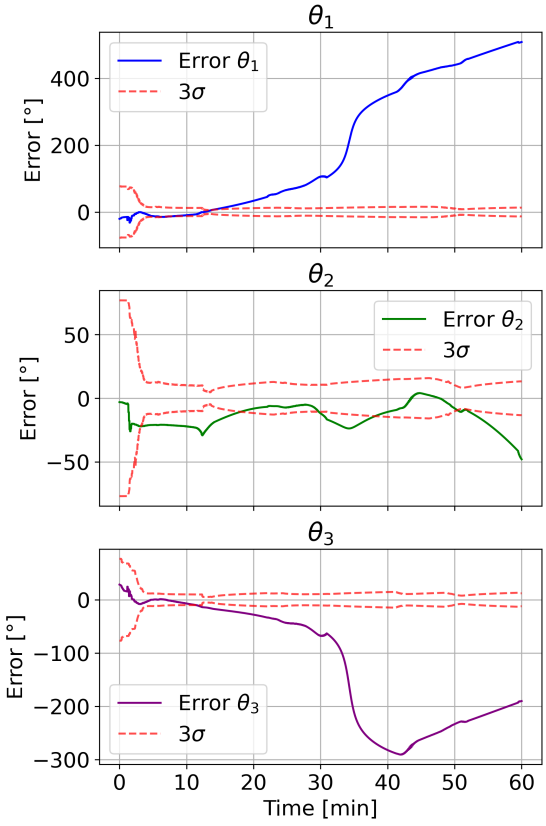


(b) Weights of group M4 for case with m43 as truth.

Next to finding the right shape model the MGS algorithm needs to estimate the attitude and orbit states of the different objects. The attitude error for the four different estimations are shown in Figure 4.4.6a, 4.4.6b, 4.4.7a and 4.4.7b.

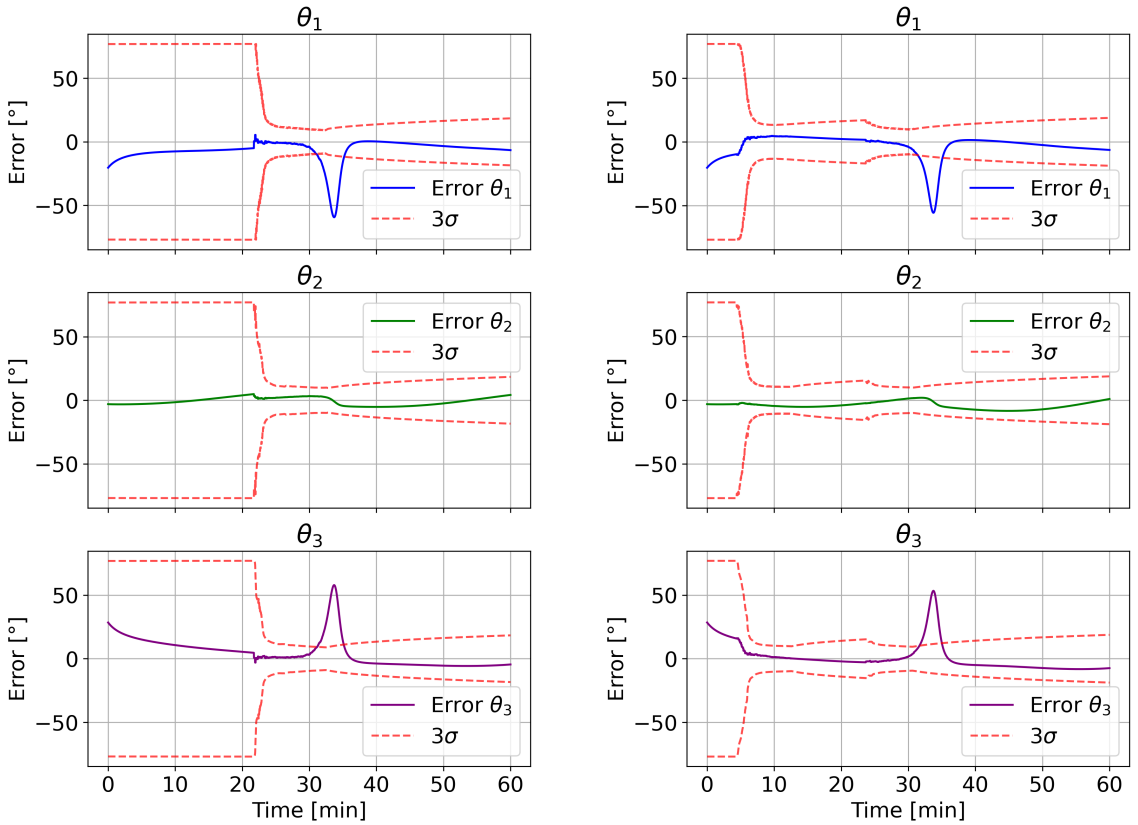


(a) Attitude error for case with m13 as truth.



(b) Attitude error for case with m24 as truth.

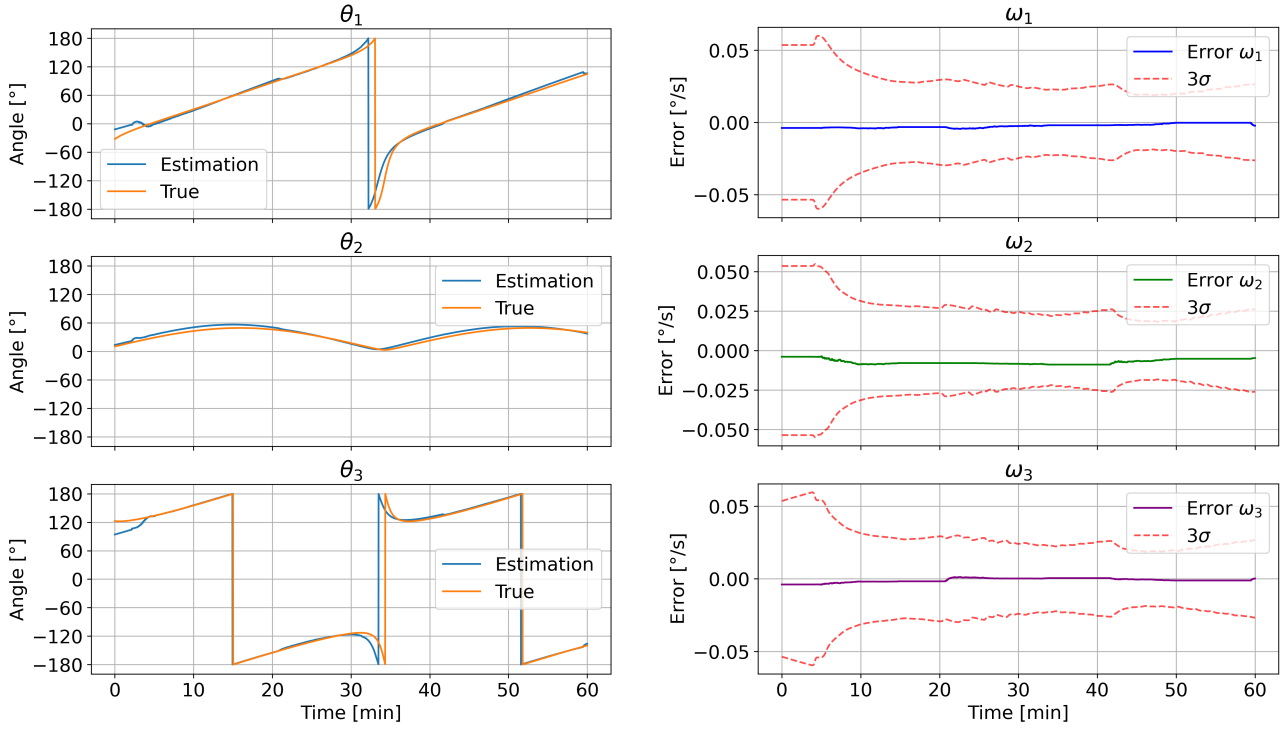
For the cases where the algorithm only activates the correct model group the attitude errors converge nicely around 0 error, bounded by the uncertainty bounds. For the m24 case the errors start low but significantly grow towards the end of the estimation. This shows that while the algorithm did find the correct shape model in the end, the act of first activating the wrong model group comes at the cost of the attitude estimation. Additionally, model m24 is a spherical model, which means it is highly symmetric. This could also make it difficult for the algorithm to properly estimate the attitude, since many attitude states would produce very similar measurements.



(a) Attitude error for case with m32 as truth.

(b) Attitude error for case with m43 as truth.

Another aspect the attitude error plots of m13, m32 and m43 have in common is a spike in the errors for θ_1 and θ_3 between 30 and 40 minutes runtime. This spike can be explained by looking at the plot of the Euler angle states next to the rotational velocity error plots in Figure 4.4.8. The ω_2 error is negative, meaning this estimated rotational velocity is larger than the truth. The algorithm therefore thinks the model spins faster than it actually does, causing it to reach the angle overshoot points in Figure 4.4.8a too soon, leading to the error spikes.

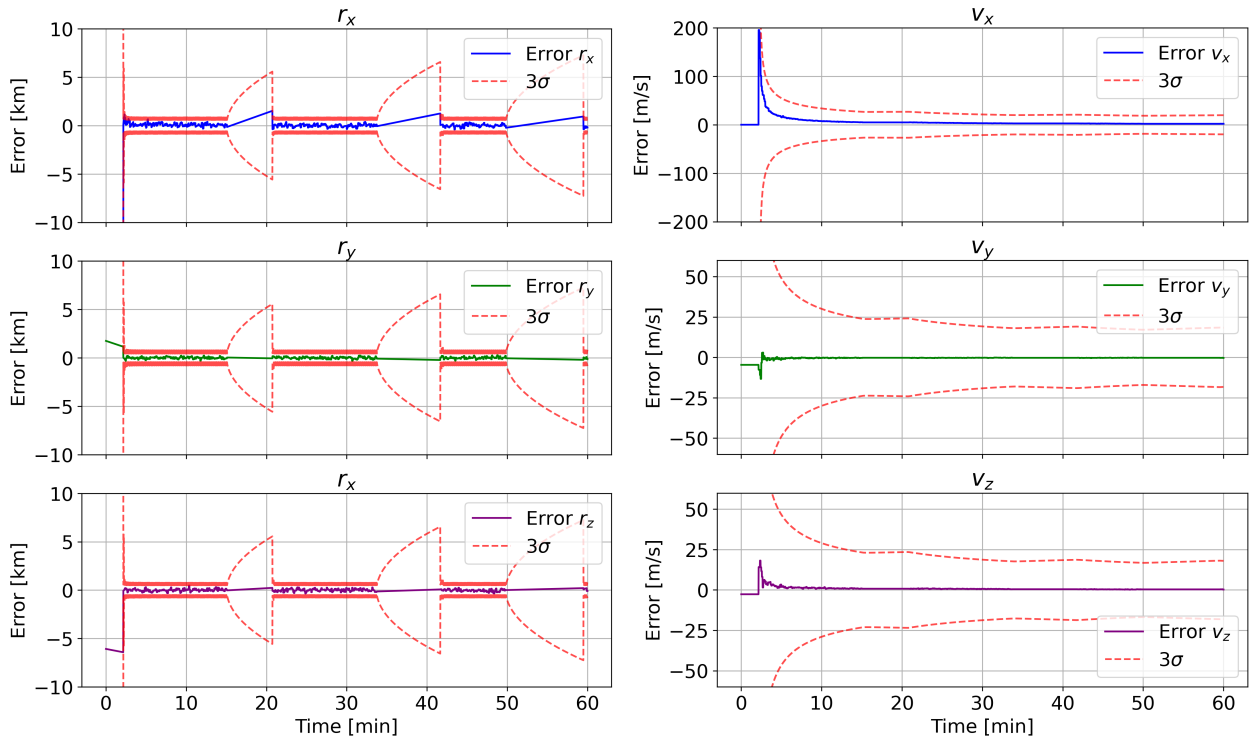


(a) Euler angle states over time.

(b) Rotational velocity errors.

Figure 4.4.8: Euler angle states and rotational velocity errors for m13 truth case.

Finally, it needs to be checked whether the orbit estimation of the MGS algorithm also conforms to expectations. The orbital position and velocity error plots for the case with m13 as truth model are shown in Figure 4.4.9.



(a) Orbital position errors.

(b) Orbital velocity errors.

Figure 4.4.9: Orbital position and velocity errors for m13 truth case.

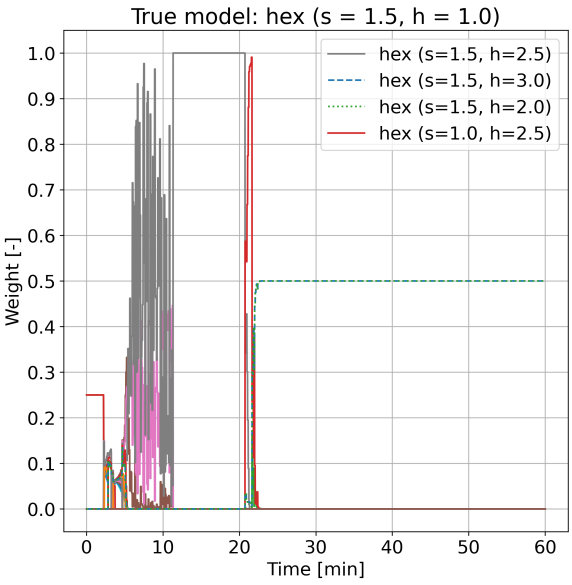
It is clear that the orbit estimation for the MGS algorithm works as well as that of the MMAE algorithm. At times with measurements the errors for both the positions and velocities are near 0. In between measurement times the uncertainty and the errors slowly increase, converging toward 0 when measurements become available again.

MGen

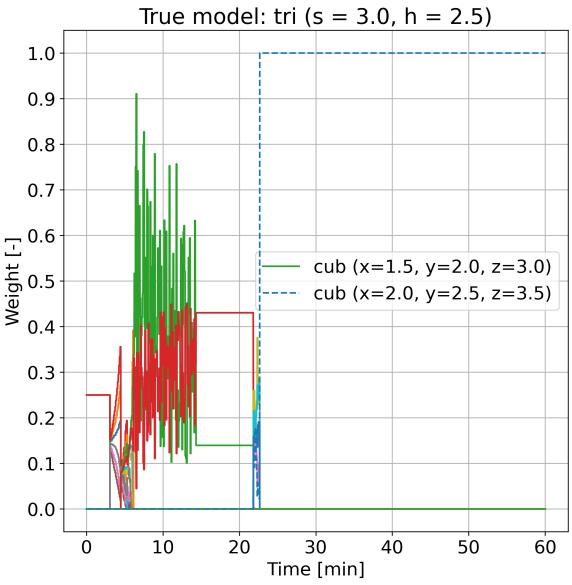
For the MGen algorithm only the central group needs to be defined. Other models are then generated from the models in the central group, based on the likelihood threshold set. Just like for the MGS verification four different shape models are included in the central group. However, instead of a sphere model a triangular prism is included. This was done because the sphere only has one dimension it can vary, its radius, while for the triangular prism the side length s and height h can be varied. This makes it possible to more extensively test the MGen algorithm. The central group hexagonal prism, triangular pyramid and cuboid have the same dimensions as in Figure 4.4.3. The triangular prism in the central group has dimensions $s = 2.0$ m and $h = 4.0$ m. To generate measurements it was decided to have the same truth shape models as in the MGS verification, m13 for the hexagonal prism, m32 for the triangular pyramid and m43 for the cuboid. For the triangular prism the true model was set to have dimensions $s = 4.0$ m and $h = 8.0$ m.

The goal for the MGen algorithm is to find the correct models by generating new models and converging on a model with the right shape and matching dimensions. The model weights of the four different truth cases are shown in Figure 4.4.10. Since a lot of different models are generated during the estimation, only models that receive a model weight larger than 0.5 at any point during the estimation or have a weight larger than 0.05 at the final time step are included in the legends. The different model shapes are abbreviated as *hex*, *tri*, *pyr* and *cub* for the hexagonal prisms, triangular prisms, triangular pyramids and cuboids, respectively.

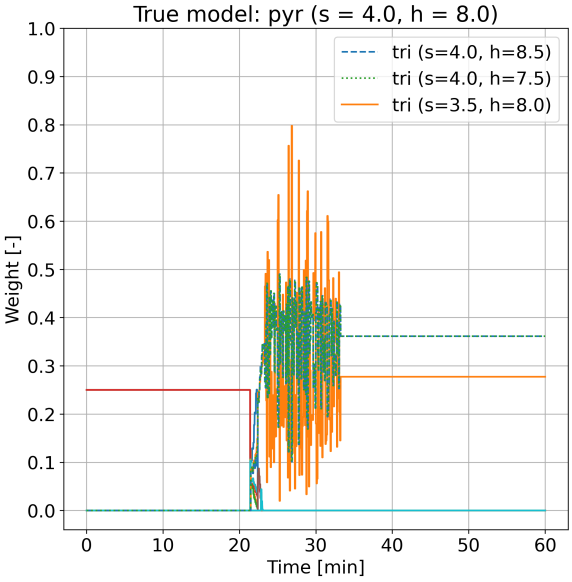
It can be seen that only for the case with the hexagonal prism (Figure 4.4.10a) and the cuboid (Figure 4.4.10d) as truth the algorithm finds the correct shape type. In both cases it is however not able to find the exact correct dimensions of the true models, but it gets close for most dimensions. The algorithm shows that it is not confident in any single model, as at the final timestep multiple models have a higher weight, with the highest weights being 0.5 for two separate models. This results stands in contrast to those found for the MMAE and MGS algorithms, with these algorithms converging on a single model each time.



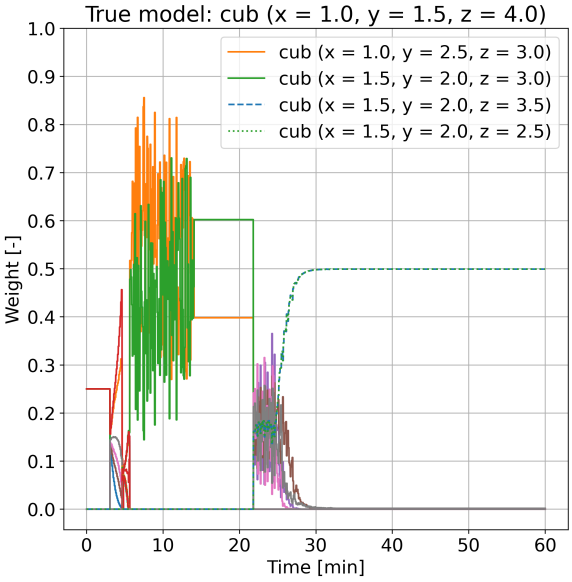
(a) Case with m13 as truth model.



(b) Case with triangular prism ($s = 3.0$ m, $h = 2.5$ m) as truth model.



(c) Case with m32 as truth model.

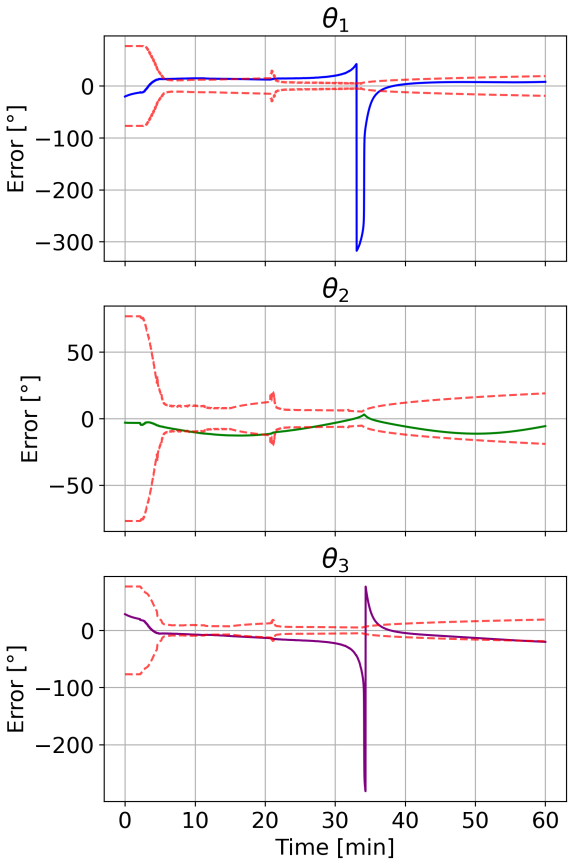


(d) Case with m43 as truth model.

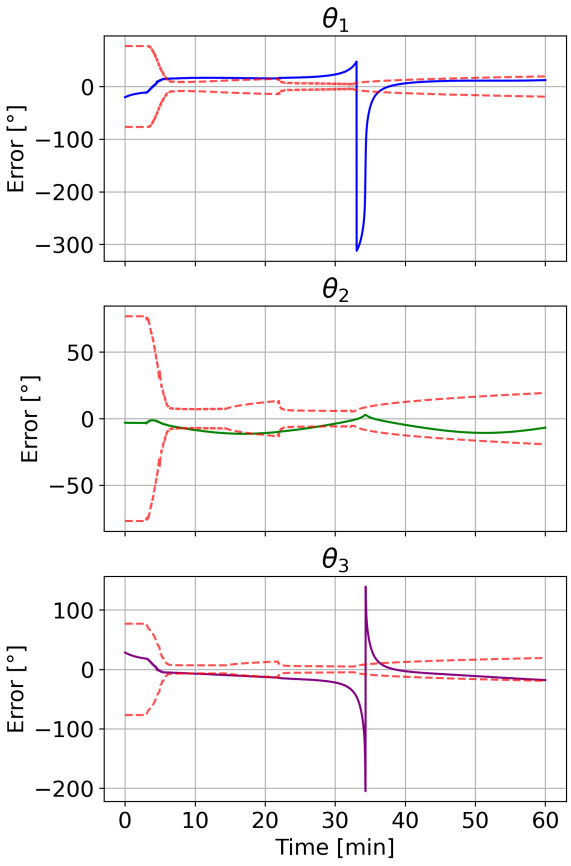
Figure 4.4.10: Weights for MGen algorithm of 4 different truth cases.

For the triangular prism (Figure 4.4.10b) and pyramid (Figure 4.4.10c) truth cases the MGen algorithm is not able to find the correct shape type, and subsequently also not the correct dimensions. The reason the algorithm only finds cubesats for the triangular prism case and triangular prisms for the pyramid case is that models that score below the likelihood threshold are removed from the model bank, because otherwise the model bank would continuously grow as new models are generated. This means that if the correct shape type does not score a high enough likelihood, it is removed, and therefore new models of this shape are also not generated.

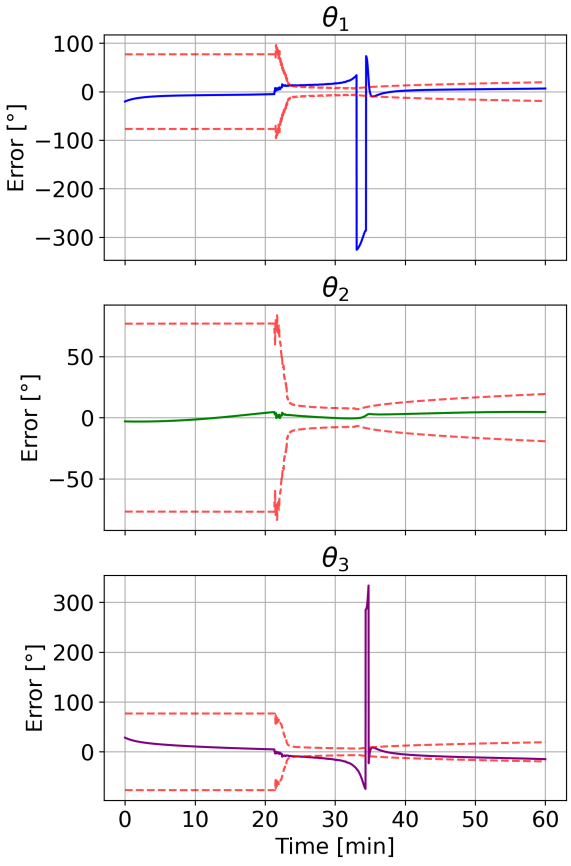
One way to avoid this could be to never remove the central group models. The results for this will be shown later. First the attitude errors of the above estimations are examined, which are shown in Figure 4.4.11. It can be seen that the attitude errors for all cases end up within the uncertainty bounds, and are close to 0 error, even for the cases where the algorithm finds the wrong shape type. The reason this happens is that the measurement plots of all the different shapes look very similar for this combination of orbits and attitude. Since the different shapes all start with the same initial guess, the algorithm can estimate the attitude states reasonably well, even if the wrong shape type and dimensions are identified.



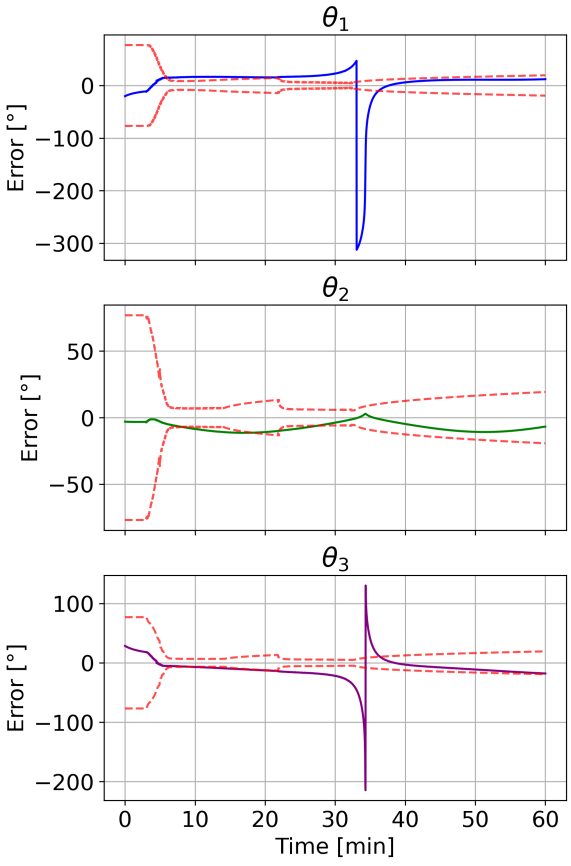
(a) Case with m13 as truth model.



(b) Case with triangular prism ($s = 3.0$ m, $h = 2.5$ m) as truth model.



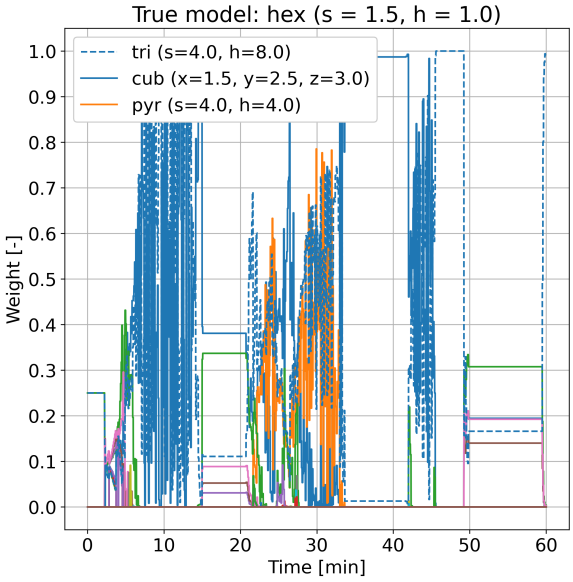
(c) Case with m32 as truth model.



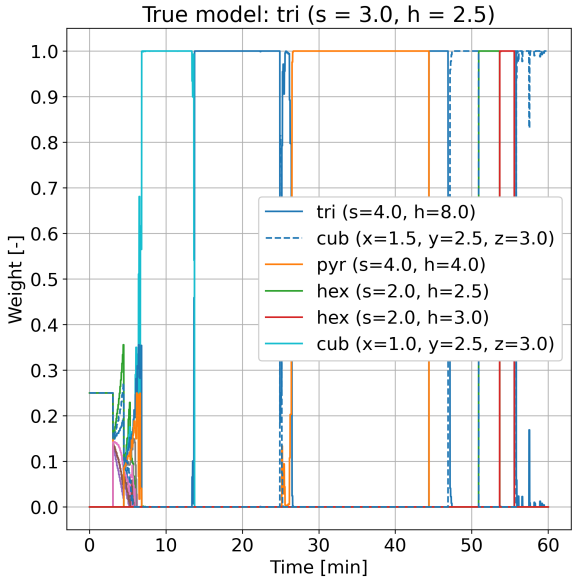
(d) Case with m43 as truth model.

Figure 4.4.11: Attitude errors for MGen algorithm.

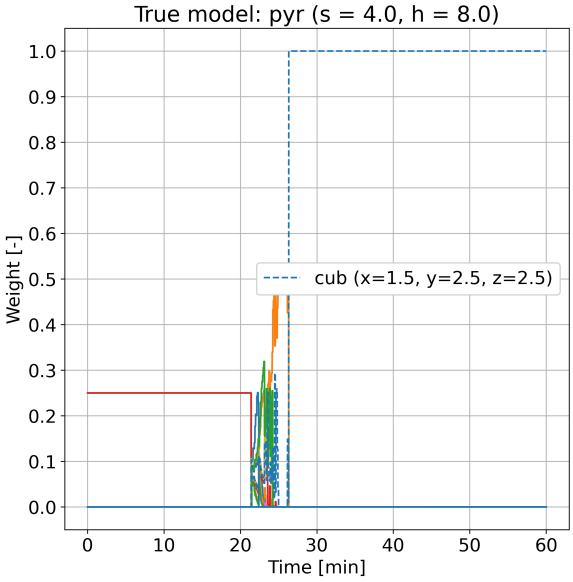
As previously mentioned, a possible solution to the MGen algorithm not being able to identify the correct shape type might be to change the algorithm so that the central models are never removed, even if their likelihoods fall below the threshold. This makes it possible for the algorithm to select the right shape type later in the estimation, and generate new models of the correct shape type. Adding this change to the algorithm while leaving the other workings exactly the same, and running the same estimation cases leads to the weight plots in Figure 4.4.12.



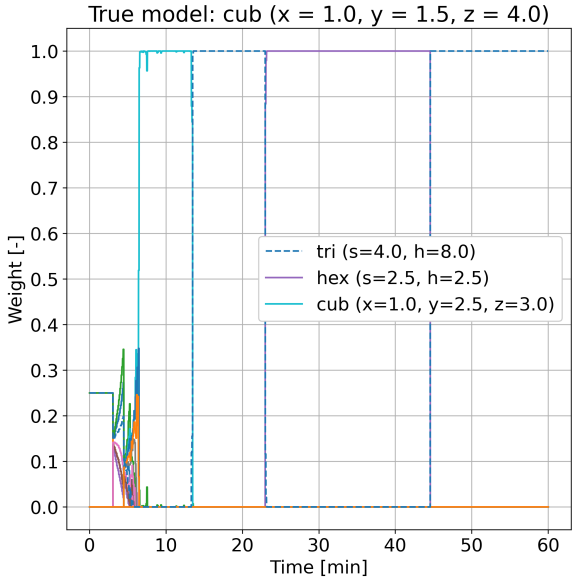
(a) Case with m13 as truth model.



(b) Case with triangular prism ($s = 3.0$ m, $h = 2.5$ m) as truth model.



(c) Case with m32 as truth model.



(d) Case with m43 as truth model.

Figure 4.4.12: Weights for MGen algorithm without removing central models.

It can be seen that the results are vastly different from those in Figure 4.4.10. Overall the results are not better than before. For example, for the case where the hexagonal prism was the true model, the MGen algorithm ended with four hexagonal prisms in Figure 4.4.10a. Now in Figure 4.4.12a no hexagonal prisms are found. Instead, only the central models of the three other shape types are assigned high weights during the estimation. For the other cases the results also do not improve compared to those of Figure 4.4.10.

Conclusion
It was shown that the MGS algorithm works as expected, with the algorithm being able to find the right models for all test cases. The attitude errors of the estimations are low except for the case with the sphere model as truth, where initially the wrong group is activated. However, it would be possible to rerun the attitude estimation with only the true model, so these errors could still be decreased for this case.

The MGen algorithm does not fully work as intended. The algorithm does not find the correct shape type in all cases, and thus also not the right shape dimensions. When the right shape type is found the algorithm gets closer to the right dimensions, but is not able to converge exactly on the correct dimensions. Changing the algorithm so the central models are never removed only made the results worse for all cases, so the original algorithm will be used in the remainder of the thesis.

In summary, the implemented MGS and MGen algorithm demonstrate that VSMM estimation is a viable technique that is able to a certain extent to correctly identify the shape and estimate the attitude of space objects. While the algorithms perform with different quality of results, both will be included in the testing and analysis of the LEO test objects defined in Chapter 3. In the next chapter it will be explained how the performance of these algorithms will be tested on these realistic test cases, where they will also be compared to the MMAE algorithm, which has previously been used in literature for the attitude estimation and shape characterisation problem. This testing will be the base on which the research questions drafted in Chapter 2 will be answered.

5 Testing setup

Now that the different algorithms mentioned in Chapter 4 are implemented their performance can be tested. The goal of this testing is to create more insight into the workings of these algorithms, with the goal to answer the research questions. As mentioned previously in Chapter 3 it was decided to test the algorithms on different LEO satellites. This is done because the majority of space debris is present in LEO, and therefore it is important to test the algorithms on this orbit regime, as this is where they can make the biggest impact. In this chapter the ways these algorithms will be tested and analysed are discussed. First in Section 5.1 the test cases on which the algorithms will be run are defined. In Section 5.2 the subsequent testing strategy is highlighted.

5.1. Test cases

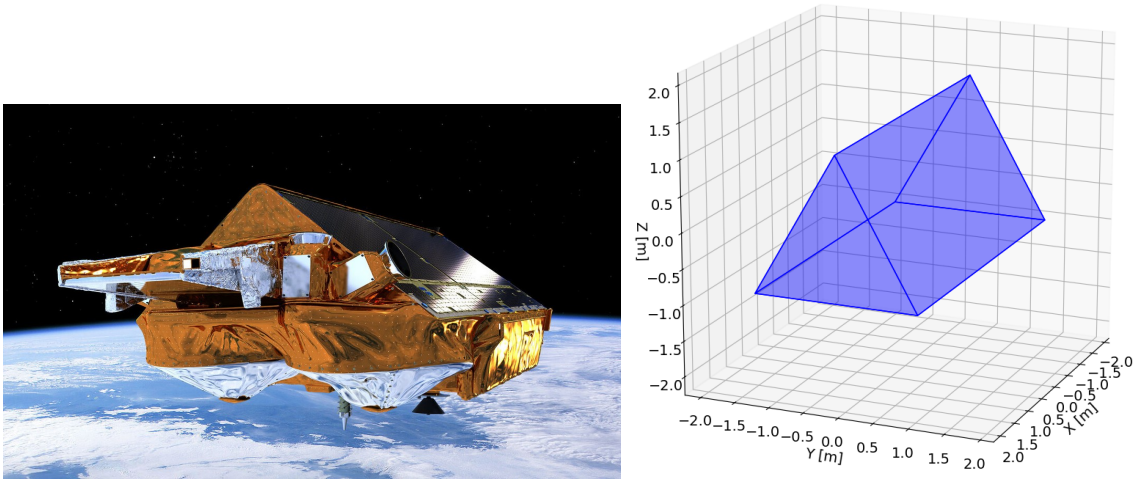
First the different test cases need to be defined. The aspects that need to be set up are how the test objects are modelled in order to simulate measurements, in terms of object shape and reflectivity coefficients of their facet surfaces. The other parameters that are required to set up the simulations are also defined.

5.1.1. Test object shapes

The objects for which light curve measurements need to be simulated were introduced in Section 3.1 and are: CryoSat 2, Swarm A/B/C, BlueWalker 3 and ACS3. As seen in the methodology chapter, the method to generate these light curves requires the object to be divided into facets. The light reflected off these facets towards the observer is then summed over all facets and the apparent brightness is calculated. It is possible to model any object accurately, as long as enough facets are used. However, it was decided to only model the test objects based on the overall larger structure shape, and not to full detail. There are two reasons behind this decision. The first is that it is difficult to find the exact dimensions of all the different parts of these satellites, as this information is not readily available to the public. The second is that the shape of a light curve is dominated by the light reflected off large surface areas. It is therefore also not necessary to model the objects to the smallest detail, as long as the large surface areas match those of the real objects closely.

CryoSat 2

The x-y-z dimensions of the spacecraft are 4.6 m x 2.34 m x 2.20 m.[1] However, the satellite is not shaped like a box, as can be seen in Figure 5.1.1a. It was decided to model the satellite as a triangular prism. This was done because the MGS and MGen algorithms work by having models of the same shape type with different dimensions. In this way the CryoSat shape model can be included in the triangular prism model group. The triangular prism model has dimensions $s = 2.30$ m and height $h = 4.60$. The simplified CryoSat shape model is shown in Figure 5.1.1b.



(a) Render of complete satellite.

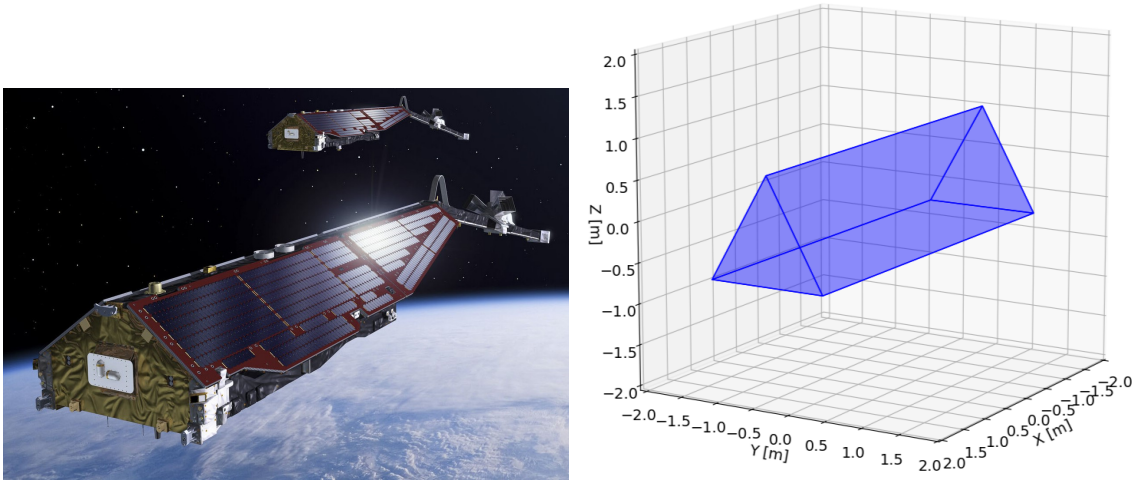
(b) Simplified shape model.

Figure 5.1.1: CryoSat 2 shape model compared to real satellite.

[1] https://www.eoportal.org/satellite-missions/cryosat-2#spacecraft. Accessed on 26-02-2025.

Swarm satellites

The Swarm satellites are all identically shaped, consisting of a main body and a trailing boom at the end, as shown in Figure 5.1.2a. The x-y-z dimensions of the satellite main body are 5 m x 1.5 m x 0.85 m.[2]. Seeing as the boom is very slender, it was assumed to not contribute much to the apparent brightness of the satellites, and is therefore not modelled. Furthermore, the satellite is tapered at the back. For simplicity it was decided to also model this satellite as a triangular prism. This shape is slightly too small at the front, but too large at the back, so these shortcomings cancel each other out, making it a good approximation to use in the simulations. The modelled dimensions are $s = 1.5$ m and height $h = 5.0$ m. The modelled shape is shown in Figure 5.1.2b.



(a) Render of two Swarm satellites.

(b) Simplified shape model.

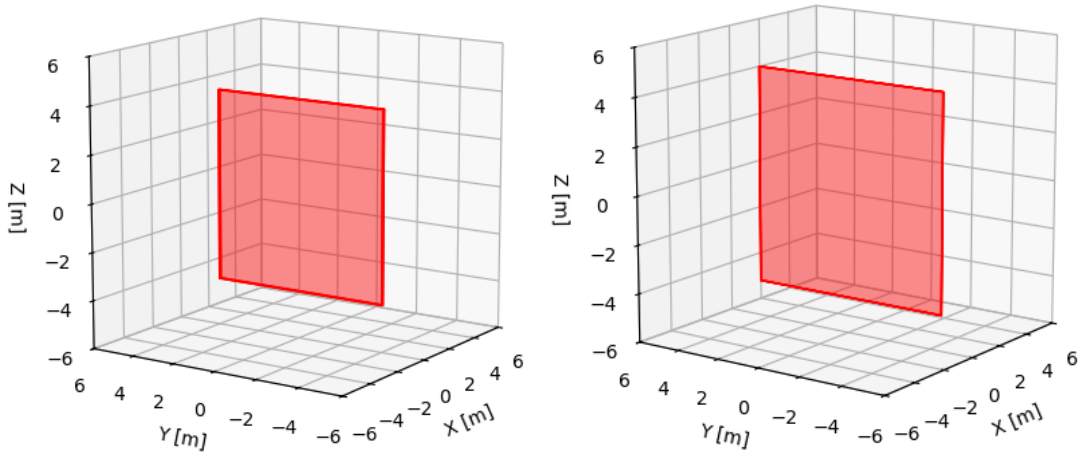
Figure 5.1.2: Swarm model compared to real satellite.

BlueWalker 3

The BlueWalker 3 satellite shape is very simple, consisting of a central cubesat surrounded by a very large communication array of 64 m^2. This large surface area dominates the apparent brightness of the satellite. Because of this the satellite is simply modelled as a large square plate with 8 m sides. Since no data could be found on the thickness of the array, it is assumed to be 0.1 m thick. The shape model is shown in Figure 5.1.3a.

ACS3

This satellite also has a simple shape, consisting of a small central cuboid with a large square solar sail with sides of 9 m. Since the large surface area of the solar sail will dominate the apparent brightness, the satellite is also modelled only as a flat plate. The sail is however much thinner than the communication array of BlueWalker 3, so the thickness is assumed to be 0.01 m. The model is shown in Figure 5.1.3b.



(a) BlueWalker 3 model.

(b) ACS3 model.

Figure 5.1.3: Shape models of BlueWalker 3 and ACS3 satellites.

[2] https://www.eoportal.org/satellite-missions/swarm#space-segment-concept. Accessed on 26-02-2025

5.1.2. Reflectivity coefficients

One aspect that needed to be determined for all models was the value of the BRDF coefficients, in particular the specular and diffuse coefficients R_{spec} and R_{diff}. These coefficients have a large impact on the shape of the light curve, with high specular coefficients leading to more sharp brightness peaks but overall lower apparent brightness, while high diffuse coefficients lead to less sharp peaks but overall higher brightness. The different facets of the models do not all have the same coefficients in reality, as some surfaces will be covered with solar panels, which reflect light more specularly, while other surfaces will reflect light more diffusely.

The information of the reflectivity coefficient values of the real satellites is not readily available, but also not necessary. This is because the shape of the light curve is what holds the most information about the attitude states, meaning the location and steepness of peaks and overall behaviour of the curve, while the actual brightness values matter less. This means that the difference in reflectivity coefficients between the different surfaces is more important than the actual values themselves. Therefore it was decided to assign values of 0.5 for both coefficients to facets which are any arbitrary surface of a satellite, while solar panels are assigned a specular coefficient $R_{\text{spec}} = 0.9$ and diffuse coefficient $R_{\text{diff}} = 0.1$. The exception to this is the solar sailing satellite ACS3. For this satellite the reflectivity coefficients were found, having a specular reflectivity coefficient of 0.74 on one side of the sail, and 0.23 on the other side [26].

5.1.3. Model bank

The full model bank used in the testing consists of 24 models, for the MGS algorithm divided into 6 model groups of different shape types, consisting of 4 models per group. The central group for the MGS and MGen algorithm therefore consists of 6 models of different shape type. The different shape types of the model groups, as well as the dimensions of the models within each group, are given in Table 5.1.1. The dimensions of all the models were selected so that in each model group there is a lot of variation between the different models.

Table 5.1.1: Model group dimensions used in testing.

(a) Group M1 - Triangular prisms

Model	s **[m]**	h **[m]**
m1	3.0	8.0
m11	4.0	4.0
m12	1.5	5.0
m13	2.3	4.6

(b) Group M2 - Spheres

Model	n_f	r **[m]**
m2	50	1.0
m21	50	0.5
m22	50	2.0
m23	50	4.0

(c) Group M3 - Flat plates

Model	x **[m]**	y **[m]**	z **[m]**
m3	0.5	6.0	6.0
m31	0.5	4.0	4.0
m32	0.1	8.0	8.0
m33	0.01	9.0	9.0

(d) Group M4 - Cuboids

Model	x **[m]**	y **[m]**	z **[m]**
m4	4.0	2.0	4.0
m41	5.0	5.0	5.0
m42	4.0	6.5	2.0
m43	2.0	2.5	8.0

(e) Group M5 - Rocket bodies

Model	r **[m]**	h_{cyl} **[m]**	h_{cap} **[m]**
m5	1.85	8.9	1.85
m51	1.5	9.0	1.0
m52	0.75	5.0	1.125
m53	2.7	4.8	2.7

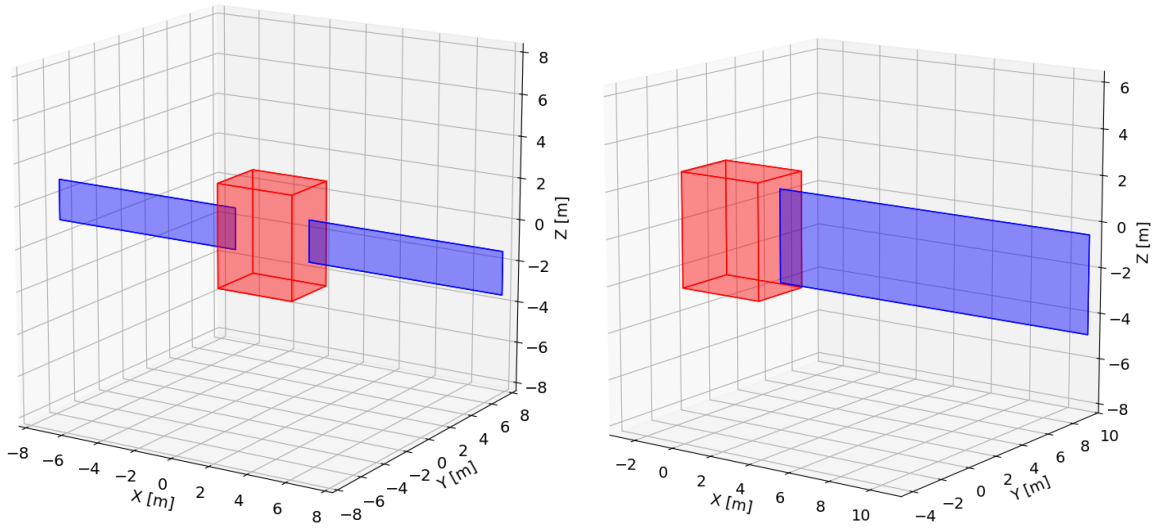
(f) Group M6 - Box-wing satellites

Model	x **[m]**	y **[m]**	z **[m]**	n_{sp}	w_{sp} **[m]**	h_{sp} **[m]**
m6	4.0	3.0	5.0	2	10.0	2.0
m61	2.8	0.1	4.6	1	8.0	2.8
m62	2.0	2.0	6.5	2	8.0	2.8
m63	4.0	3.0	5.0	1	15.0	4.0

There are two reasons the sphere, cuboid, rocket body and box-wing satellite groups are included in the simulations. The first is that all these object types are present in LEO, and could be objects that need to be characterised. Therefore it makes sense to include these groups now, so the testing mimics how the algorithms could be used in a real-life scenario. The second reason is that these shapes types are all distinct from each other, so the algorithms should be able to distinguish between them and identify the true shape and dimensions.

To further increase the realism of the simulations, it is assumed that the cuboids have two sides covered in solar panels, similar to how real cuboid satellites have sides covered in solar panels. It was decided to select the facets facing in the positive and negative y-direction of each cuboid for this. This means these facets have BRDF coefficients $R_{\text{spec}} = 0.9$ and $R_{\text{diff}} = 0.1$, as mentioned previously.

Lastly it needs to be explained how the box-wing satellite shapes are defined. The box part of these shapes are just cuboids, defined by their x-y-z dimensions. Next the number of solar panels n_{sp} can be set to 1 or 2. If this number is 2, solar panels are added to the center of the facets facing in the positive and negative x-direction. If n_{sp} is 1 only a solar panel to the center of the positive x-facing facet is added. The solar panels are assigned no thickness, only a width w_{sp} in x-direction and height h_{sp} in z-direction. An example of both a model with two solar panels (m6) and a model with only one panel (m63) are shown in Figure 5.1.4.



(a) Box-wing model m6 with two solar panels.

(b) Box-wing model with one solar panel.

Figure 5.1.4: Two box-wing satellite shape models, with $n_{sp} = 2$ and $n_{sp} = 1$. Cuboid shown in red, solar panel in blue.

The solar panels consist of two facets each, one facing in the positive y-direction and the other in the negative y-direction. The facets facing in the positive direction were chosen as the front of the solar panels, and are therefore assigned coefficients $R_{spec} = 0.9$ and $R_{diff} = 0.1$. The negative facing facets are assigned equal coefficients of 0.5, as it is assumed the backsides are not as specularly reflective as the front.

5.1.4. Simulation parameters

Next to modelling the different objects, the simulation parameters of the test cases need to be defined. These parameters are the spin cases on which the algorithms will be tested, the initial guess that is used at the start of the simulations and the dates on which the simulations performed.

Spin cases

Since space debris can have any attitude state possible, it is necessary to test the performance of the algorithms on different attitude states. For this three distinct cases were made, named **Nadir pointing**, **Single-axis spin** and **Tumbling**. The first is a case where the object has no body-axis rotational velocity relative to the orbital frame, and is therefore in a nadir pointing attitude state. This is a common attitude state for active satellites, and is therefore a relevant case to examine. The next case is where the object has a rotational velocity along one of its body axes. In the last case the object is given a rotational velocity along all three body axes, causing the object to tumble through space. For all three cases the objects are given an initial Euler 313 rotation of 10° on all three Euler angles. The subsequent rotational velocity values for each spin case are given in Table 5.1.2 below.

Table 5.1.2: Body-axis rotational velocity values for different spin cases.

Spin case	ω_1 [rad/s]	ω_2 [rad/s]	ω_3 [rad/s]
Nadir pointing	0.0	0.0	0.0
Single-axis spin	0.1	0.0	0.0
Tumbling	0.1	0.1	0.1

Initial guess

One of the research questions that these test cases will try to answer is RQ-1: *What is required for the initial guess to ensure good convergence of estimation?* This question can not be analysed without a starting point. As a start the first simulations are given an initial guess that is not too far from the true states, so that convergence is likely. The results from these simulations will then be used to investigate the requirements of the initial guess.

For all spin cases the objects will be given an initial guess that is a fixed offset from the true orbit and attitude states. For the orbit the positional states are each given an offset of 1 km, and the velocities an offset of 1 m/s. The attitude states are given an offset of 2.5° for each Euler angle axis. The rotational velocities are given an offset of 0.01 °/s on each body axis.

Dates
The simulations will be run on three different dates, for a period of 1 hour. Because the different objects are in different orbits (defined in Chapter 3) they will be visible at different times of the day. The dates and starting times of the simulations for each objects are given in Table 5.1.3. The dates are formatted as **day-month-year**, and the times are given in **hours**:**minutes** UTC.

Table 5.1.3: Dates and starting times of simulations of different test objects.

	07-09-2024	**12-10-2024**	**14-11-2024**
Object	**Time [hr:min]**	**Time [hr:min]**	**Time [hr:min]**
CryoSat 2	08:57	06:00	04:39
Swarm A	08:08	05:05	02:38
Swarm B	13:35	09:52	06:39
Swarm C	20:20	17:15	01:04
ACS3	09:10	19:01	19:28
BlueWalker 3	16:42	05:05	17:30

Three different dates are chosen for multiple reasons. The first is to check whether or not the algorithms are able to consistently find the correct shapes for each case. An algorithm might get lucky on one date and identify the correct model, but not for the other two dates, so its overall performance would be judged better than it actually is if it would only be tested on the one date. The second reason is that using multiple observation passes could be an observation strategy used in real applications to improve characterisation and estimation results.

Observer location
The Faculty of Aerospace Engineering of the Delft University of Technology, located in Delft, the Netherlands, was selected as the location of the observer from which observations are simulated. This location was selected because it serves as a realistic location where research into object characterisation takes place, so it is useful to examine whether the algorithms make this possible. The coordinates of the observer location are 51°59'24.2"N latitude, 4°22'31.1"E latitude and 0.0 m altitude.

Timesteps
The timesteps with which both the truth orbit and attitude propagations and the estimation are performed are 1.0 s. Measurements are also made available every 1.0 s. In further analysis these time steps might be changed, to see what their impact is on the performance of the models. For this especially the time step defining the rate at which measurements are generated is going to be relevant to investigate.

Covariance matrices
The initial state covariance matrix $\boldsymbol{P}_0$ and noise covariance matrix $\boldsymbol{Q}$ used in the simulation are given in Equation 5.1.1. The values are taken from the verification of the MMAE and MGS algorithms, because with these values good initial results were obtained.

$$\boldsymbol{P}_0 = \text{diag}\left[(300\text{ km})^2 * I_3 \quad (3\text{ km/s})^2 * I_3 \quad 0.2^2 * I_3 \quad (3.1 * 10^{-4}\text{ rad/s})^2 * I_3\right] \tag{5.1.1a}$$

$$\boldsymbol{Q} = \text{diag}\left[(100\text{ m})^2 * I_3 \quad (0.1\text{ m/s})^2 * I_3 \quad (2 * 10^{-4})^2 * I_3 \quad (10^{-12}\text{ rad/s})^2 * I_2 \quad (10^{-5}\text{ rad/s})^2 * I_1\right] \tag{5.1.1b}$$

Similar to the initial guess, the measurement noise is related to a research question, namely RQ-2: *What are requirements on measurement data (e.g. rate, noise, variability) to ensure correct estimates?* The impact of the measurement noise will thus be further investigated. As a starting point the same measurement noise covariance matrix $\boldsymbol{R}$ as used in the aforementioned verifications will be used. For the apparent brightness measurements a covariance value of 0.1 will be used. For the orbital position measurements covariance values of 100 m for the range and $5 * 10^{-6}$ rad for the azimuth and elevation will be used, the latter of which is close to 1 arcsecond.

5.2. Testing strategy

Now that the testing cases have been defined, the strategy of how testing will be performed and analysed is discussed next. The strategy consist of three different parts, which will be explained in this section.

5.2.1. Performance metrics

An important aspect to define first is how the performance of the different algorithms is going to be evaluated. The goal of the algorithms is to identify the true shape model used to generate the measurements, and estimate the attitude of the object with low errors. Because of this the algorithms will be evaluated by the performance metrics of root mean square error (RMSE) for the estimates of the attitude in Euler angles, as well as whether or not the true model was identified.

The latter metric is deemed to be the most important one. It was previously seen that it is possible for an algorithm to identify the correct model in the end, but first identify an incorrect model. The identifying of the incorrect model leads to the attitude estimation diverging far from the truth. If the correct model is still identified in the end, it is possible to run the estimation with just this model, which can then be expected to provide an estimation with lower errors.

5.2.2. Fixed versus variable estimation

As a baseline for the testing the test cases will be run on the MMAE, MGS and MGen algorithms. It will be checked for which cases algorithms are able to correctly identify the true shapes and converge on the true states, and why these cases and not the others. In this part of the testing the three algorithms can be directly compared to one another as well, to see where the strengths and weaknesses of the algorithms lie.

Here it will also be investigated if there are any changes that can be made to the algorithms themselves in order to improve the results. These changes could be changing the thresholds used in the MGS algorithm, or the removal of the fusion of attitude estimates in the MMAE algorithm. This part of the testing is directly related to the first research question, RQ-1: *How can the limitations of using a fixed model bank in estimation be overcome?*

5.2.3. Initial guess dependency

After the baseline testing of the algorithms it will be investigated what the impact of changing the initial guess of the estimation is on the results. This will be performed in two ways. First it will be investigated what the influence is of the initial attitude guess on the estimation results, while keeping the rotational velocities the same. This analysis is done because it is not always possible to make a guess of the initial attitude that is close to the truth. It is therefore important to analyse what happens when the initial guess is further off from the truth.

Likewise, in the second part of the analysis the impact of changing the initial rotational velocity guess will be examined. Here the initial attitude guess will be kept fixed, in order to be able to independently compare the impacts of the two aspects. The impact of the orbital parameters initial guess will not be investigated. The reason for this is that it is much easier to generate an accurate guess for the orbital positions and velocities from range and angles data, and it is more important to analyse the effects of the attitude and rotational velocities. Investigating the results on these two aspects should give a better understanding of the limitations of the algorithms, and give an answer to RQ-2: *What is required of the initial guess to ensure good convergence of estimation?*.

5.2.4. Measurement data requirements

The final part of the testing focuses on RQ-3: *What are the requirements on measurements data (e.g. noise, rate, variability* to ensure correct estimates? Here it will be investigated how the different aspects of the measurement data like the amount of noise, the rate and variability of the data impact the quality of the estimation results. It is expected that this research question will already partially be answered in the first part of the testing (Subsection 5.2.2), as the different test cases will come with different variability in the measurement curves.

Additionally, in this part it will also be examined whether the algorithms can successfully characterise the shape and estimate the states of the test objects based on real measurement data. For some of the objects (CryoSat and Swarm A/B/C) the true orbit and attitude states are available, so with these objects it will be seen if the algorithms can be validated on the real data. For the other objects it will be interesting to see what the results of the estimations will look like.

5.3. Light curve comparison

An important aspect impacting the performance of the algorithms that was previously encountered is whether light curves of different objects are distinct from each other. If the light curves of two objects with the same orbit and attitude states look very similar, the algorithms struggle to distinguish between them, leading to poor results. It is therefore important to check how the light curves of the test objects compare to each other.

For this only the light curves of the objects in the central model group will be examined. For all three spin cases the light curves will be compared to each other. This analysis is done to support the analysis of the results in the next chapter. For each spin case the orbits of one of the six test objects was selected. For this orbit the light curves of the different objects of the central group were plotted. For the single-axis spin and tumbling cases it was decided to propagate the attitude with rotational velocities of 0.01 rad/s. If these velocities are increased the light curves become to chaotic to be able to compare the different curves to each other.

5.3.1. Nadir pointing

For the Nadir pointing cases the objects were propagated for the ACS3 orbit on 14-11-2024. The resulting light curves are given in Figure 5.3.1. In the beginning of the plot the curves of the different objects show very similar behaviour, only with different apparent brightness magnitudes. After around 7 minutes there are some peaks that distinguish the curves of the different objects more, for example the large brightness peak for the triangular prism curve, and some smaller brightness peaks in the sphere curve. At the end of the plot some curves again show similar behaviour, with magnitudes that are much closer to each other than before. Especially the cuboid and box-wing satellite curves match very closely.

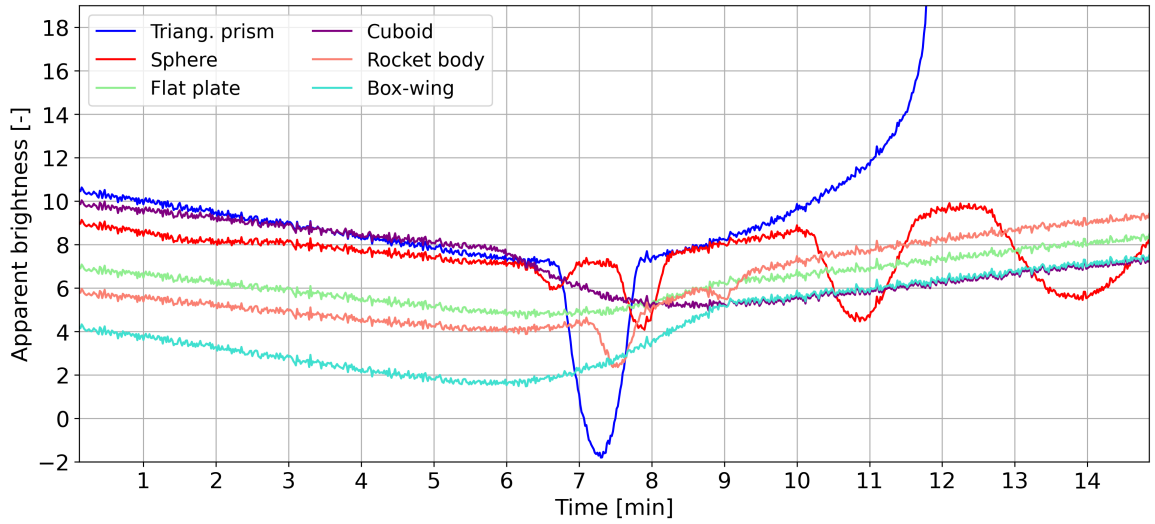


Figure 5.3.1: Comparison of light curves - ACS3 orbit 11-14-2024

The fact that the curves show similar behaviour for large part of the curves is caused by the fact that the objects do not have rotational velocities around their body-axes. This means that the only variations in the light curves are caused by the changing observation geometries caused by the objects moving through the orbits. This means overall changes in the light curve are slow, as different sides of objects only become visible gradually. The only differences between the light curves of the different objects are then purely caused by the amount of light reflected to the observer, resulting in the only major difference between the curves being the magnitudes.

The reason the cuboid and box-wing satellites initially have very different magnitudes, but at the end look very similar is because both object types are shaped like cuboids, only the box-wing satellites have solar panels. If the solar panels are in the right orientation, their large surface areas reflect a lot of light towards the observer, leading to an increased apparent brightness. However, if the orientation changes such that the panels no longer reflect light towards the observer, the box-wing satellites essentially look like cuboids from the observers' point of view. This results in the light curves of the two objects with the same orbit and attitude to look very similar.

It has been seen in the verification of the MMAE and MGS algorithms that the estimation results can suffer from light curves of different objects looking very similar, because it becomes difficult for the algorithms to distinguish between the objects, resulting in the wrong shapes being identified and the attitude estimations having larger errors. It will have to be seen in the results whether this will also happen for the test cases.

5.3.2. Single-axis spin

The Single-axis spin cases were propagated for the Swarm A orbit on 12-10-2024. The resulting curves are shown in Figure 5.3.2. It can be seen that there are more distinguishing features than in the Nadir pointing curves, with many curves having brightness peaks at different times. However, there is still some overlap present between curves of different objects. For example, the curves of the sphere and triangular prism model overlap almost exactly between 5.5 and 7.5 minutes. Between 7 and 9 minutes the rocket body and box-wing curves overlap closely as well. These similarities between the curve could negatively impact the results of the test cases.

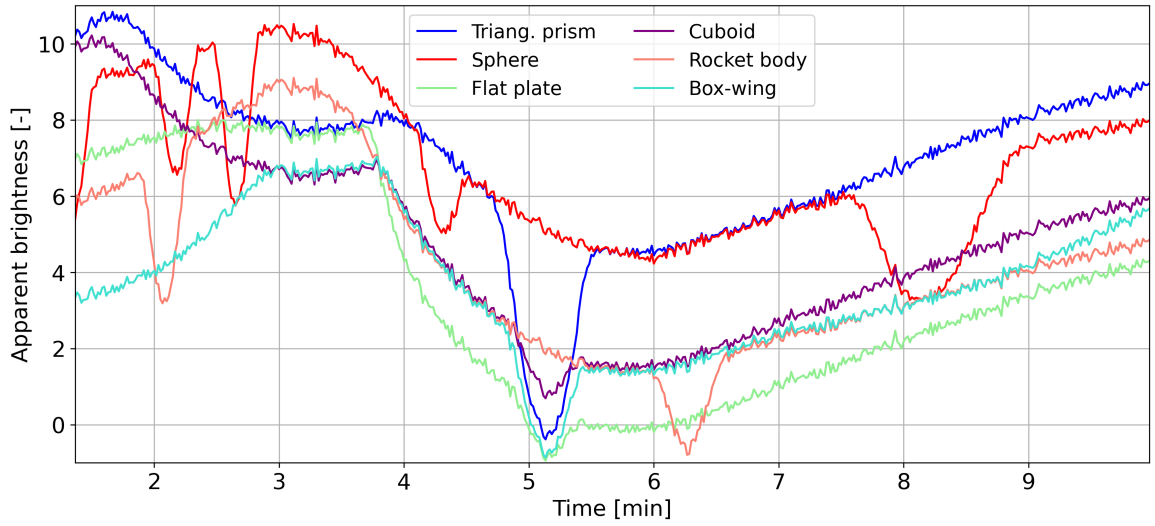


Figure 5.3.2: Comparison of light curves - Swarm A orbit 12-10-2024

5.3.3. Tumbling

For the Tumbling cases the central group objects were propagated for the Swarm B orbit on 07-09-2024. The light curves are shown in Figure 5.3.3. There are various straight lines visible in the curves. This happens because values are only plotted if they are below the visibility limit of 20. Since the objects now rotate more in different directions, it occurs more often that objects becomes less visible to the observer, attaining values larger than 20. The straight lines are a by-product of the plotting functions in Python, with the function connecting two consecutive measurements with a straight line. If an object is not visible for larger periods of time, these measurements are spaced apart, leading to the large straight lines in the plot. There are different ways of plotting, but this way shows the light curves the most clearly overall, so curves will be unchanged.

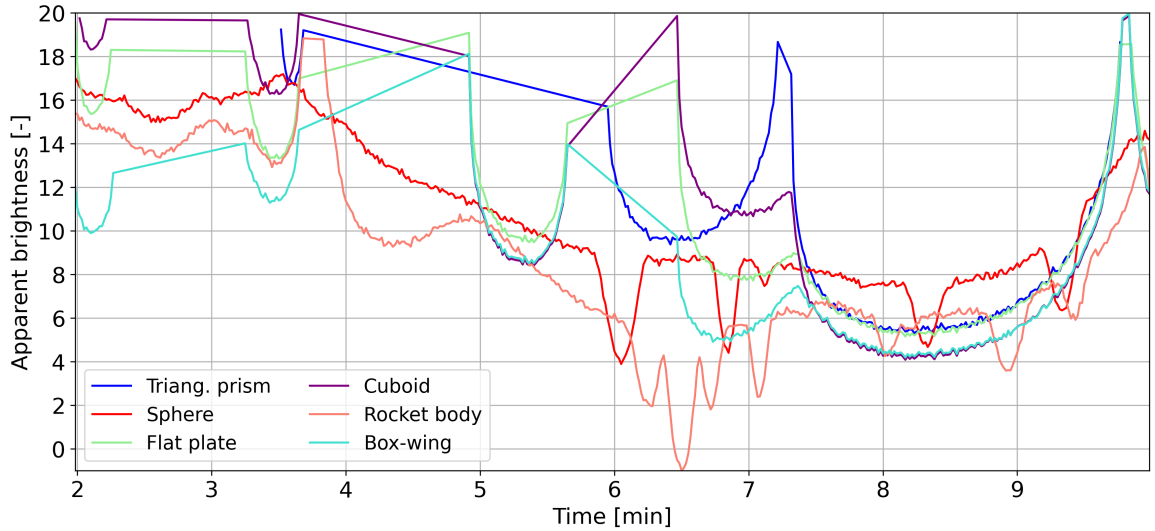


Figure 5.3.3: Comparison of light curves - Swarm B orbit 07-09-2024

In the curves themselves there are again many more distinguishing features at different points in time, caused by the objects rotating around all body-axes, resulting in more variation of the light curves for differently shaped objects. There are however still a few times where the curves overlap, for example between 7.5 and 9 minutes. Again, this might have an impact on the results in Chapter 6, which starts on the next page.

6 Results

In this chapter the results of the test cases defined in the previous chapter are presented. In Section 6.1 the results of the baseline estimations of the MMAE, MGS and MGen algorithms are discussed. Then in Section 6.2 the impact of the initial guess for a few cases is investigated. In Section 6.3 the requirements for the measurement data are looked into. Finally, in Section 6.4 a validation of the algorithms is presented.

6.1. Baseline performance analysis

As described in the previous chapter, the performance of the different algorithms is analysed based on the performance metrics of Euler angle RMSE and whether an algorithm was able to find the true shape for each case. These metrics for the baseline test case estimations are given in Appendix B.1. In the last columns of these tables it is indicated whether the algorithm was able to identify the true model for that case, or if it was able to find the right shape type but not the correct dimensions, or neither.

6.1.1. MGen

The results for the MGen algorithm in Appendix B.1.3 show that this algorithm was not able to find the true shape for any of the cases. Next to this, the attitude RMSEs are high for all cases, even the cases where the correct shape type was identified. The algorithm thus performs even less satisfactorily than during the verification. It can therefore be concluded that this algorithm is not suitable to use for the problem in its current form, and requires further development and testing to determine whether it could be a viable method, which is outside of the scope of this thesis. Further discussion of this algorithm and results are presented in Section 7.3.

6.1.2. MMAE and MGS

The results for the MMAE and MGS algorithm look much more promising, with both algorithms finding the true shapes or shape types more often. The attitude errors are also much lower for these cases, highlighting the capability of both algorithms for estimating the attitude and characterising the shape of the objects for the LEO test cases. A colour map summarising the model identification part of the results is shown in Table 6.1.1.

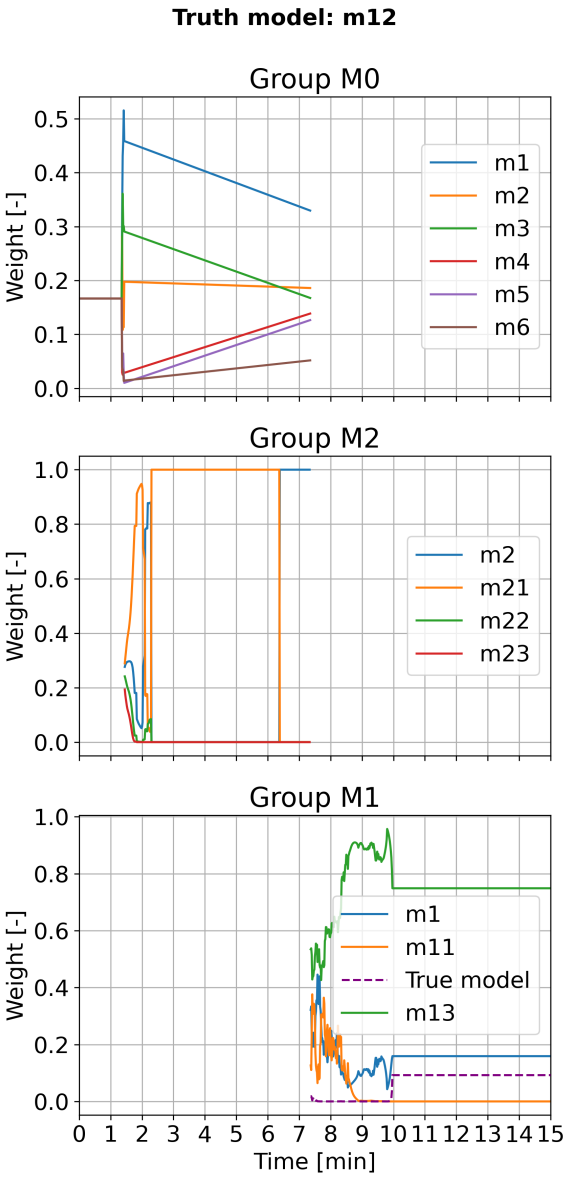
Table 6.1.1: True model identification colour map for MMAE and MGS algorithms.
Green - True shape identified. Yellow - Correct shape type. Red - Incorrect model and shape type.

		07-09-2024		12-10-2024		14-11-2024	
Satellite	**Spin Case**	**MMAE**	**MGS**	**MMAE**	**MGS**	**MMAE**	**MGS**
CryoSat 2	Nadir pointing						
	Single axis spin						
	Tumbling						
Swarm A	Nadir pointing						
	Single axis spin						
	Tumbling						
Swarm B	Nadir pointing						
	Single axis spin						
	Tumbling						
Swarm C	Nadir pointing						
	Single axis spin						
	Tumbling						
ACS3	Nadir pointing						
	Single axis spin						
	Tumbling						
BlueWalker 3	Nadir pointing						
	Single axis spin						
	Tumbling						

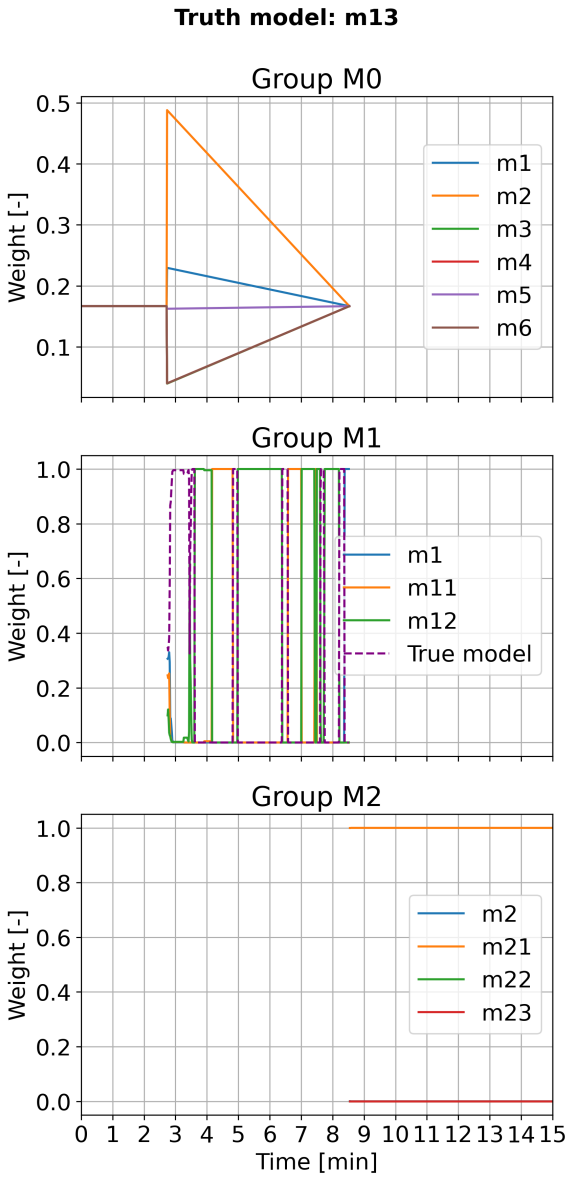
In total each algorithm ran 18 cases per date, giving a total of 54 cases per algorithm. A count shows that the MMAE algorithm was able to find the true shape in 21 of these cases, and found just the correct shape type in 6 cases. The MGS algorithm in turn found the true shape in 15 cases, and the correct shape type in 9 cases. These are quite similar results, with the MMAE algorithm performing slightly better. However, the MMAE algorithms has an average runtime of 16 minutes, while the MGS algorithm has an average runtime of 3 minutes. The MGS algorithm thus gives similar results to the MMAE algorithm much faster.

Looking more in-depth at results for the individual cases found that in a lot of cases where the MGS algorithm identified the incorrect model group, it incorrectly identified group M2, the group consisting of spherically shaped models. This is an interesting results, as there is no immediately clear reason why the MGS algorithm would consistently choose this model group over the others. Further investigation into this behaviour is therefore warranted.

In Figure 6.1.1 the weights for two cases where the algorithm incorrectly identified and activated group M2 are shown. Figure 6.1.1a shows the Swarm A - Single-axis spin case on 12-10-2024, and Figure 6.1.1b the CryoSat 2 - Single-axis spin case on 07-09-2024. In the Swarm A case the algorithm first incorrectly identifies group M2, but eventually correctly identifies group M1. In the CryoSat 2 case initially M1 is correctly identified, but eventually the algorithm incorrectly switches to group M2.



(a) Weights for Swarm A - Single-axis spin - 12-10-2024



(b) Weights for CryoSat 2 - Single-axis spin - 07-09-2024

Figure 6.1.1: Weights of MGS algorithm for Swarm A & CryoSat 2 - Single-axis spin cases.

The reason the algorithms incorrectly identify the sphere models is most likely because of what was found in Section 5.3. It was seen that there are multiple times where light curves of different objects in the central model group look very similar, or even overlap completely. When this happens the algorithms have no way of distinguishing between the models, which increases the chances that they incorrectly identify one of the spherical models as the true shape.

The spherical models in group M2 were added to the model bank to act as dummy models, to test whether the algorithms are able to correctly identify these models as incorrect. However, it seems that the inclusion of the spherical models leads to less optimal results, as the models confuse the algorithms. In the ideal case the spherical models would be included in the model bank. However, it was decided to remove the spherical models from the model bank, and run the test cases again. This makes it possible to better examine the performance of algorithms, and investigate what further improvements can be made. The raw results for the estimations without the spherical models can again be found in Appendix B.2. The resulting colour map for the shape identification for these cases is shown in Table 6.1.2.

Table 6.1.2: True shape identification colour map for MMAE and MGS algorithms. Sphere models excluded from model bank. Green - True shape identified. Yellow - Correct shape type. Red - Incorrect model and shape type.

		07-09-2024		12-10-2024		14-11-2024	
Satellite	**Spin Case**	**MMAE**	**MGS**	**MMAE**	**MGS**	**MMAE**	**MGS**
CryoSat 2	Nadir pointing						
	Single axis spin						
	Tumbling						
Swarm A	Nadir pointing						
	Single axis spin						
	Tumbling						
Swarm B	Nadir pointing						
	Single axis spin						
	Tumbling						
Swarm C	Nadir pointing						
	Single axis spin						
	Tumbling						
ACS3	Nadir pointing						
	Single axis spin						
	Tumbling						
BlueWalker 3	Nadir pointing						
	Single axis spin						
	Tumbling						

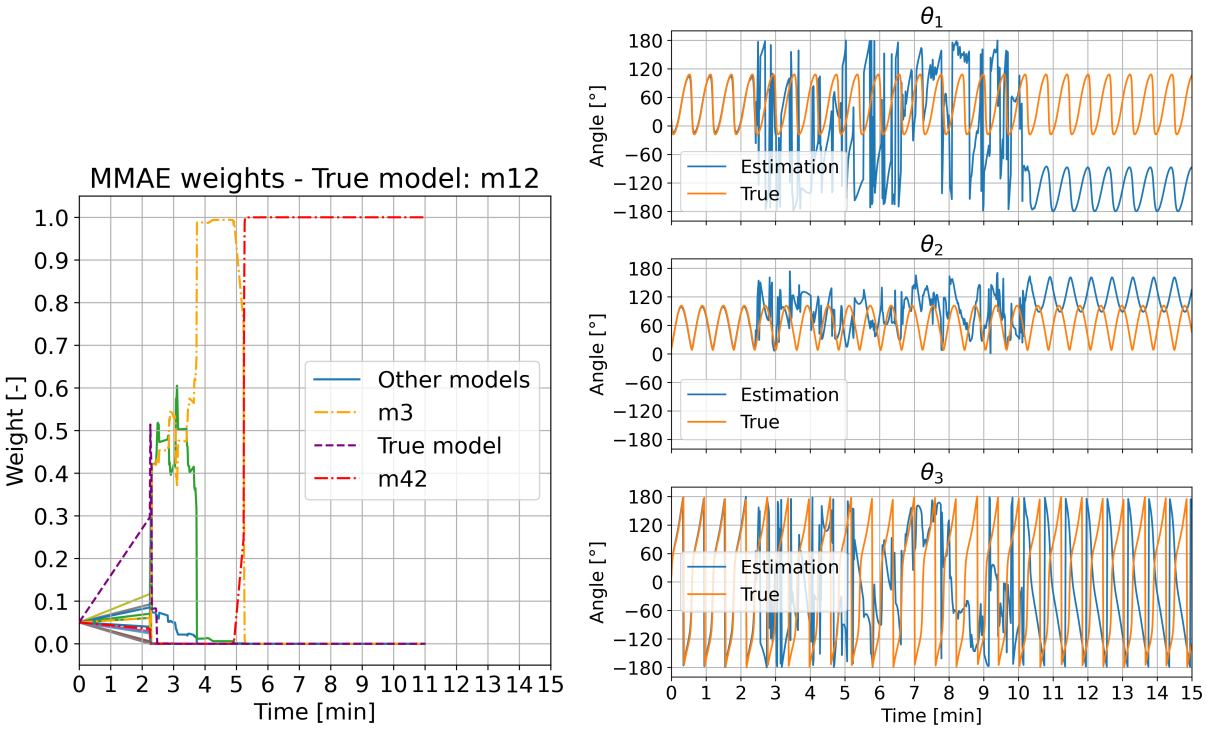
There are now a few more cases coloured green. The MMAE algorithm now identifies the correct shape in 24 cases. In 8 cases it finds only the correct shape type. The MGS algorithm finds the correct shape in 16 cases, which is only one more than previously, and in 7 cases the correct shape type. The exclusion of the spherical models thus slightly improved the results for both algorithms. However, there is still room for improvement. In order to see what further adjustment could be made to improve the results, a few of the test cases are analysed in more detail.

6.1.3. Detailed case analysis

Before cases can be analysed in detail, it needs to be decided which cases are going to be investigated. Looking back at Table 6.1.2, it can be seen that the Swarm C - Tumbling cases have inconsistent results for the three different dates. On 07-09-2024 the MMAE algorithm finds the correct shape, but the MGS algorithm not. On 12-10-2024 both algorithms do not find the correct shape or shape type, and on 14-11-2024 the MGS algorithm finds the correct shape, but not MMAE. It will be interesting to see why this happens. For this the cases on 12-10-2024 will be examined.

MMAE - Swarm C - Tumbling - 12-10-2024

The weight and Euler angle attitude for the MMAE algorithm are shown in Figure 6.1.2. In the first few minutes no measurements are available yet, so the attitude is propagated from the initial guess. Just after 2 minutes the measurements are available. In Figure 6.1.2a it can be seen that initially the correct shape is assigned a high weight. However, quickly after this the algorithm assigns high weights to incorrect models, and does not assign the correct shape a high weight again. At the same time as this happens the attitude estimation in Figure 6.1.2b becomes very chaotic, and completely diverges from the true attitude states.



(a) MMAE weights.

(b) True and estimated Euler angle attitude states.

Figure 6.1.2: MMAE weights and Euler angle attitude for Swarm C - Tumbling on 12-10-2024.

The reason the attitude estimation becomes so chaotic is because the estimated attitude states in the MMAE algorithm are fused estimates of all the estimated attitude states of all shape models included in the model bank. This means that if an incorrect shape model is assigned a high weight, its attitude estimate will start to dominate the attitude estimation. This is exactly what happens in Figure 6.1.2. Because the MMAE algorithm assigns high weights to multiple incorrect shapes, the algorithm tries to fit the measurements to multiple wrong attitude estimations, causing the overall estimation to diverge.

A possible adjustment to the MMAE algorithm that could solve this problem is to remove the fusion of the attitude states, and have a separate attitude estimation for each shape in the model bank. When an incorrect model diverges in attitude estimate from the truth, the other estimations are not affected by it. This could allow the algorithm to also find the correct shape much easier, and lead to better attitude estimations.

MGS - Swarm C - Tumbling - 12-10-2024

For the MGS algorithm the weight for the case are shown in Figure 6.1.3. It can be seen that as soon as measurements are available, the algorithm activates group M3, which is an incorrect group. The algorithm stays in this group until around 9.5 minutes. The algorithm then switches back to central group M0, only to quickly activate group M4 for the rest of the estimation, which is also an incorrect model group.

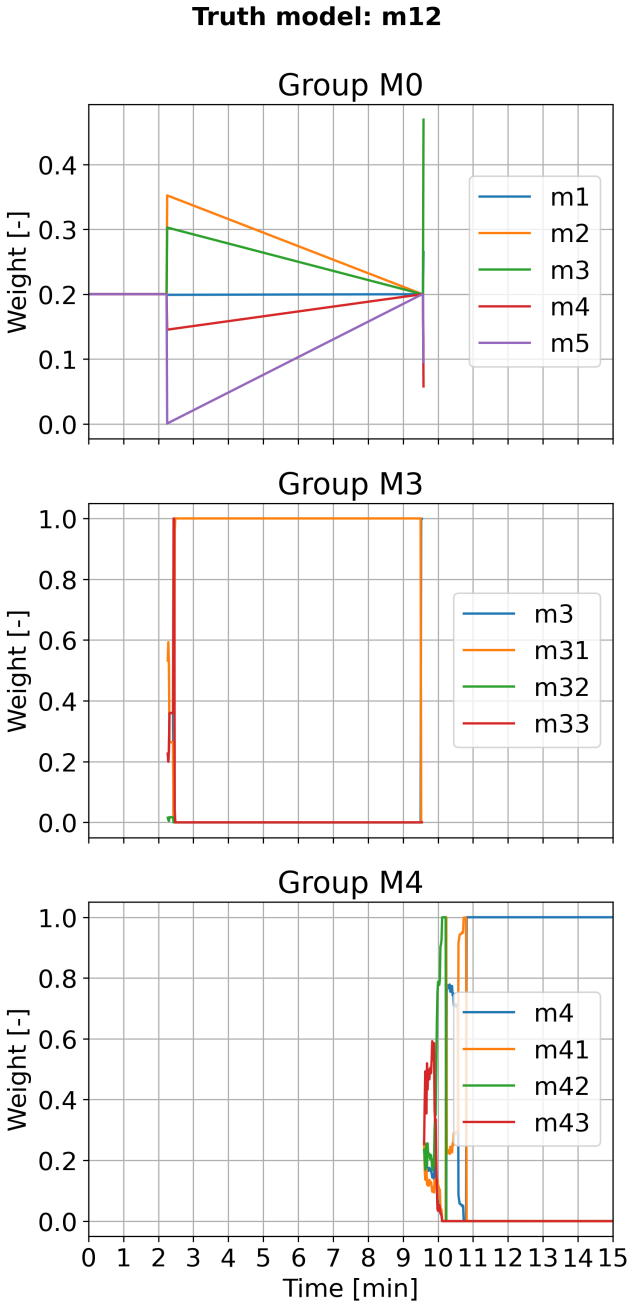


Figure 6.1.3: MGS weights for Swarm C - Tumbling on 12-10-2024.

It seems that the problem with the algorithm lies in the activation of model groups from group M0. The MGS algorithm does not keep group M0 active for very long after measurements are available, but quickly switches to one of the specific shape type groups. Even when the algorithm switches back to group M0, it quickly activates another group. This means the algorithm does not use a lot of information to make decisions to switch groups, which results in it more often activating an incorrect group than the correct group.

A way to adjust the algorithm to avoid this early switching could be to increase the weight threshold T_0. This threshold determines the weight a model in group M0 needs to have before the algorithm switches to the shape type group corresponding to that model. In the initialisation of the MGS algorithm this threshold was set to 0.5. This might be too low of a value, causing the behaviour as seen in Figure 6.1.3 to occur.

6.1.4. Algorithm improvements

Based on the case analysis in previous subsection it was decided to test whether certain changes to the MMAE and MGS algorithms can improve the shape identification and attitude estimation results from Subsection 6.1.2. The updated algorithms will then be run for the same test cases as before. Since the exclusion of the sphere models from the model banks significantly improved the results, these models are still excluded from the model bank in the following estimations. The algorithms will be compared using the shape identification and average Euler angle attitude RMSE results.

MMAE

For the MMAE algorithm the problem is that the attitude states of all the different shapes in the model bank that are run in parallel are fused to one state estimation by weighted averaging. Because the shapes in the model bank are very different from each other, the attitude estimate for each shape can differ from each other by large amounts. When these varying attitudes are fused together the overall estimate can then diverge too far from the truth, causing the algorithm to be unable to find both the correct attitude and identify the true shape. The MMAE algorithm is therefore adjusted so that this fusion of attitude states is removed. This means each model in the model bank retains its own attitude estimation. An important note is that the state fusion for the orbital states is kept in the algorithm. This is done because the orbital position measurements affect the brightness measurements much less directly, so using the averaged orbital position states would not lead to significant changes.

MGS

The issue for the MGS algorithm is that it will activate an incorrect model group in the beginning of the estimation and not switch to any other groups. There are multiple possible algorithm adjustments that could address this problem. The adjustment that is implemented is regarding the threshold T_0. This threshold determines when the algorithm initially switches from group M0 to any of the other groups. This threshold was initially set to 0.5, meaning if any models in group M0 attained a weight of 0.5 or higher the subsequent model group of this shape type is activated. Since this initial activation of the wrong model group is a recurring issue, it was decided to increase this threshold to a value of 0.9. This means the first activation of a model group should take longer, as a better fit to the measurements is required for any model in group M0 to attain a weight of 0.9 or higher.

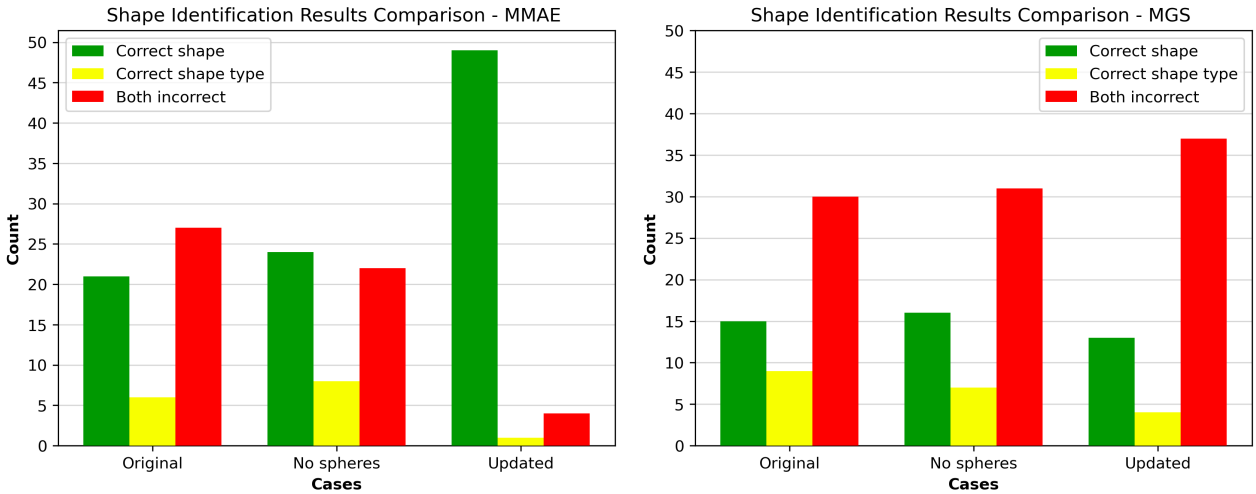
Shape identification

The raw results for the estimations with updated algorithms are found in Appendix B.3 The results for the true shape identification of are shown in Table 6.1.3.

Table 6.1.3: True shape identification colour map for MMAE and MGS algorithms. Sphere models excluded from model bank.
MMAE without attitude fusion, MGS with $T_0 = 0.9$
Green - True shape identified. Yellow - Correct shape type. Red - Incorrect model and shape type.

		07-09-2024		12-10-2024		14-11-2024	
Satellite	**Spin Case**	**MMAE**	**MGS**	**MMAE**	**MGS**	**MMAE**	**MGS**
CryoSat 2	Nadir pointing						
	Single axis spin						
	Tumbling						
Swarm A	Nadir pointing						
	Single axis spin						
	Tumbling						
Swarm B	Nadir pointing						
	Single axis spin						
	Tumbling						
Swarm C	Nadir pointing						
	Single axis spin						
	Tumbling						
ACS3	Nadir pointing						
	Single axis spin						
	Tumbling						
BlueWalker 3	Nadir pointing						
	Single axis spin						
	Tumbling						

For a clear comparison between the previous cases, the counts of how many times each algorithm was able to identify the correct shape, shape type or neither are visualised in Figure 6.1.4.



(a) MMAE **(b)** MGS

Figure 6.1.4: Shape identification results comparison for different runs of MMAE and MGS algorithms.

It is clear that the MMAE results have significantly improved. MMAE is now able to correctly identify the true shape in 49 out of the 54 cases, which is almost all of them. It is only fully incorrect in 4 cases. The removal of the fusion of attitude states can therefore be concluded to be a good improvement on the MMAE algorithm seen in literature, like the paper by Linares et al. [23].

In contrast, the increase of threshold T_0 in the MGS algorithm does not improve the results, but decreases the amount of times the algorithm finds the correct shape. The conclusion is made that further research and development is required in order to fine-tune the workings of the MGS algorithm, so that the issues encountered

with the algorithm are solved, and the results are similar to or better than those of the MMAE algorithm. This research is outside of the scope of this thesis, and will be added to the recommendations in the next chapter.

Attitude estimation

In order to compare the attitude estimation performance between the results for the Original, No spheres and Updated cases, the Euler angle attitude RMSE values per spin case are averaged for all three dates over all three angles. For the MMAE algorithm this plot of averages is shown in Figure 6.1.5. Every point on the plot is the average of 9 RMSE values, for each spin case three angles and three estimation dates. The data is plotted in this way so that the overall performance of the different cases can be compared to each other. For clarity the spin cases have been abbreviated to NPT (Nadir Pointing), SAS (Single Axis Spin) and TBL (Tumbling).

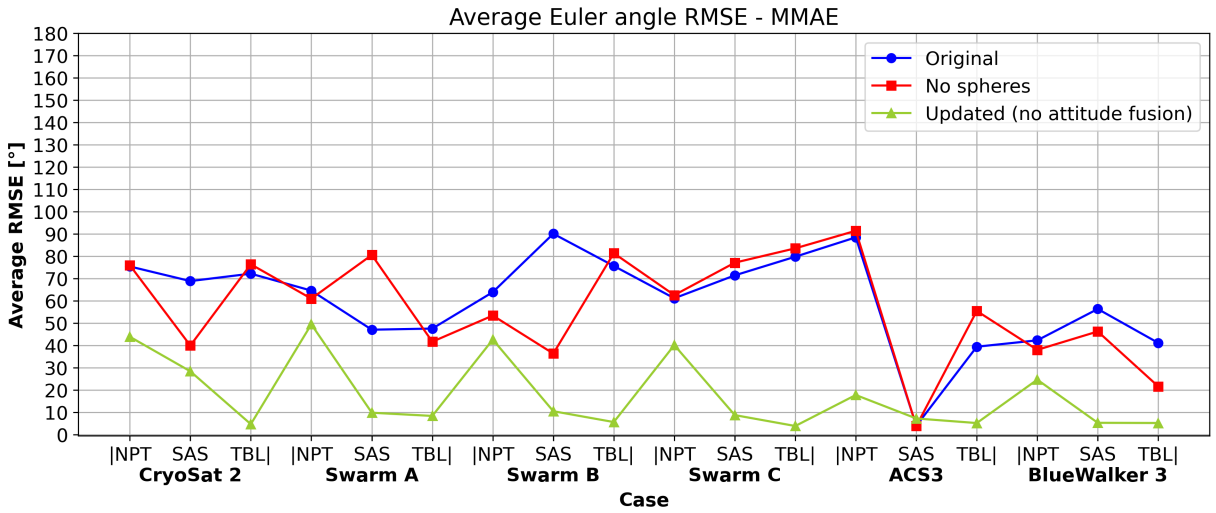
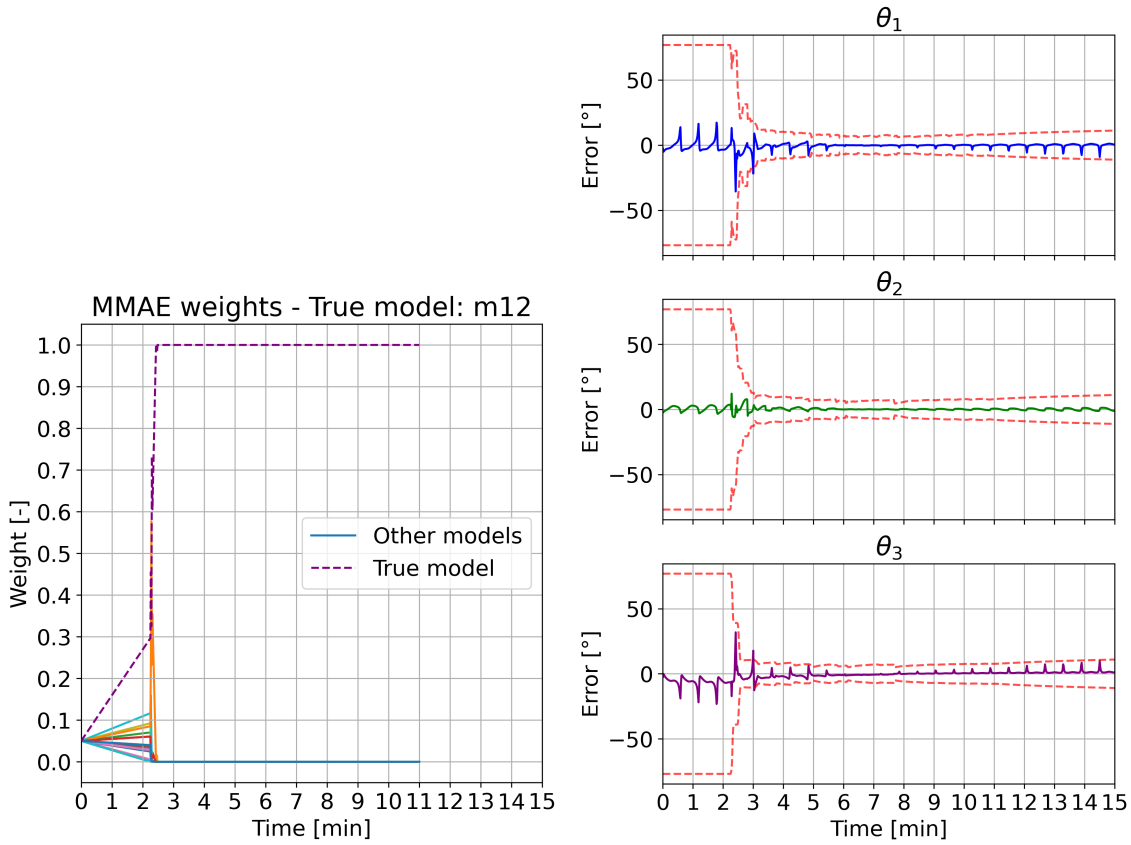


Figure 6.1.5: Averaged Euler angle attitude RMSE for different cases - MMAE

It can be seen that while the No spheres errors on average are quite similar to the Original values, the Updated algorithm performs much better in the attitude estimation. This improved performance is illustrated in the plots for Swarm C Tumbling on 12-10-2024 shown in Figure 6.1.6.



(a) MMAE weights.

(b) Euler angle attitude error.

Figure 6.1.6: MMAE weights and Euler angle attitude errors for Swarm C - Tumbling on 12-10-2024 for updated algorithm.

Comparing the results to those obtained earlier in Figure 6.1.2, it can be seen that the attitude estimation has indeed improved significantly. The algorithm is able to converge very closely on the true attitude, with a total attitude RMSE of [2.9°, 1.5°, 3.7°] for the Euler angles θ_1, θ_2 and θ_3, respectively. It is clear that the removal of attitude state fusion from the MMAE algorithm is an effective improvement to the algorithm. In the weights plot in Figure 6.1.6a the algorithm also sticks to the correct model after identifying it, and does not switch.

The average RMSE values for the MGS algorithm are plotted in Figure 6.1.7. It can be seen that the algorithm with threshold T_0 changed to 0.9 overall also performs worse in the attitude estimation, just like it performs worse in the shape identification. Overall the MGS algorithm performs similarly for all three cases.

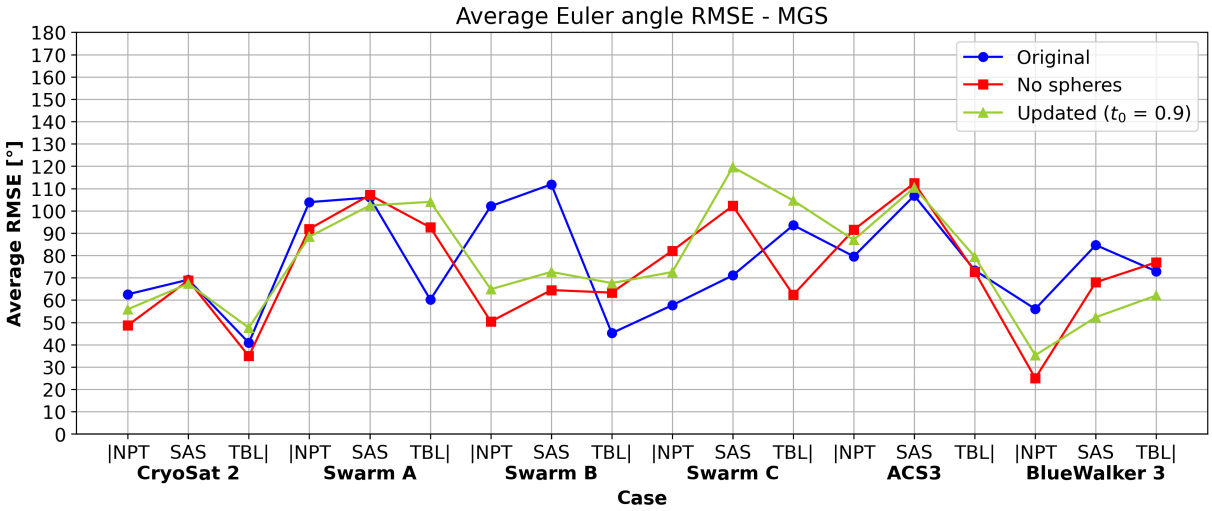


Figure 6.1.7: Averaged Euler angle attitude RMSE for different cases - MGS

6.2. Initial guess dependence

The first part of the results analysis focused on the performance of the different algorithms on the baseline test cases as defined in Chapter 5. It was seen that after a few improvements the MMAE algorithm performs very well, identifying the true shape in 49 out of 54 cases and estimating the attitude with low errors for these cases. However, this analysis only shows part of the picture. The cases were setup to start with an initial guess for the attitude states with a 2.5° offset from the true initial attitude. This is a small error in the initial guess, while in reality most of the time the true initial state is not known to such accuracy, or might not be known at all. It is therefore important to investigate what happens to the performance when the initial guess errors become larger.

For this analysis only the MMAE algorithm is looked at. This is done because this algorithm performs much better than the MGS and MGen algorithms, so it makes more sense to look into this algorithm in more detail. Secondly, the main research question is focused on the improvements of existing techniques for object characterisation and attitude estimation. Since the MGS and MGen algorithms were implemented for this thesis, they do not count as existing methods. Thus, to answer the main research question only the MMAE algorithm is further analysed.

The analysis for the initial guess dependence of the MMAE algorithm is split into two parts. In the first part the influence of the attitude initial guess is examined. In the second part the impact of the rotational velocity initial guess is looked at. This is done so the effect of the two state parameters can be investigated independently. The effect of the orbital states initial guess is not investigated, because these parameters can be estimated accurately from the angles and range measurements. This means that the initial errors for the orbital parameters will not be large, and it is not necessary to perform the investigation for these states.

6.2.1. Attitude

In the baseline test cases the initial attitude guess was set to be equal to the true initial attitude with an offset of 2.5° for all three Euler angles. For the initial guess dependence analysis estimations will be run with different values of the initial attitude offset, both in the positive and negative direction. The offset values that are investigated range from -30° up to and including +30°, in steps of 2.5°. The initial rotational velocities are kept at the same offset of 0.01 °/s from the true rotational velocities, just like for the baseline estimation.

For each test object the average Euler angle RMSE will be plotted against the value of the initial attitude offset. An important note is that an RMSE value for a case is only plotted if the MMAE algorithm finds the true shape for that test case. This is done because the RMSE values are generally much higher when the algorithm gets the shape incorrect, which would give a skewed view of the results if included in the plots.

Additionally, because many estimations had to be run and time was limited, the estimation were run with a measurement data rate 5.0s, while the estimations were run with a time step of 1.0s. This means during the estimation the UKFs in the MMAE algorithm perform 5 prediction steps before making a correction to the state estimations. In Subsection 6.3.2 it is shown that estimations with a measurement data rate of 5.0s still produce results with low errors, making it possible to run the estimations in this section with this time step.

07-09-2024

Nadir pointing

The average RMSE for different values of the initial attitude offset for the Nadir pointing case of 07-09-2024 is shown in Figure 6.2.1. What is interesting is that the algorithm finds the correct shape much more often for the negative offsets for this case. There seems no immediately clear reason as to why this is the case. For the rest it can be seen that generally the average RMSE errors increase when the offsets from the true initial attitude become larger. This is entirely in line with expectations, with it being more difficult for the algorithm to properly converge on the correct attitude when the initial guess is farther from the true initial states.

It can be seen that the errors for the CryoSat 2 satellite are much higher than for the other satellites. It is also the only satellite for which the MMAE algorithm was not able to find the correct shape for any of the positive offset values. Looking back at Table 6.1.3, it can be seen that this combination of satellite and spin case was one of the few cases where the algorithm did not find the correct shape. It appears that the measurements for the CryoSat 2 satellite for the Nadir pointing case are difficult for the algorithm to invert to good results. Still, for the negative offset values it was able to find the correct shape for almost all values, albeit with high attitude errors.

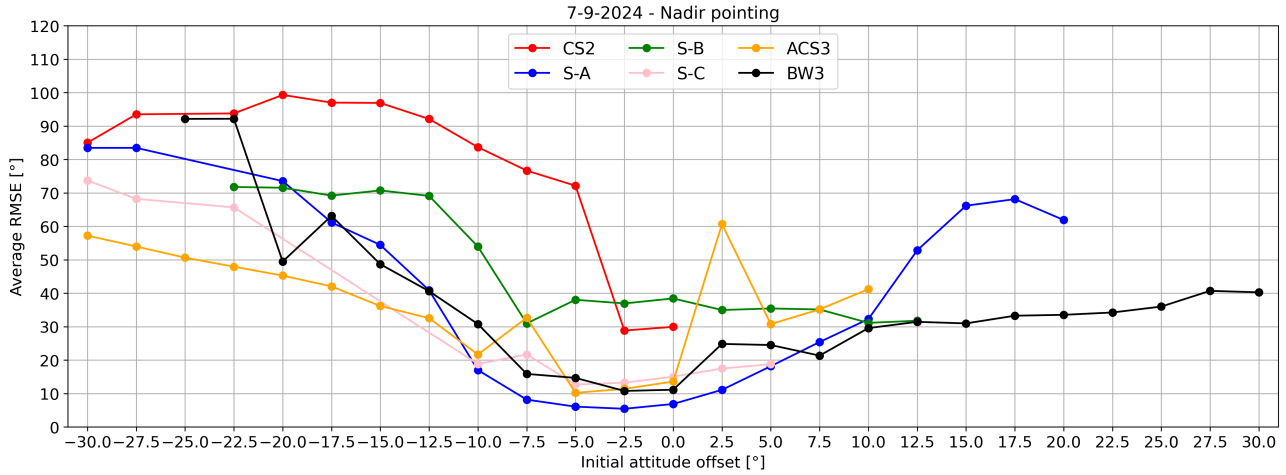


Figure 6.2.1: Average Euler angle attitude RMSE for different values of initial attitude guess offset - Nadir pointing - 07-09-2024

Single-axis spin

The RMSE plot for the Single-axis spin case is given in Figure 6.2.2. For this spin case the algorithm finds the correct shape a similar amount of times for both positive an negative offsets. The errors are slightly lower than those for the Nadir pointing case in Figure 6.2.1.

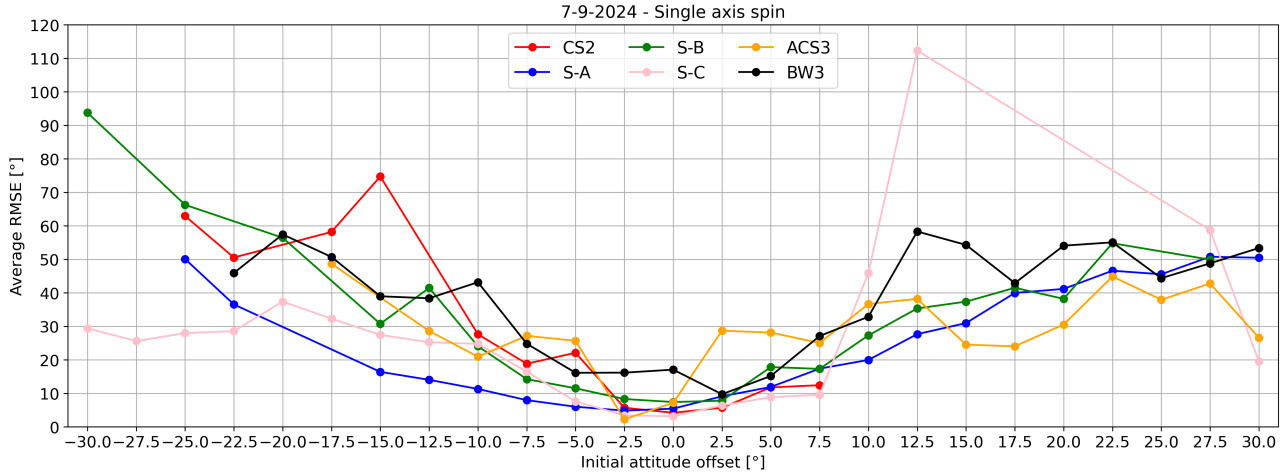
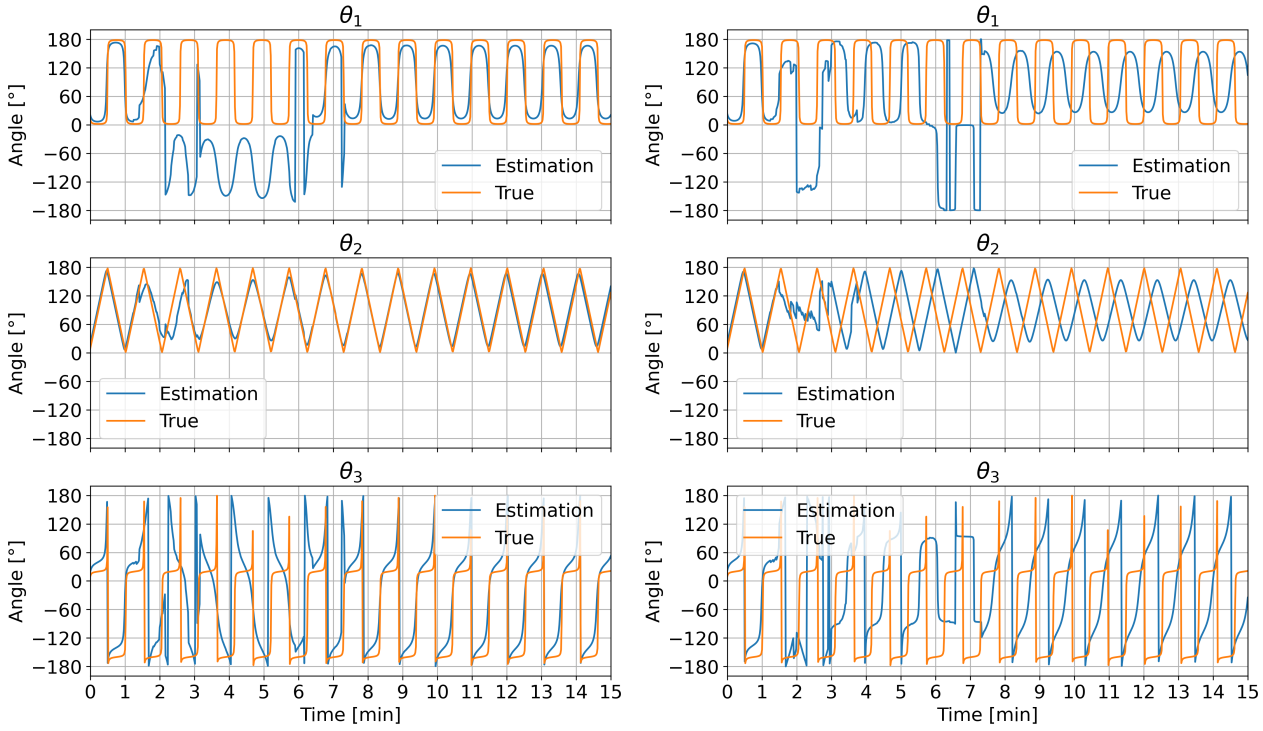


Figure 6.2.2: Average Euler angle attitude RMSE for different values of initial attitude guess offset - Single-axis spin - 07-09-2024

In the plot one outlying results is noticeable, namely the error for the Swarm C satellite at a +12.5° offset. For an offset of +10.0° the average RMSE error for this satellite is around 45°, but then the error jumps to over 110°. After this the algorithm does not find the correct shape for increased offset values until 27.5°, where the error has dropped down to 60°. To explain why this happens the Euler angle attitude plots for the offset values of 10.0° and 12.5° will be examined. The plots are shown in Figure 6.2.3.



(a) Offset value 10.0° **(b)** Offset value 12.5°

Figure 6.2.3: Truth and estimated Euler angle attitudes for initial attitude offset values of 10.0° and 12.5°.

In both plots it can be seen that when measurements become available the attitude estimation diverges from the true attitudes. However, for the 10.0° offset in Figure 6.2.3a the algorithm is able to mostly correct the estimation, resulting in an estimation that is very close to the true attitude. For the 12.5° offset this is not the case. The algorithm is not able to correct the estimation, and converges on incorrect attitude values. It has been seen at multiple times during the thesis that this is a matter of luck. Sometimes the MMAE algorithm is able to correct estimations that are diverged from the truth by a lot, and other to times not. This specific case is one example of this, and can be treated as an outlying result in the overall analysis of the initial guess dependence results.

Tumbling

The average RMSE error plot for the Tumbling cases on 07-09-2024 is shown in Figure 6.2.4. This plot shows very consistent results, with errors increasing gradually with higher offset values, and no extreme outliers.

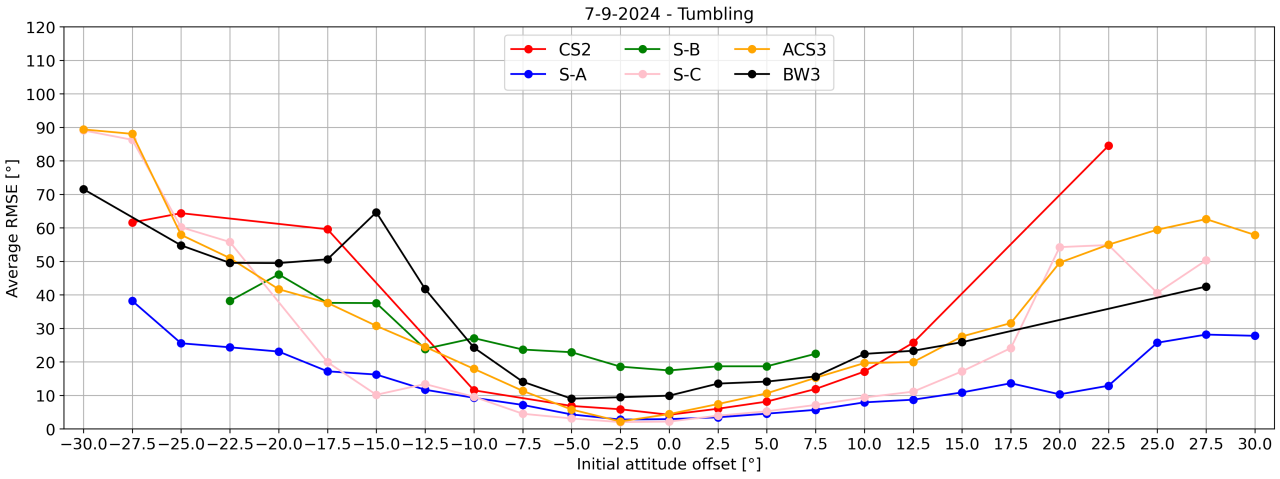


Figure 6.2.4: Average Euler angle attitude RMSE for different values of initial attitude guess offset - Tumbling - 07-09-2024

12-10-2024

Nadir pointing

The average RMSE plot for the Nadir pointing cases on 12-10-2024 is given by Figure 6.2.5. It can be seen that now the algorithm finds the correct shape for positive offset values much more often than in Figure 6.2.1. The errors for positive offsets are also slightly lower than those for negative offsets, with the exception of the Swarm A satellite, which has higher errors than all other satellites for all positive offset values. Overall the results conform to expectations.

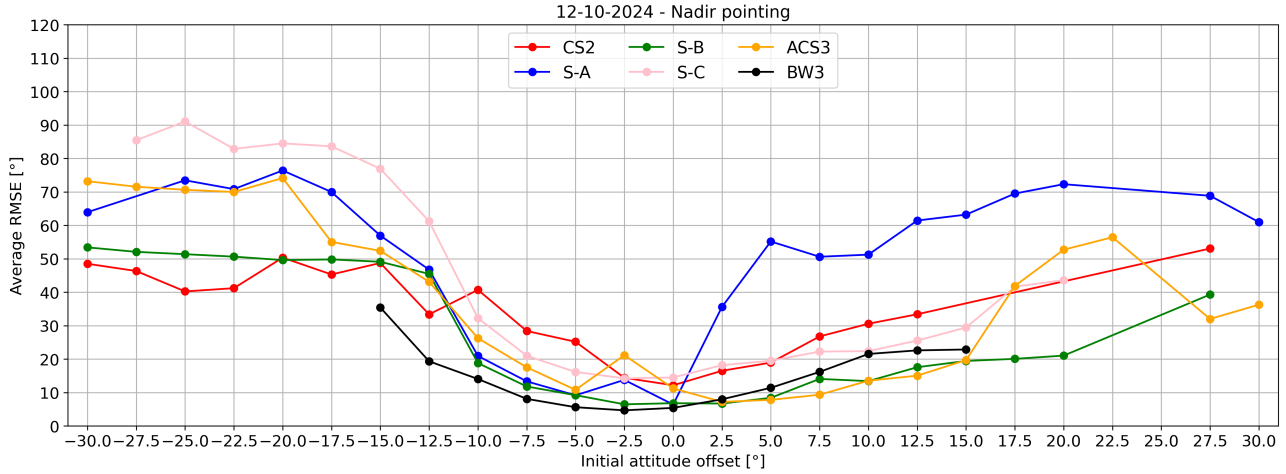


Figure 6.2.5: Average Euler angle attitude RMSE for different values of initial attitude guess offset - Nadir pointing - 12-10-2024

Single-axis spin

The Single-axis spin case RMSE plot is shown in Figure 6.2.6. Here there are some outlying points visible for positive offset values, mainly for Swarm A, Swarm C and CryoSat 2. For some offset values these satellites have a high average error, but for increased offset values these errors drop back down again, for example for Swarm A at 12.5° and 15.0° offset values. This is caused by the same behaviour as encountered in Figure 6.2.3, where the MMAE algorithm is not always able to correct a diverged attitude estimation, while for other estimations with a different initial guess it is able to convert on an attitude estimate with low errors.

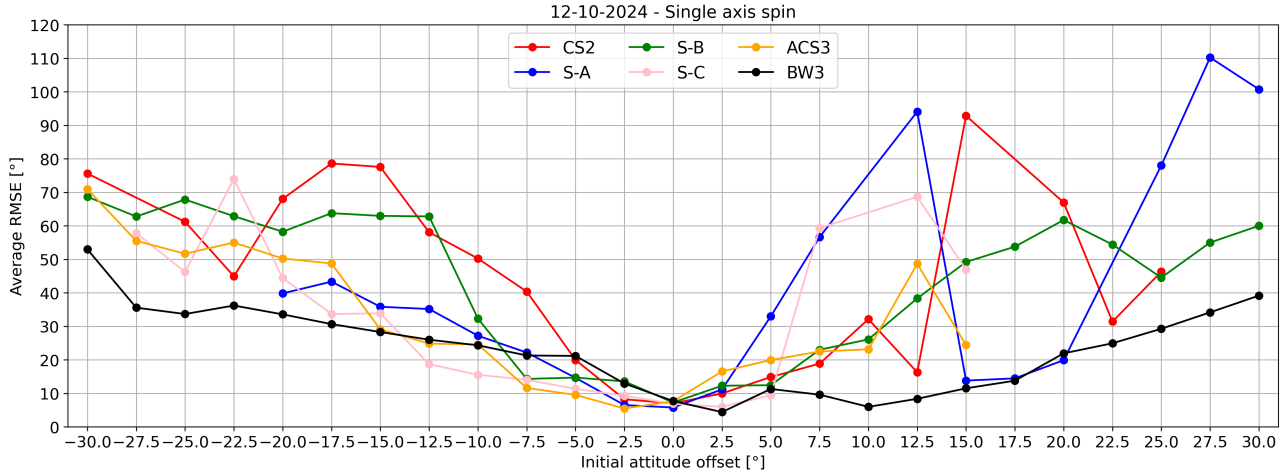


Figure 6.2.6: Average Euler angle attitude RMSE for different values of initial attitude guess offset - Single-axis spin - 12-10-2024

For the rest the plot shows similar results as before, with errors gradually increasing as offset magnitudes are increased, meaning that the initial guess of the estimations become farther from the truth at each step.

Tumbling

The average RMSE plot for the Tumbling cases on 12-10-2024 are shown in Figure 6.2.7. It can be seen that the plot looks much smoother, with errors gradually increasing for larger offset values, and not showing large outliers. This is similar to the plot for the Tumbling cases on 07-09-2024 in Figure 6.2.4. This gives an indication that the estimation results for the Tumbling cases are more robust, being affected less by changes in the initial guess than the other two spin cases.

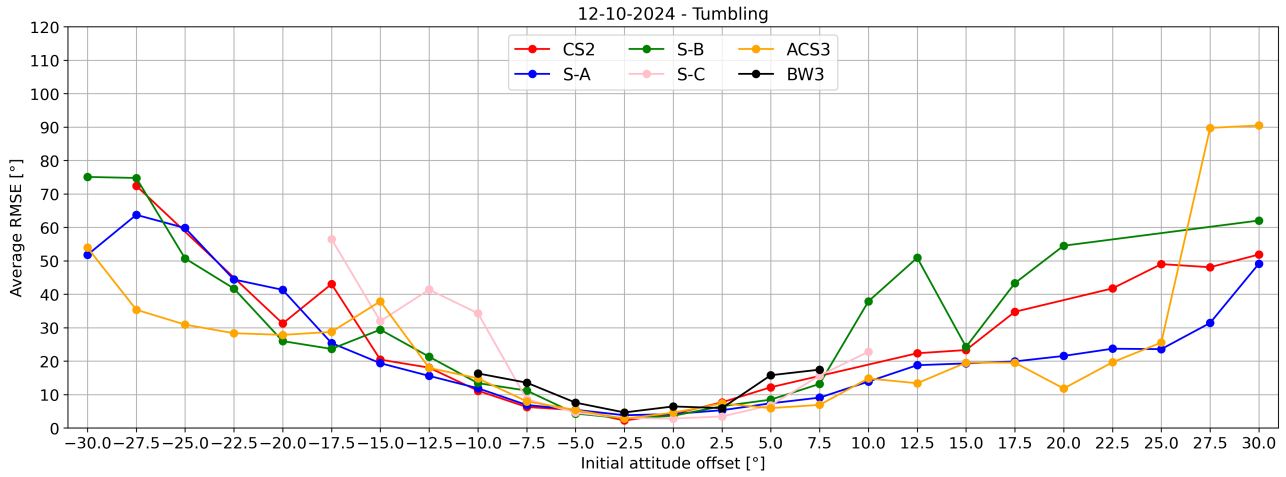


Figure 6.2.7: Average Euler angle attitude RMSE for different values of initial attitude guess offset - Tumbling - 12-10-2024

14-11-2024

Nadir pointing

For the Nadir pointing cases on 14-11-2024 the average RMSE values are plotted in Figure 6.2.8. It can be seen that the plot has much more variation than that for the Tumbling cases, for example those in Figure 6.2.7. The algorithm also finds the correct shape much less often for higher positive offsets than for negative offsets. Taking this plot and the plots for the Nadir pointing cases on the other dates it can be concluded that the Nadir pointing cases results are limitedly robust to changing initial guess values, producing the best results when the initial guess is close to the true initial state of the test objects.

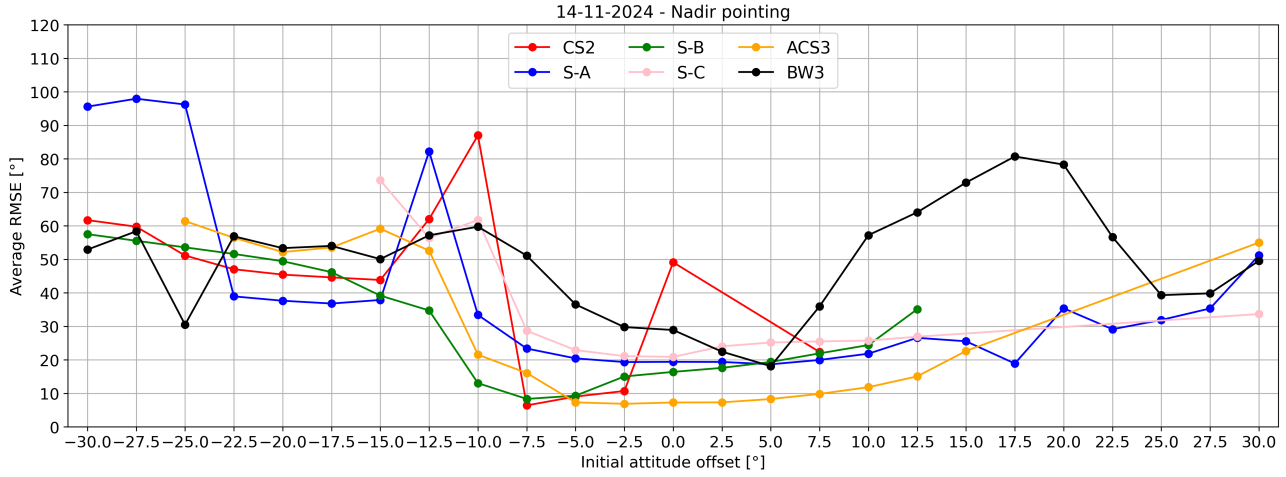


Figure 6.2.8: Average Euler angle attitude RMSE for different values of initial attitude guess offset - Nadir pointing - 11-14-2024

Single-axis spin

The average RMSE values for the Single-axis spin cases on 14-11-2024 are shown in Figure 6.2.9. While overall the errors increase for larger offset values according to expectations, it can be seen that the errors for the negative offset values are higher than those for the positive offsets. It seems that for the negative offsets the algorithm is not able to correct the diverged attitude estimations, just like was previously seen. Still, the algorithm is able to find the correct shape for a most of the offset values for all satellites.

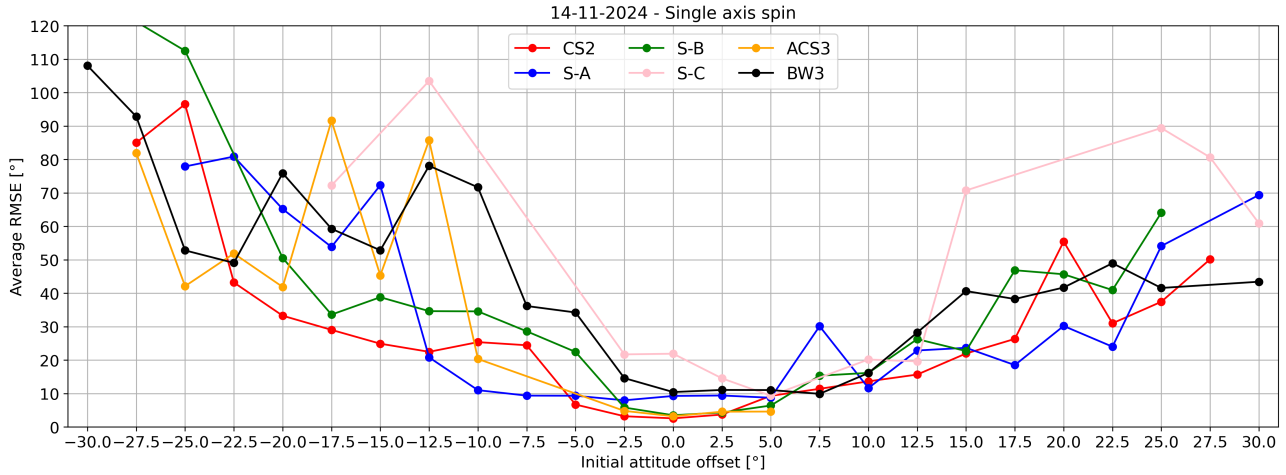


Figure 6.2.9: Average Euler angle attitude RMSE for different values of initial attitude guess offset - Single-axis spin - 11-14-2024

Tumbling

The final plot in the initial attitude offset analysis is the RMSE plot for the Tumbling cases on 14-11-2024, which is shown in Figure 6.2.10. It can again be seen that the errors for the Tumbling cases are much lower than those for the other spin case, and shows much fewer outliers. This clearly shows that the results for the Tumbling cases are much more robust to changes in the initial guess, producing low error results for a larger range of attitude offset values.

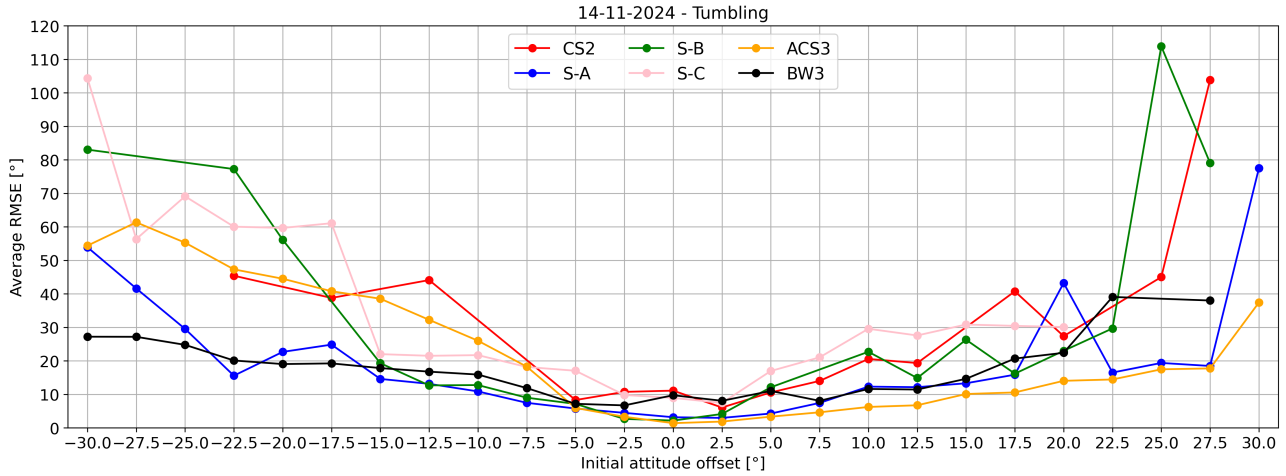


Figure 6.2.10: Average Euler angle attitude RMSE for different values of initial attitude guess offset - Tumbling - 11-14-2024

The result that the Tumbling spin case results are more robust to changing initial guess values than the Nadir pointing and Single-axis spin case results make a lot of sense. With the objects spinning around all body-axis there is a lot of variation in the simulated light curves, with apparent brightness fluctuating a lot. This means the MMAE algorithm has a lot of information to use in the attitude estimations. Conversely, the other spin cases are less robust to changing initial guess values, because there is generally less variability in their light curves, making it more difficult for the MMAE algorithm to converge on accurate estimation results for larger initial attitude offsets. A deeper investigation into the relation between measurement variability and estimation results is given in Subsection 6.3.3.

Averaged results

To create a more general overview of the initial guess dependence for the attitude offsets the results from the previous plots are averaged over all the satellites and simulation dates, for each spin case separately. With these plots an analysis of the overall behaviour of the errors with increasing attitude offsets can be made. To show the variation in the data the averaged plots per data are also shown.

Furthermore, to be able to concretely analyse the differences between the results for the different spin cases, it was decided to set an upper limit for average RMSE error. Below this limit results are deemed to be of high enough accuracy to be considered good results. With this limit set it becomes possible to identify a range of offset values that lead to these good results. The error limit is set to 20° average RMSE.

Nadir pointing

The averaged plot of the RMSE values for the Nadir pointing cases is shown in Figure 6.2.11. It can be seen that there is a slight spread between the results for the different results, but overall the average behaviour is clearly visible. For low offset values the errors are the lowest, and as the offsets become larger, so do the errors. The errors for negative offsets are on average higher than those for the positive offsets. What is also notable is that the average errors for the largest positive offsets are lower. However, the reason for this is that there were much fewer cases where the MMAE algorithm was able to find the correct shape for these positive offsets, as seen in the previous plots. This means this part of the plot shows results biased towards lower errors, while actually the performance of the algorithm for these offsets is bad.

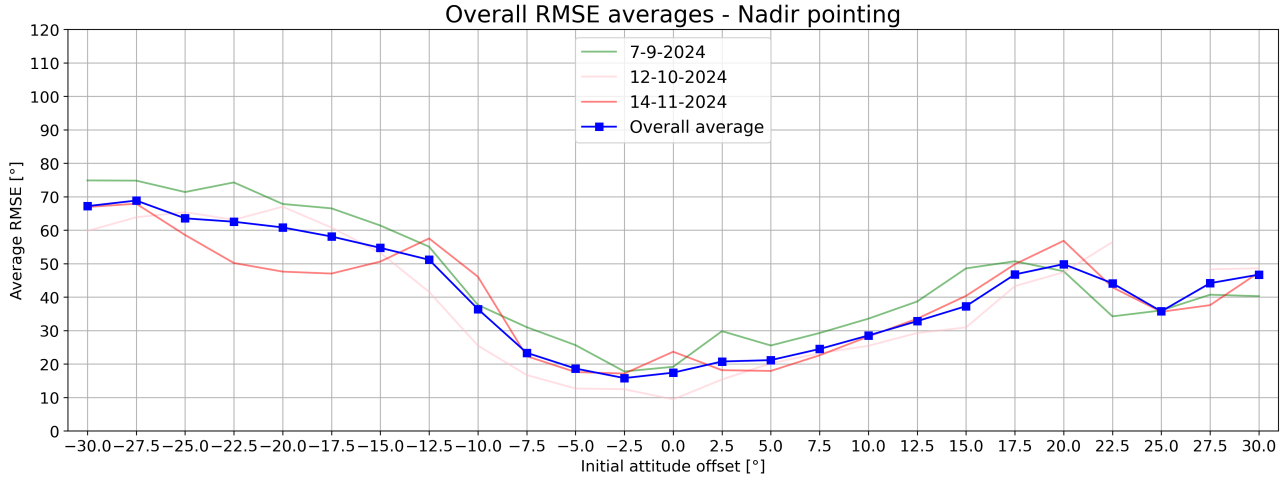


Figure 6.2.11: RMSE errors averaged over all satellites and all simulation dates - Nadir pointing

Looking at the error limit of 20°, it appears that the Nadir pointing cases only produce accurate results in an offset range between positive and negative 5.0°. This is quite a small range, and confirms the fact that the Nadir pointing case results are less robust to changes in the initial guess, and an initial guess close to the true initial guess is required to produce accurate estimation and shape characterisation results.

Single-axis spin

The averaged RMSE results for the Single-axis spin cases are plotted in Figure 6.2.12. It can be seen that the spread between the different simulation dates is small for low offset values, and increases as offset values become larger. As seen previously the errors also increase as the offsets become larger, in similar fashion for both the positive and the negative offsets.

The Single-axis spin case results appear to be slightly more robust to a change in initial guess, with an offset range between -7.5° and +7.5° producing results with average errors below the limit of 20°. This means the initial guess can be slightly more off from the true initial state and still produce accurate results. This increased robustness is caused by the fact that there is more variation in the light curves for Single-axis spin cases, which means the MMAE algorithm has more information to use in the estimations.

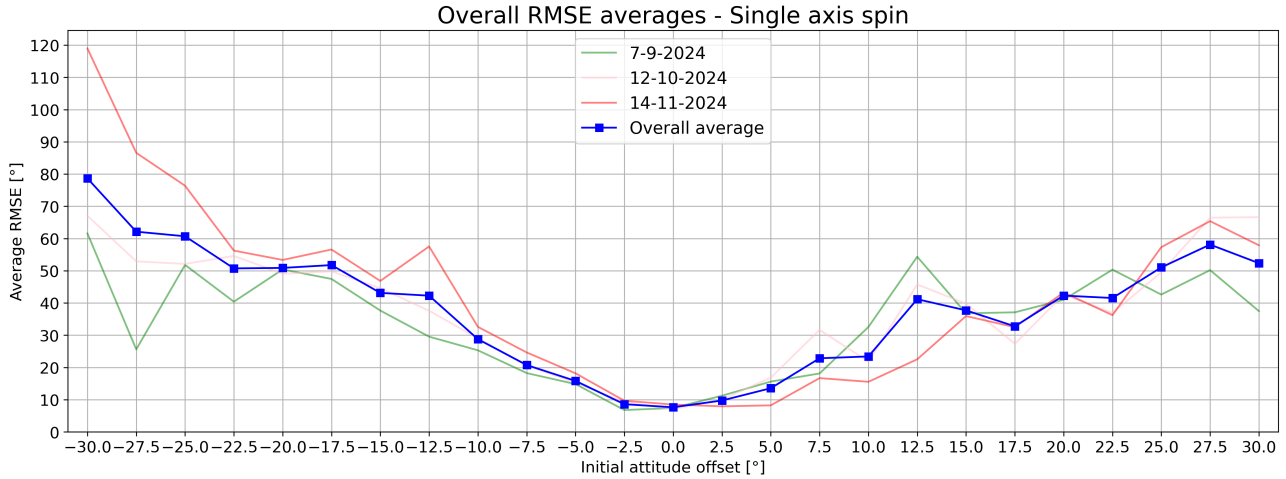


Figure 6.2.12: RMSE errors averaged over all satellites and all simulation dates - Single-axis spin

Tumbling

Lastly, the averaged RMSE values for the Tumbling cases are plotted in Figure 6.2.13. Again, the spread between the simulation dates is small for low offset values, and slightly increases with increasing offset values. The errors increase with increasing offset values in a very symmetric way, meaning that positive and negative offsets lead to results with similar errors.

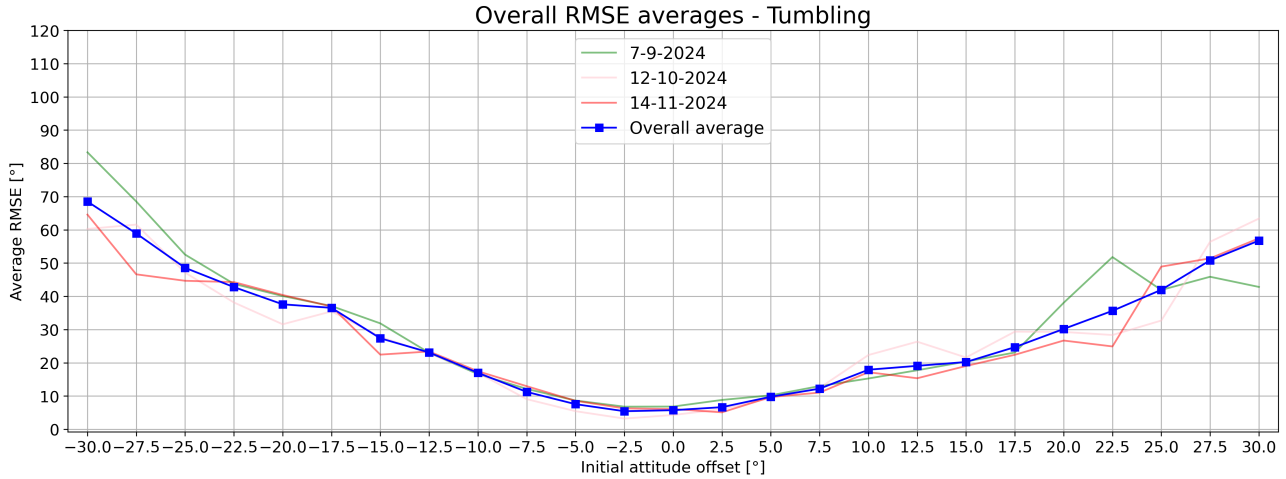


Figure 6.2.13: RMSE errors averaged over all satellites and all simulation dates - Tumbling

Looking at the error limit, it is apparent that the Tumbling cases are the most robust to changing initial guess values, as offsets between -12.0° and +12.5° produce results below the 20° error limit. This is again caused by there being more variation in the light curves of the Tumbling cases, providing a lot of information to the MMAE algorithm to use and produce accurate estimations and shape characterisations.

6.2.2. Rotational velocity

Next the influence of the rotational velocity initial guess on the attitude estimation and shape characterisation results will be examined. Similar to the initial attitude analysis, estimations will be run with the initial rotational velocity guess being offset by a fixed value from the true initial rotational velocities.

In the baseline estimations the offset from the true rotational velocities was set to 0.1 °/s. The different estimations will be run for offset values ranging from -0.1 °/s up to and including +0.1 °/s, in steps of 0.01 °/s. The initial attitude guess will be fixed to a 2.5° offset from the true initial attitude. Again for each case the average RMSE value will be plotted against the value of the initial rotational velocity offset value.

07-09-2024

Nadir pointing

The RMSE plot for the Nadir pointing cases on 07-09-2024 is shown in Figure 6.2.14. It can be seen that there are not a lot of data points for the negative offset values. This means for these values the MMAE algorithm was not able to correctly identify the correct shape for most satellites. For the positive offset values this was not the case. The average errors increase quickly for larger positive offsets, leaving only a small range of offset values that result in estimations with errors below 20°.

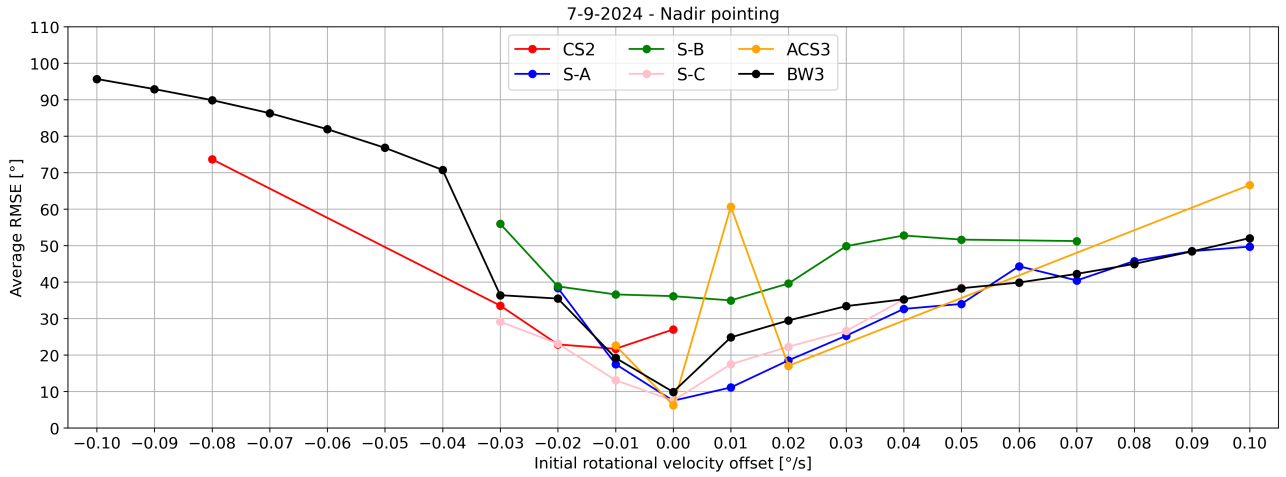


Figure 6.2.14: Average Euler angle attitude RMSE for different values of initial rotational velocity guess offset - Nadir pointing - 07-09-2024

Single-axis spin

The average RMSE values for the Single-axis spin case on 07-09-2024 are shown in Figure 6.2.15. Here there are data points for all satellites for both positive and negative offsets, meaning the algorithm found the correct shape type more often than for the Nadir pointing cases. The errors grow more slowly for larger offset values, and quite symmetric for around the zero-offset line, except for the ACS3 and BlueWalker 3 satellites. The ACS3 satellite has higher errors than other satellites overall, and no data for positive offsets above 0.06 °/s. The BlueWalker 3 satellite has higher errors for positive offsets than for negative offset values.

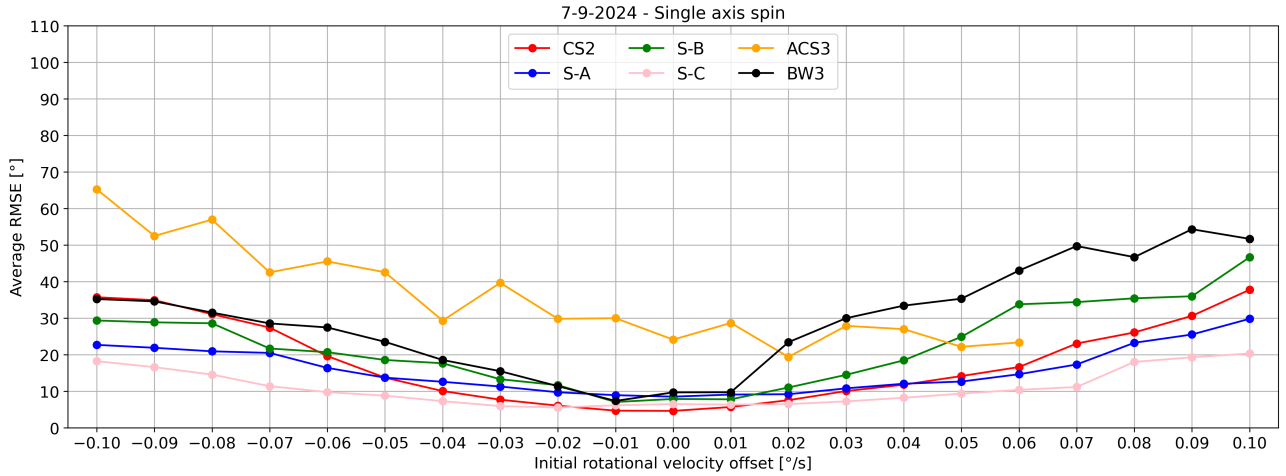


Figure 6.2.15: Average Euler angle attitude RMSE for different values of initial rotational velocity guess offset - Single-axis spin - 07-09-2024

It is curious that these outlying results are both for flat plate satellites. If this would be the case for the other spin cases and simulation dates this would warrant deeper analysis. However, it can be seen in the other plots that this is not the case. Therefore these results will be seen as unlucky combination of spin case, simulation date and test object, leading to less accurate estimation results.

Tumbling

The average RMSE values for different values of the rotational velocity offsets for the Tumbling cases on 07-09-2024 are shown in Figure 6.2.16. The results are again mostly symmetric around the zero-offset point. The errors for larger offset values, for both positive and negative offsets, are higher than those for the Single-axis spin cases in Figure 6.2.15. This could indicate that the Tumbling cases are more sensitive to the initial guess of the estimation than the Single-axis spin cases. This would not be a surprising results, seeing as the Tumbling cases have a rotational velocity around all three body-axes, while the Single-axis spin cases only have rotational velocity around on axis, so it would make sense that a large offset in initial rotational velocity affect the Tumbling cases more negatively. This result will be further investigated for the other simulation dates.

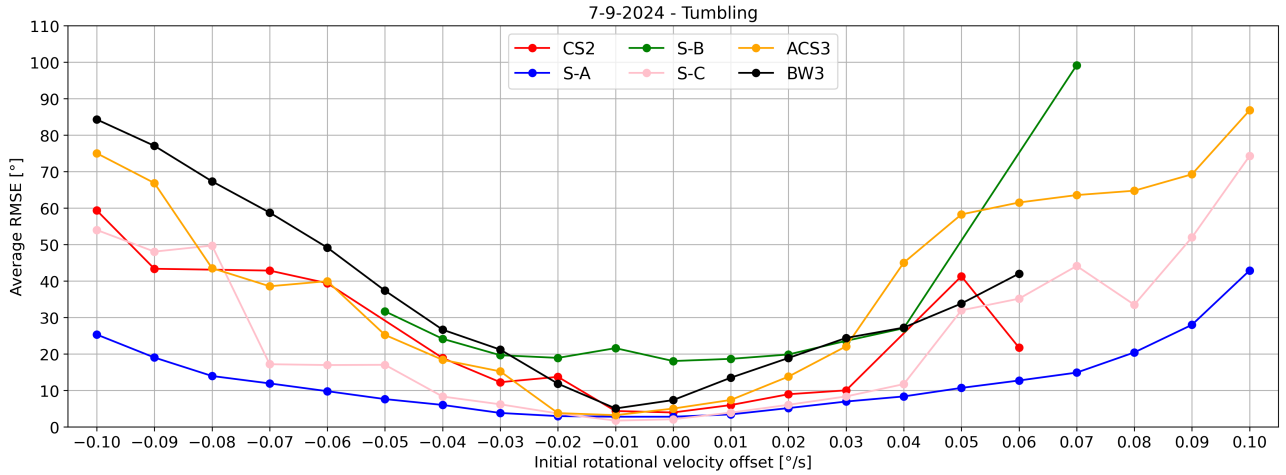


Figure 6.2.16: Average Euler angle attitude RMSE for different values of initial rotational velocity guess offset - Tumbling - 07-09-2024

12-10-2024

Nadir pointing

The Nadir pointing RMSE values are plotted in Figure 6.2.17. It can again be seen that for large values of rotational velocity offset there are fewer data points, meaning that the algorithm did not find the correct shape for these offset values. This is similar to what happend for the Nadir pointing cases on 07-09-2024, as seen in Figure 6.2.14. This gives strong indication that the accuracy of the Nadir pointing results are very dependent on the initial rotational velocity guess, and a close guess is required to produce accurate results.

This behaviour is not unexpected. The Nadir pointing cases are defined by the fact that they are not assigned body-axis rotational velocities. This means that the body-axis attitude stays constant throughout the estimations. The initial guess of the estimations are assigned rotational velocities however. This means the estimated attitudes drift away from the constant true attitude, if not corrected by the MMAE algorithm. It appears that the algorithm is able to do this correction for low values of the initial rotational velocity offsets, but not for the larger offset values, as the estimated attitudes will have drifted to far from the truth to make correction possible.

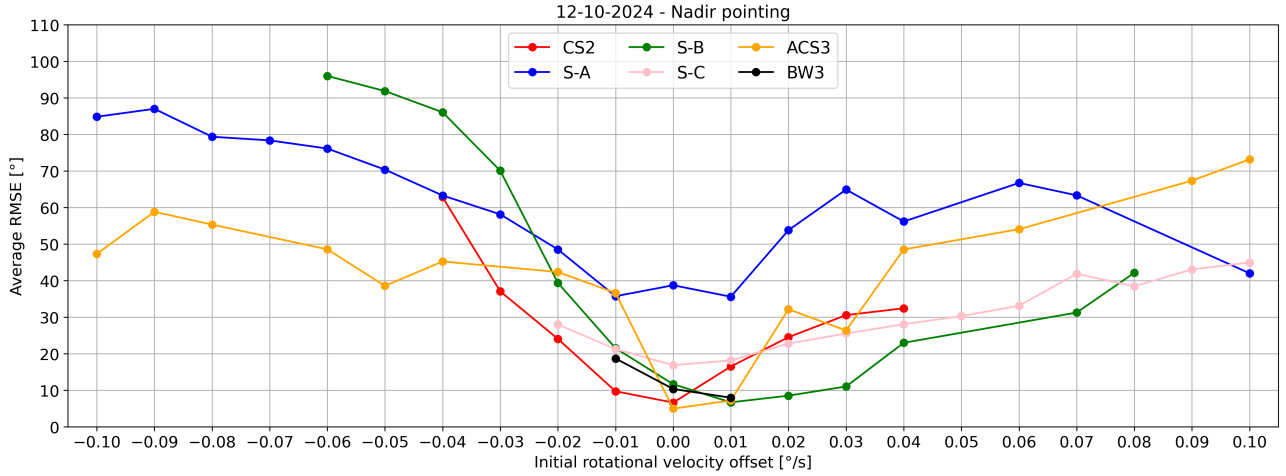


Figure 6.2.17: Average Euler angle attitude RMSE for different values of initial rotational velocity guess offset - Nadir pointing - 12-10-2024

Single-axis spin

The average RMSE values for the Single-axis spin cases on 12-10-2024 are shown in Figure 6.2.18. These errors are again much lower than those for the Nadir pointing cases, and very symmetric for both positive and negative offsets, with only a few outlying points. This result and that of 07-09-2024 indicate that the Single-axis spin case results are more robust to changes in initial rotational velocity guess, with larger offsets from the true initial guess still producing accurate estimation results.

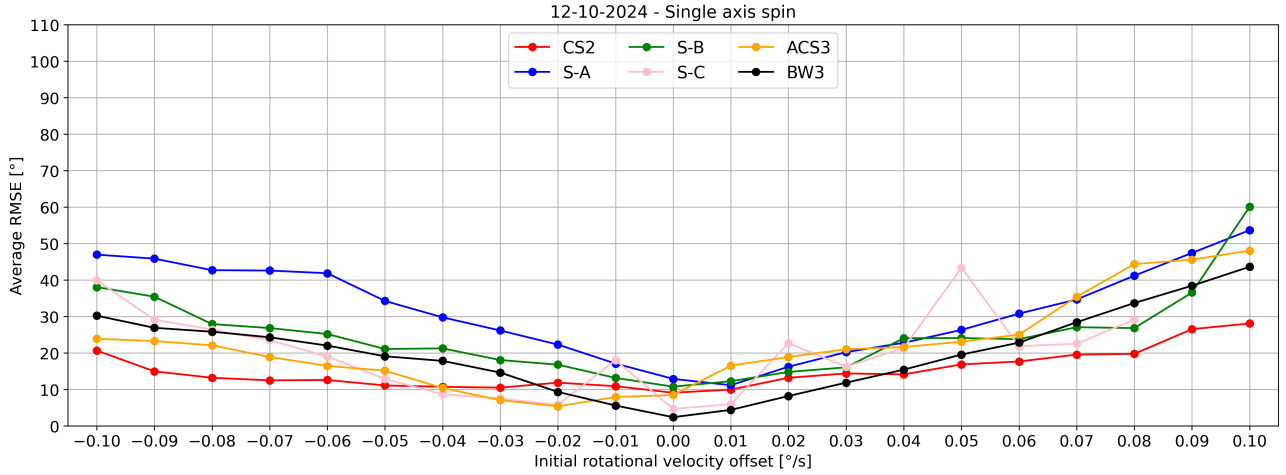


Figure 6.2.18: Average Euler angle attitude RMSE for different values of initial rotational velocity guess offset - Single-axis spin - 12-10-2024

Tumbling

In Figure 6.2.19 the RMSE values for the Tumbling cases are shown. It can be seen that for low offset values the errors are very low, and the results for the different satellites are very close to each other, but these diverge for larger rotational velocity offset values. For some satellites the errors grow much quicker with larger offset values, for example those of CryoSat 2 and Swarm B.

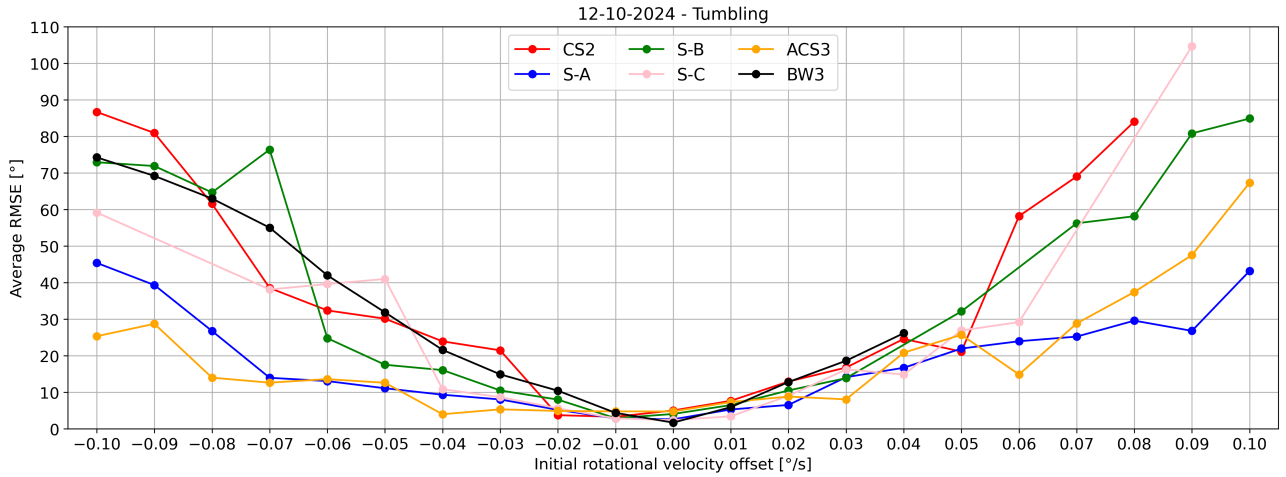


Figure 6.2.19: Average Euler angle attitude RMSE for different values of initial rotational velocity guess offset - Tumbling - 12-10-2024

Overall the errors for large offset values are higher than for the Single-axis spin cases, giving further evidence that the Tumbling cases require slightly better initial rotational velocity guesses than the Single-axis spin cases. However, the Tumbling case errors are still lower and more consistent than those of the Nadir pointing cases. This is also in line with expectations, as the Tumbling case light curves have much more variation than those of the Nadir pointing cases, which means there is much more information for the MMAE algorithm to use in the attitude estimation.

14-11-2024

The RMSE values for the Nadir pointing cases on 14-11-2024 are shown in Figure 6.2.20. There are now a lot more data points for the negative offset values. However, the errors for the negative offsets are much higher than those for the positive offsets. This behaviour falls in the same trend as the other Nadir pointing results, showing that it is more difficult for the MMAE algorithm to produce accurate estimations for these cases for negative rotational velocity offsets.

Nadir pointing

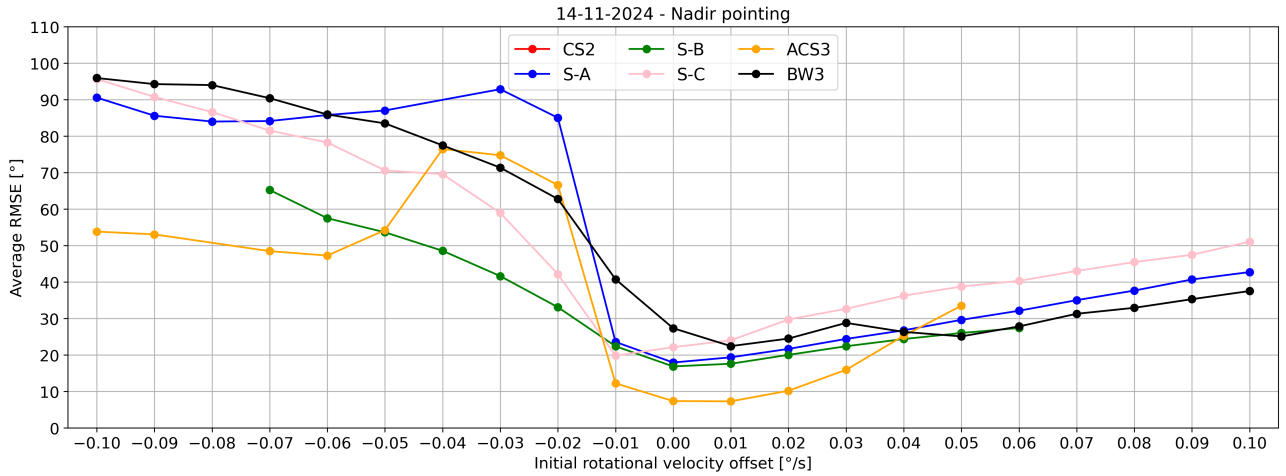


Figure 6.2.20: Average Euler angle attitude RMSE for different values of initial rotational velocity guess offset - Nadir pointing - 14-11-2024

Single-axis spin

The results for the Single-axis spin cases are plotted in Figure 6.2.21. Again, the errors are very symmetric, with both large positive and negative offsets leading to similar magnitude errors for the same satellites. The errors for this simulation date are slightly higher for larger offset values than for the Single-axis spin case results of the other dates. The results still indicate that the Single-axis spin case results are more resilient to initial rotational velocity guesses that are more offset from the true rotational velocities.

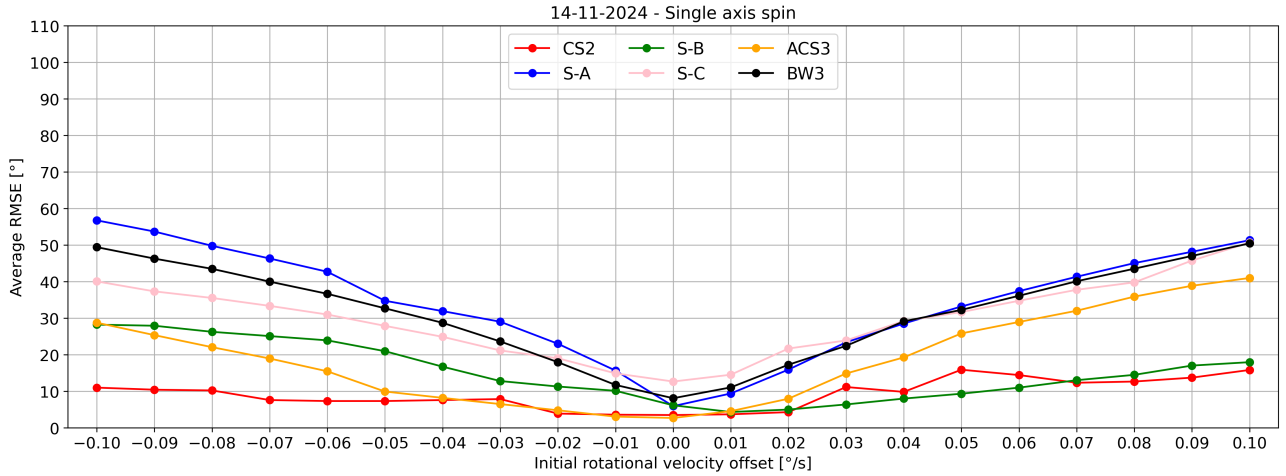


Figure 6.2.21: Average Euler angle attitude RMSE for different values of initial rotational velocity guess offset - Single-axis spin - 14-11-2024

Tumbling

Finally, the average RMSE values for different values of the initial rotational velocity offset for the Tumbling cases on 14-11-2024 are shown in Figure 6.2.22. Just like for the results on 12-10-2024, the errors for the different satellites for low offset values are low and close together. For larger offsets the values for the different satellites diverge more, and generally the errors grow larger. The errors are again larger than for the Single-axis spin cases, highlighting that the Tumbling cases are more reliant on an initial guess that is closer to the true initial state than the Single-axis spin cases.

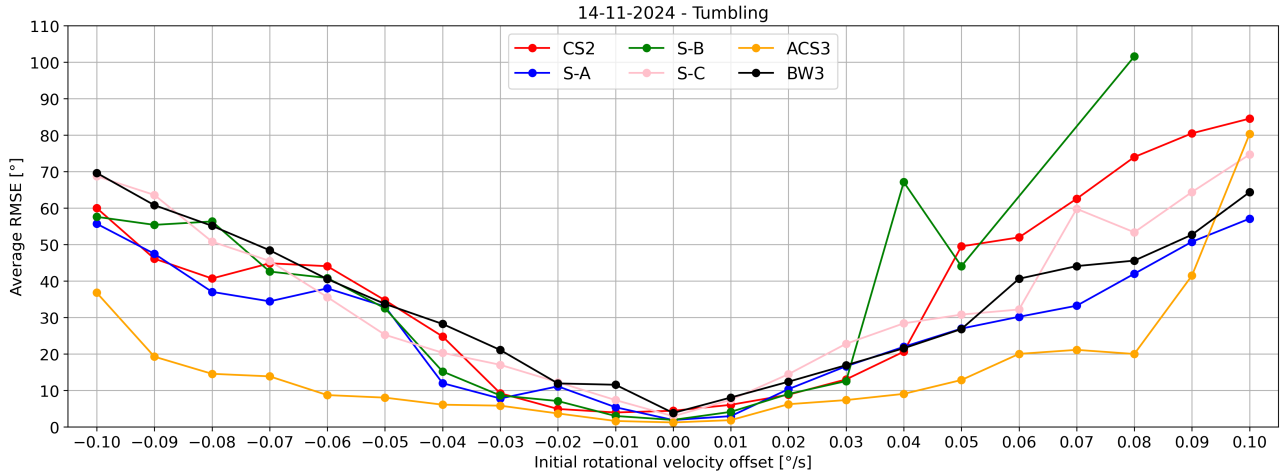


Figure 6.2.22: Average Euler angle attitude RMSE for different values of initial rotational velocity guess offset - Tumbling - 14-11-2024

Averaged results

Just like was done for the results of the initial attitude offset analysis, the average RMSE values for all the satellites and simulation dates are averaged for the different spin cases. This makes it possible to analyse the general behaviour of the results for the different spin cases with increasing rotational velocity offsets. To show the spread in the results the averaged values for the different simulation dates are also shown in each plot.

Nadir pointing

The averaged RMSE values for the Nadir pointing cases are shown in Figure 6.2.23. This plot shows the trend that was observed before, that for the Nadir pointing cases the errors are higher for negative offsets than for positive offsets. This asymmetric behaviour of the error plot is also caused by the fact that for the negative offset values the MMAE algorithm was not able to find the correct shape for most of the negative offset values. This means that the averaged results in Figure 6.2.23 for the negative offset values only consist of these high error results. Still, both aspects indicate that is is more difficult for the algorithm to produce accurate results for negative rotational velocity offsets for the Nadir pointing cases.

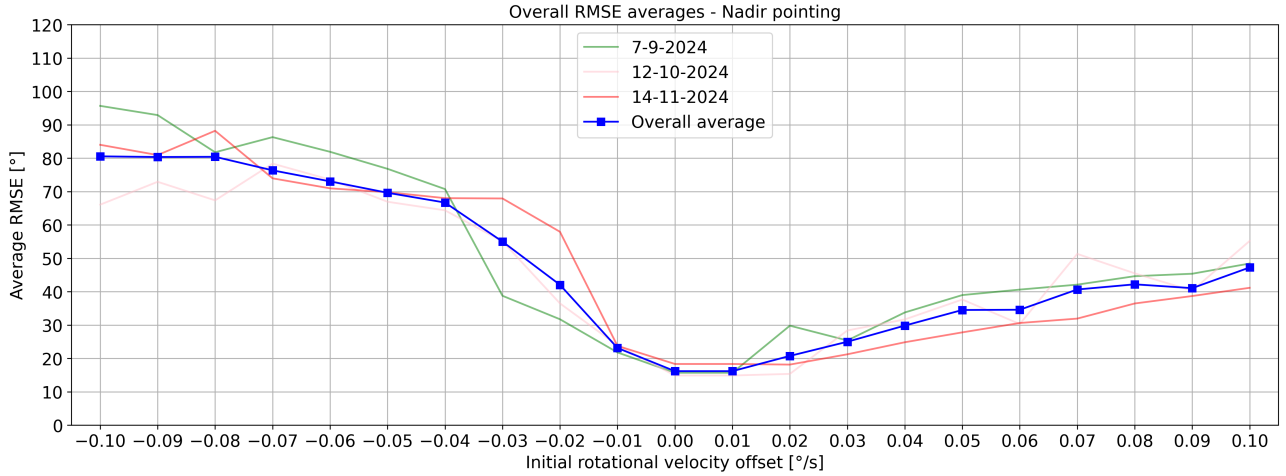


Figure 6.2.23: RMSE errors averaged over all satellites and all simulation dates - Nadir pointing

Single-axis spin

The averaged RMSE values for the Single-axis spin cases are shown in Figure 6.2.24. As was seen in the results for each simulation date separately, the errors for the Single-axis spin cases behave very symmetrically for both positive and negative offsets. The spread between the errors for the different simulation dates is also very small. This plot shows confirms that the Single-axis spin case results are very robust to changes in the initial rotational velocity guess. Offsets in the range -0.05 °/s and +0.05 °/s all produce results below the accuracy limit of 20 °, which is half of the full offset range tested.

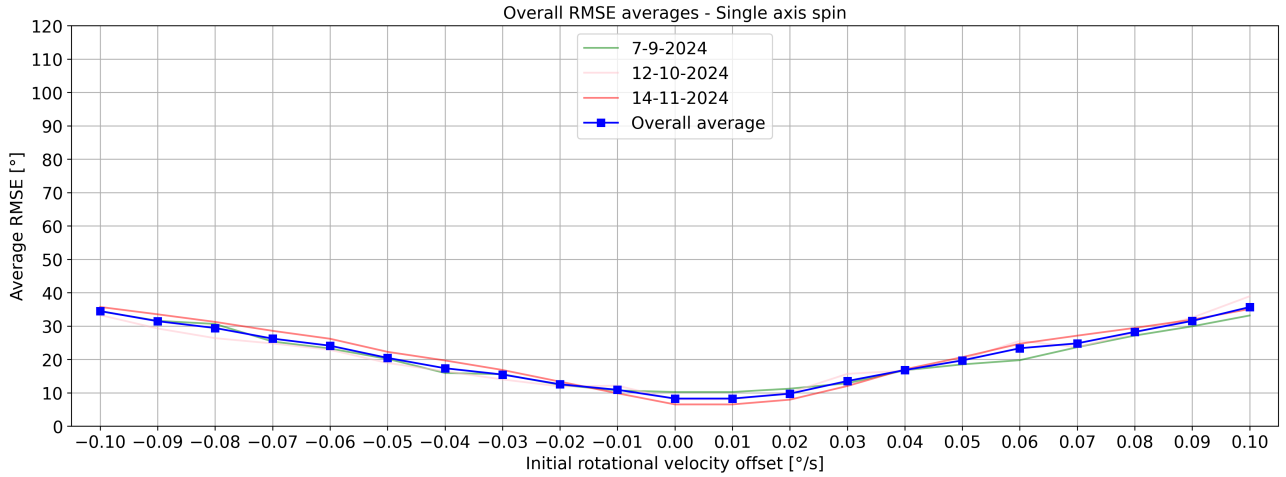


Figure 6.2.24: RMSE errors averaged over all satellites and all simulation dates - Single-axis spin

Tumbling

Finally, the averaged results for the Tumbling cases are shown in Figure 6.2.25. As was seen previously, for low offset values the errors are very low, and the spread between the simulation dates is also small. For larger offset values the errors grow more quickly compared to the Single-axis spin cases. The spread between the different simulation dates is also slightly higher for larger offset values. This confirms that the Tumbling case results are more sensitive to a change in the initial rotational velocities. However, the offset range that results in errors below the 20°limit for the Tumbling cases is around -0.045 °/s to +0.045 °/s, which is very close to the offset range for the Single-axis spin cases. So while the errors for large offset values are larger for the Tumbling cases, both cases produce the same accuracy results for a similar offset range.

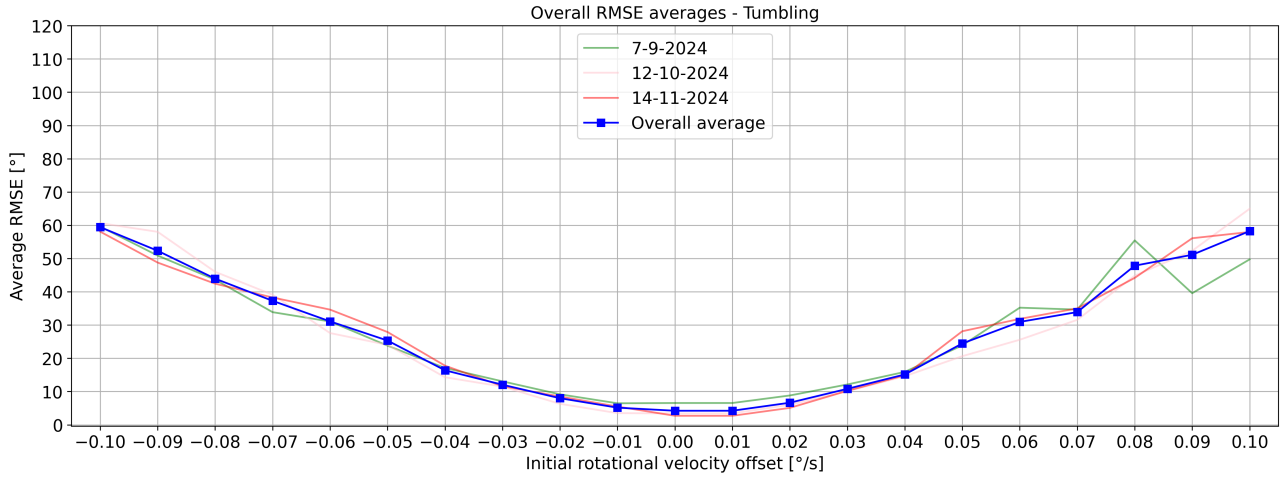


Figure 6.2.25: RMSE errors averaged over all satellites and all simulation dates - Tumbling

6.2.3. Summary

Overall it was seen that for both the initial attitude and the rotational velocity offsets the errors are low around the true initial states, but grow with growing offset values. Especially errors for the Nadir pointing cases grow quickly, showing that these cases require an initial guess close to the truth for accurate estimation results. The Single-axis spin and Tumbling cases proved to be more robust, having a wider range of offset values for which estimations with average attitude RMSE values below 20° are produced. The offset ranges for each spin case for the initial attitudes and rotational velocities are summarised in Table 6.2.1.

Table 6.2.1: Ranges of offsets from true initial state resulting in accurate results per spin case.

Spin case	Attitude offset range [°]	Rotational velocity offset range [°/s]
Nadir pointing	[-5.0, 5.0]	[-0.01, 0.02]
Single-axis spin	[-7.5, 7.5]	[-0.05, 0.05]
Tumbling	[-12.0, 12.5]	[-0.045, 0.045]

6.3. Data requirements

A vital aspect driving the estimations performed in the thesis is the measurement data that are simulated for all the different cases. Without measurements there would be no estimations to perform. In this section it will be analysed what happens to the estimation results when different aspects of the measurement data are changed. The goal of this analysis is to give an answer to the third research sub-question **RQ-3**: *What are requirements on measurement data (e.g. noise, rate, variability) to ensure accurate estimates?*

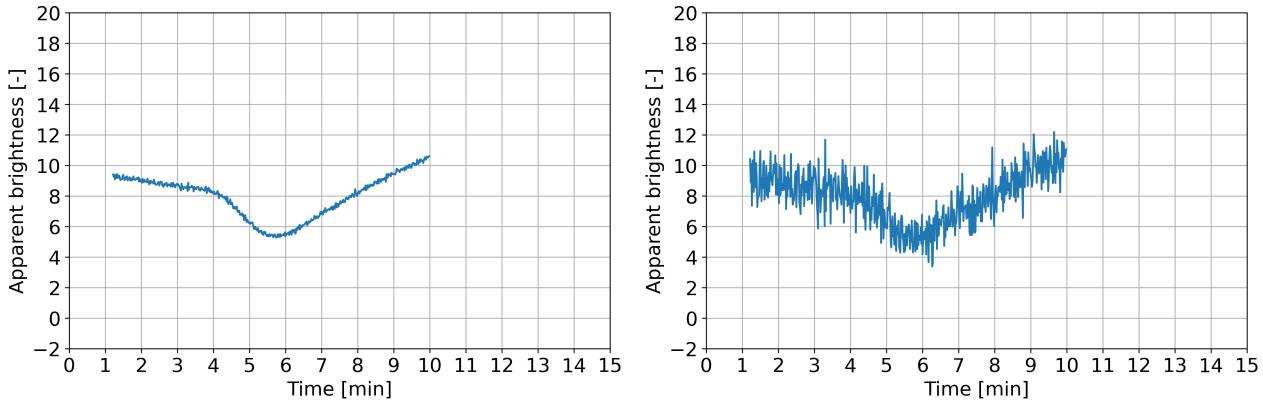
As the research question implies, there are three aspects of interest in this analysis. The first is the measurement noise. For the estimations measurements are simulated converting the true propagated states to the measurement variables and adding zero-mean Gaussian noise to the values. Depending on the magnitudes of the standard deviations that are used measurements become more or less noisy. In Subsection 6.3.1 it will be investigated what happens to the results when different levels of noisy measurements are used in the estimations.

The second aspects that is looked into is the frequency at which measurements are made available in the estimations, called the measurement data rate. In Subsection 6.3.2 the effect of different data rates on the results will be examined. The last aspect of interest is the variability of the measurements. Here variability is seen as the amount of variation in a set of measurements, for example how close subsequent measurements are to each other, if there are sharp peaks in the measurements and how large of a range the measurements span. The relation between variability and estimation results is explored in Subsection 6.3.3.

6.3.1. Noise

For the measurement noise analysis only the effect of the noise of the apparent brightness measurements will be looked at. This is done because these measurements have the most direct impact on the quality of the attitude estimation and shape characterisation results. Just like was done for the initial guess dependence analysis, estimations for all 54 test cases will be run with different values of the apparent brightness noise standard deviation σ. In the baseline estimations the standard deviation was set to a value of 0.10. Estimations will be run with both higher and lower noise values. The range of standard deviation values that are investigated is: [0.01, 0.05, 0.10, 0.25, 0.50, 0.75, 1.0, 1.5, 2.0].

To illustrate the effect of different standard deviation values on the measurements an example is given. In Figure 6.3.1 two light curves are shown for the same test case, namely the Swarm C - Nadir pointing case on 07-09-2024. In Figure 6.3.1a the measurements were simulated with a σ value of 0.10, the same value that was used in the baseline estimations. The measurements in Figure 6.3.1b were simulated using a σ value of 1.0, ten times higher than used in the baseline estimations.



(a) Apparent brightness $\sigma = 0.10$

(b) Apparent brightness $\sigma = 1.0$

Figure 6.3.1: Simulated light curves with different values of measurement noise for Swarm C - Nadir pointing - 07-09-2024

While the overall shape of the curve is still visible in the high noise plot, it is clearly less sharp than the curve with a σ of 0.10. In this section it will be examined what happens to the results of the MMAE algorithm estimations for the test cases with these different levels of noisy measurement data.

07-09-2024

The average RMSE values for different values of the apparent brightness noise for the Nadir pointing case on 07-09-2024 are shown in Figure 6.3.2. It can be seen that the errors for most satellites and noise values are above 20°, which was set to be the upper limit for accurate estimation results in the previous section. What is also interesting is that the errors for the lowest noise values are higher than those for higher noise values. This is a counter-intuitive result, as it would be expected that estimations with less noise would result in more accurate estimations. This behaviour will be investigated at the end of this subsection.

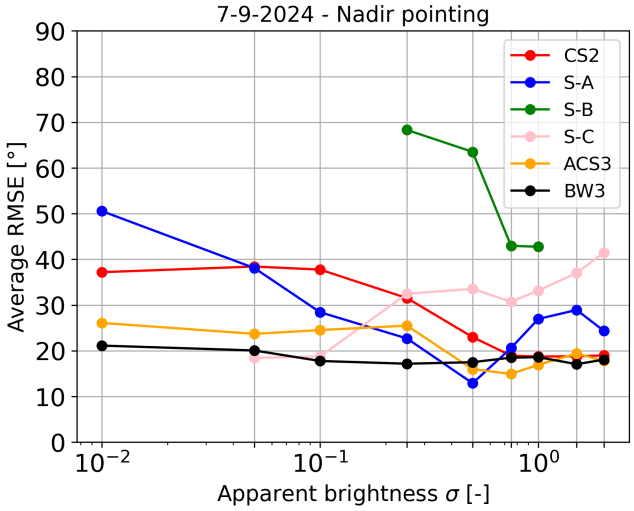
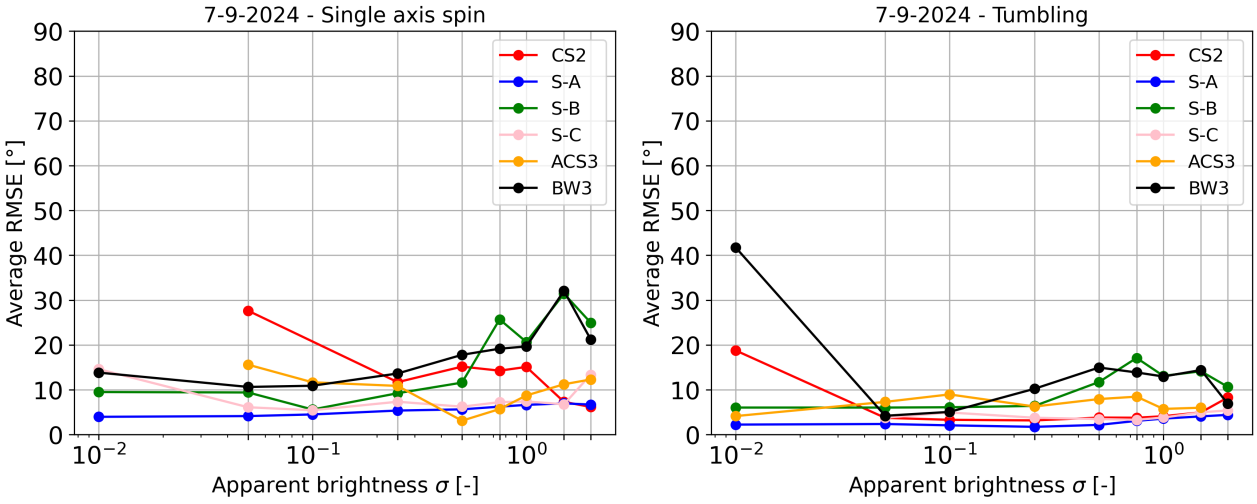


Figure 6.3.2: Average RMSE for different values of apparent brightness measurement noise - Nadir pointing - 07-09-2024

Next the results for the Single-axis spin and Tumbling cases are looked at, which are shown in Figure 6.3.3. It is clear that these errors are much lower than those of the Nadir pointing case. The errors are also more consistent for different values of the noise, showing less variation than the errors in Figure 6.3.2. However, in these plots it can also be seen that for the lowest noise value there are not successful shape characterisations for all satellites, and for a few satellites the errors are higher for low noise values.



(a) Single-axis spin

(b) Tumbling

Figure 6.3.3: Average RMSE for different values of apparent brightness measurement noise - Single-axis spin and Tumbling - 07-09-2024

12-10-2024

The RMSE values for the Nadir pointing case on 12-10-2024 are shown in Figure 6.3.4. The results are very chaotic, even more so than those for the Nadir pointing case on 07-09-2024. The errors are also even higher for most satellites and noise values. These results indicate that the Nadir pointing results are not very robust to changes in simulation parameter settings, with different settings producing vastly different results.

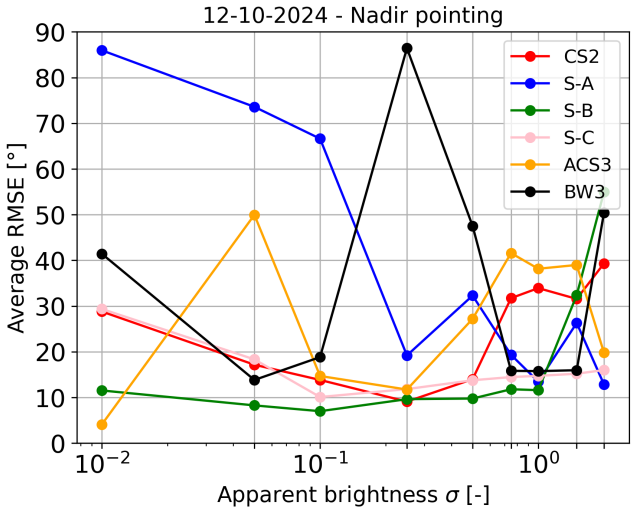
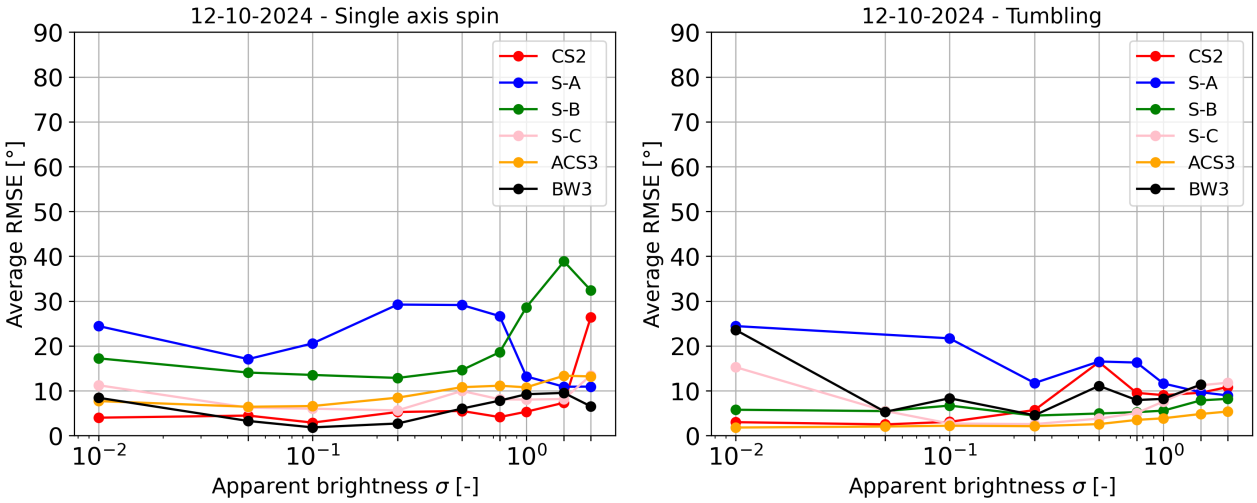


Figure 6.3.4: Average RMSE for different values of apparent brightness measurement noise - Nadir pointing - 12-10-2024

In Figure 6.3.5 the RMSE results for the Single-axis spin and Tumbling cases are shown. It is clear that these results are much more consistent compared to those of the Nadir pointing cases. Most of the errors are below 20°. In both plots it can also be seen that the errors increase slightly when the noise values become larger for most satellites. This is exactly what is expected, as more noisy measurements make it more difficult for the MMAE algorithm to accurately estimate the attitudes. However, it can also be seen that for the lowest noise values the errors are again higher for some satellites, just like has been seen before.



(a) Single-axis spin

(b) Tumbling

Figure 6.3.5: Average RMSE for different values of apparent brightness measurement noise - Single-axis spin and Tumbling - 12-10-2024

14-11-2024

The average RMSE values for the Nadir pointing cases on 14-11-2024 are shown in Figure 6.3.6. Again, errors for this spin case are very high, with only the ACS3 satellties estimations scoring below 20°. For all satellites the errors are also the highest for the lowest noise value, except for Swarm A.

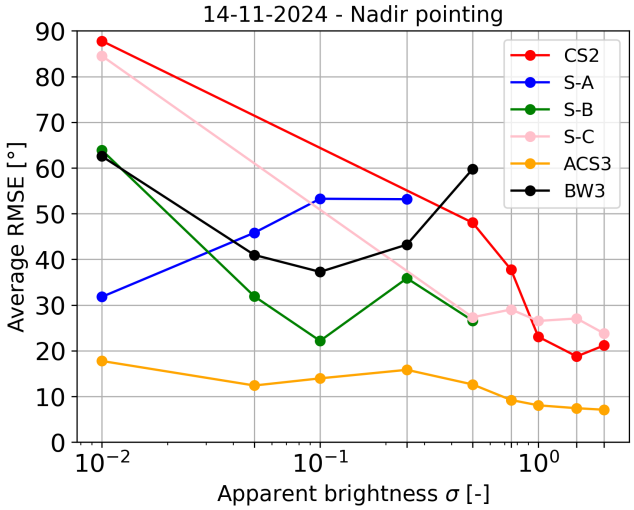
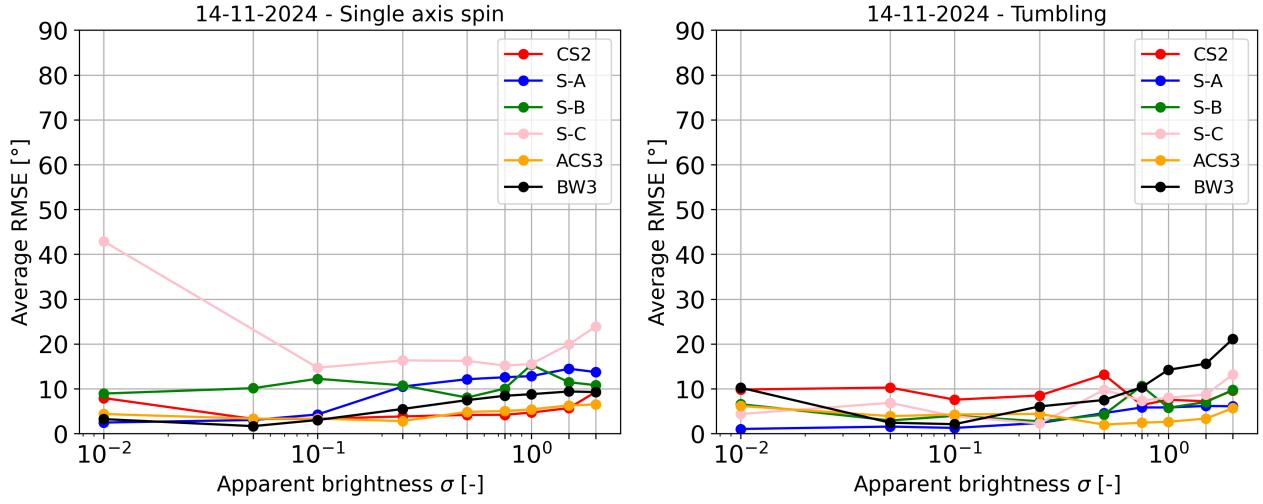


Figure 6.3.6: Average RMSE for different values of apparent brightness measurement noise - Nadir pointing - 14-11-2024

Lastly the RMSE values for different values of apparent brightness measurement noise are shown in Figure 6.3.7. The errors for these cases are again much lower than those of the Nadir pointing cases, with almost all errors being below 20°. The errors also slightly increase for larger noise values, which is in line with expectations. There is again an outlying result for the low noise values, this time the Swarm C satellite for the Single-axis spin case. This results will be investigated in more detail to explain why this occurs multiple times in the results.

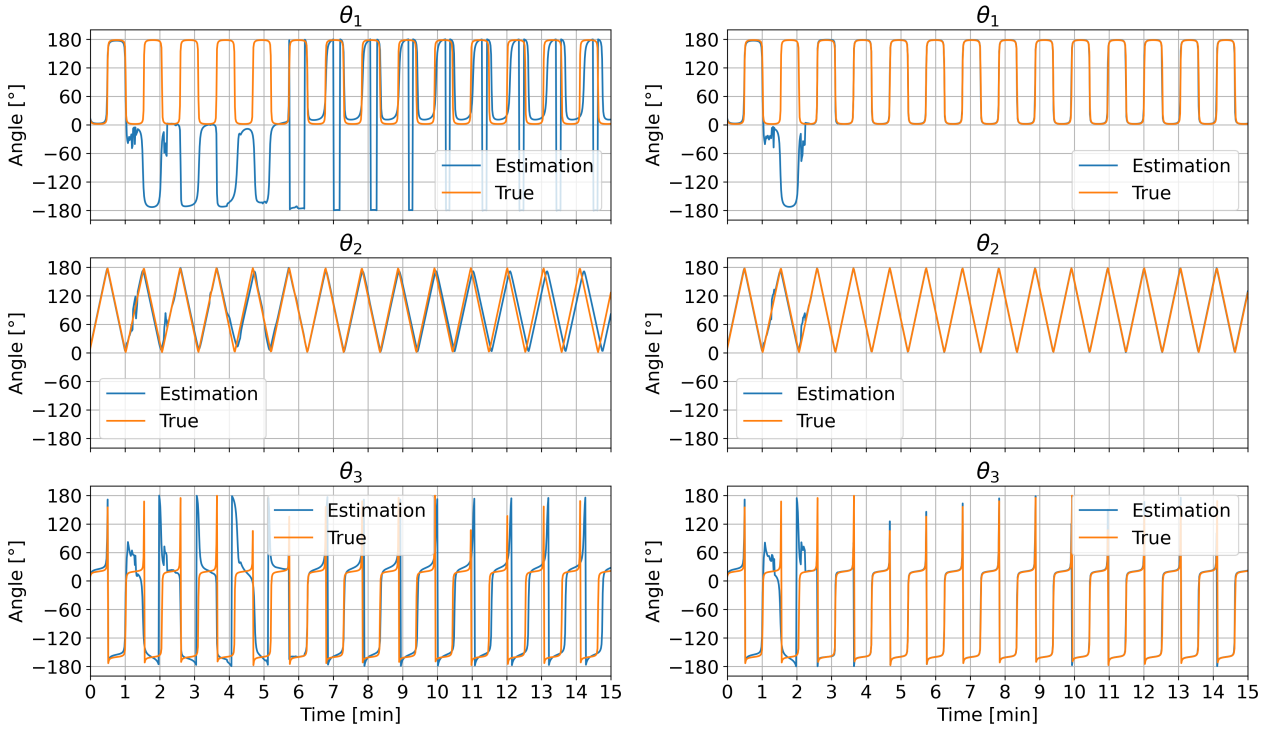


(a) Single-axis spin

(b) Tumbling

Figure 6.3.7: Average RMSE for different values of apparent brightness measurement noise - Single-axis spin and Tumbling - 14-11-2024

In Figure 6.3.8 the estimated Euler angle attitudes are plotted, as well as the true attitude states, for the Swarm C satellite. The estimation in Figure 6.3.8a was performed using a σ value of 0.10, which corresponds to the outlying result with high error in Figure 6.3.7a. The plot in Figure 6.3.8b corresponds to the estimation with a σ value of 0.10. It can be seen that at the beginning, until around 2.5 minutes, both estimations look identical, with both estimations being almost 360° offset from the truth. The fact that the estimations are almost the same makes sense., since in essence both estimations use the same measurements, only with slightly different noise values. However, after this 2.5 minutes the estimations start differing from each other.



(a) Apparent brightness $\sigma = 0.01$

(b) Apparent brightness $\sigma = 0.10$

Figure 6.3.8: True and estimated Euler angle attitude for Swarm C - Single-axis spin cases on 14-11-2024 with different noise values.

The estimation with $\sigma = 0.10$ manages to converge on the correct attitude after 2.5 minutes, and remains this way for the rest of the estimation. The estimation with $\sigma = 0.10$ on the other hand remains offset by large amount for a longer period. Around 5.5 minutes the estimation does converge more closely to the correct attitude, but remains slightly offset from the truth for the rest of the estimation. This confirms that the attitude error for the low measurement noise case in Figure 6.3.7a is higher than for the higher values. The question remains why this different behaviour of the estimation happens.

A clue is given in the plot for the θ_1 angle estimation in Figure 6.3.8a. The estimation for this angle appears to be the mirrored version of the true attitude. The reason this might happen is that the object from which the measurements are simulated, Swarm C, is a very symmetric shape (see Figure 5.1.2). This means that rotations around the same axis but in opposite direction could result in the same apparent brightness measurement. If this happens for multiple measurements sequentially the estimation could end up being the mirrored version of the true estimation, just like in Figure 6.3.8a.

The final aspect that needs explaining is why the estimation for $\sigma = 0.10$ manages to recover the estimation to the non-mirrored states, and the estimation with $\sigma = 0.01$ does not. The most likely answer is that this happens due to chance. For both estimations random noise is added to the brightness measurements. This means at each time step the estimations with different σ values receive slightly different values for the same brightness measurements. For most estimations the MMAE algorithm manages to correct the attitude estimations with these measurements, but for others, like the $\sigma = 0.01$ case, the measurement is just not right for this to happen, leading to higher estimation errors overall. This means a bit of luck can sometimes make the difference between accurate and less accurate results.

Summary

The effect of different apparent brightness measurement noise values on the attitude estimation results were investigated in this subsection. It was found that overall the Single-axis spin and Tumbling cases are more robust to changing noise values, producing results with low errors consistently for all noise values. The errors for these cases do slightly increase with higher noise values, which is in line with expectations. The Nadir pointing cases showed less robustness, with comparatively high errors, and less consistent behaviour for the different noise values. Finally, it was seen that the symmetry of the test objects can lead to the attitude estimation being mirrored from the truth. For most cases the MMAE algorithm is able to correct this, but due to chance this does not always happen, leading to outlying results with higher errors.

6.3.2. Data rate

The next aspect of the measurement data that is investigated is the frequency with which measurements are made available to the MMAE algorithm, the data rate. This is an important aspect, as the measurements are what drive the estimations. If the data rate is lower, meaning the time between consecutive measurements is higher, the same estimation will have a lower number of measurements, thus less information to correct the estimations with. However, since more measurements lead to more correction steps, runtime increases with an increased data rate. It is therefore important to analyse what happens to the baseline results of Subsection 6.1.4 for different values of the data rate.

The baseline estimations were run with an estimation time step of 1.0s, with measurements being available at the same time step, meaning for every estimation timestep there was both a prediction and a correction step in the MMAE algorithm. For the analysis estimations were run with both higher and lower time steps. The range of measurement time steps for which estimations were run is: [0.1s, 0.5s, 1.0s, 5.0s, 10.0s, 20.0s, 30.0s]. To analyse the results the average Euler angle attitude RMSE per case is plotted against the value of the measurement time step, only if the MMAE algorithm was able to correctly identify the true shape for that case.

An important note is that if the measurement time step was higher than 1.0s, the estimation time step was still kept at 1.0s. This means that if the measurement time step was 10.0s, the MMAE algorithm performed 10 prediction steps before a correction step with the new measurements was made. Conversely, if the measurement time step was lower than 1.0s, the estimation time step was set to be equal to the measurement time step, because otherwise these extra measurements would be skipped.

07-09-2024

In Figure 6.3.9 the results for the Nadir pointing cases on 07-09-2024 are plotted. What is striking is that there are only two satellites for which there is a data point for a measurement time step of 0.1 s, which means only for these two cases the MMAE algorithm found the correct shape. For the other time step values there is also not a data point for every satellite. Looking back at Table 6.1.3, it can be seen that this case was one of the few where the MMAE algorithm did not find the correct shape, namely for the CryoSat 2 and Swarm B satellites. Figure 6.3.9 shows that this combination of simulation date and spin case is difficult for the MMAE algorithm, even if more measurements and thus more information are made avalaible.

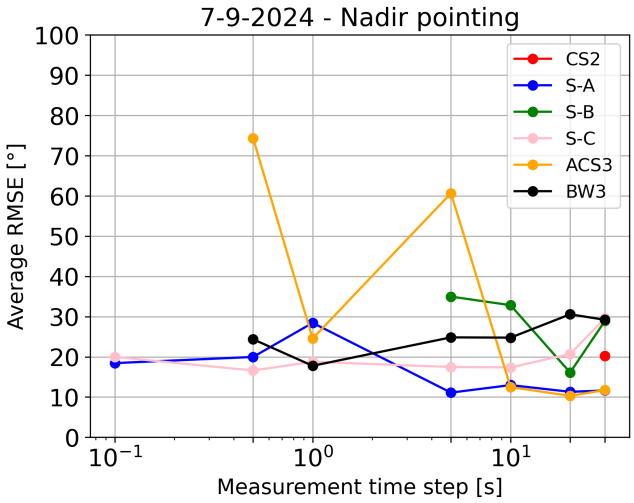
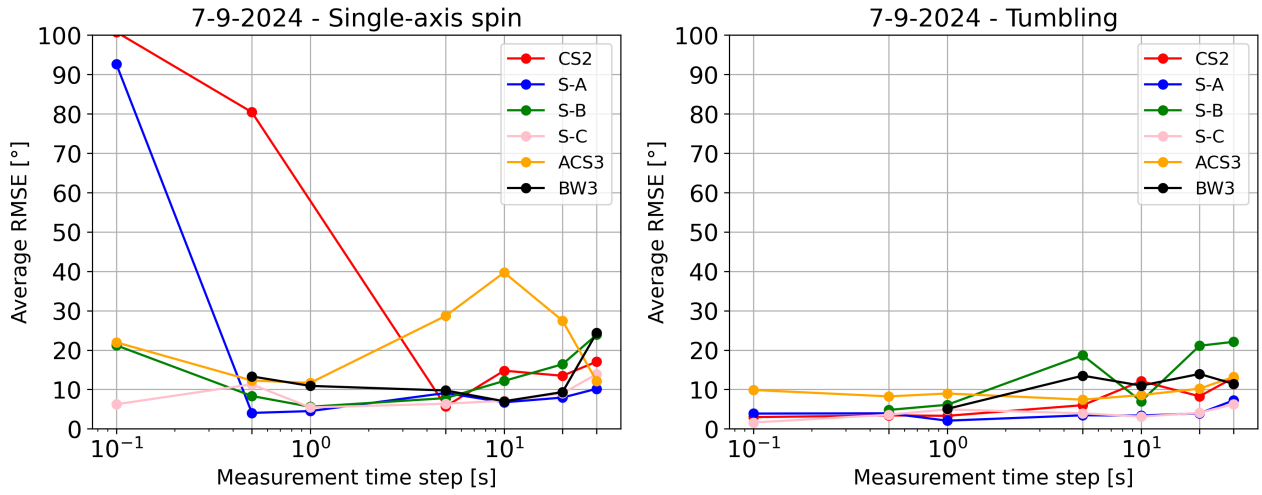


Figure 6.3.9: Average RMSE for different values of measurement data rate - Nadir pointing - 07-09-2024

In Figure 6.3.10 the results for the Single-axis spin and Tumbling cases are shown. For both cases there are more data points in the plots, also for the low measurement time steps. For the Tumbling cases the errors are low for all values of the measurement time step, with almost all errors being below 20°. For the Single-axis spin case some of the errors for the low time steps are however much higher than for the higher time steps. To see why this happens the results for Swarm A and CryoSat 2 with a measurement time step of 0.1s are looked at more closely.

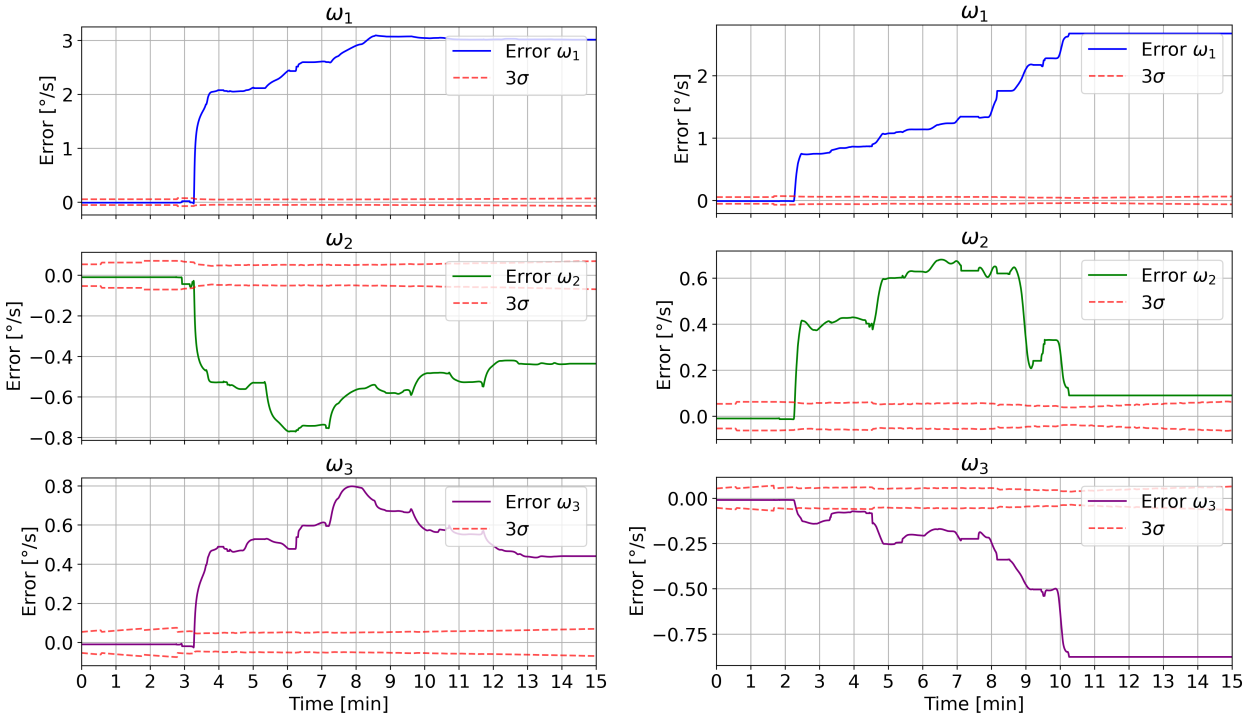


(a) Single-axis spin

(b) Tumbling

Figure 6.3.10: Average RMSE for different values of measurement data rate - Single-axis spin and Tumbling - 07-09-2024

In Figure 6.3.11 the rotational velocity errors for CryoSat 2 and Swarm A for the Single-axis spin case with a measurement time step of 0.1s are shown. It can be seen that for both curves at the times the measurements become available the errors increase drastically, especially those for ω_1, which are in the order of magnitude of full degrees per second. Because the estimated rotational velocities diverge this much and this quickly from the true states, the attitude estimation also diverges a lot, leading to the large errors seen in Figure 6.3.10.



(a) CryoSat 2

(b) Swarm A

Figure 6.3.11: Rotational velocity error plots for CryoSat 2 and Swarm A - Single-axis spin for a measurement time step of 0.1s

It is not clear why the estimated rotational velocities diverge this much for low measurements time steps. More in depth investigation would be required to figure this out, which is beyond the scope of this thesis.

12-10-2024

The RMSE plot for the Nadir pointing cases is shown in Figure 6.3.12. For this date there are more data points for this spin case compared to those on 07-09-2024 in Figure 6.3.9. Overall most errors are also lower, with the exception of the Swarm A errors and the ACS3 errors for the lowest two measurement time steps.

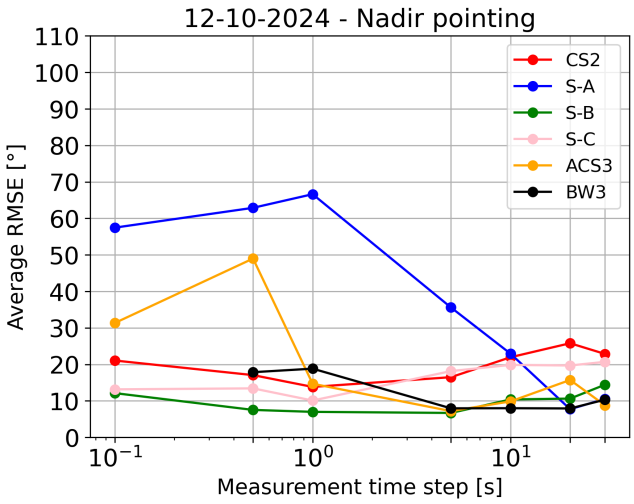
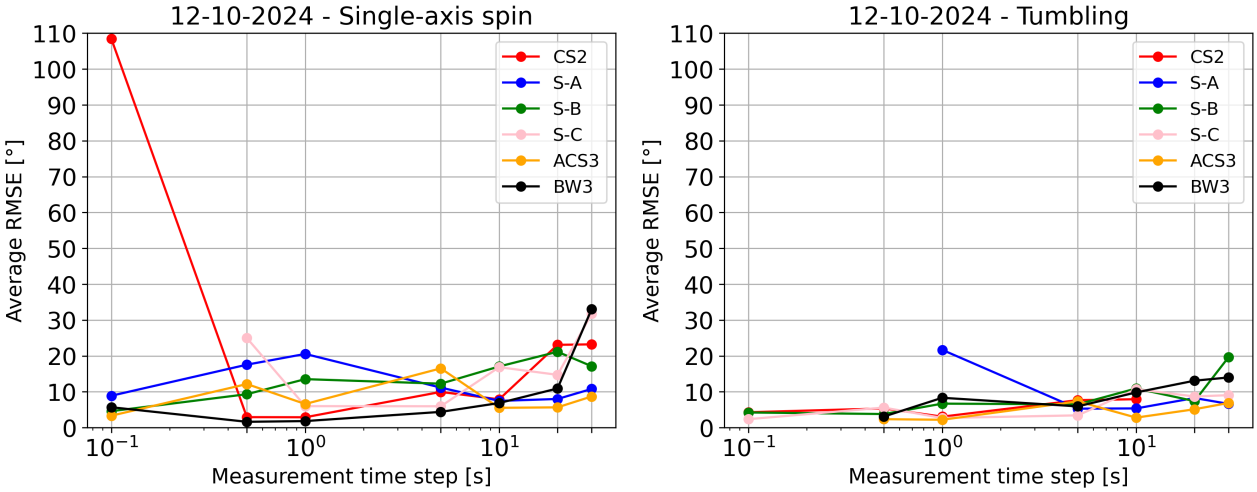


Figure 6.3.12: Average RMSE for different values of measurement data rate - Nadir pointing - 12-10-2024

The results for the Single-axis spin and Tumbling cases are shown in Figure 6.3.13. For both cases the errors are low across all time step values, noting that for the Tumbling cases there are not data points for all satellites for the lowest time steps. There is also again a large outlier in the Single-axis spin case, namely the CryoSat 2 satellite on the measurements time step of 0.1s. This is similar to the results for the Single-axis spin case on 07-09-2024, and is caused by the same divergence of the rotational velocity estimation at the start of the estimation.



(a) Single-axis spin

(b) Tumbling

Figure 6.3.13: Average RMSE for different values of measurement data rate - Single-axis spin and Tumbling - 12-10-2024

14-11-2024

The results for the Nadir pointing cases on 14-11-2024 are shown in Figure 6.3.14. It can again be seen that there are not many data points for the lowest measurement time steps, meaning that for only two satellites the MMAE algorithm was able to find the correct shape for these time steps. For the other time steps the errors range from values below 20° to above, for example for Swarm C and CryoSat 2.

Looking back at Table 6.1.3, it can be seen that this combination of simulation date, satellites and spin case resulted in the few cases where the MMAE algorithm was not able to identify the correct shape. This is reflected in the results in Figure 6.3.14. Interestingly, the algorithm was able to identify the correct shape for these satellites for the higher time steps. This shows that a low measurement time step, thus more measurements used in the estimation, does not always lead to the best results.

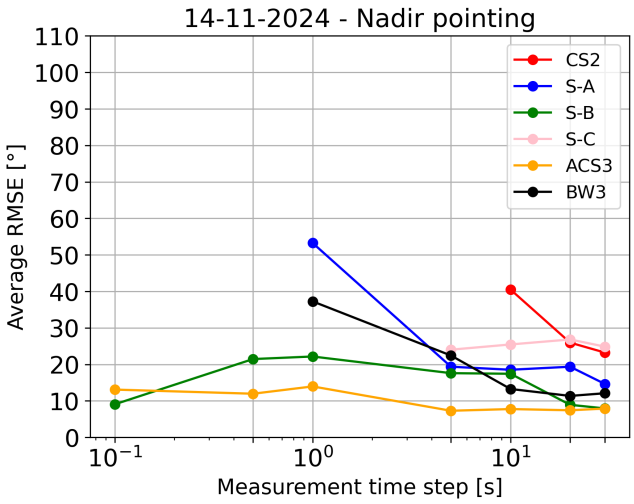
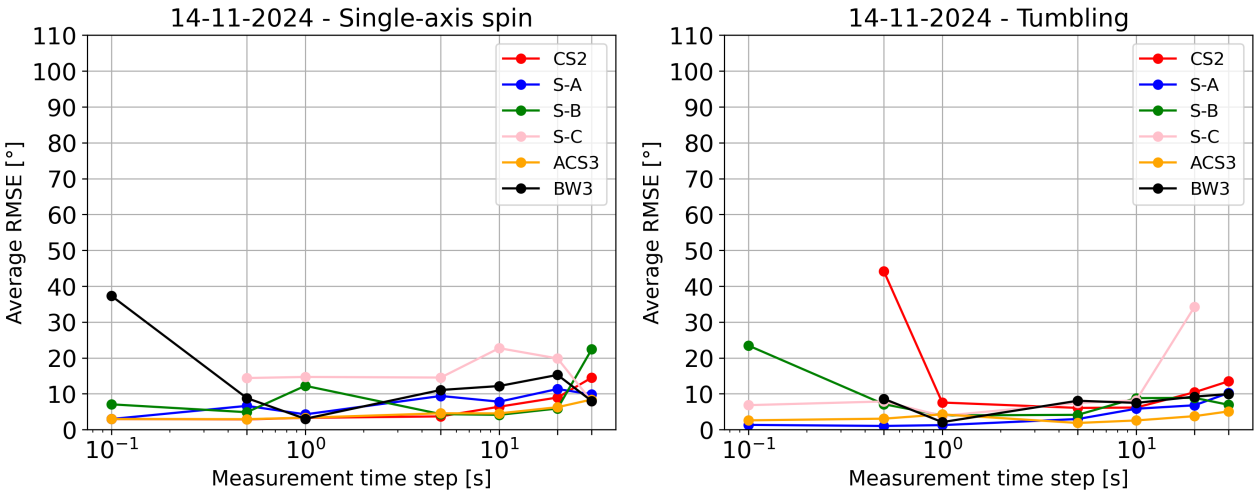


Figure 6.3.14: Average RMSE for different values of measurement data rate - Nadir pointing - 14-11-2024

Lastly, the results for the Single-axis spin and Tumbling cases on 14-11-2024 are shown in Figure 6.3.15. For both cases the errors are mostly below 20°, with a few outlying data points with higher errors. It can again be seen that for the lowest measurement time step no all satellites have data points, just like was seen before.



(a) Single-axis spin

(b) Tumbling

Figure 6.3.15: Average RMSE for different values of measurement data rate - Single-axis spin and Tumbling - 14-11-2024

Summary

The effect of different measurement timesteps on the attitude estimation results were investigated in this subsection. There are a few similarities that were seen in the results over the three simulation dates. The first is that for all spin cases the MMAE algorithm was able to identify the correct shape for the lowest measurement time step values the least often. If it did identify the correct shape, it was sometimes with high attitude errors. Analysis showed that this behaviour is caused by the rotational velocity estimation diverging at the first few measurements for these cases, the reason for which is not entirely clear.

One possible method that could be used to improve the results for the low measurement time steps is called epoch folding, a technique that is used in data analysis to extract periodicity from data [27]. It has been applied to satellite light curves before [28], and could be used to give a better initial guess of the rotational velocities, possibly solving the diverging estimations as seen in Figure 6.3.11.

Overall the errors for the Nadir pointing cases were slightly higher than those for the Single-axis spin and Tumbling cases. This result is in line with the other results examined so far, namely those of the initial guess dependence and noise characteristics. All these analyses show that the results for the Single-axis spin and Tumbling cases are more robust to changes in the simulation parameters that were used to obtain the baseline results in Subsection 6.1.4, while the Nadir pointing case results vary a lot more, with higher errors.

6.3.3. Variability

The relation between variability of measurement data and quality of estimation results is the final aspect that is investigated in this section. In order to be able to do a direct analysis between variability and results it was decided to define a variability score that can be calculated for the light curves of each satellite and spin case combination. The reasoning behind this score and the equation to calculate the score are given first.

For the variability analysis there are two main parameters of interest. The first is how close two subsequent measurements are to each other, in other words how large the difference is between measurements at two consecutive timesteps. If two consecutive measurements differ from each other by large values, the variability score should increase. This would give light curves with many sharp peaks a higher score, which is exactly one of the aspects of light curves that warrants investigation. The second aspect of interest is the overall range of brightness values that a light curve covers. If a curve only covers a small range of values it should receive a lower score compared to a curve that covers a larger range of brightness values.

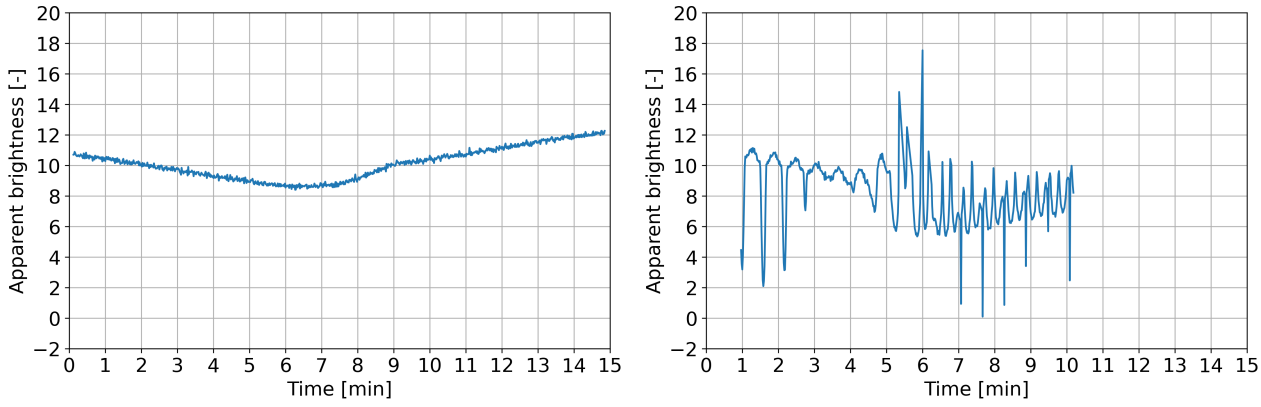
With these aspects as key parameters the variability score calculation is defined as follows. First all the differences between two pairs of consecutive measurements in a light curve are summed together. These are then divided by the total amount of measurements making up the respective light curve. This is done to normalise the scores, as the light curves of the different satellites and spin cases all have different durations, and therefore a different number of measurements included in the curve. The normalised sum of differences is then multiplied by the range of brightness values in the light curve, which is calculated by subtracting the minimum apparent brightness value from the maximum apparent brightness value. Finally, the score is multiplied by a factor of 10 and are rounded to the nearest integer, so that the scores are more convenient to analyse. The calculation of the variability score is summarised by Equation 6.3.1.

$$\text{variability score} = 10 * (m_{\text{app}}^{\text{max}} - m_{\text{app}}^{\text{min}}) \frac{1}{n} \sum_{k=0}^{n-1} |m_{\text{app}}^{t_{k+1}} - m_{\text{app}}^{t_k}| \tag{6.3.1}$$

Here n is the number of measurements making up a light curve, and m_{app} are the apparent brightness values. The superscripts max and min indicate the maximum and minimum value of the light curve, respectively. The superscripts t_k and t_{k+1} indicate measurements at two consecutive timesteps.

An example of two light curves with a different variability score is given in Figure 6.3.16. The ACS3 - Nadir pointing light curve in Figure 6.3.16a is quite flat throughout the full duration of the pass, with no peaks or bumps of fast changing brightness. Additionally, it covers only a small range of brightness values throughout, only attaining values between 8 and 12. This is thus an example of a curve with low variability, and therefore receives a low variability score of 3.

The light curve from the Swarm B - Tumbling case in Figure 6.3.16b on the other hand shows much more variation throughout the visible pass, with many sharp peaks of increased and decreased brightness. Overall the curve also span a much larger range of brightness values, attaining values between 0 and 18. This light curve therefore reaches a higher variability score of 85.



(a) ACS3 - Nadir pointing - Variability score: 3

(b) Swarm B - Tumbling - Variability score: 85

Figure 6.3.16: Two light curves on 14-11-2024 with different variability score.

Next the variability scores for all 54 test cases will be compared to each other. They will be shown per simulation date, for all satellites and spin cases.

Test case variability scores

The variability scores for the light curves of the test cases on 07-09-2024 are shown in Figure 6.3.17. As expected the Nadir pointing cases score very low on variability, as these curves were seen to be mostly flat and covering only a small range of apparent brightness values. The Single-axis spin and Tumbling cases score much higher, with the Tumbling cases for CryoSat 2, Swarm A and B coring the highest overall. Interestingly, the Tumbling cases do not always score higher than the Single-axis spin cases.

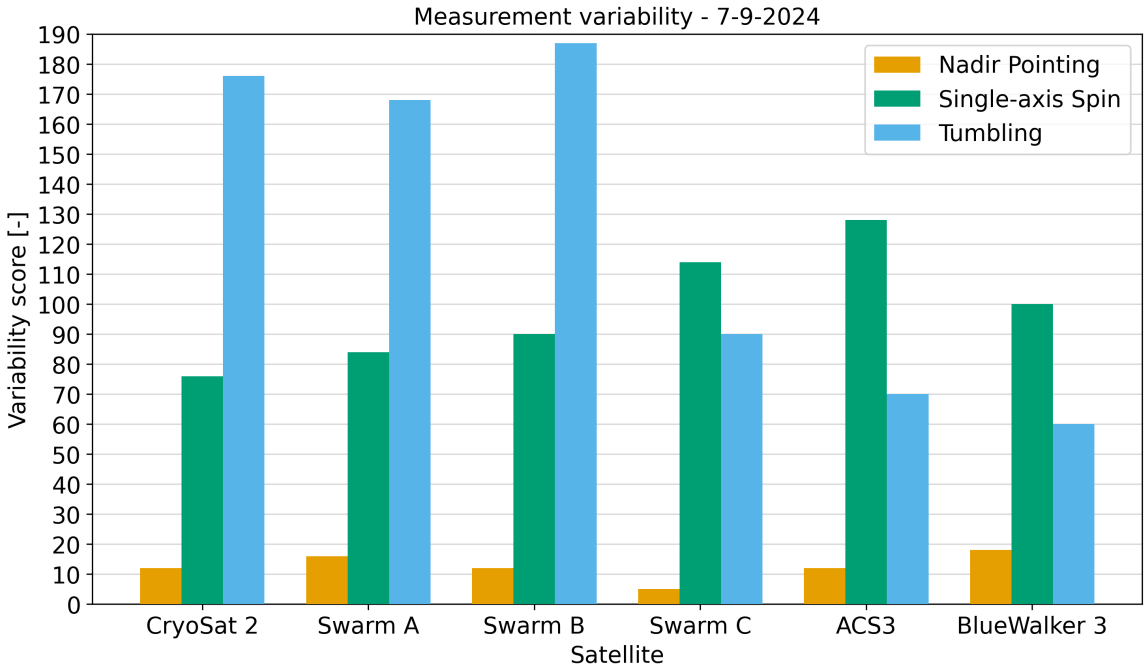


Figure 6.3.17: Variability scores for test cases on 07-09-2024.

In Figure 6.3.18 the variability scores for the test case light curves on 12-10-2024 are shown. It can be seen that overall the light curves score lower than those on 07-09-2024. The Nadir pointing cases still score very low for all satellites. The Single-axis spin cases score higher than the Tumbling cases for all satellites except for Swarm A. To explain why this happens the light curves for the Single-axis spin and Tumbling cases for Swarm C are examined.

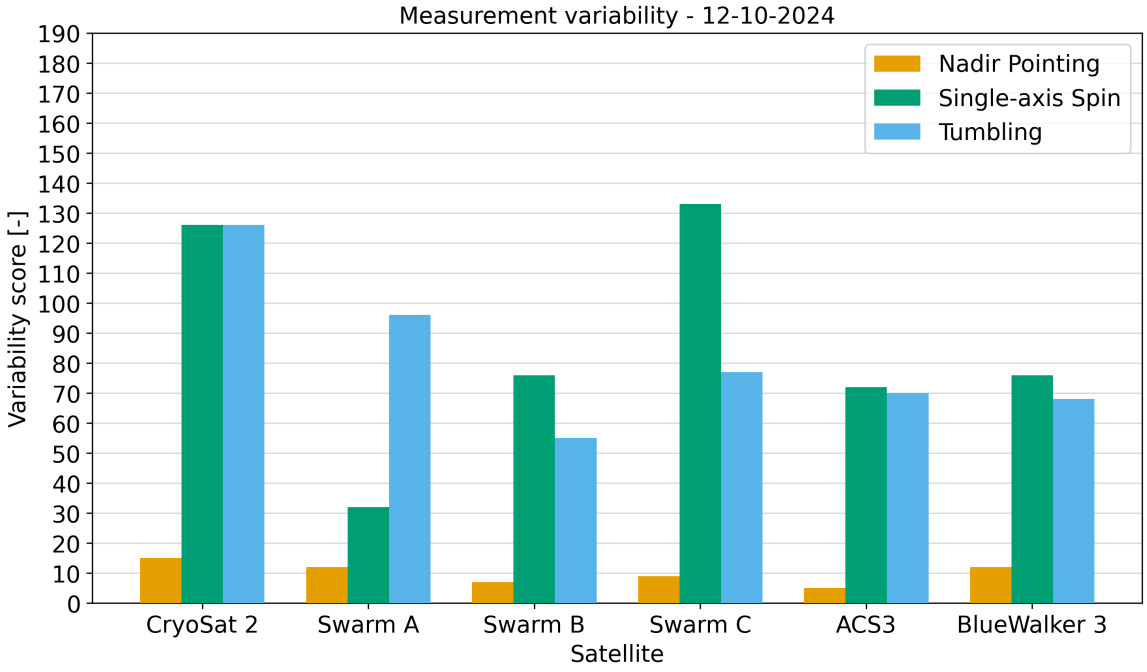
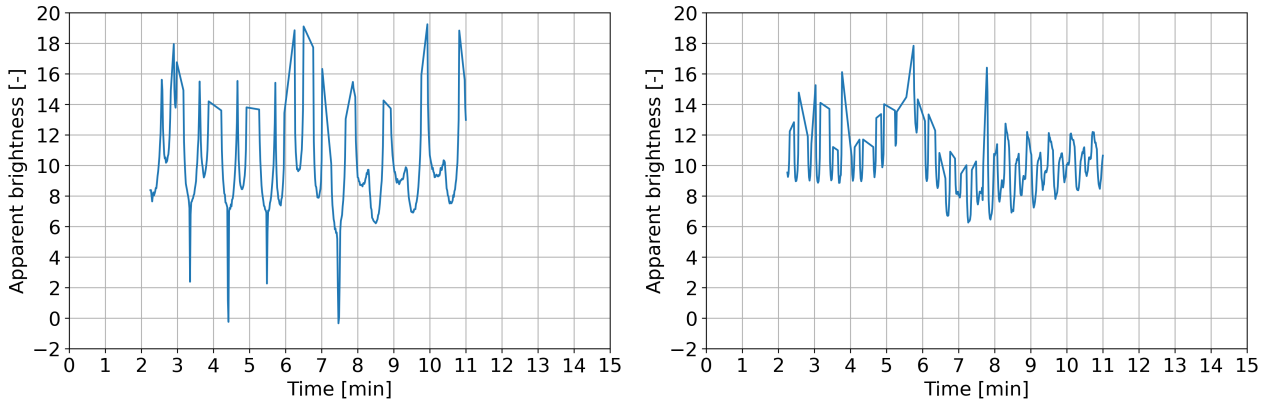


Figure 6.3.18: Variability scores for test cases on 12-10-2024.

The Single-axis spin and Tumbling light curves for Swarm C on 12-10-2024 are shown in Figure 6.3.19. It can be seen that the Tumbling curve in Figure 6.3.19b has faster changing brightness values than the Single-axis spin curve in Figure 6.3.19a. However, the Single-axis spin curves has a few very sharp increases in brightness. These are caused by a short-term increase in received specular reflection, where the observation geometry is aligned such that the specular reflection points directly towards the observer. It appears that for the Tumbling case this alignment does not happen, and the sharp peaks do not appear in the light curve. These sharp peaks and the fact that the light curve covers a larger range of brightness values are the reason that the Single-axis spin light curve has a higher variablity score than the Tumbling curve.



(a) Single-axis spin - Variability score: 133

(b) Tumbling - Variability score: 77

Figure 6.3.19: Light curves for Swarm C satellite on 12-10-2024 - Single-axis spin and Tumbling

Lastly, the variability scores of the light curves from the test cases on 12-11-2024 are shown in Figure 6.3.20. It can be seen that for half of the satellites the Tumbling cases score higher than the Single-axis spin cases, and for the other half the opposite is the case. The Nadir pointing cases still have very low variability scores, just like for the other dates.

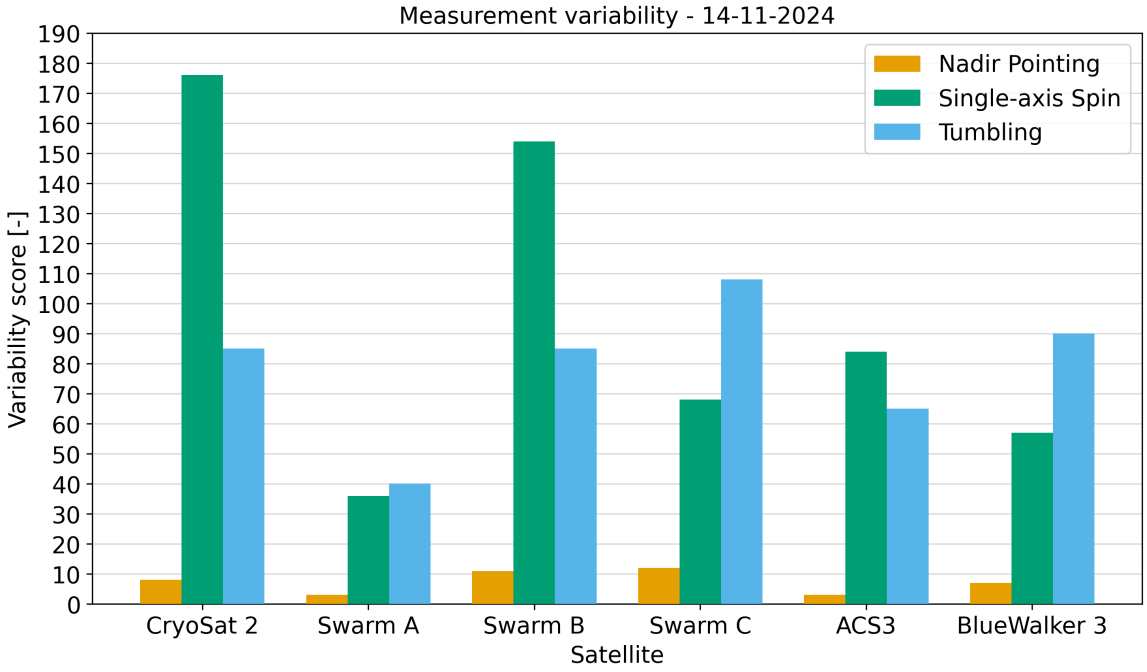


Figure 6.3.20: Variability scores for test cases on 14-11-2024.

Overall there is a large variety in variability scores for all the different test cases. Next it will be analysed if there is a relationship between variability scores and successful shape characterisation and attitude estimation.

Shape characterisation

The first aspect that is analysed is the shape characterisation. To get a wider view of the relation between variability and successful shape identification, the results for all of the cases obtained in the initial guess analysis from Section 6.2 were included in this analysis. The results were divided up in variability ranges of variability score size 10, and it was counted how many times the MMAE algorithm was able to correctly identify the true shape in each variability range. The results of this analysis is shown in Figure 6.3.21.

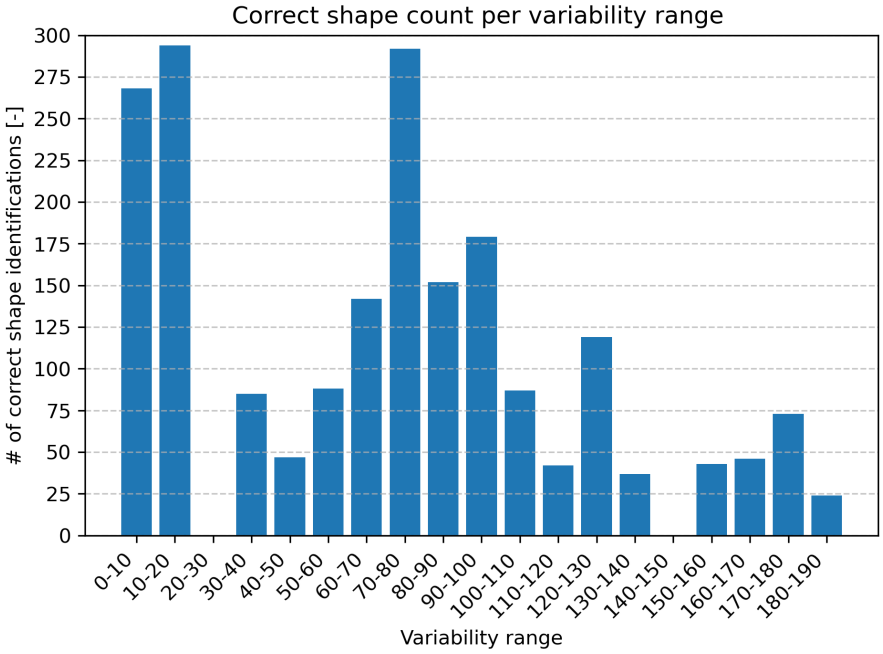


Figure 6.3.21: Count of correct shape identifications per variability score range.

From these results it looks like estimations with low variability scores are more likely to get a correct shape identification, with only scores between 70 and 80 correctly identifying shapes a similar number of times. In actuality this plot gives a skewed view of the results. To see this the number of light curves that have a variability score within each variability range are plotted in Figure 6.3.22. This plot has almost the exact same pattern as the previous plot. To get a proper idea of the relation the results in Figure 6.3.21 are normalised by dividing the counts by the results from Figure 6.3.22. The normalised results are shown in Figure 6.3.23.

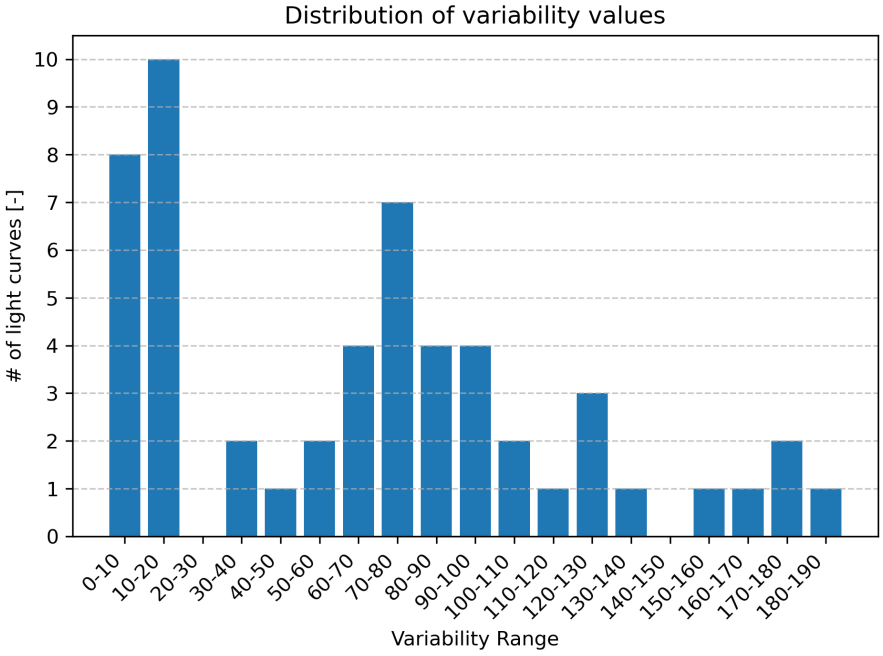


Figure 6.3.22: Number of light curves per variability range

The normalised results look vastly different from the first plot. On average the cases with the lowest (0-20) and highest (180-190) variability scores identified the correct shape the fewest times. The results for all scores in between are very similar, mostly ranging between 35 and 45 correct identifications. These results only indicate a slight relation between variability score and shape characterisation, with scores ranging between 30 and 180 performing slightly better than scores above and below this range.

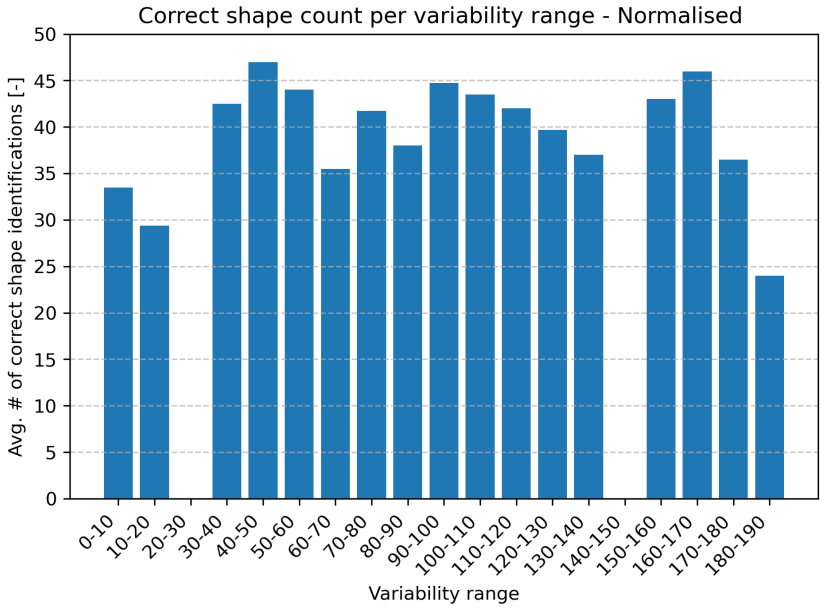


Figure 6.3.23: Normalised count of correct shape identifications per variability score range.

Attitude estimation

In order to analyse the relation between variability and accuracy of attitude estimation results the attitude RMSE values are plotted in Figure 6.3.24 against the variability score for the respective test case for the 54 baseline estimation results for the updated MMAE algorithm. The RMSE values are shown for all three Euler angles separately, to give a wider picture of the error values. The y-axis of the plot is given in a logarithmic scale. This is done to make it easier to compare the magnitudes of the errors for the different variability scores.

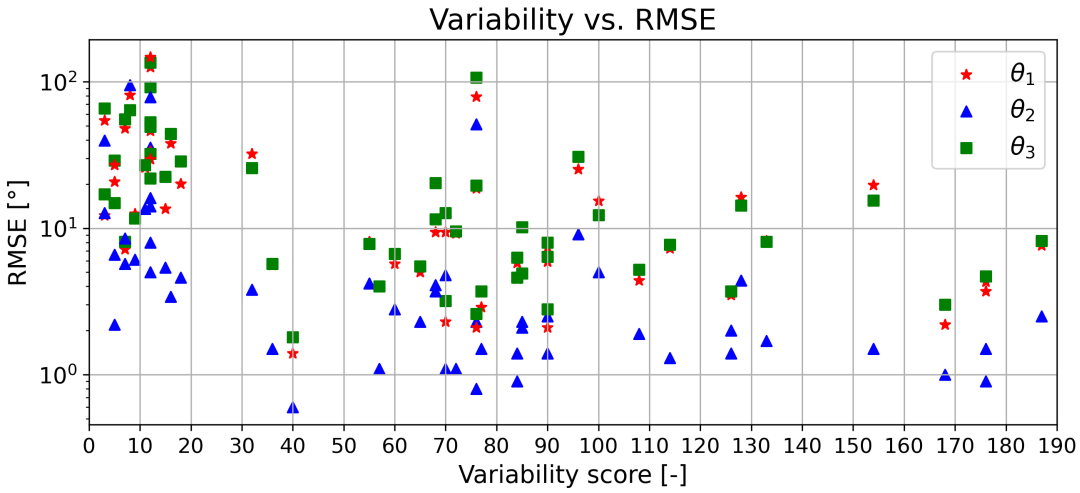


Figure 6.3.24: Euler angle RMSE plotted against variability score of the respective test case.

It can be seen that for low variability scores, between 0 and 20, the majority of error values lie between magnitudes 10° and 100°. Only a few times the errors are below 10°, and those times are mostly for the θ_2 angle, which can be seen to have lower errors for all variability scores. For variability scores higher than 20 most errors lie between 1° and 10°. A few times the errors are even below 1°, but again only for the θ_2 angles. For a few cases the error values for higher variability scores are above 10°. From this plot it can be concluded that variability of light curve measurements is an important factor impacting the accuracy of attitude estimation results.

6.4. Algorithm validation

At the start of this thesis (Chapter 2) the lack of validation with real measurement data of methods for shape characterisation and attitude estimation of space objects through light curves was identified as one of the research gaps in literature. One of the goals for this thesis was to perform such validation for the methods implemented in the thesis. In this section the setup and results of this validation are discussed.

However, first it has to be stated that performing the validation proved more challenging than initially expected. The validation was planned to be done towards the end of the thesis work, and due to setbacks there was less time available than desired to properly perform this validation. Additionally, only an attempt was made to validate the updated MMAE algorithm, since this is the algorithm that most of the results in Chapter 6 are focused on. The results that were achieved are still presented in this section, to highlight the difficulties inherent to the validation, and provide extra input to the discussion and recommendations for future research in Chapter 7.

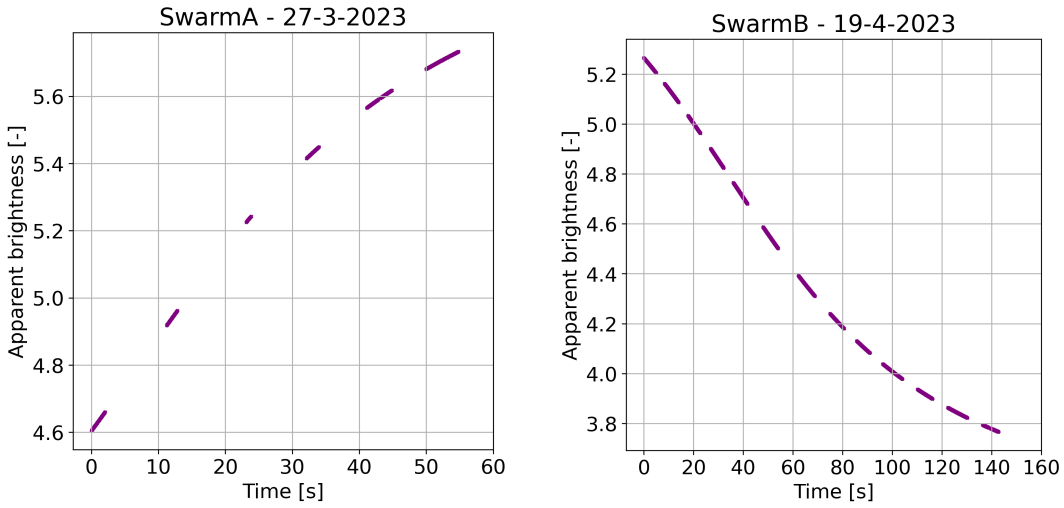
In Subsection 6.4.1 it is discussed how the validation was set up, highlighting the measurement and truth data that were used to validate the MMAE algorithm, as well as the settings of the estimation parameters that were used to set up the estimations. Subsection 6.4.2 covers the results of the respective validations.

6.4.1. Setup

Measurement data

Measurements were available for the Swarm A and B satellite on multiple dates. The measurements consisted if apparent brightness measurements, as well as azimuth, elevation and range data. The range data in the measurement files were obtained from TLEs. The time between consecutive measurements in the data files is 0.04s. However, there are several time gaps present in the data, so there are periods with no measurements in between.

Results for the validation will be shown for a single date for both satellites. For Swarm A the measurements for 27-03-2023 were used for validation, for Swarm B the measurements on 19-04-2023. The raw measurement data can be found in the GitHub repository. Here the truth data that was used in the validations are also located. The real brightness measurements for both validation cases are shown in Figure 6.4.1.



(a) Swarm A

(b) Swarm B

Figure 6.4.1: Real apparent brightness measurements of Swarm A and B used in validation estimations

Truth data

To calculate estimation errors truth data of the orbit and attitude states are required. Both are publicly available on an online ESA database[1]. However, this data is stored in different reference frames than used in the estimations. For example, the orbit data is stored in the ITRF frame, while the estimation is performed in the ECI frame.

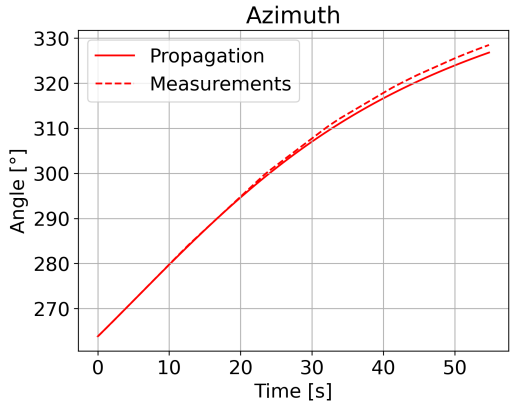
With some assistance it was possible to convert the attitude truth data into body-axis quaternions, which is the same frame in which the attitude is estimated in the MMAE algorithm. This data was then linearly interpolated

[1] https://earth.esa.int/eogateway/missions/swarm/data

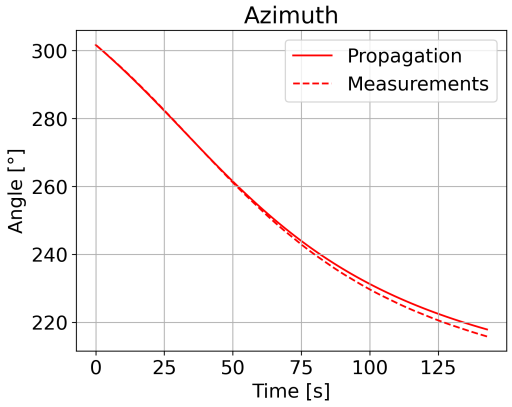
to obtain data for all the time points required. However, in the limited time available it was not possible to convert the orbit data to the ECI frame such that azimuth, elevation and range data from the measurement files fit the truth data. The following was done to still have truth data to calculate estimation errors with.

The first data entries for the azimuth, elevation and range data were converted into an initial orbit position state in the ECEF frame, which was then converted into the ECI frame. This was also done for the next data entry. The two position states were then used to calculate the initial orbit velocity, by subtracting them from each other and dividing by the time between the data entries. The initial orbital position and velocity were then propagated in Tudat for the full duration of the measurement files. This resulted in a 'pseudo-truth' orbit, which was used to calculate the estimation errors of the respective validations.

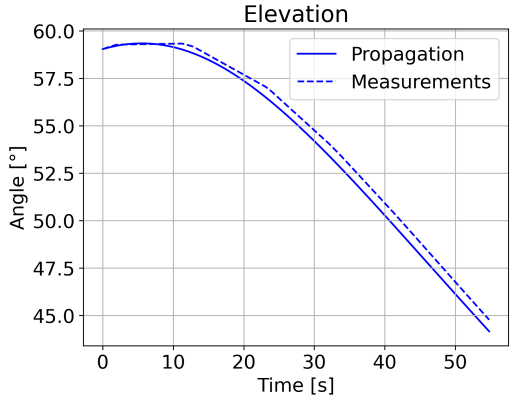
To justify this approach the azimuth, elevation and range values obtained from the propagated pseudo-truth orbit are plotted next to those from the measurement files. For both the Swarm A and B these plots are shown in Figure 6.4.2. In all the figures it can be seen that the values for the propagation and the measurements are initially very close to each other, but diverge slightly towards the end of the time periods. The differences are however considered to be small enough to allow the pseudo-truth orbit to be used as truth data in the validation.



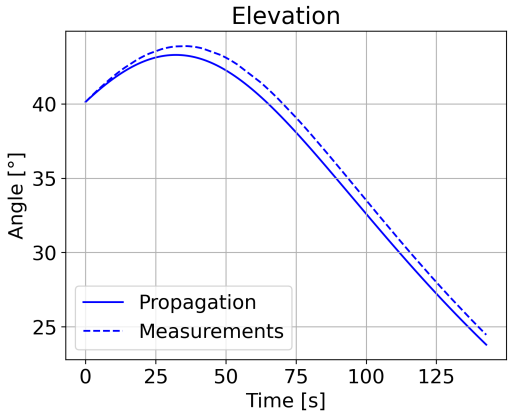
(a) Azimuth - Swarm A



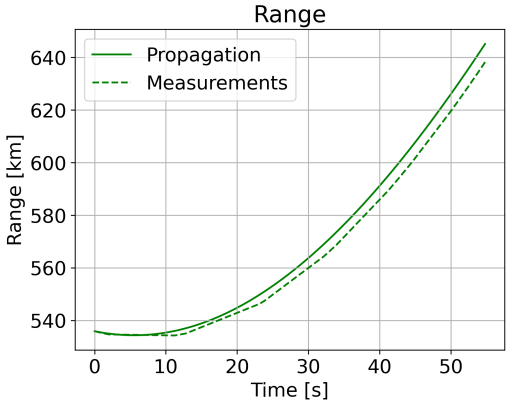
(b) Azimuth - Swarm B



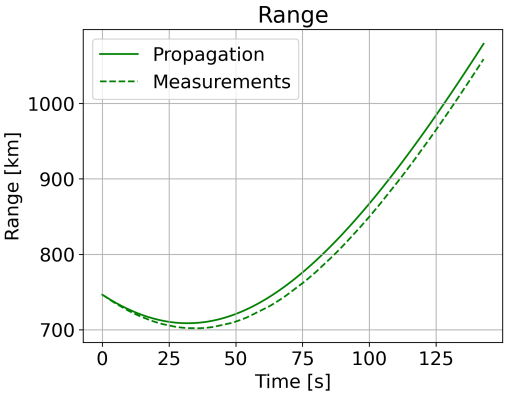
(c) Elevation - Swarm A



(d) Elevation - Swarm B



(e) Range - Swarm A



(f) Range - Swarm B

Figure 6.4.2: Propagated angles and range values compared to measurement data for Swarm A and Swarm B.

Estimation parameters

While the measurements data is available at time steps of 0.04s, it was decided to run the validation estimations with a time step of 0.4s. This was done because it was found in Subsection 6.3.2 that low values for the measurement timesteps sometimes lead to very high error results. The cause for this was not exactly determined, so to avoid this behaviour the slightly higher estimation time step was selected.

The validation estimations were performed with the same model bank as used in the testing, defined in Chapter 5, with the sphere models excluded, because this is the same model bank with which the best results for the MMAE algorithm were obtained in Subsection 6.1.4. The model bank thus consists of 20 models.

The next parameters are the covariance matrices. For the initial covariance matrix $\boldsymbol{P}_0$ and the process noise covariance matrix $\boldsymbol{Q}$ the same values that were used to produce the results in the previous sections were kept for the attitude and rotational velocity values. For the orbital position and velocity the values in the $\boldsymbol{P}_0$ matrix were slightly adjusted. Since the difference between the measurements and pseudo-truth orbit grows over time, it was decided to reflect this in the $\boldsymbol{P}_0$ matrix. For the orbital positions a σ value of 10 km was assigned, and for the orbital velocities a value of 10 m/s. The covariance matrices $\boldsymbol{P}_0$ and $\boldsymbol{Q}$ are given by Equation 6.4.1.

$$\boldsymbol{P}_0 = \text{diag}\left[(10\text{km})^2 * I_3 \quad (10\text{ m/s})^2 * I_3 \quad 0.2^2 * I_3 \quad (3.1 * 10^{-4}\text{ rad/s})^2 * I_3\right] \tag{6.4.1a}$$

$$\boldsymbol{Q} = \text{diag}\left[(100\text{ m})^2 * I_3 \quad (0.1\text{ m/s})^2 * I_3 \quad (2 * 10^{-4})^2 * I_3 \quad (10^{-12}\text{ rad/s})^2 * I_2 \quad (10^{-5}\text{ rad/s})^2 * I_1\right] \tag{6.4.1b}$$

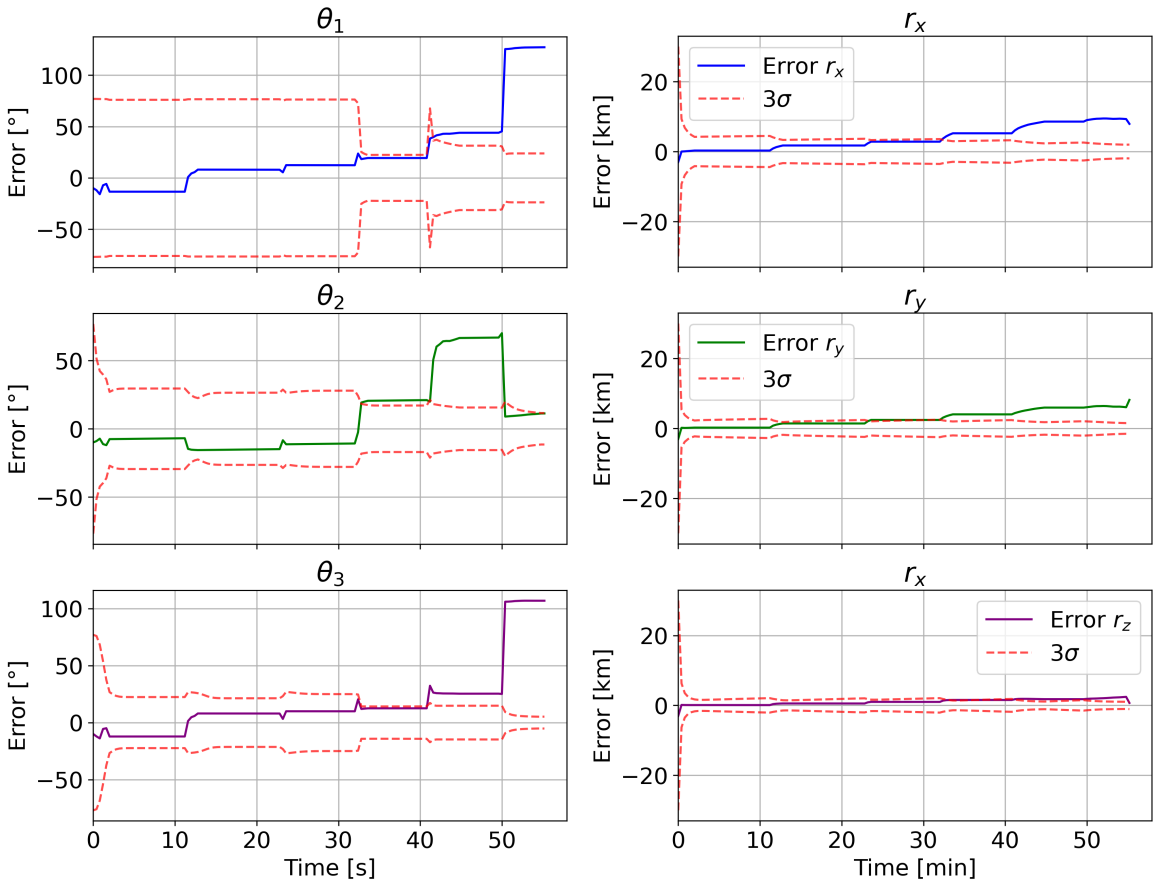
The last matrix that is defined is the measurement noise covariance matrix $\boldsymbol{R}$. To reflect the differences between the measurements and the pseudo-truth orbit, the σ values for the azimuth and elevation angle measurements were set to 0.5°, or about 0.0087 radians. The range noise standard deviation was set to 1.0 km. For the brightness measurements the standard deviations differ for each measurement. However, the MMAE algorithm as implemented works with fixed measurement noise values. Therefore it was decided to assign the average standard deviation value from each validation case, so Swarm A - 27-03-2023 and Swarm B - 19-04-2023, as the respective values in the noise covariance matrix $\boldsymbol{R}$. This was done both because there was not enough time to update the algorithm, and because the algorithm was verified and tested with this functionality, so for proper validation it is best to keep the algorithm unchanged. For the Swarm A validation the average standard deviation is 0.135, and for Swarm B it is 0.183.

The initial orbital states guess that were used as input for the estimations are 3 km offset from the initial positions of the pseudo-truth orbit, and 1 m/s for the initial velocities calculated from the first two pseudo-truth orbital positions. The attitude initial guess was set to be the true initial attitude from the interpolated ESA data, offset by 10° for each Euler angle. Since the true initial rotational velocities of the real satellite are very small, the initial rotational velocity guess was set to 0.0001 rad/s for all three body-axes.

6.4.2. Results

Swarm A - 27-03-2023

In Figure 6.4.3 the estimation errors for the attitude and orbital position states for the Swarm A validation estimation are plotted. In both plots the errors initially stay around 0 error, but diverge later on in the estimation. Especially the attitude estimation diverges by a lot, ending with errors larger than 100° for the θ_1 and θ_3 angles. The final orbital position errors are not as large, but do go beyond the covariance bounds of the estimation.



(a) Euler attitude error

(b) Orbital position error

Figure 6.4.3: Attitude and orbital position errors for the Swarm A validation estimation

These results are much less accurate than were achieved with the simulated data for the test cases in the previous sections of this chapter. To explain this first the MMAE weights for the estimation in Figure 6.4.4 are looked at.

It can be seen that the algorithm assigns higher weights to many different models throughout the estimation. Only at the end of the estimation it converges on a single model. This is not the model the Swarm satellites were modelled as, but it is a model of the correct shape type, namely a triangular prism. The algorithm is thus able to get close to the shape of the Swarm satellites from the real measurements.

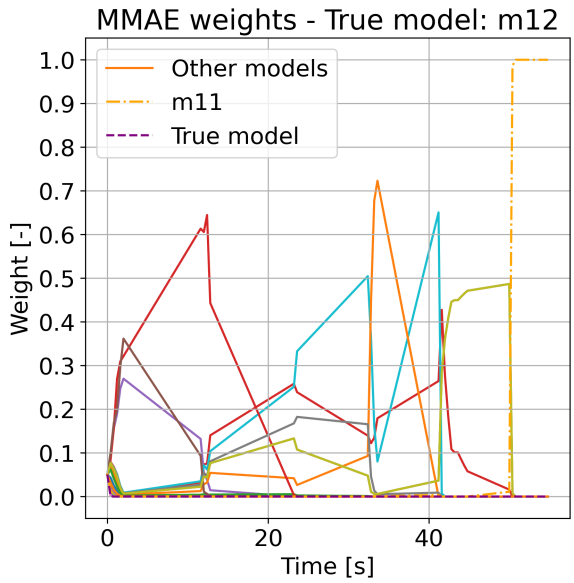


Figure 6.4.4: MMAE weights for Swarm A validation estimation

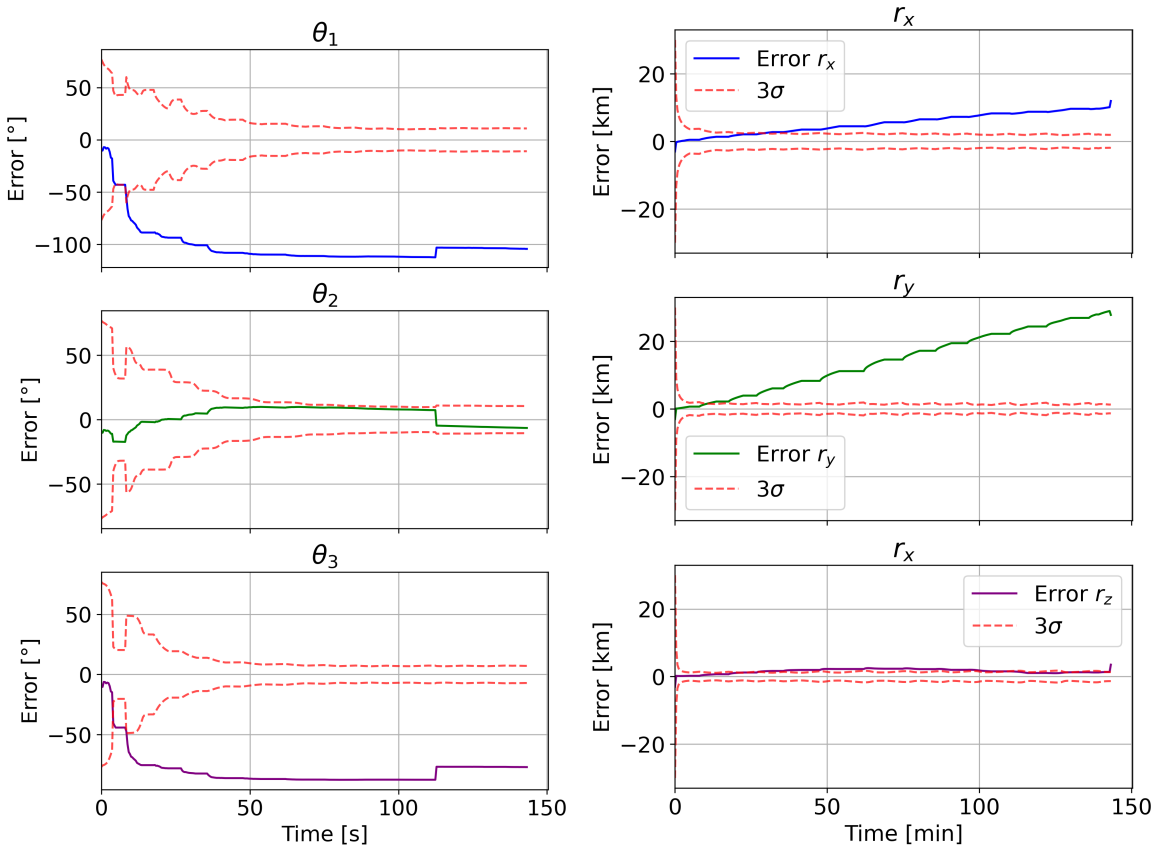
It does need to be kept in mind that the real Swarm satellites are not completely shaped like a triangular prism, so there is a natural mismatch between the real measurements and the predicted measurements calculated in the MMAE algorithm, which has a significant impact on how well the algorithm can characterise the object shape and estimate the attitudes. This is reflected in the attitude error plot in Figure 6.4.3a. The reason the errors are so high is most likely caused by the mismatch between the simplified shape model of the Swarm satellite and the real shape of the satellite from which the measurements originate. This means that even though the algorithm performs a separate attitude estimation for each shape in the model bank, the difference between simulation and reality are too big to lead to accurate results.

The reason that the orbital position errors in Figure 6.4.3b are less extreme, but still high, is most likely caused by the differences between the pseudo-truth and measurements as seen in Figure 6.4.2. The measurements the MMAE algorithm uses for the estimation are made from the real orbit, and since the pseudo-truth orbit is not the same as this orbit the estimation errors that are calculated with the pseudo-truth will be larger. The errors in Figure 6.4.3b are around the same order of magnitude as the differences in range values in Figure 6.4.2e, so it can safely be assumed that this is indeed what causes the large errors. This means that the estimated orbital positions could in fact represent the true orbit better than the pseudo-truth orbit. This would however need to be confirmed with the true orbit data from ESA, which was unfortunately not possible in the remaining time for the thesis.

Swarm B - 19-04-2023

The errors for the estimated attitude and orbital positions for the Swarm B validation estimation are shown in Figure 6.4.5. It can be seen that the attitude errors very quickly diverge from the truth, with only the θ_2 errors ending within the covariance bounds.

The orbital position errors also diverge from the pseudo-truth orbit, with larger errors than the Swarm A validation estimation. However, it can be seen in Figure 6.4.2f that the differences between the range values of the pseudo-truth orbit and the measurements are also larger than those for Swarm A. This difference is reflected by the higher position errors in Figure 6.4.5, since these are calculated with the pseudo-truth orbital positions. Again, proper validation with the ESA orbit data would be required to provide better insight into the orbital position errors.



(a) Euler attitude error

(b) Orbital position error

Figure 6.4.5: Attitude and orbital position errors for the Swarm B validation estimation

To explain the high attitude errors the estimation weights in Figure 6.4.6 are examined. For almost the entire estimation the weights are quite low. This means the algorithm has low confidence that any one model is the correct model, because the estimations do not fit the measurements well. Only at the end the algorithm converges on one model, which is a model of the wrong shape type, namely a box-wing satellite. This is a similar result as what was seen for Swarm A, where the simplified shape model used in the model bank does not match reality close enough, so when real measurements are used in the estimation the MMAE algorithm is not able to find the correct model, and is thus also not able to accurately estimate the attitude.

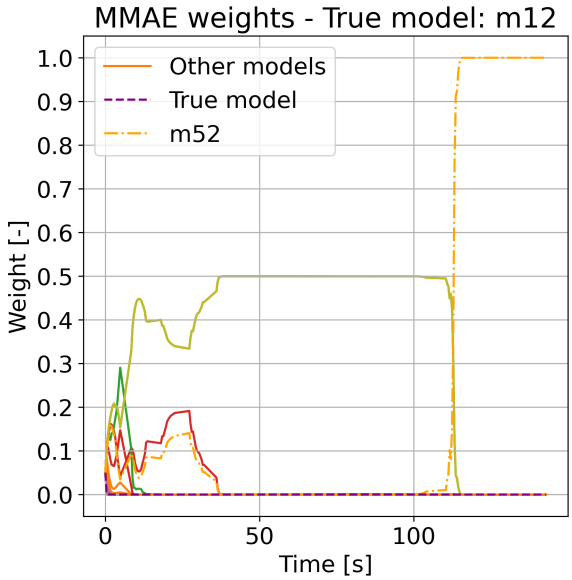


Figure 6.4.6: MMAE weights for Swarm B validation estimation

7 Conclusions

In this thesis the problem of object characterisation and attitude estimation of LEO objects through light curve inversion stood central. In particular, the main research question that drove the activities performed throughout the thesis period was the improvement of existing techniques for this problem. The first step that was taken was determining the objects and orbits on which the characterisation and estimation would be performed in Chapter 3, as well as setting up the simulation environment in which this would be done.

Next, in Chapter 4 the method for simulating light curve measurements was implemented, namely the Ashikmin-Shirley BRDF model. After this a method that is commonly used for the characterisation problem in literature called MMAE was implemented. One of the research sub-questions was centred around the limitations of fixed model estimation methods like the MMAE algorithm. Therefore two new methods were implemented, called MGS and MGen, that are variable model estimation methods. In the following chapter, Chapter 5, the test cases with which the three algorithms would be tested and analysed were defined, as well as the strategy as to how the testing would be performed. Finally, in Chapter 6 the results of this testing were presented and explained.

In this chapter the conclusions that were made based on these results are presented, with the goal to answer the research questions that drove the thesis research. First, in Section 7.1 the three research sub-questions will be answered. Then in Section 7.2 an answer to the main research question will be given. In Section 7.3 a discussion of the results and conclusions is presented. Section 7.4 closes off the thesis by giving recommendations for future research that could be undertaken to further advance the field of object characterisation and attitude estimation of LEO objects through light curve inversion.

7.1. Answers to sub-questions

In Chapter 2 the main research question was drafted based on the knowledge gaps identified in literature. Since this question was quite broad, three supporting sub-questions were set up to guide the research activities of the thesis. These questions, labelled RQ-1, RQ-2 and RQ-3, are individually answered in this section.

7.1.1. Fixed model bank limitations

The first sub-question addresses the limitations that inherent to fixed model estimation techniques like MMAE:

RQ-1: How can the limitations of using a fixed model bank in estimations be overcome?

The VSMM type algorithms MGS and MGen were implemented in order to see if variable model estimation methods could outperform traditional fixed bank estimators such as MMAE. The MGen algorithm, which generates new object shapes based on model likelihoods determined from measurements, does not perform well in its current form on the test cases. The algorithm achieves poor results in both object characterisation and attitude estimation. One main reason for this is that models that score below a certain likelihood threshold are removed from the model bank. This often resulted in the model of the right shape type to be removed from the model bank, which made identifying the true shape impossible. Even if the correct shape type is not removed from the model bank, the algorithm does not yet have the capability to converge on the correct dimensions of the true shape. Major reworking of the algorithm would be required to improve its capabilities, but even then it is unsure if this method could improve on the results of the established MMAE algorithm.

The MGS algorithm shows more promise, but still underperforms compared to the MMAE algorithm. The algorithm is computationally more efficient than the MMAE algorithm because it only runs smaller groups of models in parallel, compared to the MMAE algorithm which runs all models in the model bank simultaneously. However, the idea behind the MGS algorithm is that different model groups are activated based on which shape model in the central model group best fits the measurements at a certain time. Unfortunately, it often occurs that the algorithm initially activates the incorrect model group, and does not switch back to the central model group, making it impossible to identify the true shape. The algorithm was however able to correctly find the true shape for quite a few cases, and has potential to be an algorithm that can effectively be used in the shape characterisation field. Further development would be required to further improve the results of this algorithm on the object characterisation and state estimation through light curve problem.

Lastly, the limitations of the MMAE algorithm were addressed for the algorithm itself. It was found that removing the fusion of weighted average attitude state estimation resulted in significant improvement of performance of the MMAE algorithm, with the adjusted algorithm finding the correct shape for almost all test

cases with low attitude errors. With the fusion of attitude states removed each for shape model in the model bank a separate attitude estimation is performed, instead of one averaged estimation for all shape models. This makes it possible for the MMAE algorithm to find the true shape much more often, as the estimated states for incorrect models will over time diverge much more from the measurements than the shape models that are close to the true shape. This proved to be a very effective change to the existing MMAE algorithm, and is considered as the most important result obtained in the thesis. The remaining sub-questions can be seen as a sensitivity analysis on the baseline results of the test cases achieved by the MMAE algorithm.

7.1.2. Initial guess dependence

The next sub-question focuses on the initial guess of the estimations, specifically:

RQ-2: What is required of the initial guess to ensure good convergence of estimation?

To answer this question only the performance of the improved MMAE algorithm was analysed, as this model performed much better than the newly implemented VSMM methods, and because the main research question focuses on existing techniques, which only applies to the MMAE algorithm. It was analysed what the effects on the baseline results were of changing the offset of the initial guess from the true initial state for the attitude and rotational velocities. In essence this analysis functioned as a sensitivity analysis of the results on the initial guess used in the estimations. The effect of changing the initial attitude and rotational velocity guess was investigated separately.

It was deemed that estimations with average error values below 20° were accurate enough to be considered good estimates. The analysis found that the different spin cases allow different ranges of offset values from the true initial states that still produce results that are considered accurate. The table containing the offsets for each spin case is repeated in Table 7.1.1. The Nadir pointing case results showed to be the most sensitive to changes in the initial guess, requiring an initial guess that is close to the true initial state to produce estimations with errors below the limit of 20°. The Single-axis spin and Tumbling case results proved to be more robust, allowing larger ranges for both the initial attitude and rotational velocity offsets that still lead to accurate results.

Table 7.1.1: Ranges of offsets from true initial state resulting in accurate estimations, per spin case.

Spin case	Attitude offset range [°]	Rotational velocity offset range [°/s]
Nadir pointing	[-5.0, 5.0]	[-0.01, 0.02]
Single-axis spin	[-7.5, 7.5]	[-0.05, 0.05]
Tumbling	[-12.0, 12.5]	[-0.045, 0.045]

7.1.3. Measurement data requirements

The last research sub-question addresses the measurement data with which estimations are made:

RQ-3: What are requirements on measurement data (e.g. noise, rate, variability) to ensure correct estimates?

To answer this question two types of analyses were performed. For the noise and data rate a similar analysis as for the initial guess dependence was performed, where estimation were run with values for the noise and data rate that were different from the values used in the baseline estimations. In both analyses a similar result as the initial guess dependence result was found, namely that the Single-axis spin and Tumbling case results are more robust to changes to the parameters used to simulate the measurements, producing estimations with low attitude errors for a large range of data rate and apparent brightness noise values. The Nadir pointing results again proved to be less resilient, producing estimations with much higher errors for different values of the measurement parameters. Interestingly, it was found that for all spin cases the lowest values of data rate and measurement noise do not directly lead to more accurate results.

The link between measurement variability and estimation results was investigated on two aspects. To allow this analysis a variability score was defined, that was calculated for the light curve of each of the 54 test cases. First the relation between this variability score and the ability of the MMAE algorithm to correctly identify the true shape was examined. This was done by looking at all the results from the initial guess dependence analysis, to have a larger set of data to investigate. It was found that there is only a small direct link between variability score and successful shape characterisation, with low variability scores resulting in slightly fewer cases with successful shape identifications.

The link between attitude estimation accuracy and variability score was only examined for the baseline results of the 54 test cases from the improved MMAE algorithm. Here it was seen that there exists a stronger relation between measurement variability and estimation errors, with higher variability scores consistently leading to

results with lower attitude errors. This relation was indirectly already seen in the initial guess dependence and the noise and data rate analysis results. The Single-axis spin and Tumbling case light curves have higher variability scores than the Nadir pointing curves for all test cases. This is the reason that the Single-axis spin and Tumbling case results are more resilient to changes in the simulation parameters, as light curves with more variability have more information for the MMAE algorithm to use to produce accurate attitude estimations.

7.1.4. Validation

Lastly, it was attempted to validate the MMAE algorithm with real measurement data for the Swarm A and B satellites. Lack of validation with real measurement data was namely identified as one of the research gaps in the literature. For both satellites a validation estimations was performed on one observation date.

Both validation estimations were unsuccessful in both shape characterisation and state estimation. For the Swarm A estimation the MMAE algorithm identified a shape of the correct shape type as the simplified Swarm satellite model, but with high attitude and orbital position errors. For the Swarm B estimation the algorithm found a completely incorrect shape, namely a box-wing model, and produced even larger estimation errors. These unsuccessful validations highlight the differences between reality and simulation, and difficulties in applying estimation methods like MMAE for real-life applications, as will be further discussed in Section 7.3.

7.2. Answer to main research question

The main research question driving the tasks performed in the thesis research is the following:

What improvements can be made on existing techniques for space debris object characterisation of LEO objects through light curve inversion to estimate attitudes and characterise shapes?

In this thesis it was found that adjusting the existing technique MMAE such that the attitude of each shape model in the estimation model bank is performed separately in the algorithm, without fusing the attitude estimates of all the shapes, leads to improved results on the shape characterisation and attitude estimation of LEO test objects for simulated measurements. Extensions of this technique to a VSMM method like MGS and MGen proved to be ineffective at improving on these results. Validation of the improved MMAE algorithm with real measurement data was unsuccessful, and highlighted difficulties that still need to be overcome in order to use the method in real-life applications.

7.3. Discussion

Range measurement simulation

There are a few aspects of the results and conclusions that need to be addressed. The first is that all the estimation results obtained in this thesis included simulated range measurements. As mentioned in Subsection 4.3.4, in reality most apparent brightness observations are made with passive optical sensors, that do not measure range. In the limited time available it was not achieved to get the implemented methods working without including these range measurements however. For further development and research on these methods this would be an important aspect that would need to be solved to make the methods applicable to practical use in real estimations.

Validation

Secondly, the validation attempts of the MMAE algorithm highlighted a crucial shortcoming to the results obtained. Some of the shapes included in the model bank that represented real satellites, like the Swarm and CryoSat 2 satellites, were modelled in a highly simplified way using basic shapes. In reality these satellites have a more complex and detailed shape, which would produce different measurements than the simplified models. The validation attempts showed that this disconnect between the simplified shape models and real measurements is large enough to lead to poor results when using real data.

Relevance of research

The last aspect that needs to be addressed is the title of this thesis: *Space Debris Object Characterisation through Light Curve Inversion*. The first two words are space debris, but in all the test cases and results only existing, active satellites were looked at, no debris objects. The question thus stands: how is any of the research in this thesis relevant to space debris?

There are two parts to answering this question. The first is that real satellites, like the ones investigated in this thesis, can turn into debris objects once they are no longer active. This could happen due to a number of reasons, like critical malfunctions leaving the satellite uncontrollable, or simply because a satellite has exceeded its lifetime and becomes no longer active. Once these satellites are no longer actively controlled, non-conservative forces like aerodynamic drag and solar radiation pressure, and torques caused by these forces, will overtime

lead to these objects drifting from their original orbit and attitude states. Since they are no longer active, information on these new states is no longer downlinked to the ground, making these states unknown. This is what makes techniques like shape characterisation and state estimation through light curves important, as they allow these states to be estimated so that predictions can be made about their future states, which is crucial for the operational of other still-functioning satellites.

Additionally, satellites could be impacted by other debris objects, both while they are still active and when they have become debris objects. These impacts are always high-energy collisions, which can drastically alter the orbit and attitude states of a satellite, as well as its shape. This is again where the techniques covered in this thesis become relevant, as they make it possible to estimate these post-collision states and characterise the shape of the satellite after these collisions.

The second part of the answer to why the research in this thesis is relevant is the fact that the methods covered are still very much in development, and have mostly only be used in theoretical settings. In order to become proper methods that could be used to characterise the shapes and estimate the states of real debris objects these methods need to be tested and validated. It is basically impossible to use real debris objects for this, because their true shapes and states are almost always unknown. It is thus not possible to assess whether these methods work properly, unless they are tested and validated on objects for which the shapes and states are known, which are currently active satellites.

All this being said, there are still a lot of aspects on which the methods implemented in this thesis could be improved. A few of these aspects are covered in the next and final section of this thesis.

7.4. Recommendations for future work

There is still a lot of potential to improve the methods discussed and implemented in this thesis to tackle the problem of object characterisation and attitude estimation. The aspects of most importance or with the most potential are presented here.

VSMM

The variable model methods implemented in this thesis did not perform better than the existing fixed model method MMAE. However, they could be improved in different ways to potentially match or even improve on this algorithm. The MGS method for example might deliver better results if a delay was implemented in the mechanism of switching between model groups. Doing this would give the algorithm more time to identify which model in the central group best fits the measurements, before switching to this group. The MGen algorithm needs a lot of refining before it performs as well as the MMAE or MGS algorithms. One such refinement could be to run estimations of different shape types completely parallel to each other. This would result in many models generated for each shape type, which could improve the chances of the algorithm finding the correct shape.

Glint

A major obstacle in the attitude estimation is the small range of initial guesses that result in good convergence of estimations, as was seen in the initial guess dependence analysis. One method that could improve on this aspect uses the sharp peaks in light curves caused by short dominance of specular reflectance, called glint. Matsushita et al. [29][30] use this information in the light curve to constrain the possible attitudes the objects can attain, and project states that are outside of this constraint on the constraint boundary. They found that they could correctly estimate attitudes that started with much larger initial attitude errors.

Validation

As has been mentioned multiple times, the methods in this thesis will only become of practical use for real-life applications once they have been validated with real measurement data. Future research should focus on what is required to ensure estimations run with real data produce accurate estimation results and correct shape characterisations. One way to do this would be to include more detailed shape models of objects from which measurements are made, like the Swarm and CryoSat 2 satellites. This would bring reality and simulation closer together, making it possible to identify the areas in the methods that need to be further improved.

References

[1] Nodir Adilov et al. "An Analysis of Launch Cost Reductions for Low Earth Orbit Satellites". In: *Economics Bulletin* (Mar. 2024). URL: https://www.researchgate.net/publication/379335390_An_Analysis_of_Launch_Cost_Reductions_for_Low_Earth_Orbit_Satellites_accepted_for_publication_Economics_Bulletin_2022.

[2] ESA Space Debris Office. *ESA's Annual Space Environment Report*. Tech. rep. Version 8.0. European Space Agency, July 2024. URL: https://www.sdo.esoc.esa.int/environment_report/Space_Environment_Report_latest.pdf.

[3] Claude R. Phipps et al. "A Laser Optical System to Remove Low Earth Orbit Space Debris". In: *Proceedings of the 6th European Conference on Space Debris*. Ed. by L. Ouwehand. Vol. 6. 1. Conference paper. ESA, 2013. URL: https://conference.sdo.esoc.esa.int/proceedings/sdc6/paper/29/SDC6-paper29.pdf.

[4] Charles J. Wetterer and Moriba Jah. "Attitude Determination from Light Curves". In: *Journal of Guidance, Control, and Dynamics* 32.5 (2009), pp. 1648–1651. DOI: 10.2514/1.44254. eprint: https://doi.org/10.2514/1.44254. URL: https://doi.org/10.2514/1.44254.

[5] Ingrid van Houten-Groeneveld and C.J. van Houten. "Photometrics Studies of Asteroids. VII." In: *Astrophysical Journal* 127 (Apr. 1958), p. 253. DOI: 10.1086/146459.

[6] M. Kaasalainen and J. Torppa. "Optimization Methods for Asteroid Lightcurve Inversion: I. Shape Determination". In: *Icarus* 153.1 (2001), pp. 24–36. ISSN: 0019-1035. DOI: https://doi.org/10.1006/icar.2001.6673. URL: https://www.sciencedirect.com/science/article/pii/S0019103501966734.

[7] Jiří Šilha et al. "Space debris observations with the Slovak AGO70 telescope: Astrometry and light curves". In: *Advances in Space Research* 65.8 (2020), pp. 2018–2035. ISSN: 0273-1177. DOI: https://doi.org/10.1016/j.asr.2020.01.038. URL: https://www.sciencedirect.com/science/article/pii/S0273117720300727.

[8] D. Hall et al. "Separating Attitude and Shape Effects for Non-Resolved Objects". In: *The 2007 AMOS Technical Conference Proceedings*. Conference paper. Kihei, Maui, HI: Maui Economic Development Board, Inc., 2007, pp. 464–475. URL: https://amostech.com/TechnicalPapers/2007/NROC/Hall.pdf.

[9] Alexander Burton and Carolin Frueh. "Multi-Hypothesis Light Curve Inversion for Space Object Attitude Determination". In: *Proceedings of the 8th European Conference on Space Debris*. Vol. 8. Conference paper. Purdue University. ESA Space Debris Office, 2021. URL: https://conference.sdo.esoc.esa.int/proceedings/sdc8/paper/18/SDC8-paper18.pdf.

[10] Joanna C. Hinks, Richard Linares, and John L. Crassidis. "Attitude observability from light curve measurements". In: *AIAA Guidance, Navigation, and Control (GNC) Conference*. American Institute of Aeronautics and Astronautics Inc., 2013. ISBN: 9781624102240. DOI: 10.2514/6.2013-5005.

[11] Andrew Dianetti and John Crassidis. "Resident Space Object Characterization Using Polarized Light Curves". In: *Journal of Guidance, Control, and Dynamics* 46 (Oct. 2022), pp. 1–18. DOI: 10.2514/1.G006847.

[12] Richard Linares et al. "Astrometric and Photometric Data Fusion for Resident Space Object Orbit, Attitude, and Shape Determination Via Multiple-Model Adaptive Estimation". In: *AIAA Guidance, Navigation, and Control Conference*. Aug. 2010. ISBN: 978-1-60086-962-4. DOI: 10.2514/6.2010-8341.

[13] Xiao-Rong Li, Yaakov Bar-Shalom, and William Dale Blair. "Engineer's guide to variable-structure multiple-model estimation for tracking". In: *Multitarget-multisensor tracking: Applications and advances.* 3 (2000), pp. 499–567. URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ee1365d32d153bf1f60d99d27859180c4f30066e.

[14] Richard Linares, Roberto Furfaro, and Vishnu Reddy. "Space Objects Classification via Light-Curve Measurements Using Deep Convolutional Neural Networks". In: *Journal of the Astronautical Sciences* 67 (3 Sept. 2020), pp. 1063–1091. ISSN: 21950571. DOI: 10.1007/s40295-019-00208-w.

[15] Liam Robinson and Carolin Frueh. "(AAS 22-574) Light Curve Inversion for Reliable Shape Reconstruction of Human-Made Space Objects". In: Conference paper. Aug. 2022. URL: https://www.researchgate.net/publication/366781300.

[16] Alexander Burton. "Attitude Estimation Using Light Curves". In: *Purdue University* (July 2024). DOI: 10.25394/PGS.26383684.v1. URL: https://hammer.purdue.edu/articles/thesis/ATTITUDE_ESTIMATION_USING_LIGHT_CURVES/26383684.

[17] Sangeetha Nandakumar et al. “The high optical brightness of the BlueWalker 3 satellite”. In: *Nature* 623.7989 (2023), pp. 938–941. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06672-7. URL: https://doi.org/10.1038/s41586-023-06672-7.

[18] Andreja Gomboc, Martin Horvat, and Uroš Kostić. *ESA PECS Study Arrangement Report: Relativistic GNSS*. 2014. URL: https://www.esa.int/gsp/ACT/doc/STUDIES/ACT-RPT-PHY-PEC-400010374111NLKML-Relativistic_GNSS_Final_Report.pdf.

[19] Sangho Jo et al. “S-LIGHT: Synthetic Dataset for the Separation of Diffuse and Specular Reflection Images”. In: *Sensors* 24.7 (2024). ISSN: 1424-8220. DOI: 10.3390/s24072286. URL: https://www.mdpi.com/1424-8220/24/7/2286.

[20] Charles J. Wetterer et al. “Refining space object radiation pressure modeling with bidirectional reflectance distribution functions”. In: *Journal of Guidance, Control, and Dynamics* 37 (1 2014), pp. 185–196. ISSN: 15333884. DOI: 10.2514/1.60577.

[21] Michael Ashikhmin and Peter Shirley. “An Anisotropic Phong Light Reflection Model”. In: *Journal of Graphics Tools* 5 (Jan. 2001). URL: https://www.researchgate.net/publication/2523821_An_Anisotropic_Phong_Light_Reflection_Model.

[22] John L. Crassidis and F. Landis Markley. “Unscented filtering for spacecraft attitude estimation”. In: *Journal of Guidance, Control, and Dynamics* 26 (4 2003), pp. 536–542. ISSN: 15333884. DOI: 10.2514/2.5102.

[23] Richard Linares et al. “Space object shape characterization and tracking using light curve and angles data”. In: *Journal of Guidance, Control, and Dynamics* 37 (1 2014), pp. 13–25. ISSN: 15333884. DOI: 10.2514/1.62986.

[24] Byron D. Tapley, Bob E. Schutz, and George H. Born. “Chapter 4 - Fundamentals of Orbit Determination”. In: *Statistical Orbit Determination*. Ed. by Byron D. Tapley, Bob E. Schutz, and George H. Born. Burlington: Academic Press, 2004, pp. 85–86. ISBN: 978-0-12-683630-1. DOI: https://doi.org/10.1016/B978-012683630-1/50023-0.

[25] Eric Bellm and Shrinivas Kulkarni. “The unblinking eye on the sky”. In: *Nature Astronomy* 1.3 (Mar. 2017). ISSN: 2397-3366. DOI: 10.1038/s41550-017-0071. URL: http://dx.doi.org/10.1038/s41550-017-0071.

[26] Keats Wilkie and Johnny Fernandez. *Advanced Composite Solar Sail System (ACS3) Mission Update*. Presentation slides. Presented at The 6th International Symposium on Space Sailing, New York, NY, USA, June 5-9, 2023. Hampton, Virginia, United States: NASA Langley Research Center, June 2023. URL: https://ntrs.nasa.gov/api/citations/20230008378/downloads/Wilkie_ACS3_mission_update_ISSS_2023_20230530_rev_a.pdf.

[27] Larsson, S. “Parameter estimation in epoch folding analysis”. In: *Astron. Astrophys. Suppl. Ser.* 117.1 (1996), pp. 197–201. DOI: 10.1051/aas:1996150. URL: https://doi.org/10.1051/aas:1996150.

[28] Jean-Noël Pittet et al. “Space debris attitude determination of faint LEO objects using photometry: SwissCube CubeSat study case”. In: May 2017.

[29] Y. Matsushita et al. “Light Curve Analysis and Attitude Estimation of Space Objects Focusing on Glint”. In: *First International Orbital Debris Conference*. Vol. 2109. LPI Contributions. Dec. 2019, 6091, p. 6091. URL: https://ui.adsabs.harvard.edu/link_gateway/2019LPICo2109.6091M/PUB_PDF.

[30] Yuri Matsushita et al. “Conceptual Study of Improved Photometric Attitude Estimation Using Glint”. In: *Transactions of the Japan Society for Aeronaturical and Space Sciences, Aerospace Technology Japan* 22 (2024), pp. 59–65. DOI: 10.2322/tastj.22.59.

A Test object TLE data

The TLE data for all satellites were retrieved from the *space-track.org* website for the date 14-11-2024.

Table A.1: CryoSat 2

	CryoSat 2
Line 1	1 36508U 10013A 24318.77506330 .00000661 00000-0 16335-3 0 9990
Line 2	2 36508 92.0154 137.2273 0003801 110.1730 249.9886 14.51895234773864

Table A.2: Swarm A

	Swarm A
Line 1	1 39452U 13067B 24319.15485474 .00014358 00000-0 40044-3 0 9993
Line 2	2 39452 87.3377 268.9331 0003430 92.0643 268.1003 15.36630797616804

Table A.3: Swarm B

	Swarm B
Line 1	1 39451U 13067A 24318.81141188 .00005961 00000-0 25517-3 0 9993
Line 2	2 39451 87.7501 157.8044 0002801 87.9629 272.1938 15.22616637594743

Table A.4: Swarm C

	Swarm C
Line 1	1 39453U 13067C 24319.08965172 .00014308 00000-0 39903-3 0 9991
Line 2	2 39453 87.3378 267.5457 0003440 92.2500 267.9147 15.36632294616759

Table A.5: BlueWalker 3

	BlueWalker 3
Line 1	1 53807U 22111AL 24319.06106888 .00011899 00000-0 41321-3 0 9998
Line 2	2 53807 53.2357 221.1658 0008002 173.3430 186.7679 15.30184266121145

Table A.6: ACS3

	ACS3
Line 1	1 59588U 24077B 24319.07684548 .00001775 00000-0 19527-2 0 9992
Line 2	2 59588 97.5074 353.0061 0034766 189.6307 170.4184 13.72806455 27912

B Baseline estimation results

B.1. Original cases

B.1.1. MMAE

Table B.1.1.1: Performance metrics for MMAE - 7-9-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	51.2	11.7	58.8	Correct shape type
CryoSat 2	Single axis spin	136.2	63.6	113.4	No
CryoSat 2	Tumbling	15.9	8.9	17.2	Correct shape type
Swarm A	Nadir pointing	87.1	102.3	118.8	No
Swarm A	Single axis spin	120.9	65.9	147.9	No
Swarm A	Tumbling	19.8	5.5	17.9	Yes
Swarm B	Nadir pointing	136.9	46.6	126.8	No
Swarm B	Single axis spin	91.1	12.6	64.5	No
Swarm B	Tumbling	132.2	63.8	77.	No
Swarm C	Nadir pointing	121.7	89.1	99.	No
Swarm C	Single axis spin	136.	61.	98.3	No
Swarm C	Tumbling	8.	4.9	7.	Yes
ACS3	Nadir pointing	73.8	89.5	43.	No
ACS3	Single axis spin	2.3	0.7	2.3	Yes
ACS3	Tumbling	5.9	2.8	4.3	Yes
BlueWalker 3	Nadir pointing	101.3	103.8	36.6	No
BlueWalker 3	Single axis spin	120.2	63.1	109.6	No
BlueWalker 3	Tumbling	72.7	48.2	112.3	Yes

Table B.1.1.2: Performance metrics for MMAE - 12-10-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	73.4	89.4	82.5	No
CryoSat 2	Single axis spin	110.1	66.	118.9	No
CryoSat 2	Tumbling	108.4	53.2	154.6	No
Swarm A	Nadir pointing	58.3	27.8	57.3	Yes
Swarm A	Single axis spin	6.2	1.3	6.5	Yes
Swarm A	Tumbling	3.1	1.4	4.2	Yes
Swarm B	Nadir pointing	52.6	45.	97.	No
Swarm B	Single axis spin	151.9	67.	120.3	No
Swarm B	Tumbling	79.9	42.8	84.4	No
Swarm C	Nadir pointing	46.7	22.	48.9	Correct shape type
Swarm C	Single axis spin	8.6	2.	9.1	Yes
Swarm C	Tumbling	129.6	55.1	136.3	No
ACS3	Nadir pointing	86.1	79.2	88.6	No
ACS3	Single axis spin	9.4	1.1	6.3	Yes
ACS3	Tumbling	8.3	2.	8.4	Yes
BlueWalker 3	Nadir pointing	27.4	3.9	18.9	Yes
BlueWalker 3	Single axis spin	26.4	5.7	24.3	Yes
BlueWalker 3	Tumbling	2.8	1.3	2.9	Yes

Table B.1.1.3: Performance metrics for MMAE - 14-11-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	130.4	58.3	123.	No
CryoSat 2	Single axis spin	5.	1.2	5.3	Yes
CryoSat 2	Tumbling	72.6	46.2	172.6	Correct shape type
Swarm A	Nadir pointing	12.2	67.4	50.	No
Swarm A	Single axis spin	35.3	3.7	35.6	Yes
Swarm A	Tumbling	132.7	64.1	179.2	No
Swarm B	Nadir pointing	26.1	17.1	27.	Yes
Swarm B	Single axis spin	132.6	50.5	120.4	No
Swarm B	Tumbling	81.2	41.8	77.7	No
Swarm C	Nadir pointing	58.	30.4	34.5	Correct shape type
Swarm C	Single axis spin	120.4	52.3	155.1	Yes
Swarm C	Tumbling	135.4	66.7	175.1	No
ACS3	Nadir pointing	134.4	85.	116.7	No
ACS3	Single axis spin	6.1	1.	5.8	Yes
ACS3	Tumbling	106.9	51.	165.	Yes
BlueWalker 3	Nadir pointing	14.2	31.1	43.2	Correct shape type
BlueWalker 3	Single axis spin	53.2	26.5	78.	Yes
BlueWalker 3	Tumbling	57.3	18.7	53.8	No

B.1.2. MGS

Table B.1.2.1: Performance metrics for MGS - 7-9-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	125.4	70.9	74.2	No
CryoSat 2	Single axis spin	124.	64.1	114.5	No
CryoSat 2	Tumbling	117.7	49.	136.8	No
Swarm A	Nadir pointing	56.5	96.1	119.3	No
Swarm A	Single axis spin	118.3	60.7	131.1	No
Swarm A	Tumbling	13.9	1.9	13.4	Yes
Swarm B	Nadir pointing	76.1	55.9	159.1	No
Swarm B	Single axis spin	140.8	70.1	135.2	No
Swarm B	Tumbling	111.4	51.5	139.9	No
Swarm C	Nadir pointing	121.2	87.2	128.1	Correct shape type
Swarm C	Single axis spin	128.9	70.8	120.3	Correct shape type
Swarm C	Tumbling	62.5	33.4	162.1	Correct shape type
ACS3	Nadir pointing	90.1	74.	56.4	No
ACS3	Single axis spin	142.7	83.3	158.1	No
ACS3	Tumbling	86.3	57.7	68.5	Correct shape type
BlueWalker 3	Nadir pointing	105.	76.9	19.1	No
BlueWalker 3	Single axis spin	84.3	42.9	99.5	Yes
BlueWalker 3	Tumbling	54.	36.3	115.7	No

Table B.1.2.2: Performance metrics for MGS - 12-10-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	13.3	5.6	21.7	Yes
CryoSat 2	Single axis spin	88.1	57.3	154.5	No
CryoSat 2	Tumbling	5.5	2.8	6.1	Yes
Swarm A	Nadir pointing	142.9	98.4	157.1	No
Swarm A	Single axis spin	136.7	70.6	107.3	Correct shape type
Swarm A	Tumbling	99.6	53.3	108.9	Correct shape type
Swarm B	Nadir pointing	136.2	53.	137.	No
Swarm B	Single axis spin	103.3	59.5	150.3	Yes
Swarm B	Tumbling	31.6	12.8	34.3	Yes
Swarm C	Nadir pointing	19.3	11.	47.1	No
Swarm C	Single axis spin	75.6	39.7	78.7	Yes
Swarm C	Tumbling	139.9	60.8	121.3	No
ACS3	Nadir pointing	77.9	67.3	62.	No
ACS3	Single axis spin	85.2	52.8	130.	No
ACS3	Tumbling	113.3	44.6	147.	No
BlueWalker 3	Nadir pointing	82.6	47.8	86.2	Yes
BlueWalker 3	Single axis spin	107.4	69.	130.4	No
BlueWalker 3	Tumbling	98.2	44.9	107.8	Correct shape type

Table B.1.2.3: Performance metrics for MGS - 14-11-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	128.7	83.1	40.6	No
CryoSat 2	Single axis spin	6.6	1.6	11.5	Yes
CryoSat 2	Tumbling	19.3	8.1	22.5	Yes
Swarm A	Nadir pointing	63.8	86.6	114.5	Yes
Swarm A	Single axis spin	121.6	63.8	143.7	Correct shape type
Swarm A	Tumbling	56.3	23.9	170.6	Yes
Swarm B	Nadir pointing	129.9	69.7	102.	No
Swarm B	Single axis spin	150.4	67.6	129.6	No
Swarm B	Tumbling	11.1	2.9	11.3	Yes
Swarm C	Nadir pointing	24.8	75.6	5.3	Correct shape type
Swarm C	Single axis spin	60.6	13.2	51.9	Yes
Swarm C	Tumbling	98.7	45.8	117.7	Yes
ACS3	Nadir pointing	91.3	86.8	110.6	No
ACS3	Single axis spin	99.7	55.7	153.9	No
ACS3	Tumbling	55.4	36.8	50.8	No
BlueWalker 3	Nadir pointing	47.9	21.8	16.9	No
BlueWalker 3	Single axis spin	103.1	42.1	83.3	No
BlueWalker 3	Tumbling	79.3	40.7	79.2	No

B.1.3. MGen

Table B.1.3.1: Performance metrics for MGen - 7-9-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	80.3	86.7	88.9	No
CryoSat 2	Single axis spin	143.3	63.1	121.7	Correct shape type
CryoSat 2	Tumbling	106.8	52.8	133.9	No
Swarm A	Nadir pointing	52.8	99.2	83.8	Correct shape type
Swarm A	Single axis spin	132.7	67.6	118.4	No
Swarm A	Tumbling	132.9	61.3	114.8	Correct shape type
Swarm B	Nadir pointing	86.5	74.9	50.	No
Swarm B	Single axis spin	116.	62.8	128.3	Correct shape type
Swarm B	Tumbling	106.9	55.7	148.8	No
Swarm C	Nadir pointing	140.	91.3	114.5	No
Swarm C	Single axis spin	95.4	62.9	132.8	No
Swarm C	Tumbling	147.2	58.3	129.9	No
ACS3	Nadir pointing	84.4	71.3	125.1	No
ACS3	Single axis spin	111.7	59.2	163.1	No
ACS3	Tumbling	119.1	57.4	143.9	No
BlueWalker 3	Nadir pointing	61.4	71.1	114.5	No
BlueWalker 3	Single axis spin	118.4	53.3	139.9	No
BlueWalker 3	Tumbling	88.8	43.1	150.9	No

Table B.1.3.2: Performance metrics for MGen - 12-10-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	113.2	105.4	65.1	No
CryoSat 2	Single axis spin	115.2	59.1	126.6	No
CryoSat 2	Tumbling	108.6	55.5	123.4	No
Swarm A	Nadir pointing	94.3	83.9	104.3	Correct shape type
Swarm A	Single axis spin	118.2	50.5	145.4	Correct shape type
Swarm A	Tumbling	95.	46.8	103.2	Correct shape type
Swarm B	Nadir pointing	86.1	91.6	93.4	No
Swarm B	Single axis spin	129.9	57.1	116.7	Correct shape type
Swarm B	Tumbling	126.3	59.1	124.	No
Swarm C	Nadir pointing	87.4	62.9	110.8	No
Swarm C	Single axis spin	141.8	59.	134.1	No
Swarm C	Tumbling	79.1	39.7	114.3	No
ACS3	Nadir pointing	117.7	76.3	113.7	No
ACS3	Single axis spin	147.2	49.8	117.3	Correct shape type
ACS3	Tumbling	114.5	54.4	135.4	No
BlueWalker 3	Nadir pointing	32.3	69.4	90.8	No
BlueWalker 3	Single axis spin	134.6	74.2	115.9	No
BlueWalker 3	Tumbling	85.5	47.9	159.4	No

Table B.1.3.3: Performance metrics for MGen - 14-11-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	100.1	86.5	103.	No
CryoSat 2	Single axis spin	94.9	55.4	139.1	No
CryoSat 2	Tumbling	105.2	49.6	137.6	No
Swarm A	Nadir pointing	128.4	87.6	130.5	No
Swarm A	Single axis spin	122.7	67.8	126.3	No
Swarm A	Tumbling	112.	59.7	136.2	Correct shape type
Swarm B	Nadir pointing	71.4	85.	138.8	No
Swarm B	Single axis spin	126.3	70.3	119.2	Correct shape type
Swarm B	Tumbling	125.8	62.5	125.9	No
Swarm C	Nadir pointing	82.8	98.2	121.7	No
Swarm C	Single axis spin	123.1	69.9	138.	No
Swarm C	Tumbling	102.3	55.4	133.4	No
ACS3	Nadir pointing	61.4	44.7	104.9	No
ACS3	Single axis spin	133.5	61.7	128.4	No
ACS3	Tumbling	101.8	47.6	119.7	No
BlueWalker 3	Nadir pointing	120.2	91.2	91.1	No
BlueWalker 3	Single axis spin	104.8	54.7	150.8	No
BlueWalker 3	Tumbling	94.7	34.2	103.4	No

B.2. No sphere cases

Table B.2.0.1: Performance metrics for MMAE - No spheres - 7-9-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	51.2	11.7	58.8	Correct shape type
CryoSat 2	Single axis spin	8.	1.8	8.7	Yes
CryoSat 2	Tumbling	28.6	17.8	26.4	No
Swarm A	Nadir pointing	55.4	83.5	130.6	No
Swarm A	Single axis spin	166.1	54.9	119.2	Yes
Swarm A	Tumbling	3.2	1.2	3.5	Yes
Swarm B	Nadir pointing	95.8	13.	105.5	No
Swarm B	Single axis spin	15.3	1.9	12.6	Yes
Swarm B	Tumbling	9.3	5.4	9.1	Yes
Swarm C	Nadir pointing	107.6	103.1	107.5	No
Swarm C	Single axis spin	125.5	64.3	117.7	Correct shape type
Swarm C	Tumbling	2.3	3.	2.5	Yes
ACS3	Nadir pointing	69.7	56.	75.6	No
ACS3	Single axis spin	1.9	0.6	1.9	Yes
ACS3	Tumbling	6.	2.9	4.5	Yes
BlueWalker 3	Nadir pointing	106.9	78.2	21.5	No
BlueWalker 3	Single axis spin	22.6	5.1	27.9	Yes
BlueWalker 3	Tumbling	20.2	11.2	29.3	Yes

Table B.2.0.2: Performance metrics for MMAE - No spheres - 12-10-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	102.7	53.1	90.7	No
CryoSat 2	Single axis spin	123.4	73.8	131.2	No
CryoSat 2	Tumbling	100.7	54.5	178.4	No
Swarm A	Nadir pointing	60.4	30.1	59.3	Yes
Swarm A	Single axis spin	10.2	1.3	6.7	Yes
Swarm A	Tumbling	2.9	1.3	4.	Yes
Swarm B	Nadir pointing	60.	60.7	78.5	No
Swarm B	Single axis spin	99.	58.2	124.2	Correct shape type
Swarm B	Tumbling	143.	66.2	153.8	No
Swarm C	Nadir pointing	44.6	28.	49.3	Correct shape type
Swarm C	Single axis spin	8.	1.8	8.3	Yes
Swarm C	Tumbling	137.1	57.7	134.3	No
ACS3	Nadir pointing	117.1	92.4	105.5	No
ACS3	Single axis spin	9.4	1.1	6.3	Yes
ACS3	Tumbling	80.6	39.	85.4	No
BlueWalker 3	Nadir pointing	26.1	3.6	17.2	Yes
BlueWalker 3	Single axis spin	22.1	6.	22.9	Yes
BlueWalker 3	Tumbling	2.9	1.3	3.	Yes

Table B.2.0.3: Performance metrics for MMAE - No spheres - 14-11-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	130.3	58.5	126.8	No
CryoSat 2	Single axis spin	5.4	1.3	5.7	Yes
CryoSat 2	Tumbling	92.7	48.7	139.7	Correct shape type
Swarm A	Nadir pointing	12.1	67.3	50.1	No
Swarm A	Single axis spin	157.3	59.4	150.1	No
Swarm A	Tumbling	155.1	59.8	143.8	Yes
Swarm B	Nadir pointing	26.2	13.6	27.2	Yes
Swarm B	Single axis spin	7.4	0.8	7.6	Yes
Swarm B	Tumbling	133.4	59.7	152.4	Correct shape type
Swarm C	Nadir pointing	58.	30.4	34.5	Correct shape type
Swarm C	Single axis spin	158.6	79.	130.8	No
Swarm C	Tumbling	164.8	75.5	174.8	No
ACS3	Nadir pointing	98.4	81.7	126.5	No
ACS3	Single axis spin	6.1	1.	5.8	Yes
ACS3	Tumbling	108.2	50.7	121.7	Yes
BlueWalker 3	Nadir pointing	14.2	31.1	43.2	Correct shape type
BlueWalker 3	Single axis spin	104.9	69.8	134.9	No
BlueWalker 3	Tumbling	56.	17.9	52.4	No

B.2.1. MGS

Table B.2.1.1: Performance metrics for MGS - No spheres - 7-9-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	32.5	13.3	45.5	Correct shape type
CryoSat 2	Single axis spin	100.6	53.9	141.7	Correct shape type
CryoSat 2	Tumbling	2.9	1.3	4.1	Yes
Swarm A	Nadir pointing	57.8	73.8	10.2	No
Swarm A	Single axis spin	121.7	61.6	151.7	No
Swarm A	Tumbling	152.3	60.1	135.9	Yes
Swarm B	Nadir pointing	41.5	7.	47.3	Yes
Swarm B	Single axis spin	7.3	1.4	8.	Yes
Swarm B	Tumbling	109.7	48.7	98.5	No
Swarm C	Nadir pointing	121.2	87.2	128.	Correct shape type
Swarm C	Single axis spin	117.7	63.7	113.2	No
Swarm C	Tumbling	91.5	53.2	141.5	No
ACS3	Nadir pointing	57.	122.	59.3	Correct shape type
ACS3	Single axis spin	124.5	68.5	148.6	No
ACS3	Tumbling	139.6	59.4	158.6	Correct shape type
BlueWalker 3	Nadir pointing	26.8	59.1	34.7	No
BlueWalker 3	Single axis spin	87.1	57.5	127.4	Yes
BlueWalker 3	Tumbling	54.	36.3	115.7	No

Table B.2.1.2: Performance metrics for MGS - No spheres - 12-10-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	13.3	5.6	21.7	Yes
CryoSat 2	Single axis spin	88.4	57.4	157.5	No
CryoSat 2	Tumbling	5.5	2.8	6.1	Yes
Swarm A	Nadir pointing	142.9	98.4	157.1	No
Swarm A	Single axis spin	129.	38.7	131.8	No
Swarm A	Tumbling	59.	25.	90.	Yes
Swarm B	Nadir pointing	100.	62.5	127.7	No
Swarm B	Single axis spin	119.3	70.5	96.	No
Swarm B	Tumbling	92.3	51.3	144.1	No
Swarm C	Nadir pointing	86.8	75.9	134.2	No
Swarm C	Single axis spin	80.8	43.9	133.4	No
Swarm C	Tumbling	82.3	48.	134.	No
ACS3	Nadir pointing	104.1	98.2	44.	Yes
ACS3	Single axis spin	100.4	63.8	123.3	No
ACS3	Tumbling	82.4	31.9	98.2	No
BlueWalker 3	Nadir pointing	8.2	2.	7.6	Yes
BlueWalker 3	Single axis spin	39.4	8.7	35.1	Yes
BlueWalker 3	Tumbling	113.3	54.6	181.1	Correct shape type

Table B.2.1.3: Performance metrics for MGS - No spheres - 14-11-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	143.9	23.8	138.7	No
CryoSat 2	Single axis spin	6.6	1.6	11.5	Yes
CryoSat 2	Tumbling	91.8	33.7	166.	No
Swarm A	Nadir pointing	94.8	97.	95.	No
Swarm A	Single axis spin	157.	85.6	86.8	No
Swarm A	Tumbling	109.5	54.5	146.9	Yes
Swarm B	Nadir pointing	26.5	13.9	27.4	Yes
Swarm B	Single axis spin	98.6	44.5	134.9	No
Swarm B	Tumbling	11.2	3.	11.4	Yes
Swarm C	Nadir pointing	24.8	75.6	5.3	Correct shape type
Swarm C	Single axis spin	158.1	78.8	130.6	No
Swarm C	Tumbling	3.8	1.8	5.	Yes
ACS3	Nadir pointing	99.	92.5	147.5	No
ACS3	Single axis spin	152.4	78.5	151.8	No
ACS3	Tumbling	33.3	15.9	35.	No
BlueWalker 3	Nadir pointing	47.9	21.8	16.9	No
BlueWalker 3	Single axis spin	93.2	46.6	116.8	No
BlueWalker 3	Tumbling	53.6	25.1	58.8	No

B.3. Updated cases

B.3.1. MMAE

Table B.3.1.1: Performance metrics for MMAE - Updated - 7-9-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	46.2	14.1	53.	Correct shape type
CryoSat 2	Single axis spin	78.7	51.1	106.9	No
CryoSat 2	Tumbling	3.7	1.5	4.7	Yes
Swarm A	Nadir pointing	37.8	3.4	44.1	Yes
Swarm A	Single axis spin	5.8	1.4	6.3	Yes
Swarm A	Tumbling	2.2	1.	3.	Yes
Swarm B	Nadir pointing	125.3	35.4	135.3	No
Swarm B	Single axis spin	7.4	1.4	8.	Yes
Swarm B	Tumbling	7.6	2.5	8.2	Yes
Swarm C	Nadir pointing	20.7	6.6	29.	Yes
Swarm C	Single axis spin	7.3	1.3	7.7	Yes
Swarm C	Tumbling	5.9	2.5	6.4	Yes
ACS3	Nadir pointing	33.5	8.	32.1	Yes
ACS3	Single axis spin	16.3	4.4	14.3	Yes
ACS3	Tumbling	9.4	4.8	12.7	Yes
BlueWalker 3	Nadir pointing	20.1	4.6	28.6	Yes
BlueWalker 3	Single axis spin	15.4	5.	12.3	Yes
BlueWalker 3	Tumbling	5.7	2.8	6.7	Yes

Table B.3.1.2: Performance metrics for MMAE - Updated - 12-10-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	13.6	5.4	22.5	Yes
CryoSat 2	Single axis spin	3.6	1.4	3.7	Yes
CryoSat 2	Tumbling	3.5	2.	3.7	Yes
Swarm A	Nadir pointing	92.2	16.1	91.6	Yes
Swarm A	Single axis spin	32.2	3.8	25.7	Yes
Swarm A	Tumbling	25.3	9.1	30.7	Yes
Swarm B	Nadir pointing	7.2	5.7	8.1	Yes
Swarm B	Single axis spin	18.7	2.3	19.6	Yes
Swarm B	Tumbling	8.1	4.2	7.8	Yes
Swarm C	Nadir pointing	12.5	6.1	11.7	Yes
Swarm C	Single axis spin	8.2	1.7	8.1	Yes
Swarm C	Tumbling	2.9	1.5	3.7	Yes
ACS3	Nadir pointing	27.	2.2	14.9	Yes
ACS3	Single axis spin	9.2	1.1	9.5	Yes
ACS3	Tumbling	2.3	1.1	3.2	Yes
BlueWalker 3	Nadir pointing	29.6	5.	21.9	Yes
BlueWalker 3	Single axis spin	2.1	0.8	2.6	Yes
BlueWalker 3	Tumbling	9.4	4.1	11.5	Yes

Table B.3.1.3: Performance metrics for MMAE - Updated - 14-11-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	81.2	94.9	63.9	No
CryoSat 2	Single axis spin	4.3	0.9	4.7	Yes
CryoSat 2	Tumbling	10.2	2.3	10.2	Yes
Swarm A	Nadir pointing	54.3	39.8	65.7	Yes
Swarm A	Single axis spin	5.6	1.5	5.7	Yes
Swarm A	Tumbling	1.4	0.6	1.8	Yes
Swarm B	Nadir pointing	26.	13.6	26.9	Yes
Swarm B	Single axis spin	19.7	1.5	15.5	Yes
Swarm B	Tumbling	4.9	2.1	4.9	Yes
Swarm C	Nadir pointing	146.9	78.2	49.4	No
Swarm C	Single axis spin	20.1	3.7	20.3	Yes
Swarm C	Tumbling	4.4	1.9	5.2	Yes
ACS3	Nadir pointing	12.2	12.7	17.	Yes
ACS3	Single axis spin	4.5	0.9	4.6	Yes
ACS3	Tumbling	5.	2.3	5.5	Yes
BlueWalker 3	Nadir pointing	48.	8.5	55.3	Yes
BlueWalker 3	Single axis spin	4.	1.1	4.	Yes
BlueWalker 3	Tumbling	2.1	1.4	2.8	Yes

B.3.2. MGS

Table B.3.2.1: Performance metrics for MGS - Updated - 7-9-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	69.	12.9	74.2	Yes
CryoSat 2	Single axis spin	116.2	52.6	114.5	No
CryoSat 2	Tumbling	22.5	13.3	24.	No
Swarm A	Nadir pointing	62.8	102.3	102.4	No
Swarm A	Single axis spin	106.6	66.1	118.7	Correct shape type
Swarm A	Tumbling	122.9	57.4	181.6	No
Swarm B	Nadir pointing	103.	15.3	107.5	No
Swarm B	Single axis spin	7.5	1.6	8.2	Yes
Swarm B	Tumbling	121.8	52.6	120.9	No
Swarm C	Nadir pointing	121.2	87.2	128.	Correct shape type
Swarm C	Single axis spin	171.7	76.1	127.9	No
Swarm C	Tumbling	125.6	64.1	118.9	Correct shape type
ACS3	Nadir pointing	62.9	90.2	58.6	Yes
ACS3	Single axis spin	147.	61.9	132.2	No
ACS3	Tumbling	119.5	57.1	151.3	No
BlueWalker 3	Nadir pointing	87.1	96.	32.1	No
BlueWalker 3	Single axis spin	17.8	6.7	19.4	Yes
BlueWalker 3	Tumbling	23.	12.6	22.7	No

Table B.3.2.2: Performance metrics for MGS - Updated - 12-10-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	13.3	5.6	21.7	Yes
CryoSat 2	Single axis spin	88.4	57.4	157.5	No
CryoSat 2	Tumbling	5.5	2.8	6.1	Yes
Swarm A	Nadir pointing	93.7	83.3	87.6	No
Swarm A	Single axis spin	141.8	50.5	100.3	No
Swarm A	Tumbling	133.1	58.4	119.6	No
Swarm B	Nadir pointing	100.	62.5	127.7	No
Swarm B	Single axis spin	119.3	70.5	96.	No
Swarm B	Tumbling	92.3	51.3	144.1	No
Swarm C	Nadir pointing	93.6	83.5	34.	No
Swarm C	Single axis spin	126.6	80.3	145.2	No
Swarm C	Tumbling	129.2	57.	130.	No
ACS3	Nadir pointing	57.	71.	104.	No
ACS3	Single axis spin	132.3	64.3	139.	No
ACS3	Tumbling	120.4	62.2	120.3	No
BlueWalker 3	Nadir pointing	8.2	2.	7.6	Yes
BlueWalker 3	Single axis spin	39.4	8.7	35.1	Yes
BlueWalker 3	Tumbling	150.2	68.8	143.8	No

Table B.3.2.3: Performance metrics for MGS - Updated - 14-11-2024

Satellite	Spin Case	RMSE θ_1[°]	RMSE θ_2[°]	RMSE θ_3[°]	True shape identified
CryoSat 2	Nadir pointing	143.9	23.8	138.7	No
CryoSat 2	Single axis spin	6.6	1.6	11.5	Yes
CryoSat 2	Tumbling	168.3	58.7	127.6	No
Swarm A	Nadir pointing	66.8	91.9	104.7	No
Swarm A	Single axis spin	164.5	64.1	109.1	No
Swarm A	Tumbling	101.3	56.4	105.7	No
Swarm B	Nadir pointing	26.5	13.9	27.4	Yes
Swarm B	Single axis spin	148.2	58.4	144.1	No
Swarm B	Tumbling	11.2	3.	11.4	Yes
Swarm C	Nadir pointing	24.8	75.6	5.3	Correct shape type
Swarm C	Single axis spin	151.2	65.9	131.6	No
Swarm C	Tumbling	116.2	59.3	141.3	No
ACS3	Nadir pointing	99.	92.5	147.5	No
ACS3	Single axis spin	144.1	62.5	110.8	Yes
ACS3	Tumbling	33.3	15.9	35.	No
BlueWalker 3	Nadir pointing	47.4	21.6	15.8	No
BlueWalker 3	Single axis spin	115.4	71.8	156.7	Yes
BlueWalker 3	Tumbling	53.6	25.1	58.8	No

C Research proposal

Literature Review

Object Characterisation by Light Curve Inversion

Wieger Verbeek

Delft University of Technology

TU Delft

Literature Review

Object Characterisation by Light Curve Inversion

by

Wieger Verbeek

Student number: 4675150
Date: Thursday 10th October, 2024
Thesis supervisors: Dr. S. Gehly & Dr. M. Langbroek

TU Delft

Contents

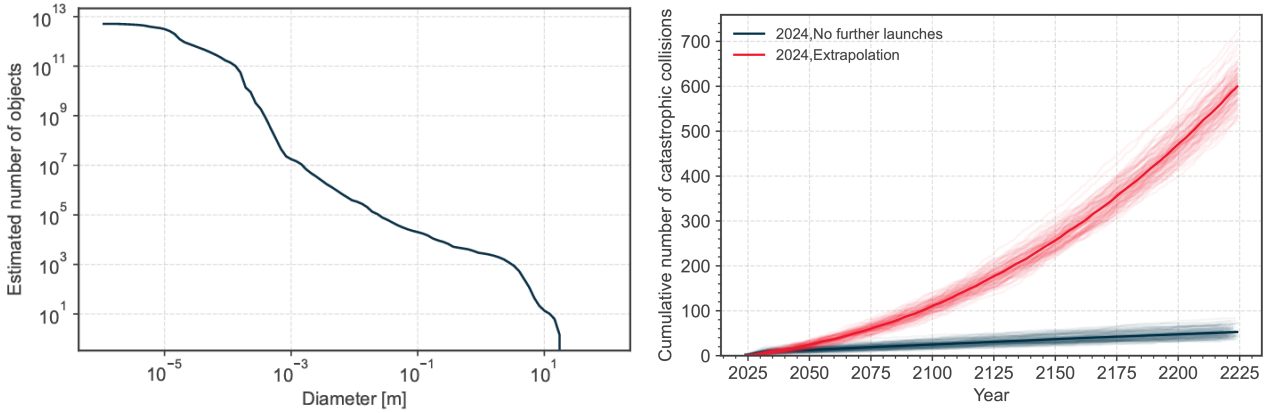
Nomenclature

Abbreviations

ADR	Active Debris Removal
AMR	Area-to-Mass Ratio
BRDF	Bidirectional Reflectance Distribution Function
DL	Deep Learning
EGI	Extended Gaussian Image
FSMM	Fixed-Structure Multiple-Model
GA	Genetic Algorithm
LEO	Low Earth Orbit
ML	Machine Learning
MLI	Multi-Layer Insulation
MMAE	Multiple-Model Adaptive Estimation
PSO	Particle Swarm Optimiser
SRP	Solar Radiation Pressure
UKF	Unscented Kalman Filter
VSMM	Variable-Structure Multiple-Model

1 Introduction

In recent decades the amount of space debris objects orbiting the Earth has drastically increased. It is estimated that there are now over 30.000 debris objects larger than 10 cm, 900.000 objects with size between 1 and 10 cm, and around 128 million objects smaller than 1 cm in geocentric orbit, as per the 2024 edition of the Annual Space Environment Report by ESA [1]. The current estimated distribution of the amount of objects per size is shown in Figure 1.1a.



(a) Distribution of amount of objects in Earth orbit per size.

(b) Expected number of catastrophic collisions with and without mitigation.

Figure 1.1: Figures highlighting the magnitude of the space debris problem [1].

All these objects pose a threat to existing and future space missions, as even the smaller objects could inflict catastrophic damage on a satellite, leading to the end of mission. Additionally, with the amount of objects ever increasing annually the risk of losing access to the space environment entirely becomes ever larger. This phenomenon is commonly known as the Kessler syndrome, and is a scenario where the number of objects especially in Low Earth Orbit (LEO) is so large, that even if no new objects are launched, the existing debris population is still dense enough to produce collisions, which would result in more debris and again more collisions [2]. This effect is shown in Figure 1.1b, which is the expected number of catastrophic collisions in orbit for the scenario where no satellites are launched after 2023, and for the case where the current rate of launching satellites is extrapolated. It is clear that in both cases mitigation is required to ensure a viable space environment and continuous access to space.

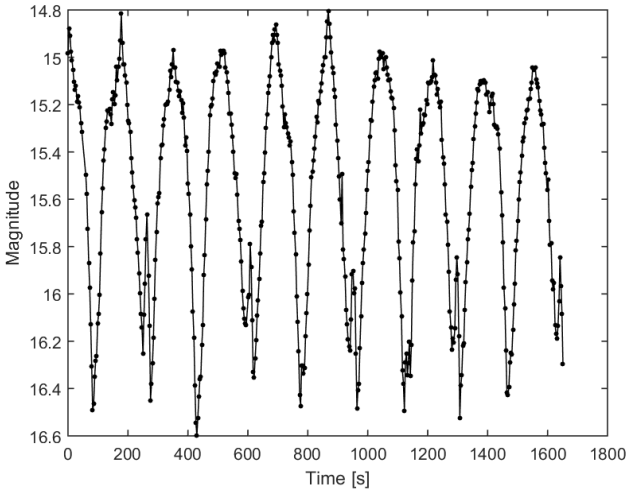
There are two ways that space debris can be dealt with to reduce the risk of collision with an active satellite: collision avoidance and active debris removal (ADR). In both cases it is necessary to know the state of a debris object. Estimating these states is a difficult task because of a few different reasons. The first is that there are so many objects that not all can be continuously tracked. Secondly, the majority of objects are too small (< 10 cm) to be tracked from the ground [3]. The last reason is that debris objects are not active anymore, so they do not transmit any information about their states, so state estimation can solely be done externally.

This problem of estimating the state and other parameters of a space object is called object characterisation. There are many parameters that could be characterised, like the shape, mass, attitude, angular velocities, surface properties and the orbit of an object. This characterisation is important because the more is known about an object, the more accurate it can be modelled. Accurate modelling is necessary for ADR missions and collision avoidance, the first because it is very difficult to capture an object in-orbit, especially if it is rotating, and the latter because risk analyses depend on the accuracy of modelled orbits.

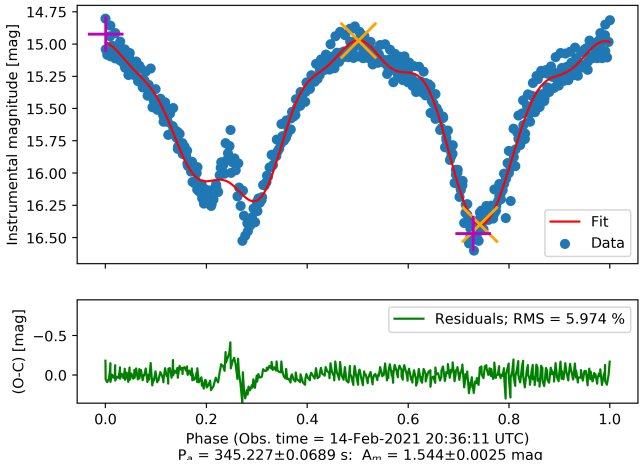
One method that has been the topic of many research papers the last few years is the use of light curves of space objects to estimate certain states and parameters, like in Linares et al. [4] and Šilha et al. [5]. This object characterisation with light curves will be investigated in this literature review. First, in Chapter 2 the concept of light curve inversion will be explained. Then in Chapter 3 the different parameters that can be estimated and characterised using light curve inversion will be discussed. In Chapter 4 different methods that have previously been researched and used in characterisation will be highlighted, as well as a novel method. Finally, in Chapter 5 the research questions and objectives that will form the basis of the MSc. thesis following the literature review are drafted, and the research planning is shown.

2 Light curve inversion

A light curve is the time history of the observed brightness of an object in orbit, measured from the ground [6]. Changes in this observed brightness could be caused by the rotation of the space object, with different sides of for example a satellite reflecting a different amount of light towards an observer, leading to a variation in measured brightness. An example of a light curve is given in Figure 2.1a.



(a) Example of the light curve of a rocket body. A periodic variation in observed brightness is clearly visible.



(b) Synodic rotation period extracted from light curve.

Figure 2.1: Example of a light curve and a parameter that can be extracted from it [7].

It is possible to use the light curve to extract information about the space object in question, like shape [4], attitude [8] and surface parameters [9]. This extraction of parameters is called light curve inversion, and there are different methods to characterise different aspects of a space object, which will be covered in the next chapters. One example is given in Figure 2.1b, where the apparent or synodic rotation period is extracted from the light curve shown in Figure 2.1a. As Šilha et al. [10] say: *"The apparent (synodic) rotation period [...] is a period of an object with respect to the observer. It can be directly extracted from the light curve and it is a direct function of the object's own rotation period in the inertial frame also referred to as the sidereal period."* This is an example of how relatively simple information can be extracted from a light curve. For other parameters, like shape and attitude, this is less straightforward.

The reason that properties like shape and attitude are important to estimate is that these are required to more accurately model the orbits of space objects, as non-conservative forces like aerodynamic drag (mainly at LEO) and solar radiation pressure (SRP) highly depend on these properties. This accurate modelling is important for both ADR missions and collision avoidance.

One main difficulty in the characterisation of space objects using light curve inversion is mentioned by Burton and Frueh [11]: *"There are many different orientations for any diffuse measurement that could all produce that exact measurement, and there are multiple attitude-time histories that could have produced any individual light curve."* It is therefore not trivial to obtain complex information like attitude and tumble rates from a light curve. Often additional *a-priori* information is needed to make detailed estimations as Hinks et al. mention [12].

The aforementioned problem has to do with a property called observability. In orbit estimation this property refers to the ability to obtain a unique estimate of the spacecraft state, as defined by Tapley, Schutz and Born [13]. A problem is not observable if multiple possible solutions exist for a given set of measurements. Friedman et al. [14] investigated observability for a few different cases. In their results it is clear that the more measurements there are over a longer time period, the better the observability and therefore the easier it becomes to estimate parameters. Unfortunately, in real life this is not always possible, as for example LEO satellites are only visible for short passes at a time, decreasing the observability of parameters and making the light curve inversion problem increasingly difficult. In the following chapters it will be highlighted how it is still possible to extract information and estimate parameters from light curves, and what the limitations are.

3 Characterisation parameters

In this chapter an overview is given of parameters that can be characterised using light curve inversion, along with one or multiple examples. The parameters that are discussed are class (Section 3.1), mass and area (Section 3.2), surface properties (Section 3.3), shape (Section 3.4) and attitude and angular velocities (Section 3.5).

3.1. Class

One aspect that might be important to characterise is to what class a space object belongs to, e.g. active satellite, defunct satellite, rocket body or debris fragment. This is precisely what Linares et al. [15] have tried. Using a method called Multiple-Model Adaptive Estimation (MMAE), which will be discussed in more detail in Chapter 4, they use simulated light curve data to give a classification to a space object. The different classes they use are shown in Figure 3.1.

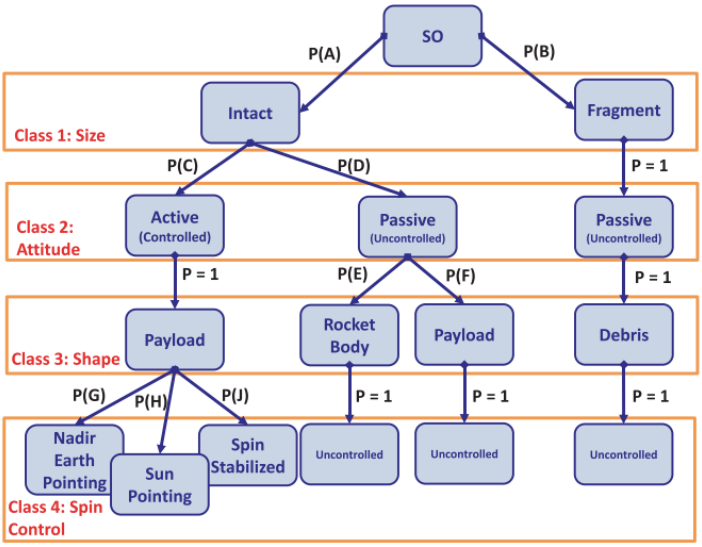


Figure 3.1: Overview of classes used by Linares et al. [15].

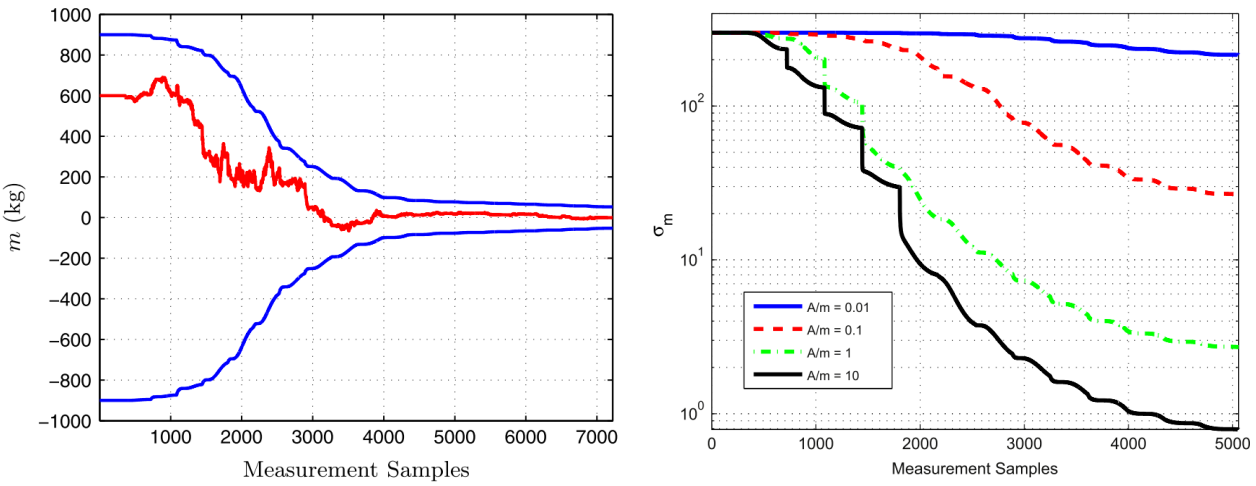
To test their method they simulated light curves for four different spacecraft: a spin stabilised bus, an uncontrolled bus, a nadir pointing bus and a rocket body. The results look promising, as the algorithm was able to correctly classify each of the 4 spacecraft. Their method was however not validated using real data, so it remains to be seen how well the method holds up in that case. Additionally, it was not tested whether the model could correctly identify a debris fragment, which would be important if this method were to be deployed for real object classification.

3.2. Mass and area

Other parameters that might be of interest to estimate are the mass and (albedo-)area of a space object. The reason for this is that the effects of non-conservative forces like aerodynamic drag and SRP are highly dependent on the mass that is accelerated and the area that light is reflected off. Linares et al. [16] estimate these parameters by combining photometric data, meaning light curves, and astrometric data, meaning observations of the orbital positions and velocities like range and/or angles measurements, in order to make the parameters observable.

In the paper they apply 3 different methods to estimate combinations of the different parameters. The first method only estimates mass using an assumed value for the albedo coefficients of the simulated object, a cuboid satellite with the same albedo for all 6 sides. The second method estimates mass, areas and albedo-areas, and the third method mass, albedo-areas and albedo coefficients.

Figure 3.2a shows the error plot of the mass estimate of the method with assumed albedo coefficient values. Clearly, the estimate converges to a low error for the mass estimate, with 3σ bounds of 55 kg. However, it can be seen that this required a very large amount of measurement samples, over 7000. They achieved this by placing the simulated space object in a near-geosynchronous orbit, so that each observation data arc was 3 hours, with 30 second measurement intervals for 20 orbits.



(a) Mass estimation of method using assumed albedo coefficient values.

(b) Mass covariances for different AMR objects against # of measurements.

Figure 3.2: Figures showing mass estimate and mass estimate covariances from Linares et al. [16].

In Figure 3.2b the covariance values for mass estimates for objects they simulated with a different area-to-mass ratio (AMR) are shown plotted against the number of measurement samples in the estimates. They found that the estimates for high AMR objects converge much faster, as the perturbing effect of SRP is higher for these objects, leading to the mass and areas becoming observable sooner. It can however be seen that over 2000 measurement samples are still required to get an estimate with a covariance below 10 kg. This method therefore does not have a broad use case, as most debris objects reside in LEO, where visible arcs only last a few minutes.

3.3. Surface properties

For some applications it could be of interest to determine what materials the surface of a space objects consists of. This is what Zigo et al. [17] have looked into. The underlying assumption they use is that space debris objects are inactive, and therefore do not emit any light, so the only measured light is reflected sunlight. As they put it: *"This assumption means that any measured changes in the spectrum of the visible light or in the colour indices will be caused by the surface properties and materials of the target body."* In their paper they attempt to do this with a combination of reflectance spectroscopy and BVRI photometery, in order to extract information about the surface materials from the colours. Their results show a limited ability to distinguish between copper-gold multi-layer insulation (MLI), aluminium and solar panels.

Other surface properties that can be estimated are reflectance coefficients. These coefficients are commonly used to simulate light curves using Bidirectional Reflectance Distribution Functions (BDRFs), which will be discussed in more detail in the next chapter. These functions work by assigning diffuse and specular reflectance coefficients to the surfaces that make up a space object. In Wetterer et al. [9] they instead add these coefficients, in a slightly modified way, to the state vector to be estimated using an Unscented Kalman Filter (UKF). To test their method they simulate the light curve of a cube shaped satellite, with the same surface reflectance parameters for all sides. Their results show reasonably good convergence for the reflectance parameters, but it is important to keep in mind that this was tested using a simple case. More complex satellites have different materials at different surfaces, with different reflectance coefficients, so this method might not produce as good results then.

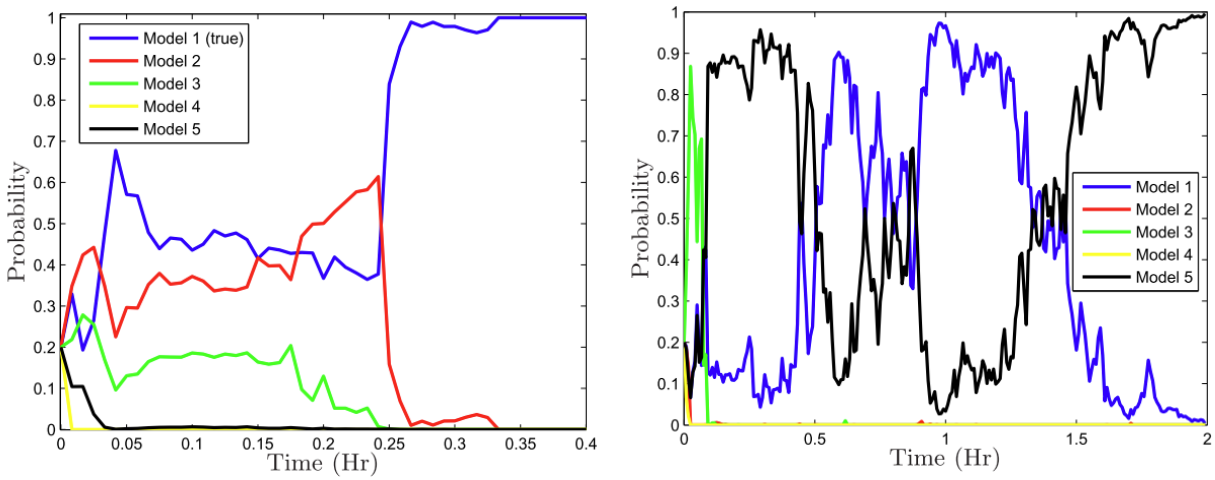
A different approach was taken by Linares et al. [16], as mentioned in the previous section. They use the albedo and albedo-area as parameters to be estimated in the UKF. Their results show good convergence on the true albedo values except for one side of the satellite, where the albedo value is not observable due to the specific observation geometries, and thus an estimate can not be made by the algorithm. These results were however also achieved by simulating a cuboid shaped satellite with the same reflectance values for all sides, which is an unlikely case for real space objects, where different sides could be covered with MLI or solar panels. The method used by Zigo et al. could perhaps be combined with this method, to identify the materials on each side based on the colour, and giving each side a better initial guess for surface parameters like diffuse and specular reflectance coefficients.

3.4. Shape

One parameter which is often important to be estimated is the shape of a space object. This is again related to accurately modelling aerodynamic drag and SRP, but it could also be important to know the shape of a satellite after a collision, which could be of importance to ADR missions.

Again, Linares et al. [4][18] have attempted to do this using MMAE. In both papers they randomly generate dimensions for a few different reference shapes to be used in the shape models of the UKF's in the MMAE algorithm. This algorithm then estimates both the orbit states, attitude, rotation rates and the shape of the object.

They ran the algorithm for two cases. In the first case the shape model selected as the truth was included in the model bank the MMAE algorithm uses to estimate the shape. In the second case the true model is not included, but a few models that are close to the truth are included. They found that in the first case the algorithm is able to quickly converge on the true shape, as shown in Figure 3.3a. In the second case the algorithm does not converge on a single model, but fluctuates between the two models that are closest to the truth, as can be seen in Figure 3.3b. This means that even if the true shape is not included, the algorithm is able to root out the models that are far from the truth, and still provide information about the true shape. This information could be used in further refinement of the estimate, like they mention: *"This information could be further used to redefine the model bank to find a better shape model estimate or for classifying a piece of debris (e.g., such as a sheared-off solar panel)"*.



(a) Shape estimate with true shape in model bank.

(b) Shape estimate without true shape in model bank.

Figure 3.3: Plots showing the estimates of the object shape for two cases [18].

While the results shown above look very good, it is important to interpret them in the right context. The shape models used in the model bank in both cases only included very similar shapes, only with different sizes. In [4] they used a blend of 25 regular polygon prisms and 25 cuboids with randomly generated dimensions, and the algorithm shows similar results as in Figure 3.3. It remains to be seen how well the algorithm performs when many different shape models are included.

A different approach of shape estimation is taken by Robinson and Frueh [19]. They use Extended Gaussian Image (EGI) for light curve inversion of shapes. This way they try to recover both convex and non-convex shapes from light curves. Non-convex shape recovery is important because most satellites are non-convex shaped, for example by having extended solar panels like in a box-wing design. Figure 3.4 shows their attempt to reconstruct such a shape from a simulated light curve. On the left is the true shape, and on the right are the convex and non-convex guesses of their algorithm, respectively. It can be seen that even the non-convex guess is not very accurate, highlighting the difficulty of accurately reconstructing these shapes from light curves.

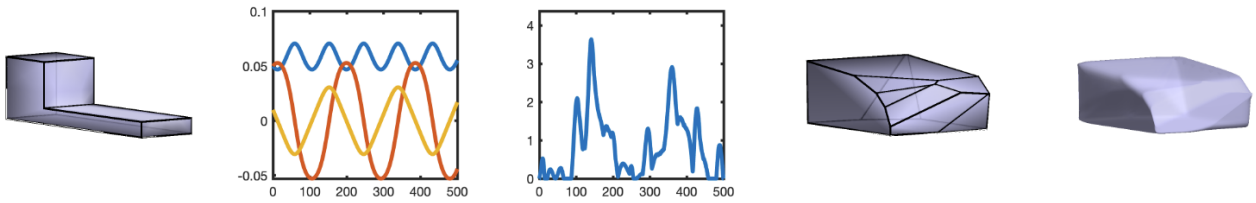
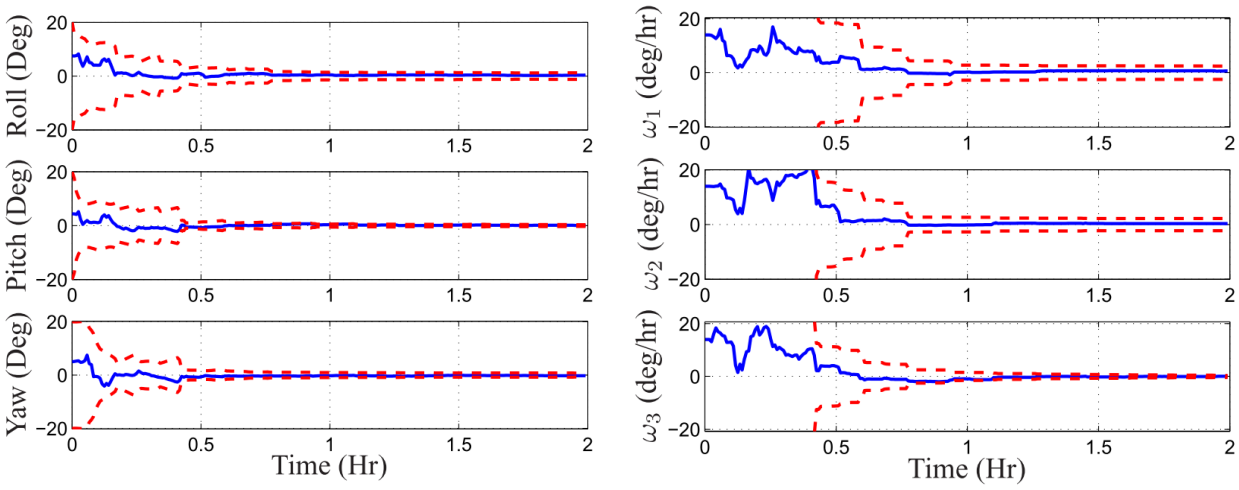


Figure 3.4: Light curve inversion of non-convex box-wing shape by Robinson and Frueh [19]. From left to right: true shape, angular velocities, normalised light curve, convex guess and non-convex guess.

3.5. Attitude and angular velocities

The attitude and angular velocities of space objects are the most frequently estimated parameters using light curve inversion, as these largely determine the accelerations the objects experience due to drag and SRP, combined with shape. One of the methods to estimate these parameters is MMAE, as seen before in Linares et al. [18]. It is also possible to use just a single UKF, like in Wetterer et al. [6] and Linares et al. [20]. In all cases the attitude, angular velocities, position and velocity states are estimated using UKFs, sometimes with extra parameters like shape, mass and surface parameters, as discussed in the previous sections. One set of error results for attitude and angular velocities estimation is shown in Figure 3.5.



(a) Attitude errors.

(b) Angular rate errors.

Figure 3.5: Error plots for the attitude and angular rate estimations by Linares et al. [20].

It can be seen that the errors converge very nicely around 0 in all cases, which is the case in the results from most papers. However, it has been shown by Dianetti et al. [21] that the convergence of these filter methods is very sensitive to initial guess in attitude and angular velocities, and if the guesses are too far from the truth they do not converge. This is quite an important limitation, which will be discussed further in the next chapter. Additionally, Linares et al. [18] discovered that the attitude and angular rate estimation are very sensitive to the shape model used in the estimation, and that a wrong shape model leads to high errors in both parameter estimations.

A different approach to estimate angular velocities is taken by Binz et al. [22]. They analyse the light curves using frequency based method, like the Fourier transform, in order to estimate an upper bound of the possible angular velocities of retired GEO satellites, from real brightness measurements of these objects. They introduce an interesting new method, which they have named the cross-residual technique. It works by taking a part of the light curve and overlaying this on other parts, subtracting the curves from each other and calculating the squared and average residuals. In this way several possible rotational periods are extracted from the signal where the curves overlapped the most, and by some extra analysis a most likely upper bound rotation rate is determined. This method is subject to some ambiguities, as high-frequency peaks that are due to repeated reflections in a single rotation, like those from solar panels on opposite sides, are also picked out, but that is why they determine upper bounds, with the comment that actual rotation rates are most likely (much) lower.

Others have attempted to estimate only the spin axis orientation. Santoni et al. [23] estimate the spin axis orientation for cylindrical objects, i.e. rocket bodies, assuming the object is undergoing a flat spin, meaning that the rotation axis is perpendicular to the cylinder central axis. They observed a real rocket body for multiple passes, and made an estimate for the spin axis by overlapping the possibilities for the different light curves. The results look to be of good quality, but this method is of course only valid for rocket bodies undergoing simple rotation.

Koshkin et al. [24] estimate the spin axis for more complicated satellite shapes. Similar to Binz et al. they make use of similar patterns in light curves, but this time from different light curves from different observation stations. They call these photometric patterns, and they use these to extract the spin axis orientation based on the geometries of the observations. Their method seems to work well, but mainly due to the fact that they use multiple observation stations. This increases the observability of the parameters, but it takes more effort to measure an object with multiple observation stations, and this is also not always possible. Ideally, parameters are estimated using a single observation station, from a single observation pass. This is however the other extreme, and would most likely lead to poor observability, and is not likely to be possible in real life.

4 Characterisation methods

In this chapter a few methods used in space object characterisation using light curve inversion will be discussed in more detail. Some methods have already been mentioned previously, while others have not been covered in this literature review yet. The methods that will be discussed are light curve simulation (Section 4.1), Multiple-Model Adaptive Estimation (Section 4.2), Variable-Structure Multiple-Model Estimation (Section 4.3), Machine Learning (Section 4.4), Particle Swarm Optimisation (Section 4.5) and Extended Gaussian Image (Section 4.6).

4.1. Light curve simulation

In most papers the light curves that are used for characterisation are simulated. This is because some sort of truth is required in order to assess the performance of a method, and in most cases this is not available for real objects, like attitude data of inactive satellites. One difficulty in simulating the light curve of a satellite as mentioned by Robinson and Frueh [19] is that often these objects are non-convex, which can cause self-shadowing. This makes it not straightforward to accurately calculate the amount of light an object reflects to an observer. A very accurate way of dealing with this is ray tracing, like used by Kobayashi and Frueh [25], where the paths of individual light rays are followed to determine if they are reflected to an observer. This method is very computationally expensive, and therefore not suited for many applications.

One commonly used method to model the light reflections are bidirectional reflectance distribution functions (BDRFs) as described by Wetterer et al. [26]. For this method the shape of a space object is divided in different facets. For each facet the amount of light reflected diffusely and specularly is determined based on geometry and the reflective properties of the surface. The amount of light reflected to an observer is then summed over all the facets comprising the object, and a light curve measurement is created. In this way self-shadowing can be modelled to a certain extent. There are different types of BDRFs, like non-Lambertian [27], Phong [28] and Cook-Torrance [6][29]. These methods differ in the way the diffuse and specular reflectances are calculated.

4.2. Multiple-Model Adaptive Estimation

A method that is frequently used in space object characterisation, mainly by Linares et al. [4][15][18], is called Multiple-Model Adaptive Estimation (MMAE). The workings of the algorithm is as follows. An unknown system is modelled with different physical models, like different shapes and spin states. Each physical model is implemented in a respective estimator. Most commonly used is the Unscented Kalman Filter (UKF), as it is well suited for non-linear problems like light curve inversion [30]. All the different estimator implementations with their respective physical models are called the model bank. The MMAE algorithm tries to fit the observations, so measurements of the real system, to the different physical models in the estimator. This is done by calculating conditional probabilities of how well each model fits the observations. This process is repeated for all observations, with the goal for a single model to converge on a probability approaching 1, meaning that that model matches the real system the best. A flow chart of this process is shown in Figure 4.1.

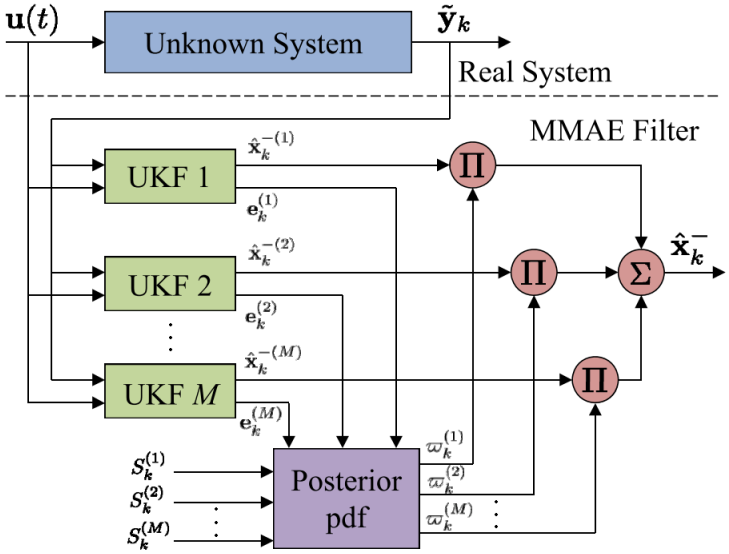


Figure 4.1: Flow chart visualising MMAE process [18].

An important characteristic of this method is that it uses a fixed amount of models in the model bank. This can be called a Fixed-Structure Multiple-Model (FSMM) estimator. If too few models are included in the model bank, this could lead to the algorithm not converging, according to Li et al. [31], but if too many models are included this could overcrowd the algorithm, next to increasing the computational cost, which would also deteriorate the performance. It was seen in Section 3.4 that Linares et al. [18] only used a few shape models based on similar reference shapes, and good convergence was obtained. This might not be the case anymore if more models and models with different shapes are added to the model bank. Next section introduces a possible solution to this.

4.3. Variable-Structure Multiple-Model Estimation

This method can be seen as a continuation on MMAE. As the name implies, a Variable-Structure Multiple-Model (VSMM) estimator does not have a fixed structure, meaning that the models it uses in the model bank during estimation can change. At every time instance the estimator determines which models should be included in the filters, and which should be removed or 'switched off'. This could be a good way to improve on the shortcomings of MMAE discussed previously. At the time of writing there has been no study yet on the use of VSMM for space object characterisation.

There are a couple of ways VSMM could improve on the performance of MMAE. It is mentioned by Li et al. in [31] that a VSMM estimator has the capability of using additional information encountered during estimation, apart from the measurements. They give the example of a car driving on a highway. When driving on a straight section of the road, it is not necessary to include models for turning in the model bank, as this results in unnecessary computational cost. When the car encounters an intersection however, it is possible that the car will make a turn, so then it becomes necassary to include turning models in the model bank.

For the light curve inversion problem this additional information could be glint, which is defined as a rapid change or a spike in brightness in a light curve. Matsushita et al. [32] have shown that this phenomenon can be used to improve attitude estimations, as glint only occurs for specific observation geometric conditions. Using this information it is possible to correct attitude estimates, especially when the initial guess used in the estimation is far from the truth, which has been previously established as a problem for filter based estimation methods in Section 3.5.

Another powerful characteristic of VSMM that could be used is the way it is determined which models are included in the model bank and thus used in the estimation filters. According to Li et al. [31] there are two main categories that can be distinguished: active model-set and model-set generation. In active model-set all the models that could be used in the estimation, called the total model-set, are determined in advance. This means the total model-set is finite. The VSMM estimator then selects at each time which models are used in the estimation and which are not, with different possibilities on how these decisions are made. In model-set generation new models are generated in real time during the estimation process, so the total model-set can not be determined in advance.

Especially a VSMM estimator of model-set generation type could be a powerful tool in the space object characterisation of for example shapes, where it is not possible to predetermine all possible shapes and sizes a space object could have. This could be a solution to a problem encountered in Section 3.4, where in Linares et al. [18] the MMAE could not converge on a single shape model, but alternated between two shape models that were close to the true model. With model-set generation new shapes could be generated starting from these two close shapes in order to converge on the true shape.

4.4. Machine Learning

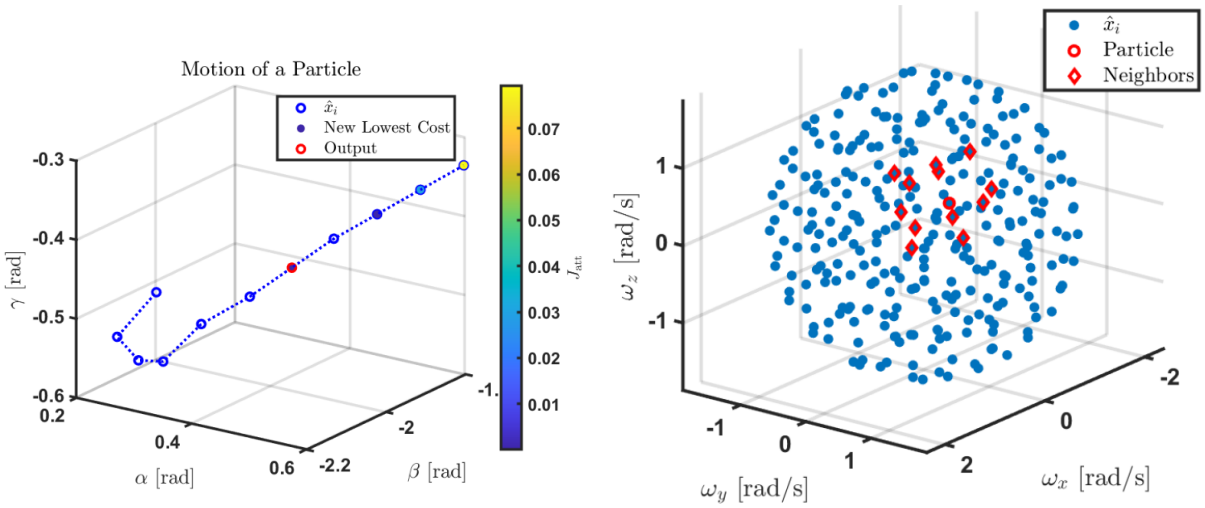
As with many fields in the current day and age, the concept of Machine Learning (ML) and more specifically Deep Learning (DL) has been applied to space object characterisation from light curves. So has Linares et al. [33], an author who has been encountered in many papers in this literature review. One major downside of using ML is that it requires a lot of training data. For space object characterisation this would mean a lot of light curves, with knowledge about the true states of the satellites in question. These are not available in the quantities required for ML training, so use is made of simulated data to train the ML algorithm.

While the ML algorithms are able to quite accurately characterise the test objects using simulated test data, the performance on real data is significantly worse. Reasons for this are that the real data is subject to noise, gaps in data and a lot of different lighting conditions. A good example of this was encountered by Antón et al [28]. They observed the same satellite throughout different periods of the year, and for some cases the light curves had significantly different features due to shadowing of the satellite by the Earth. All of these effects would need to be included in the generation of simulated training data, otherwise the ML algorithm does not know how to deal with them. While running ML algorithm itself is computationally not expensive, the proper training of the algorithm to accurately handle real data becomes a very labour intensive and non-trivial task.

4.5. Particle Swarm Optimisation

A method that looks to tackle a few of the challenges previously mentioned about filter based estimation methods is the Particle Swarm Optimiser (PSO) implemented by Burton [34]. More specifically, they tackle the issue of filter methods' convergence depending on an initial guess that is close to the true initial state. The PSO itself is an optimisation method that uses particles and neighbours that move through a state space in order to minimise a cost function.

Their attitude estimation method works by first generating possible initial attitude vectors corresponding to the first (or if necessary another) measurement of a light curve using a specific PSO. The motion of one particle in this process is shown in Figure 4.2a. After this a full state PSO generates possible initial angular velocity state vectors within a certain constrained bound, as shown in Figure 4.2b. Each initial angular velocity vector is paired with each of the initial attitude vectors found by the first PSO, and these pairs are used in the optimisation to estimate the true initial state, both attitude and angular velocities, that best fits the respective light curve. The best possible initial states found by the optimiser are then put into a refinement algorithm to further improve the fit to the light curve. The results show the method is able to find initial states matching the light curves of differently shaped objects quite well.



(a) Motion of a particle through attitude space.

(b) Generated initial angular velocity vectors.

Figure 4.2: Plots showing the workings of PSO attitude estimation [34].

A similar method is used by Cimino et al. [35]. They however use two optimisation algorithms to estimate the initial state to fit a light curve, one based on a PSO and one based on a Genetic Algorithm (GA). Their results also come close to finding the correct initial states for different satellites. It could be interesting to explore if such a method can be combined with MMAE or VSMM, where a PSO algorithm generates possible initial guesses for the filters, which perform the rest of the parameter estimation using these initial guesses.

4.6. Extended Gaussian Image

A method that was seen in Section 3.4 was the Extended Gaussian Image (EGI) for the inversion of shapes from light curves. Robinson and Frueh [19] found that this method was able to accurately recover the shape of convex objects, but not those of non-convex objects, such as the box-wing shapes common to satellites. A similar conclusion was made by Vallverdú Cabrera et al. [36]. Instead, they attempted to determine an equivalent convex shape for a box-wing satellite. The result is shown in Figure 4.3, with the satellite shown in black and the equivalent convex shape in red. While this might not be the desired outcome in shape estimation, it could still provide information about the approximate dimensions and shape of an object.

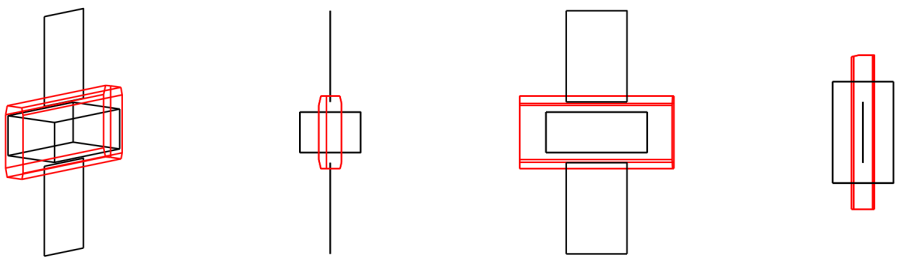


Figure 4.3: Equivalent convex shape estimation of a non-convex box-wing satellite [36]. Left: 3D visualisation. Rest: plane projections.

5 Research proposal

5.1. Research questions

Based on the literature reviewed in this report it is clear that space object characterisation through light curve inversion is a field where lots of research is being performed. Over the years performances and results have improved, for example by making use of advanced estimation techniques like MMAE and ML. There is still a lot of room for improvement however. Next to this there is also need for this improvement, as the space debris problem continues to grow every year, and more effort is required to characterise objects in orbit that are a potential threat to missions. Based on this the following main research question for the MSc. thesis is drafted:

What improvements can be made on existing techniques for space object characterisation via light curve inversion (MMAE, ML, PSO and others) to estimate attitudes, angular velocities and shapes?

It was decided to focus on the parameters of attitude, angular velocity and shapes because it was shown, for example in Wetterer et al. [9], that these parameters are strongly related, and the most important parameters to estimate in order to more accurately model the orbits of these objects for ADR and collision avoidance purposes.

Additionally, there are two research questions drafted that are related to the main question, which will be attempted to be answered during the thesis if time allows it. The first additional research question is stated as:

How can filtering methods be made less dependent on a close initial guess for good convergence?

This question is based on the fact that filtering methods, like the UKFs used in MMAE by Linares et al. [18], are very powerful methods to estimate object parameters, but have as main flaw that they require an initial guess that is close to the true initial state in order to properly converge on an estimate. In real life scenarios this might not be possible, for example if an unknown object is detected and needs to be characterised. Therefore it is important to investigate whether this dependence can be reduced, so that these methods can be applied in more scenarios. The second additional research question that is drafted is as follows:

What are the requirements on light curve measurements (quality, data rate, amount) to get correct estimates?

The reason this question is added is that throughout the literature light curve data is dealt with in numerous different ways. In most cases the data is simulated, but sometimes with thousands of measurements and hours of observations, and sometimes without any noise. In real life scenarios there are often much less preferable observation conditions, with a lot of noise or very short observation passes. It is therefore of interest to investigate what limits there are on these conditions that still make it possible to provide accurate enough estimations.

5.2. Research objectives

Based on the research questions a few research objectives are made. They are listed in Table 5.1 below, in order of priority for the thesis. The main focus will lie on the development of a VSMM estimation algorithm that can estimate the attitude, angular velocities and shape of a space object, and the comparison of the performance from this algorithm with the performance of some of the other methods encountered in this literature review.

Next, validation is an important step in the development of any algorithm. It is therefore desired to validate the VSMM estimation algorithm with real light curve data, in order to prove or disprove its use in real life space object characterisation. It has been seen in the literature that this is difficult to do, but an attempt will be made nonetheless. After this the focus is on the two additional research questions, where the question related to the initial guess of filtering methods was deemed to be of higher importance than the second additional question.

Table 5.1: Research objectives.

#	Objective
1	Develop a VSMM estimation algorithm to estimate space object shape, attitude and angular velocity.
2	Compare VSMM estimation results with current techniques like MMAE.
3	Validate the VSMM estimation algorithm with real light curve data.
4	Look into the viability of combining PSO initial guess generation with filter methods to improve initial guess independence.
5	Investigate different aspects that influence measurements, like quality, amount and rate, in order to assess minimum performance requirements.

The research objectives can be broken down in distinct tasks, which are given in Table 5.2 below. The task number corresponds to the research objective in Table 5.1. In the table there is an additional Task 0, which corresponds with setting up the simulation environment to be used in the subsequent estimations. A few decisions for this environment need to be made, based on some analyses, like the choice of integrator and propagator to be used in the simulations.

Table 5.2: Research tasks overview

(a) Task 0: Simulation environment

Task	Description
0.1	Determine test object orbits
0.2	Perform integrator/propagator analysis
0.3	Select force models
0.4	Select degrees of freedom (DOF) (uncoupled 3DOF or coupled 6DOF)

(b) Task 1: Base estimations

Task	Description
1.1	Implement attitude propagation & estimation
1.2	Verify attitude estimation
1.3	Implement light curve simulation
1.4	Verify light curve simulation
1.5	Implement shape estimation from light curves

(c) Task 2: VSMM estimation

Task	Description
2.1	Implement MMAE algorithm
2.2	Verify MMAE shape estimation
2.3	Develop VSMM algorithm
2.4	Compare VSMM results to MMAE

(d) Task 3: Validation

Task	Description
5.1	Prepare real measurement data for validation
5.2	Validate MMAE algorithm
5.3	Validate VSMM algorithm
5.4	Compare validation results

(e) Task 4: PSO initial guess generation

Task	Description
3.1	Implement PSO algorithm
3.2	Verify PSO initial guess generation
3.3	Combine PSO with VSMM
3.4	Validate PSO-VSMM algorithm

(f) Task 5: Measurement requirements

Task	Description
4.1	Investigate influence of data quality
4.2	Investigate influence of data amount
4.3	Investigate influence of data rates
4.4	Perform cross-analysis

It can be seen that there are also some verification tasks. In order to check that implemented algorithms work as expected it will be attempted to verify the algorithms with those found in papers. For the attitude estimation in Task 1.2 this would be comparing the results with those from Crassidis et al. [30]. The light curve simulation verification in Task 1.4 might be compared to Wetterer et al. [26]. For Task 2.2 the shape estimation can be compared to that in Linares et al. [18]. The PSO verification of Task 3.2 can be done with comparison to [34].

Not all tasks will take up the same amount of time. It is for example expected that the implementation of the MMAE (Task 2.1) and the development of the VSMM algorithm (Task 2.3) will take up the majority of the time available until the Midterm. The goal is to have the first results from these algorithms available for the Midterm. It will then be evaluated how many of the remaining tasks can be executed in the time remaining for the thesis.

5.3. Research planning

In the plots below the planning for the MSc. thesis research is shown, divided in the time before and after the midterm, in Figure 5.1 and Figure 5.2 respectively. The planning is broken down in hours spent per week on the three main research activities: literature, code and writing.

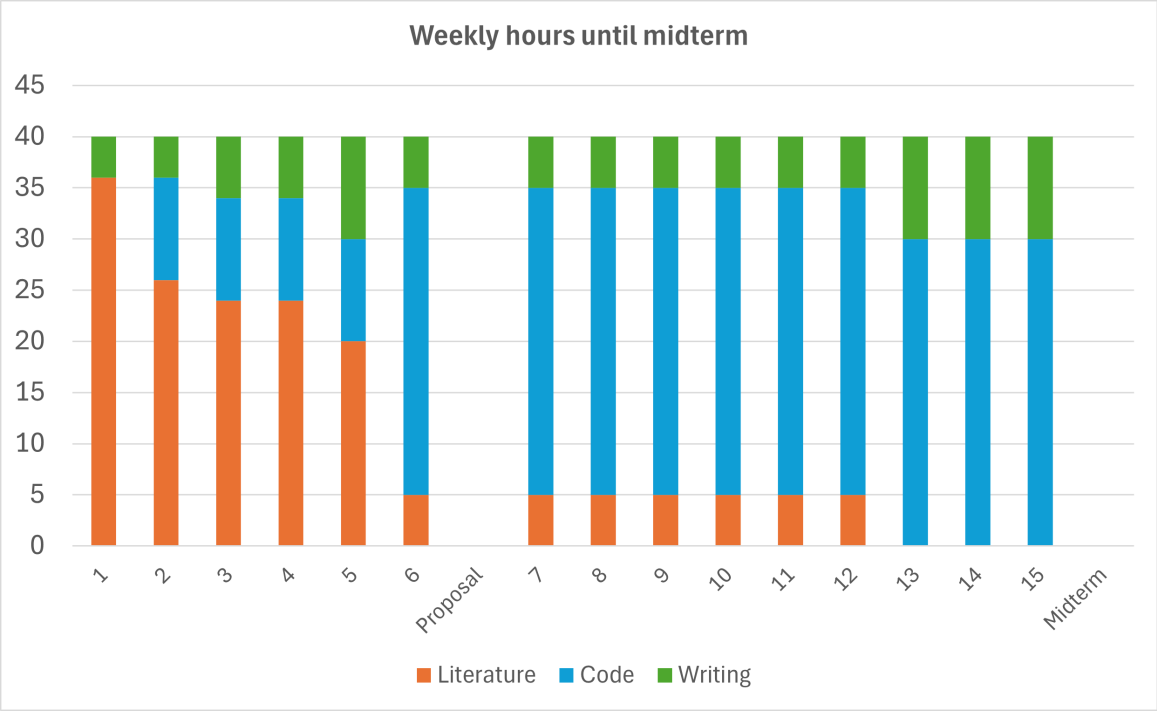


Figure 5.1: Research planning until midterm.

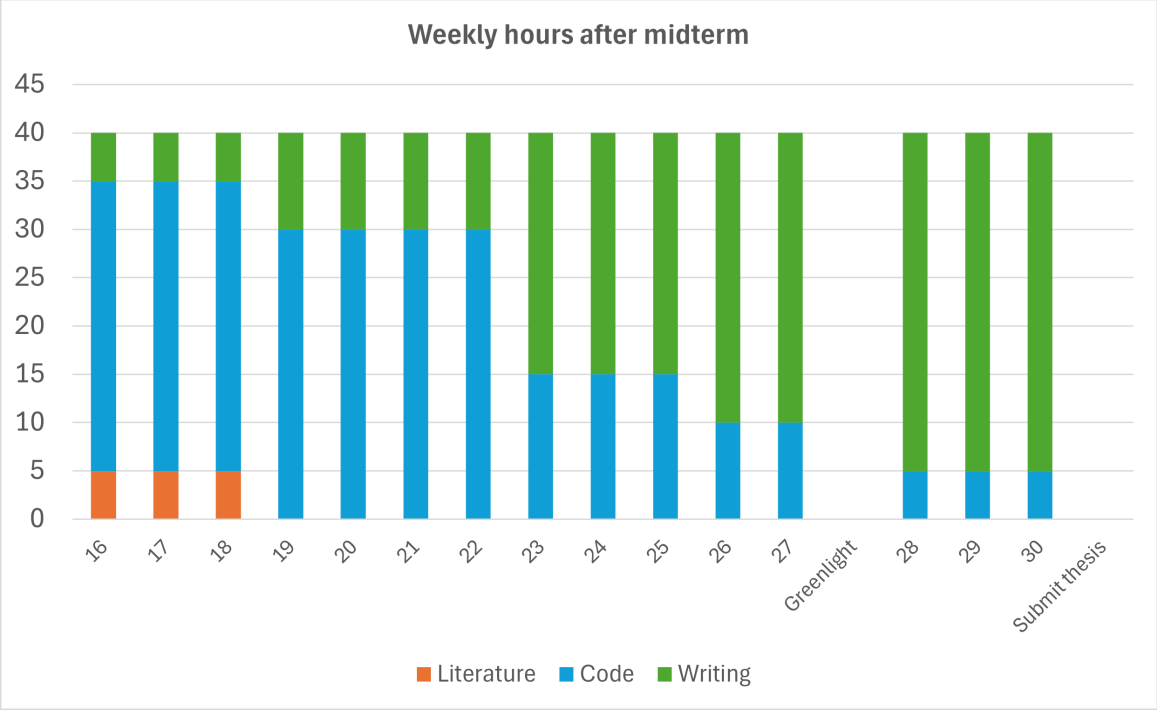


Figure 5.2: Research planning after midterm.

References

[1] ESA Space Debris Office. *ESA's Annual Space Environment Report*. Tech. rep. Version 8.0. European Space Agency, July 2024. URL: https://www.sdo.esoc.esa.int/environment_report/Space_Environment_Report_latest.pdf.

[2] D. J. Kessler and B. G. Cour-Palais. "Collision Frequency of Artificial Satellites: The Creation of a Debris Belt". In: *Journal of Geophysical Research* 83.A6 (1978), pp. 2637–2646. DOI: 10.1029/JA083iA06p02637.

[3] Mark Matney. *Space Debris Reentry and Mitigation*. Conference Paper JSC-CN-37432-1. Presented at VKI Lee Series, Space Debris Reentry and Mitigation, Brussels, Belgium, September 12-14, 2016. NASA Johnson Space Center, Sept. 2016. URL: https://ntrs.nasa.gov/citations/20160011226.

[4] Richard Linares et al. "Space object shape characterization and tracking using light curve and angles data". In: *Journal of Guidance, Control, and Dynamics* 37 (1 2014), pp. 13–25. ISSN: 15333884. DOI: 10.2514/1.62986.

[5] Jiri Šilha et al. "Light Curves Application to Space Debris Characterization and Classification". In: *Proceedings of the 8th European Conference on Space Debris*. Vol. 8. 1. Conference paper. Comenius University. European Space Operations Centre (ESOC), Darmstadt, Germany: ESA Space Debris Office, 2021. URL: https://conference.sdo.esoc.esa.int/proceedings/sdc8/paper/168/SDC8-paper168.pdf.

[6] Charles J. Wetterer and Moriba Jah. "Attitude estimation from light curves". In: *Journal of Guidance, Control, and Dynamics* 32 (5 2009), pp. 1648–1651. ISSN: 15333884. DOI: 10.2514/1.44254.

[7] Jiří Šilha et al. "Space debris observations with the Slovak AGO70 telescope: Astrometry and light curves". In: *Advances in Space Research* 65.8 (2020), pp. 2018–2035. ISSN: 0273-1177. DOI: https://doi.org/10.1016/j.asr.2020.01.038. URL: https://www.sciencedirect.com/science/article/pii/S0273117720300727.

[8] M. Jah and R. Madler. "Satellite Characterization: Angles and Light Curve Data Fusion for Spacecraft State and Parameter Estimation". In: *Advanced Maui Optical and Space Surveillance Technologies Conference*. Ed. by S. Ryan. Jan. 2007, E49. URL: https://www.researchgate.net/publication/242579202_Satellite_Characterization_Angles_and_Light_Curve_Data_Fusion_for_Spacecraft_State_and_Parameter_Estimation.

[9] Charles J. Wetterer et al. "Simultaneous position, velocity, attitude, angular rates, and surface parameter estimation using astrometric and photometric observations". In: *Proceedings of the 16th International Conference on Information Fusion*. 2013, pp. 997–1004. URL: https://ancs.eng.buffalo.edu/pdf/ancs_papers/2013/FUSION2013.pdf.

[10] Jiří Šilha et al. "Apparent rotation properties of space debris extracted from photometric measurements". In: *Advances in Space Research* 61 (3 Feb. 2018), pp. 844–861. ISSN: 18791948. DOI: 10.1016/j.asr.2017.10.048.

[11] Alexander Burton and Carolin Frueh. "Multi-Hypothesis Light Curve Inversion for Space Object Attitude Determination". In: *Proceedings of the 8th European Conference on Space Debris*. Vol. 8. Conference paper. Purdue University. ESA Space Debris Office, 2021. URL: https://conference.sdo.esoc.esa.int/proceedings/sdc8/paper/18/SDC8-paper18.pdf.

[12] Joanna C. Hinks, Richard Linares, and John L. Crassidis. "Attitude observability from light curve measurements". In: *AIAA Guidance, Navigation, and Control (GNC) Conference*. American Institute of Aeronautics and Astronautics Inc., 2013. ISBN: 9781624102240. DOI: 10.2514/6.2013-5005.

[13] Byron D. Tapley, Bob E. Schutz, and George H. Born. "Chapter 4 - Fundamentals of Orbit Determination". In: *Statistical Orbit Determination*. Ed. by Byron D. Tapley, Bob E. Schutz, and George H. Born. Burlington: Academic Press, 2004, pp. 159–284. ISBN: 978-0-12-683630-1. DOI: https://doi.org/10.1016/B978-012683630-1/50023-0.

[14] Alex M. Friedman and Carolin Frueh. "Observability of Light Curve Inversion for Shape and Feature Determination Exemplified by a Case Analysis". In: *Journal of the Astronautical Sciences* 69 (2 Apr. 2022), pp. 537–569. ISSN: 21950571. DOI: 10.1007/s40295-021-00293-w.

[15] Richard Linares, John L. Crassidis, and Moriba K. Jah. "Space object classification and characterization via Multiple Model Adaptive Estimation". In: *17th International Conference on Information Fusion (FUSION)*. 2014, pp. 1–7. URL: https://apps.dtic.mil/sti/pdfs/AD1015362.pdf%5D.

[16] Richard Linares et al. "Astrometric and photometric data fusion for inactive space object mass and area estimation". In: *Acta Astronautica* 99 (1 June 2014), pp. 1–15. ISSN: 00945765. DOI: 10.1016/j.actaastro.2013.10.018.

[17] Matej Zigo et al. "Combined Effort of Reflectance Spectroscopy and BVRI Photometry in the Field of Space Debris Characterization". In: *Proceedings of the 8th European Conference on Space Debris*. Vol. 8. 1. Conference paper. Comenius University. ESA Space Debris Office, 2021. URL: https://conference.sdo.esoc.esa.int/proceedings/sdc8/paper/198/SDC8-paper198.pdf.

[18] Richard Linares et al. "Astrometric and Photometric Data Fusion for Resident Space Object Orbit, Attitude, and Shape Determination Via Multiple-Model Adaptive Estimation". In: *AIAA Guidance, Navigation, and Control Conference*. Aug. 2010. ISBN: 978-1-60086-962-4. DOI: 10.2514/6.2010-8341.

[19] Liam Robinson and Carolin Frueh. "(AAS 22-574) Light Curve Inversion for Reliable Shape Reconstruction of Human-Made Space Objects". In: Conference paper. Aug. 2022. URL: https://www.researchgate.net/publication/366781300.

[20] Richard Linares, Moriba Jah, and John Crassidis. "Inactive space object shape estimation via astrometric and photometric data fusion". In: *Advances in the Astronautical Sciences* 143 (Jan. 2012), pp. 217–232. URL: https://www.researchgate.net/publication/287186964_Inactive_space_object_shape_estimation_via_astrometric_and_photometric_data_fusion.

[21] Andrew Dianetti and John Crassidis. "Resident Space Object Characterization Using Polarized Light Curves". In: *Journal of Guidance, Control, and Dynamics* 46 (Oct. 2022), pp. 1–18. DOI: 10.2514/1.G006847.

[22] C. Binz et al. "Optical Survey of the Tumble Rates of Retired GEO Satellites". In: *Advanced Maui Optical and Space Surveillance Technologies Conference*. Ed. by S. Ryan. Sept. 2014, E61, E61. URL: https://amostech.com/TechnicalPapers/2014/Poster/BINZ.pdf.

[23] Fabio Santoni, Emiliano Cordelli, and Fabrizio Piergentili. "Determination of disposed-upper-stage attitude motion by ground-based optical observations". In: *Journal of Spacecraft and Rockets* 50 (3 2013), pp. 701–708. ISSN: 15336794. DOI: 10.2514/1.A32372.

[24] Nikolay Koshkin et al. "Determination of the spacecraft's spin axis orientation. Photometric patterns method". In: *Advances in Space Research* (2024). ISSN: 18791948. DOI: 10.1016/j.asr.2024.08.038.

[25] Daigo Kobayashi and Carolin Frueh. "Compressed Sensing for Satellite Characterization – Improved atmospheric modeling and sensing matrix realism". In: *8th European Conference on Space Debris*. Apr. 2021, 10, p. 10. URL: https://conference.sdo.esoc.esa.int/proceedings/sdc8/paper/19/SDC8-paper19.pdf.

[26] Charles J. Wetterer et al. "Refining space object radiation pressure modeling with bidirectional reflectance distribution functions". In: *Journal of Guidance, Control, and Dynamics* 37 (1 2014), pp. 185–196. ISSN: 15333884. DOI: 10.2514/1.60577.

[27] Richard Linares and John L. Crassidis. "Space-object shape inversion via adaptive hamiltonian Markov Chain Monte Carlo". In: *Journal of Guidance, Control, and Dynamics* 41 (1 2018), pp. 47–58. ISSN: 15333884. DOI: 10.2514/1.G002296.

[28] Alfredo M. Antón et al. "Artificial Intelligence for Space Resident Objects Characterisation with Lightcurves". In: *Proceedings of the 8th European Conference on Space Debris*. Vol. 8. 1. Conference paper. GMV, Beechleaf Consulting, Space Insight Ltd. ESA Space Debris Office, 2021. URL: https://conference.sdo.esoc.esa.int/proceedings/sdc8/paper/116/SDC8-paper116.pdf.

[29] Laurence David James Blacketer. "Attitude characterisation of space objects using optical light curves". PhD thesis. University of Southampton, Mar. 2022. URL: https://eprints.soton.ac.uk/457200/1/Laurence_Blacketer_Thesis_Final_28_03_2022.pdf.

[30] John L. Crassidis and F. Landis Markley. "Unscented filtering for spacecraft attitude estimation". In: *Journal of Guidance, Control, and Dynamics* 26 (4 2003), pp. 536–542. ISSN: 15333884. DOI: 10.2514/2.5102.

[31] Xiao-Rong Li, Yaakov Bar-Shalom, and William Dale Blair. "Engineer's guide to variable-structure multiple-model estimation for tracking". In: *Multitarget-multisensor tracking: Applications and advances.* 3 (2000), pp. 499–567. URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ee1365d32d153bf1f60d99d27859180c4f30066e.

[32] Yuri Matsushita et al. "Conceptual Study of Improved Photometric Attitude Estimation Using Glint". In: *Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan* 22 (May 2024), pp. 59–65. DOI: 10.2322/tastj.22.59.

[33] Richard Linares, Roberto Furfaro, and Vishnu Reddy. "Space Objects Classification via Light-Curve Measurements Using Deep Convolutional Neural Networks". In: *Journal of the Astronautical Sciences* 67 (3 Sept. 2020), pp. 1063–1091. ISSN: 21950571. DOI: 10.1007/s40295-019-00208-w.

[34] Alexander Burton. "Attitude Estimation Using Light Curves". In: *Purdue University* (July 2024). DOI: 10.25394/PGS.26383684.v1. URL: https://hammer.purdue.edu/articles/thesis/ATTITUDE_ESTIMATION_USING_LIGHT_CURVES/26383684.

[35] Lorenzo Cimino et al. “Genetic Algorithm for Space Debris and Space Objects Attitude Motion Reconstruction through Optical Measurements”. In: *Proceedings of the 2nd NEO and Debris Detection Conference*. Vol. 2. 1. Conference paper. Sapienza University of Rome. ESA Space Debris Office, 2023. URL: https://conference.sdo.esoc.esa.int/proceedings/neosst2/paper/106/NEOSST2-paper106.pdf.

[36] David Vallverdú Cabrera, Jens Utzmann, and Roger Förstner. *Inversion of the shape of space debris from non-resolved optical measurements within SPOOK*. 2021. URL: https://amostech.com/TechnicalPapers/2021/Non-Resolved-Object-Characterization/Vallverdu-Cabrera.pdf.