Monocular-based Pose Estimation Of Unknown Uncooperative Targets

B. Kevers

TU Delft

Monocular-based Pose Estimation Of Unknown Uncooperative Targets

by

B. Kevers

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday December 15, 2021 at 09:30 AM.

Student number:	4436423	
Project duration:	March 1, 2021 – December 15, 2021	
Thesis committee:	Dr. A. Cervone,	TU Delft, Committee chair
	Dr. J. Guo,	TU Delft, Supervisor
	Prof. Dr. G.C.H.E. de Croon,	TU Delft, External examiner

Abstract

The current commercialisation of spaceflight significantly boosts the amount of objects in space, resulting in an increasing risk of involuntary collisions between space debris and operational spacecraft. Collisions are catastrophic for the orbital environment, as they create thousands of pieces of debris flying around that will put other satellites in danger for decades. To sustain a healthy space environment, feasible and cost-effective Active Debris Removal (ADR) and On Orbit Servicing (OOS) solutions are required that can be implemented on a wide scale. Ideal ADR and OOS solutions require no prior information on the target. This research aims to support development in that field by investigating the use of monocular-based Simultaneous Localisation And Mapping (SLAM) for pose estimation around unknown targets.

The research in this work is guided by a designed technical research framework that aims to assess the robustness of monocular SLAM under realistic orbital scenarios. The pose estimator requires robustness to images with high contrast and noise, but also to adverse and changing illumination conditions. Furthermore, it requires the capability to function in dynamic environments, where objects in the background are not fixed in the target body frame. This work also analyses the hardware requirements to obtain stable performance of the pose estimator. Apart from those challenges related to space imagery, it needs to be able to obtain pose estimates at accuracies specified by the mission type.

Due to the complexity and novelty of the field, spacecraft pose estimation of unknown uncooperative targets is still in early phases, as the state-of-the-art algorithms have been tested on simulations and not yet been demonstrated in real-life. In this work, The algorithm of Campos Martínez et al.[12] is adapted for spacecraft relative navigation. Background robustness is achieved by implementing a distance threshold for feature storing. Features that are located further from the camera than a predefined threshold are discarded, thereby separating fore- and background features. Additionally, pose initialisation is optimised by enhancing the robustness under low-parallax conditions, while maintaining real-time performance. Finally, the feature descriptor distance threshold is tuned to improve the accuracy of the algorithm for spacecraft relative navigation purposes. The designed pose estimator is tested on synthetically and lab-generated datasets, both capturing the Delfi-n3Xt satellite. The designed datasets allow for analysis of relative motion effects, environmental effects, like brightness changes and background objects, and for the analysis of hardware requirements.

The suitability of the pose estimator for spacecraft relative navigation is assessed on three factors. Firstly, achievement of robustness to orbital relative navigation conditions is required. This includes relative motion, background, space image quality, and illumination robustness. The robustness to these factors is demonstrated in this report. The second factor is based on the capability of this system to work in real-time on space hardware. In this work, it is found that a lower-limit of 800x600 pixels on image resolution guarantees stable performance of the pose estimator. Furthermore, the frame rate should be high enough such that the target's orientation propagation between frames does not exceed 1 deg/frame. When observing hardware used on Vision-Based Navigation (VBN) systems, it is noticed that the RNS camera in [49] could capture resolutions of 1024x1024px at a frame rate of 3 Hz. Based on the fact that the design of that system was developed more than ten years ago, and the fact that technology advances significantly over the years, it is deemed feasible that the algorithm could be implemented on space hardware. The third and last point is related to the achievable accuracy. On average, this system achieved an accuracy of sub-degree level in orientation, and centimetre-level in relative position. For particular mission phases of Rendezvous and Proximity Operations (RPO), strict accuracy requirements need to be met. For reference, the relative navigation requirements for the e.Deorbit mission [59] and the Servicing Mission 4 of the Hubble telescope [49] are observed. Since the required accuracies of both the e.Deorbit mission and Servicing Mission 4 have been achieved in this work, the designed pose estimation system is deemed suitable for spacecraft relative navigation.

The novelties of this work related to robustness analysis of the implemented pose estimator to spacecraft relative motion, the capability of handling background objects in the frame, and the analysis of hardware requirements, provide a relevant contribution to the work already performed in this field.

Preface

This thesis has been conducted as part of the MSc Aerospace Engineering at TU Delft. This work concludes my time as a student and is the result of accumulated knowledge and experience gathered over the past six years. In those six years, I have worked with great passion, enthusiasm and dedication to develop myself as both a human being and engineer. We, as humans, are perpetually innovating to improve the quality of life. I truly believe that life on Earth will greatly benefit from increasing access to space, and that technology advancements in the area of Active-Debris Removal and On-Orbit Servicing are necessary to maintain a sustainable space environment. With this thesis I aim to contribute to this field.

This thesis did not come without any setbacks and has been a continuous learning process. Therefore, I would like to express my gratitude to my supervisor, Dr. J. Guo, who has guided me from start to end. The quality of this thesis would not have been as it is without his supervision and guidance.

Finally, I would like to thank my parents for allowing me to fearlessly chase my passion and ambitions. I would not be where I am now without their unconditional support.

B. Kevers
Delft, December 2021

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

AAM Active Appearance Models. 103

ADR Active Debris Removal. iii, 2, 3, 6–8, 17, 40, 73, 83–85, 104, 105

AGAST Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. 34

AKAZE Accelerated-KAZE. 20, 32, 34

AM Amplitude-modulated. 89

ARX Autonomous Rendezvous Experiment. 89

ASC Advanced Scientific Concepts. 90

ATV Automatic Transfer Vehicle. 90

AVGS Advanced Video Guidance Sensor. 12, 13, 92, 93, 100

BA Bundle Adjustment. 31, 32, 39, 97, 98

BRIEF Binary Robust Independent Elementary Features. 33, 34

BRISK Binary Robust Invariant Scalable Keypoints. 20, 32, 34

C-W Clohessy-Wiltshire. 42

CCR Corner-cube reflector. 9, 12, 86, 92, 100, 101

CNN Convolutional Neural Network. 14, 94

CPD Coherent Point Drift. 27–29, 40, 46, 62, 68, 73

CSA Canadian Space Agency. 89

CW Continuous-wave. 89

DART Demonstration of Autonomous Rendezvous and Docking. 12, 13, 92

DOF Degrees Of Freedom. 7, 11, 13, 22, 84, 91, 93, 101, 104, 105

DoG Difference-of-Gaussians. 32, 33

DTU Technical University of Denmark. 13, 93

ECI Earth Centered Inertial. 3

EKF Extended Kalman Filter. 18, 26, 43, 45, 74, 106, 107

EKPF Extended Kalman Particle Filter. 18, 107

EO Electro-Optical. viii, ix, 9–11, 86–88, 99

ESA European Space Agency. 6, 83

ETS-VII 7th mission of the Engineering Test Satellite Program. 11, 91

FAST Features from Accelerated Segment Test. 33, 34

FED Fast Explicit Diffusion. 34

FF Formation Flying. 11, 13, 91, 93, 101

FM Frequency-modulated. 89

FOV Field Of View. 12, 88, 89, 91

GMM Gaussian Mixture Model. 28

GNC Guidance, Navigation & Control. 4, 8, 16, 26, 44, 68, 85, 89, 98

GNFIR Goddard Natural Feature Image Recognition. 104

GNSS Global Navigation Satellite System. 9, 86

GPS Global Positioning System. 13, 92

HST Hubble Space Telescope. 7, 85

IMU Inertial Measurement Unit. 40, 107, 108

IP Image Processing. 14, 94, 99

IR Infrared. 6, 84

ISS International Space Station. 2, 6, 83, 84, 89, 90, 104

JPL Jet Propulsion Laboratory. 89

LAMP Laser Mapper. 89

LARS Laser Range Scanner. 89

LCS Laser Camera System. 89

LDRI Laser Dynamic Range Imager. 89

LED Light Emission Diode. 9, 11–13, 86, 91, 93, 100, 101

LIDAR Light Detection and Ranging. ix, x, 3, 6–8, 10, 11, 17, 18, 40, 73, 84, 85, 87–91, 104, 106–108

LoG Laplacian-of-Gaussian. 33

LOS Line Of Sight. 7, 13, 84, 93, 99

MLDB Modified Local Difference Binary. 34

MOI Moment of Inertia. 18, 107

MRPs Modified Rodrigues Parameters. 43

MSFC Marshall Space Flight Center. 12, 92

NASA National Aeronautics and Space Administration. 6, 83

NASDA National Space Development Agency of Japan. 11, 91

NRM Newton-Raphson method. 105

OE Orbital Express. 8, 12, 13, 85, 92, 100

OOS On Orbit Servicing. iii, 2, 3, 6–8, 11, 40, 73, 83–85, 91, 101

ORB Oriented FAST and Rotated BRIEF. 20, 27, 32–35, 39, 40, 45, 47, 48, 65, 74, 76

PCA Principle Component Analysis. 103

PnP Perspective-n-Point. 99, 102

PXS Proximity Operation Sensor. 11, 12, 91, 92, 100

REG Rendezvous Entry Gate. 7, 84

RF Radio-Frequency. 9, 86

RNS Relative Navigation Sensor. 13, 93, 104

RPO Rendezvous and Proximity Operations. iii, x, 2–7, 71, 75, 83, 84

RVS Rendezvous and Docking Sensor. 90

SEPS Simultaneous Estimation of Pose and Shape. viii, ix, 17, 18, 106

SFM Structure from Motion. viii, 32

SIFT Scale Invariant Feature Transform. 20, 32–34, 39

SLAM Simultaneous Localisation And Mapping. iii, viii, 3–5, 16–19, 25–37, 39, 40, 45–48, 60, 64, 66, 73–76, 97, 98, 105–107

SNL Sandia National Laboratories. 89

SRF Sensor Reference Frame. 14, 94, 100

SURF Speeded Up Robust Features. 20, 32, 33

TM Template Matching. ix, 14, 95, 102–104

TOF Time Of Flight. 89, 90

TRF Target Reference Frame. 14, 94, 99, 101

TRIDAR Triangulation LIDAR sensor. 90

VBN Vision-Based Navigation. iii, viii, ix, 2, 3, 5, 6, 8, 10, 13, 16, 23, 51, 71, 75, 83–87, 93, 98, 99, 101, 107

VBS Visual Based System. 13, 93

VNS Vision Navigation System. 90

I

Introduction and Overview

1 Introduction

For most of the history of spaceflight, orbits around Earth have been treated as an infinite resource and satellites as disposable articles. With the increasing number of objects in orbit, and the space infrastructure shifting to *mega constellations*, the environment around Earth may not remain sustainable [74]. As a result, spacecraft Rendezvous and Proximity Operations (RPO) require further advancements and technology developments. This thesis aims to contribute to the future of spacecraft RPO. The motivation for this work is further outlined in section 1.1. Subsequently, the background information for the research is described in section 1.2. Section 1.3 provides an overview of the research objective and questions, after which the structure of the report is shown in section 1.4.

1.1. Motivation

Ever since the inception of the space age, there has been more space debris in orbit than operational satellites [23]. With the current commercialisation of spaceflight, the amount of objects in space rises significantly, resulting in an increasing risk of involuntary collisions between operational payloads and space debris. The first time that two satellites have collided in orbit happened during the Iridium 33 and Cosmos 2251 collision in 2009 [35]. This collision, occurring at a speed of 11,700 m/s, created thousands of pieces of debris larger than 1 cm, putting many other satellites in similar orbits at increased risk for decades to come. The Cosmos 2251 satellite had been out of service since 1995. However, this has not been the only fragmentation of space debris. Over the last two decades, on average 12 accidental fragmentations occur in space each year [23]. To improve the health of the space environment, feasible and cost-effective Active Debris Removal (ADR) solutions are required.

An additional method to improve the health of the space environment is through On Orbit Servicing (OOS). By means of OOS, spacecraft's operational lifetime can be extended through maintenance or repair. Various studies have shown the cost-effectiveness of in-orbit repair of damaged spacecraft [31]. The realisation of OOS systems is an active topic of research, as the main technical challenges include hardware complexity, flexibility and autonomy, in particular when the target to be serviced is uncooperative [4].

Both OOS and ADR fall under the branch of spacecraft RPO. This branch of space engineering has been critical in enabling challenging and ambitious space programmes, like the Apollo programme and the International Space Station (ISS), and will play a major role in sustaining a healthy space environment around Earth, as well as in further exploration of our solar system.

One of the greatest challenges in spacecraft RPO lies in the problem of relative navigation. Navigation in these mission domains is typically done by means of Vision-Based Navigation (VBN) systems, which rely on optical sensors to provide state estimates relative to the target. A lot of research is focused around VBN with respect to known targets which would either be cooperative or uncooperative. However, VBN around completely unknown targets is still a relatively unexplored field of study. This work aims to contribute to technology developments within that field.

1.2. Background

The problem of relative navigation in the domain of spacecraft RPO concerns the estimation of the chaser's state relative to its target. In this context, the state is defined as the relative pose (position and orientation) and its derivatives (translational and angular relative velocity). The relative navigation problem is illustrated in Figure 1.1. There, the Earth Centered Inertial (ECI) frame is represented by $\varepsilon = \{E; \hat{e}_1, \hat{e}_2, \hat{e}_3\}$. The origins of the chaser and target satellite frames are respectively represented by S and T. Then, the objective is to find the transformation matrix between the chaser and target frame at each time step. During RPO, this requires full autonomy of the processes, as the time delay with ground station intervention is substantially too big to fulfil the strict mission requirements. Hence, autonomous solutions are necessary.

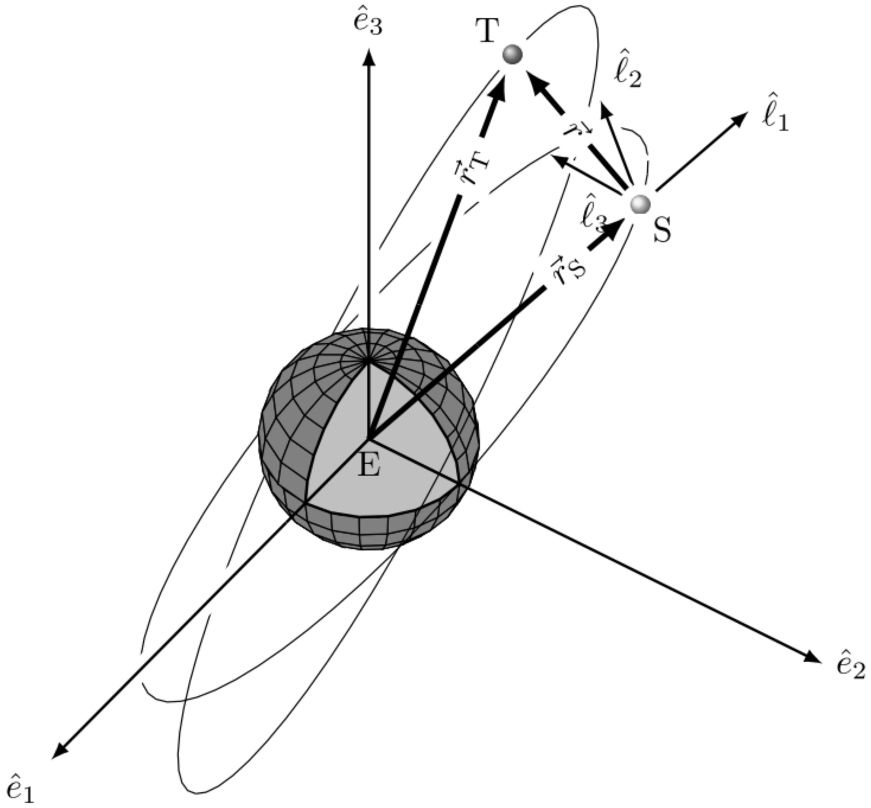


Figure 1.1: The Orbital Relative Navigation Problem [20]

Inherent to the complexity of the mission domain, intricate and costly hardware is imposed to fulfil the mission objectives. In order to stimulate sustainable solutions that can be adapted on a wide scale, cost-effective technologies have to be researched and developed. Currently, relative navigation is mainly performed by means of Light Detection and Ranging (LIDAR) sensors. Those sensors are robust under various illumination conditions and have a high accuracy, but they introduce high cost, mass, power and hardware complexity. Recent advancements in the field of computer vision, robotics, and machine learning have allowed the utilisation of passive monocular cameras as main sensor for VBN. Monocular cameras are superior in terms of cost, mass, power and hardware complexity. Currently, various learning-based algorithms are being researched that implement monocular cameras for relative state estimation. However, due to the working principle of machine learning, those algorithms typically require a known model of its target and a large amount of datasets for training. Since real imagery of spacecraft in orbit is scarce, those datasets have to be created artificially, imposing an uncertainty on the functioning of the algorithm in real environmental conditions. For ideal ADR and OOS missions, no apriori information of the target should be required, meaning that the target is *unknown* by the navigation system. For this, Simultaneous Localisation And Mapping (SLAM) algorithms would be suitable, as they would reconstruct the target and measure its relative pose simultaneously. The research on the application of monocular SLAM algorithms for spacecraft relative navigation is still in early phases. Mainly simulation and laboratory tests have been performed so far. Due to the many unknowns in the suitability of monocular SLAM for the application of spacecraft relative navigation, this is considered the area of interest for this thesis.

The challenge in relative state estimation lies in the determination of the target's pose. Therefore, the focus of this research will be on pose estimation of the target relative to the chaser satellite. Furthermore, the challenge of monocular-based pose estimation for spacecraft relative navigation is based around the information processing of monocular space imagery. Space imagery is characterised by high noise, high contrast, and adverse illumination conditions. Hence, robustness of the pose estimator to these phenomena is essential for the suitability of the pose estimation algorithm for spacecraft applications.

In the context of this work, the pose estimation design will be optimised for close-proximity operations around an unknown target, where the full relative pose is desired to be known. An unknown target is defined as a target object of which no information about their shape, appearance and mass properties is known. Typically, close-proximity is a relative distance smaller than 200 meters, where there is still a safe distance between the chaser and target, but the target is large enough to achieve pose estimation by the on-board sensors. The suitability for the pose estimation system in any other phases of RPO missions is not considered in this research. Neither will the integration of the navigation system in a full Guidance, Navigation & Control (GNC) system be analysed, as the primary focus is on the pose estimation system.

The analysis and testing of the pose estimation will be performed using the Delfi-n3Xt as target. This satellite is a 3U cubesat developed by TU Delft for demonstration of various on-board subsystems. It was launched in 2013 and operated successfully to complete all primary mission objectives. A 1:1 mock-up model of the satellite is available at the University for laboratory tests, as well as a CAD Model for simulated tests. An image of the mock-up model can be seen in Figure 1.2.

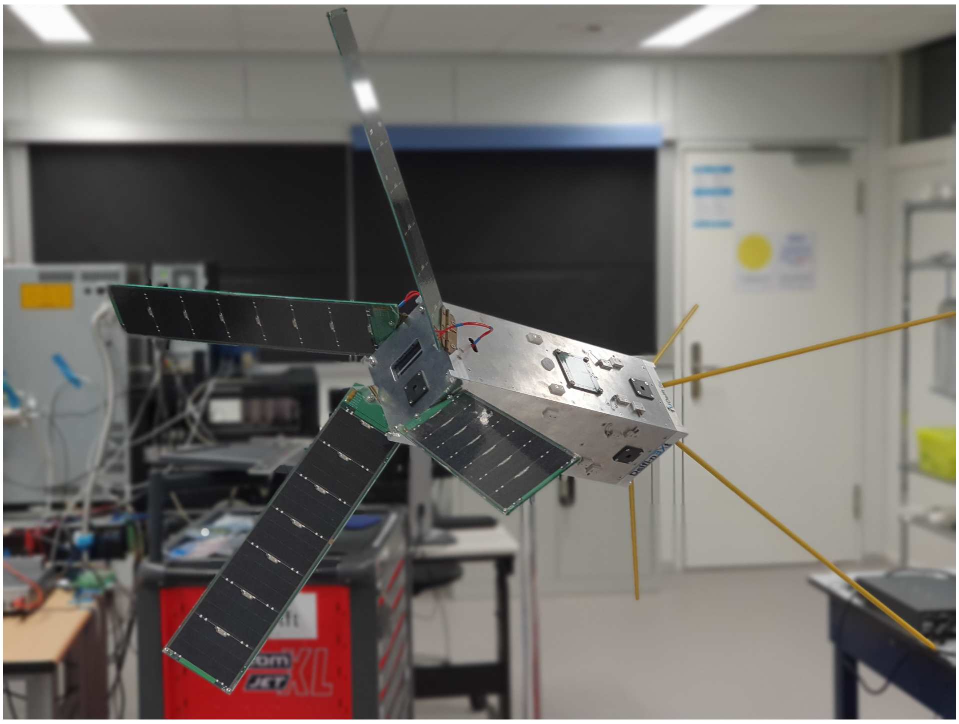


Figure 1.2: Delfi-n3Xt mock-up model

1.3. Research Overview

This research is preceded by a literature study, which is appended in Appendix A. From this literature study, knowledge gaps were found in the field of monocular-based relative navigation in space. The knowledge gaps that provide the basis and motivation of this work are listed below.

- Analysis of unknown target pose estimation based on realistic space imagery,
- Analysis of hardware requirements for unknown target pose estimation,

- Analysis of robustness of the designed pose estimation system under various orbital scenarios,
- Analysis of the integration of the pose estimator with a state estimation system.

These gaps form the primary thesis objective: *To support technology development in the field of spacecraft RPO by researching the suitability of monocular SLAM for spacecraft relative navigation around an unknown target.* This objective results in the primary research question stated as:

How can monocular-based pose estimation be achieved to enable relative navigation towards an unknown target in space?

In order to answer this question and guide the research, the following subquestions are formulated:

RQ-1 What monocular-based pose estimation algorithm is most suitable for spacecraft relative navigation towards unknown targets ?

a. What criteria define the suitability of a pose estimation system for space applications?
b. What are the current approaches to enable pose estimation in unknown environments?
c. How can the scale of the environment be recovered?
d. How can the pose estimation pipeline be integrated in a state estimator for target state tracking?

RQ-2 How does the designed pose estimation algorithm perform?

a. What test scenarios are required to assess the suitability for spacecraft relative navigation?
b. How can the tests be realised?
c. How can the realistic quality of the tests be verified?
d. What hardware specifications would be required to guarantee performance?
e. What are the limiting environmental factors?
f. How does the designed pose estimation system perform compared to the state-of-the-art?

1.4. Thesis Outline

This document comprises five main parts. Part I encapsulates chapter 1 and chapter 2. Chapter 1 outlines the primary motivation and background information to the thesis, as well as an overview of the research objective and questions. Chapter 2 provides a brief overview of VBN. It begins with a description of the missions that exploit VBN systems, after which the monocular state and pose estimation systems in the current state-of-the-art are discussed in more detail. At last, the topic of feature detection, description, and matching is introduced, as well as a brief outline of current relative navigation facilities.

Part II documents the design of the research framework and the navigation system. Chapter 3 offers a high-level overview of the technical research framework that is adapted in this work. Chapter 4 discusses the design decisions of the pose estimation system. Finally, The datasets on which the designed algorithm is tested is elaborated on in chapter 5.

Part III consists of chapter 6 and analyses the obtained results. Furthermore, the chapter is concluded by an analysis of the suitability of SLAM for monocular-based spacecraft relative navigation.

Finally, the thesis is round up in Part IV. Chapter 7 concludes the work and provides recommendations for future studies. The appendices in Part V comprise the literature study in Appendix A, the software work flow in Appendix B, an extensive provision of the results in Appendix C, and finally a written scientific paper about the thesis in Appendix D.

2

Vision-based Navigation Overview

This chapter provides the reader with background knowledge of RPO missions and VBN systems, and the existing literature that this work is based on. First, an overview of typical RPO missions is provided in section 2.1. Subsequently, Vision-based navigation systems are discussed in more detail in section 2.2, after which monocular cameras and monocular-based navigation are described in section 2.3 and section 2.4, respectively. Afterwards, the existing literature is researched on the topic of monocular-based pose estimation of unknown uncooperative targets in section 2.5. The basics of feature detection, description, and matching are given in section 2.6, and an overview of existing relative navigation facilities is provided in section 2.7. Finally, the chapter is concluded in section 2.8.

2.1. Missions Overview

The purpose of this section is to gain an overview of how a generic mission that exploits VBN in RPO looks like. This section starts with a brief overview of the history of RPO. Subsequently, a discussion of ADR missions is presented, including insight in a typical ADR mission. Afterwards, the history of OOS is provided. Finally, asteroid exploration missions are outlined.

2.1.1. Proximity Operations and Rendezvous

Proximity operations date back to the start of the space age, with their roots at the Gemini program in the 1960s. The spacecraft mission involving Gemini VI and Gemini VII was the first mission to perform rendezvous and proximity operations. The operations were performed manually by astronauts by visual monitoring with the support of ground station. This led to the first ever mission which utilised proximity operations in order to meet operational requirements: the Apollo 9 program. Around the same time (1967), the first fully autonomous RPO and docking was executed by two Soviet spacecraft, Kosmos-186 and Kosmos-188, by means of an active inter-spacecraft communication link. From then onward, the advancements in proximity navigation grew. This led to the initiation of orbiting space stations, of which the ISS is the prime example.

2.1.2. Active Debris Removal

In order to stabilise the growth of space debris, ADR missions are necessary. Studies by European Space Agency (ESA) and National Aeronautics and Space Administration (NASA) show that the number of debris objects in orbit would continue to grow even under idealised circumstances, where there are no launches, no debris release, and no explosions [1]. This is caused by an assumed collision rate between objects of one collision per ten years. Consequently, debris mitigation measures alone are not enough to obtain a sustainable, healthy orbital population. This stresses the importance of ADR missions, that reduce the number of large and massive objects in space.

RemoveDEBRIS successfully performed key ADR technology demonstrations, after it was launched in 2018 from the ISS. It demonstrated the use of both a net and a harpoon for capturing, a dragsail deployment for accelerated deorbital rate of itself, and its VBN system [24]. Its VBN comprises a combination of LIDAR, Infrared (IR), and optical cameras. It has been tested by ejecting a second CubeSat at very low velocity (2 cm/s) out of orbit plane, to comply with safety constraints and VBN demonstration needs (lightning, background, range).

Future ADR missions include the ELSE-D mission [10] and the ClearSpace-1 mission.

ADR Mission Profile
In order to get better insight into how and when vision-based navigation is performed in ADR and OOS missions, it is essential to observe a generic ADR mission in more detail. The e.Deorbit mission will be used as a baseline for this. This mission's main objective was to safely remove Envisat (a former remote sensing satellite) from orbit [59]. Its various mission phases are illustrated in Figure 2.1. The chaser spacecraft is injected into a relative orbit, after which it performs several manoeuvres to reach the Rendezvous Entry Gate (REG), about 8 km away from its target, by means of absolute navigation. From that point, the rendezvous phase is initiated, and thus the relative navigation system. It uses Line Of Sight (LOS) measurements from a narrow-angle camera, and LOS and range measurements from a LIDAR sensor, to reduce the distance from 8 km to about 100 meters, when it reaches its Parking Hold Point. At this relative distance, the target is sufficiently large for 6-Degrees Of Freedom (DOF) pose estimation to be initialised. Once the pose estimate has been initialised, it performs state tracking on the target along the V-bar, and executes a target inspection fly-around. It then tries to synchronise its motion with the target. Once this is achieved and it is close enough, it captures the target after which it stabilises and disposes the coupled satellites.

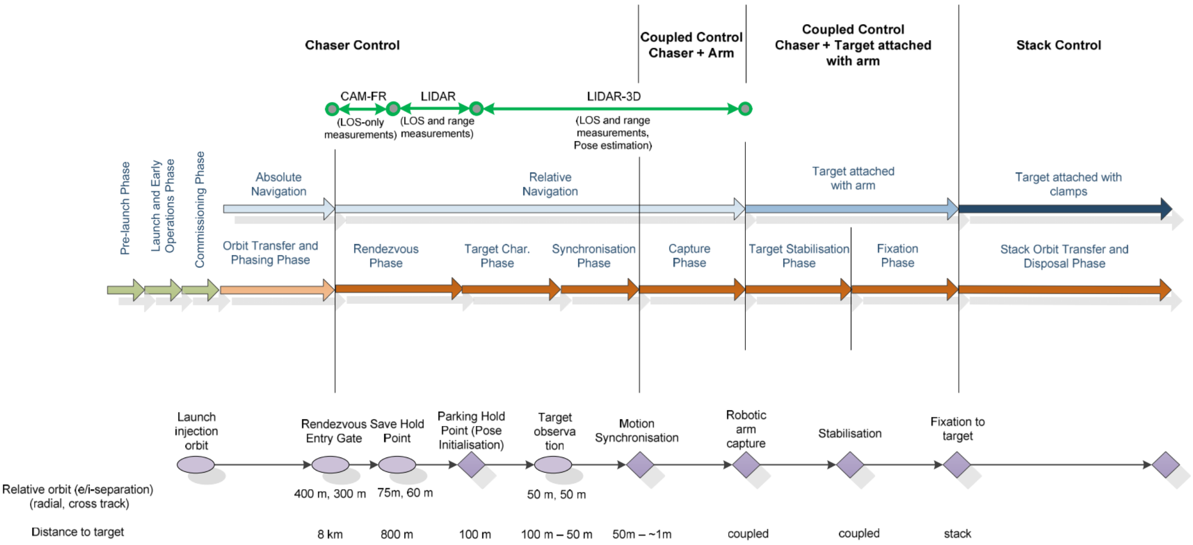


Figure 2.1: e.Deorbit mission phases [59]

The state estimation accuracy requirements during the e.Deorbit mission are listed in Table 2.1. It should be mentioned that these requirements are very mission dependent.

Table 2.1: e.Deorbit RPO performance requirements in close proximity [59]

Performance Parameter (per axis)	Rendezvous, closing (100m)	Rendezvous, Sync. Capture (50m)
Relative Position [m]	10	0.05
Relative Velocity [m/s]	0.1	0.01
Relative Attitude [deg]	5	2
Relative Angular rate [deg/s]	0.5	0.5

2.1.3. On Orbit Servicing

The purpose of OOS is to perform maintenance or repair in order to extend the lifetime of spacecraft. This benefits the health of the space environment as it reduces the amount of debris in orbit. The repair of a jammed solar panel on Skylab-2 was the first real OOS operation. This was done by a crew with the help of ground support [17]. The consequent milestone was achieved by on orbit servicing of the Hubble Space Telescope (HST) by the space shuttle, which performed semi-autonomous rendezvous and docking using its on-board relative navigation system. The close-proximity and docking was performed manually by the astronauts. Advancements in the field of GNC resulted in the first generation of fully autonomous OOS missions, like the DARPA's Orbital Express (OE) mission [52]. This mission executed autonomous GNC from a relative range of 200 km until docking and subsequent servicing functions like fuel transfer, computer and battery replacement. It used laser-based tracking and a monocular sensor for close-proximity navigation. Recently, the Northrop Grumman MEV-1 achieved the first commercial servicing mission on a target not designed for docking. The MEV-1 extended the life of the target spacecraft by five years.

2.1.4. Asteroid Exploration

Asteroid exploration is typically performed to research its materials. This material can be explored to obtain information about the asteroid's composition. This gives insight in the formation of the solar system, but these materials can also be exploited for further exploration and colonisation of the solar system, by using them for in orbit manufacturing or refueling.

The first mission in which a spacecraft performed a fly-by of an asteroid was NASA's Galileo mission, where the spacecraft flew past asteroids Gaspara in 1991 and Ida in 1993. From then onward, there were numerous missions that studied asteroids. The NASA's NEAR-Shoemaker mission studied asteroids Mathilde and Eros; the Rosetta mission observed Steins and Lutetia, Deep Space 1 and Stardust both had close encounters with asteroids, and NASA's Dawn spacecraft orbited asteroid Vesta for more than a year.

Active asteroid exploration missions include OSIRIS-REx, which is an asteroid study and sample-return mission. It was launched in 2016, after which it rendezvoused with Bennu in 2018. It spent the next months studying the surface in order to find a suitable location from which to extract a sample. It successfully collected a sample in 2020. It is expected to be back on Earth in 2023. Its VBN system uses both star fields and asteroid landmarks for navigation, captured by its optical cameras [75].
The Japanese Hayabusa was launched in 2003 and returned asteroid samples from Itokawa in 2010. Its vision-based navigation system was ground-based, meaning the pictures taken from the asteroids were downlinked and the relative pose of the spacecraft was determined on ground [29]. It had multiple visual cameras, as well as LIDARs on board. However, its landing process was autonomous, due to the communication delay of 30 min. From an altitude of 40 m above the surface, a target marker was released, which was used as relative navigation target. This reduces the computational load and increases the image processing certainty.
The successor of Hayabusa, Hayabusa2 was launched in 2014. After launch, it rendezvoused with asteroid Ryugu in 2018 and returned the asteroid samples to Earth in 2020. The relative navigation system is based on the one of Hayabusa, where they released target markers on the asteroid.

Mission Profile

Asteroid exploration missions generally have slight differences compared to ADR and OOS missions. Therefore, a mission profile of a typical asteroid exploration mission will be described: the OSIRIS-REx mission, in particular its proximity operations. Its mission is outlined in Figure 2.2.

After launch and outbound cruise, OSIRIS-REx enters the approach phase. This begins when the asteroid is just bright enough to be detected by the VBN system. It then approaches the asteroid gently in order to optically acquire the asteroid with its on-board cameras [69], and also observing the vicinity of the asteroid for any potential hazards. When close enough, it takes images from the asteroid to start shape model development of the asteroid. The shape modelling is done using two independent strategies, one concerning stereo cameras, and one based on range images. Finally, it selects two target sites for touch down and obtain samples, which are reviewed for suitability.

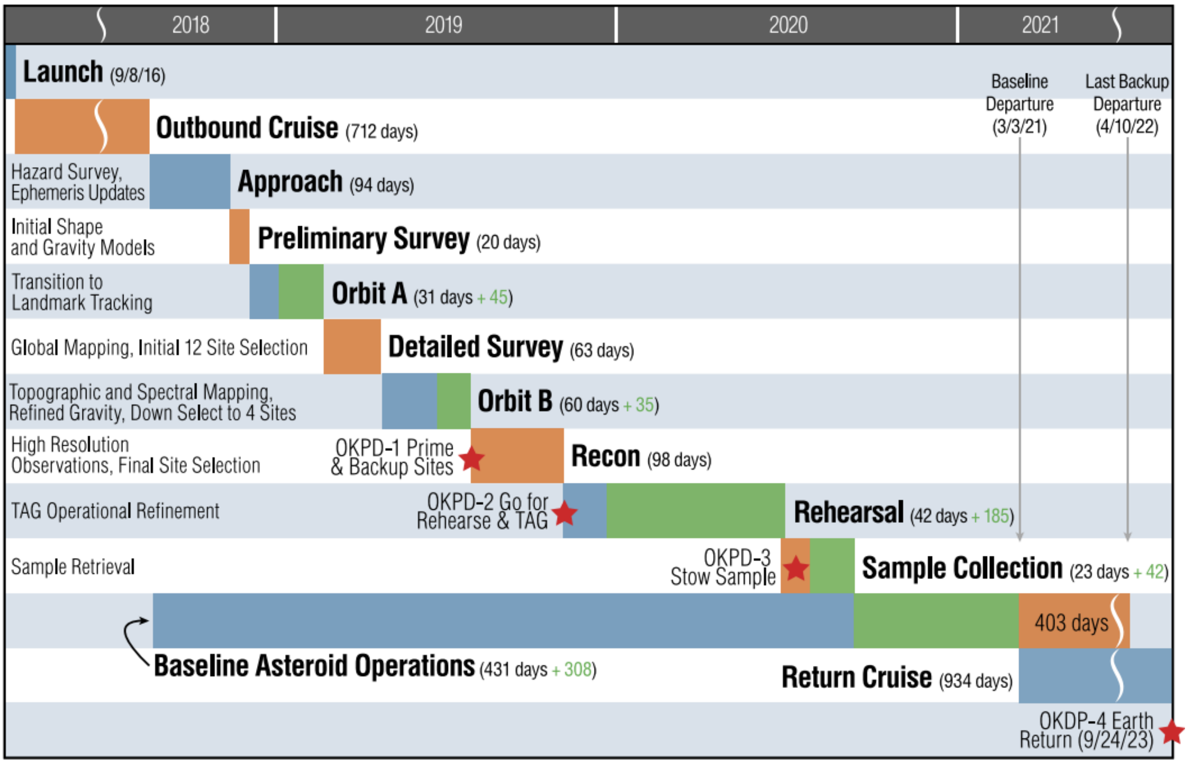
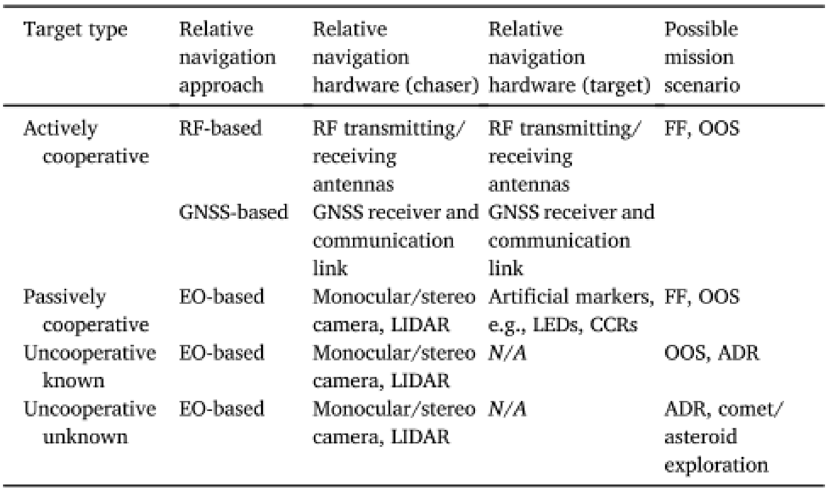


Figure 2.2: Mission outline OSIRIS-REx [69]

2.2. Vision-based Navigation

This section gives a general overview of relative navigation systems in close-proximity of various targets. Close-proximity is typically a distance less than 200 meters between the chaser and target, such that the target is large enough for pose estimation while maintaining a safe distance [4]. In this range, Electro-Optical (EO) sensors are the best option to ensure pose estimation, because of the limitations of Radio-Frequency (RF)-based and Global Navigation Satellite System (GNSS)-based technologies [53]. The taxonomy of space-borne relative navigation approaches and scenarios is illustrated in Figure 2.3. Targets for relative navigation in close-proximity are generally classified as cooperative or uncooperative. Cooperative targets can either be passively or actively cooperative. A passively cooperative target contains artificial markers which can be detected and recognised by the chaser satellite. The markers themselves can be both passive (e.g. Corner-cube reflector (CCR)s) and active (e.g. Light Emission Diode (LED)s). Relative navigation to an uncooperative target is more difficult as there is a lack of cooperative information from the target. Uncooperative targets are further subdivided into known and unknown targets. Known targets are defined as targets of which the geometry/model is known.



Target type	Relative navigation approach	Relative navigation hardware (chaser)	Relative navigation hardware (target)	Possible mission scenario
Actively cooperative	RF-based	RF transmitting/ receiving antennas	RF transmitting/ receiving antennas	FF, OOS
	GNSS-based	GNSS receiver and communication link	GNSS receiver and communication link	
Passively cooperative	EO-based	Monocular/stereo camera, LIDAR	Artificial markers, e.g., LEDs, CCRs	FF, OOS
Uncooperative known	EO-based	Monocular/stereo camera, LIDAR	*N/A*	OOS, ADR
Uncooperative unknown	EO-based	Monocular/stereo camera, LIDAR	*N/A*	ADR, comet/ asteroid exploration

Figure 2.3: Overview of relative navigation scenarios [53]

A VBN system determines the relative state from its sensor measurements. In general, it does so by first determining its pose from the sensor data, after which the relative pose, and potentially features from the sensor data itself are inserted in the state estimator. The pose estimation pipeline generally contains image processing algorithms which extract certain features from the images. The basic components are visualised in Figure 2.4. These blocks will be further elaborated on in the next sections.

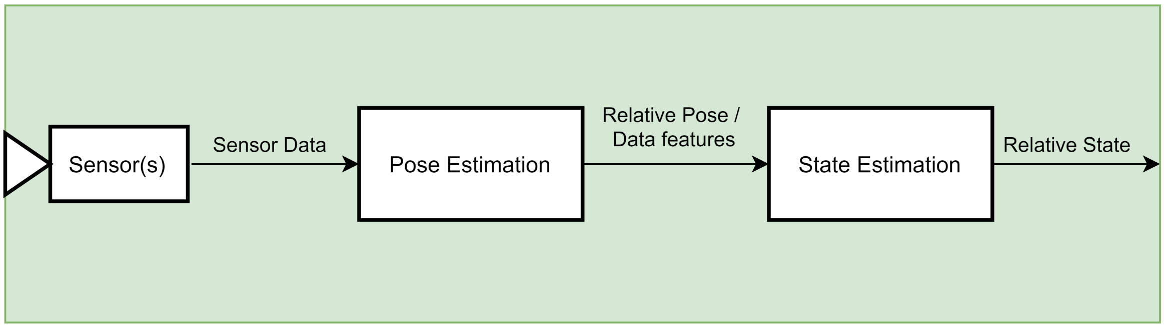


Figure 2.4: High-level overview of a generic VBN system

2.3. Monocular Cameras for Relative Navigation

Monocular cameras belong to the class of electro-optical sensors (together with LIDAR sensors and stereo cameras) that can be used for full relative pose estimation. The taxonomy of EO sensors for spacecraft applications is illustrated in Figure 2.5. In this section, monocular cameras that have been used in space are reviewed.

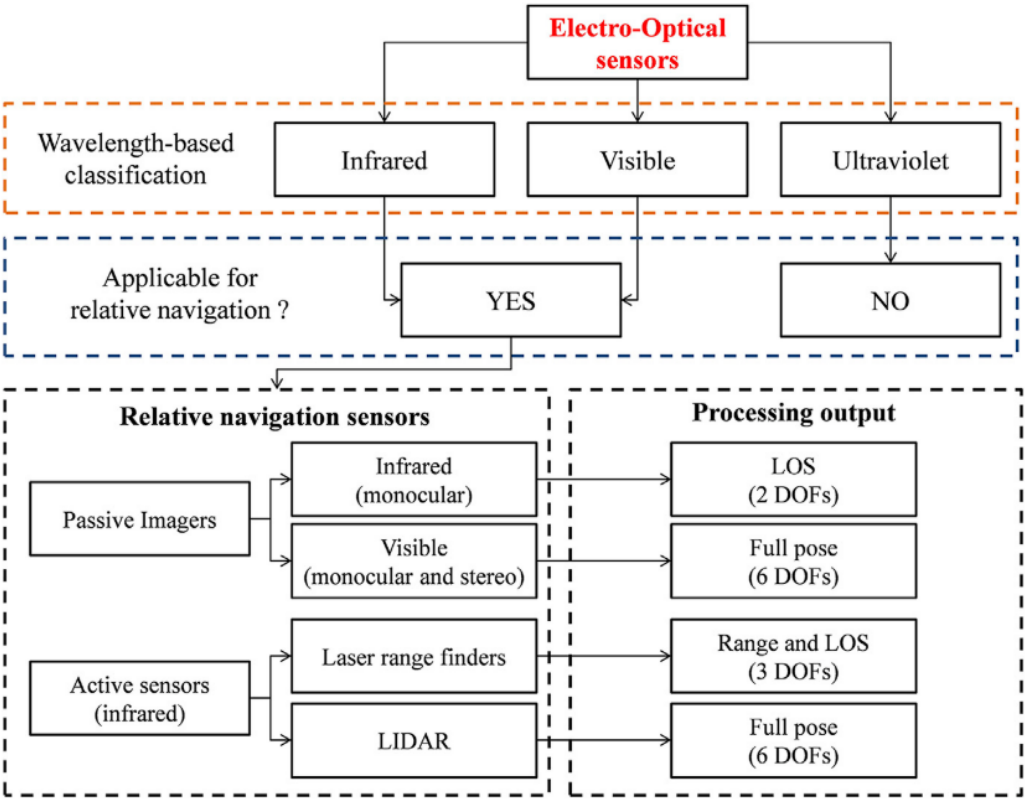


Figure 2.5: Overview of EO sensors for relative navigation [53]

Monocular cameras are generally passive, but can be made active by including light sources to illuminate the target under poor illumination conditions. This would increase the power budget, but that would typically remain below the power budget of LIDARs and stereo cameras.

Over the last decades, there have been three various vision-based systems developed and tested in space which are able to determine a cooperative target's full pose. These systems had as purpose to demonstrate OOS and Formation Flying (FF) capabilities.

The first system was launched in 1999 as part of the 7th mission of the Engineering Test Satellite Program (ETS-VII), developed by the National Space Development Agency of Japan (NASDA) [44]. The target and chaser satellite were launched together as one satellite into a circular orbit at 550 km altitude. When the orbit was reached, the target satellite was released from the chaser until a distance of about 9 km was obtained between both satellites. Then, the chaser satellite started its rendezvous procedures after which docking was successfully achieved. In the docking phase, the Proximity Operation Sensor (PXS) was used as navigation sensor. The PXS estimated the relative position and attitude between the docking frames of the chaser and target. The chaser contained a single visible camera and a LED array, pulsing visible light, at a 640-nm wavelength, within a cone of 30° in order to illuminate the target under poor visibility conditions. On the target's docking interface, a set of seven non-coplanar, passive, round shaped markers were installed. The PXS could measure the relative position from a distance of 9 m from the target, and the relative attitude when it was within 3 m. Just before docking, the attitude control system of the target was disabled. Then the chaser approached the target with 6-DOF control at a velocity of 0.01 m/s to align the reference frames of both docking systems.
Two types of optical interference were taken into account in the design of the PXS: sunlight interference and intersensor interference. Sunlight interference occurs when the sunlight is in direct view of the sensor, or when it is reflected via the target's surface. Intersensor interference happens when the emission of light of the sensor interfered with the other sensors on-board, or when the sensor confuses its own reflectors with others. The times the sunlight interferes with the sensor depends on the orbital position of ETS-VII. In its circular orbit (550 km altitude, 35 deg inclination), the angle between the orbit plane and direction of the sun (beta angle) changes over a period of approximately 50 days, as the J2 invariant rotates the orbital plane. Di-

rect sunlight interference occurs at the local sun time of around 18:00, and the beta angle becomes less than the Field Of View (FOV) of the navigation sensor. The reflected sunlight interference is more difficult to avoid, due to this type of interference occurring at various beta angles and local sun times. Hence, countermeasures were taken to avoid interference of reflected sunlight:

1. Use a bandpass filter in front of the optical sensor receiver, that allows light of the same wavelength as the optical navigation sensor to penetrate.
2. Eliminate the disturbance image by taking the LED on/off images and subtracting these images.

To avoid intersensor interference, the PXS used different wavelengths for their light emitting sensors on board.

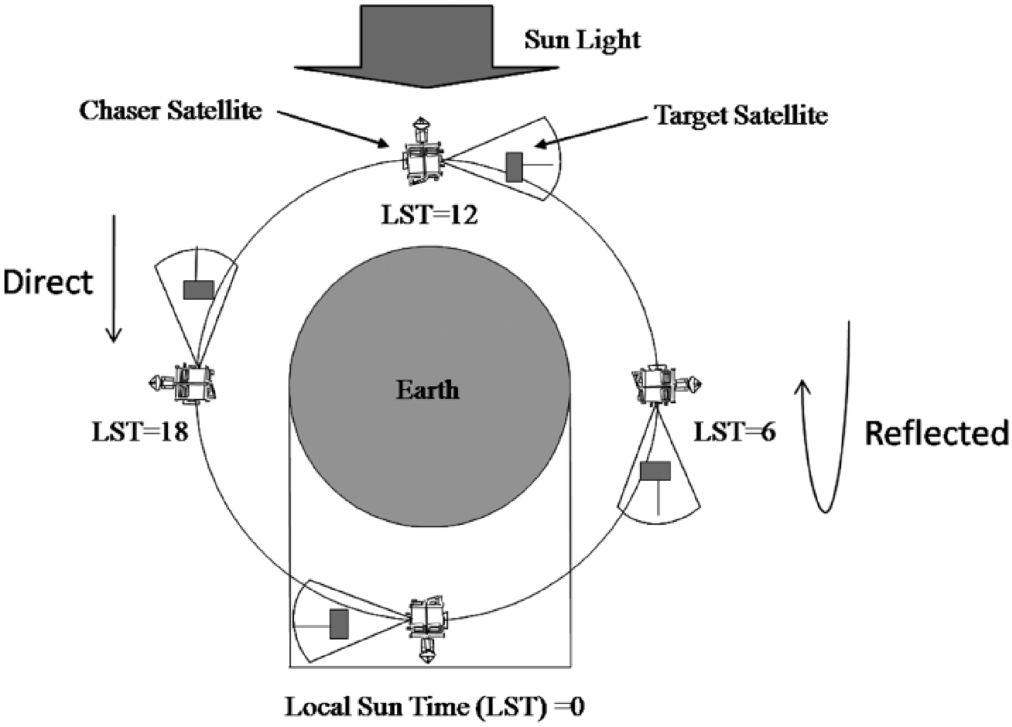


Figure 2.6: Sunlight interference and local sun time [44]

The second system, the Advanced Video Guidance Sensor (AVGS), was tested in 2005 during the Demonstration of Autonomous Rendezvous and Docking (DART) mission as one of the main relative navigation sensors, and in 2007 during the OE mission as the primary docking sensor. The sensor was developed by the Marshall Space Flight Center (MSFC).

Apart from the necessary software and hardware to process the sensor images, the AVGS comprises two sets of laser diodes at different wavelengths (800 and 850 nm) and a camera that captures the return from the laser [30]. This sensor is designed to interact with a retro-reflective target, similar to the PXS. The two sets of four CCRs on the target have filters that allow one wavelength to pass through to be reflected, while blocking the other wavelength. The sensor captures a first image by firing one set of laser diodes, and then a second image by firing the other laser diodes. Then, by subtracting the second image from the first and including an intensity threshold, all background noise is eliminated. The left-over data is transformed into a set of spots which are compared to the target pattern, after which the relative position and attitude is determined between the chaser and target.

DART is launched in a circular parking orbit at 500 km. Afterwards, it will perform several manoeuvres to place it 1 km behind its target. From that point, it will execute two sets of -V Bar and +R Bar manoeuvres to evaluate AVGS performance and its proximity operations algorithms [63]. The -V Bar manoeuvres include two approaches to 5 meters on the target vehicle docking axis, a simulated collision avoidance manoeuvre, and an evaluation of the maximum tracking range of the AVGS. The target vehicle of DART is MUBLCOM, an

experimental communications satellite. It contains retroreflectors for use with the AVGS. During the mission, the target satellite is 3-axis stabilised in a fixed yaw position and its X-axis aligned with the velocity vector. This way the retroreflectors can be observed from the -X axis, which means the chaser can approach from behind the target.
The DART mission was not fully successful [50]. During the manoeuvres designed to evaluate AVGS performance, DART transitioned its navigational data source from Global Positioning System (GPS) to AVGS as planned. During this transition, DART overshot a waypoint that would have triggered the final transition to full AVGS capability. Due to this, the range measurements supplied by the AVGS were not accurate. As a consequence, the chaser was able to steer towards its target, but eventually crashed into it as it was not able to accurately determine the relative distance.
In the other mission (OE mission), the AVGS was used as primary docking sensor between ASTRO, chaser satellite developed by Boeing, and NEXTSat, the target satellite provided by Ball Aerospace [25]. Both satellites were inserted in a circular orbit at 492 km as a physically linked pair. In orbit, the satellites undocked and redocked several times. First to a maximum distance of 10 m, which was successfully performed, after which a 30 m separation was planned. This operation experienced a failure of the primary sensor computer, and both satellites ended up separately at a distance of 6 km. Eventually, the operation was recovered by sending new commands to the chaser satellite, and the AVGS successfully navigated and docked the chaser with its target. The next scenario was a fly-around of the NEXTSat at a range of 120 m and redocking. This was successfully performed and marked the first time that a fly-around of another spacecraft was performed using primarily passive sensors, without any active exchange of relative navigation information and without intervention or control from the ground station. The following scenario let the chaser satellite departure to 4km and return autonomously to redock. This has also been successfully achieved.

More recently, in 2010, the PRISMA in-orbit testbed was launched, which made use of the Visual Based System (VBS) developed by Technical University of Denmark (DTU). This system was one of the relative navigation technologies for FF applications aboard PRISMA, and is used for several experiments: Homing and rendezvous (relative distance 10 m to 100 km), proximity operations (5 to 100 m), and final approach and recede (0 to 5 m) [56]. VBS consists of two cameras on the chaser and a set of active (LED) markers on the target. The far-range camera is used for LOS-only navigation for a relative distance of 100 m up to 100 km. The short-range camera is able to carry out pose determination by imaging the markers on the target, the workable distance goes from tens of meters to a few centimetres. The visibility of the LEDs is enhanced by a band pass filter in front of the short-range camera. During this mission, the VBS performance was successful.

The characteristics of the above discussed systems are summarised in Table 2.2. It shows that monocular-based sensors operating in the visible band are suitable for pose determination of cooperative targets in space in a sufficiently wide operational range.

Table 2.2: Monocular systems for cooperative pose estimation [53]

System (year)	Mission (Developer)	Operational Range	FOV	Data rate	Weight	Power consumption
PXS (1997)	ETS-VII (NASDA)	below 3 m	16° x 12° (conic)	2 Hz	-	-
AVGS (2007)	OE (MSFC)	below 150 m	16° x 16° (conic)	5 Hz	9 kg	14 W stand-by 30 W tracking
VBS (2010)	PRISMA (DTU)	below 50 m	22° x 16° (rectangular)	1 Hz	0.425 kg	3 W

With regard to uncooperative spacecraft, VBN rely exclusively on sensors installed on the chaser spacecraft. The VBS camera described above, can also be used towards a known uncooperative target. Several uncooperative pose determination algorithms have been tested on the real images collected during the PRISMA mission [19]. Another sensor, the Relative Navigation Sensor (RNS), which consists of three monocular cameras operating at different ranges, was designed to perform 6-DOF pose estimation of the Hubble telescope by means of vision-based algorithms only. It has been tested as an independent experiment on the space shuttle mission STS-125 [49]. Another camera has been tested in a simulation environment in a combination with uncooperative pose estimation algorithms [42]. The specifications of these cameras are listed in Table 2.3

Table 2.3: Examples of monocular cameras for uncooperative spacecraft pose estimation [53]

Monocular sensor	Pixel size (μm)		Pixel nr.		FOV (°)		IFOV (°)		Focal length (mm)
	hor.	ver.	hor.	ver.	hor.	ver.	hor.	ver.	
VBS camera	8.6	8.3	752	582	22.3	16.8	0.03	0.03	16.4
RNS3 camera	6.4	6.4	1024	1024	23	23	0.02	0.02	16
Simulated camera[42]	7.4	7.4	640	480	49.1	37.8	0.08	0.08	5.2

2.4. Monocular-based Navigation

Navigation systems comprise a pose estimator and a state estimator. Both aspects will be discussed in this section.

2.4.1. Pose Estimation

Pose determination is the estimation of the relative position and attitude of a Target Reference Frame (TRF) with respect to a Sensor Reference Frame (SRF). Pose estimators can be classified as model-dependent and model-agnostic architectures, depending on whether the 3D model of the target is known. Model-dependent pose estimation requires knowledge of the target's shape, while model-agnostic does not need any a-priori information. An overview of the various architectures discussed below is illustrated in Figure 2.7.

Learning-based Pose Estimation

Learning-based pose estimation has recently obtained great popularity due to the introduction of Convolutional Neural Network (CNN)s. In general, learning based pose estimation is centred around a CNN model, which is trained for predicting outputs of a specific type, like keypoints or parameterised pose [4]. The CNN models produce a complex non-linear mapping between input images and the output information. This mapping is created and fine-tuned by training the model with a database of representative images, in this case of the target spacecraft or object. This training data is not available at the scale required for deep learning due to the lack of representative spacecraft images. Hence, such models rely on synthetic images of the target spacecraft.

Model-dependent Feature-based Pose Estimation

This architecture is based on extracting features from the 2D image and subsequently matching the features with the ones in the 3D model of the target. These features can, for instance, be corners or edges. Once the features are matched, they are inserted into a pose solver to obtain the relative pose. Consequently, this architecture comprises two tasks: feature detection and matching, and pose solving.

Image Processing (IP)-based feature detection is the conventional method to extract features from the image. The IP subsystems enhance or transform the initial image data and pixel information in order to expose features, like corners, edges and keypoints which can then be extracted. The extracted features are matched with known features in the 3D of the spacecraft, resulting in a matching matrix between features in the 2D input image and the 3D model. This is subsequently solved using a PnP or pose solver. Feature correspondence algorithms like RANSAC, Soft assign and feature groups, generally test multiple correspondence hypotheses when the pose is being initialised from a lost-in-space state. The computational complexity of this makes them very slow. Another challenge of IP systems is related to the quality of real space images, which are characterised by adverse illumination, high contrast and high image noise.

Model-dependent End-to-end Pose Estimation

Feature-based pose estimation described here fore consists of two distinct subsystems: feature extraction and pose solving. End-to-end pose estimation –in contrast to feature-based systems– is generally represented as one block whose input is a 2D image and output is the relative pose. This is not the case for feature-based systems, which interact with the input data at different stages of the process. If the learning-based methods are disregarded, the realisation of end-to-end pose estimation is solely obtained by exploiting Template Matching (TM).

Template matching is based on matching the input image with a template, stored in a database. This database is generated offline by sampling the six-dimensional pose space of the 3D model. The input image is then matched by finding correlations between the image and the stored templates by exploiting correlation techniques, for instance the sum of absolute difference or normalised cross-correlation. The estimated pose is hence based on the template that has the highest degree of similarity. The main drawbacks come from memory and computation inefficiency, by storing and cross-correlating high resolution pose samples.

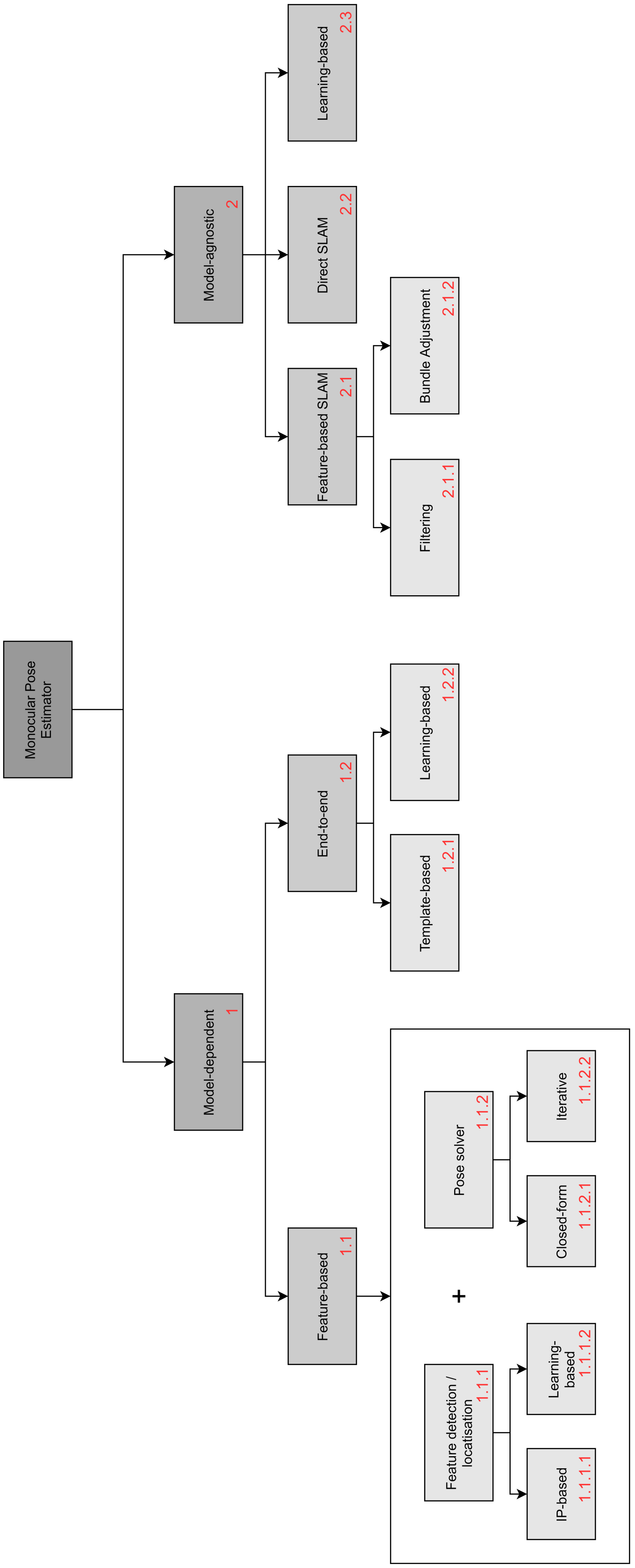


Figure 2.7: Pose estimation architectures overview

Model-agnostic Pose Estimation

Model agnostic pose estimation is typically performed by exploiting SLAM algorithms. SLAM is the real-time process of an object moving through an unknown environment and simultaneously computing its trajectory (localisation) and building a map from the environment (mapping). In most cases, the environment is static. This method can also be applied in a non-static environment when it is composed of rigid objects. Hence, this can be adopted to autonomous space applications involving a spacecraft flying in close-proximity of an unknown target.

2.4.2. State Estimation

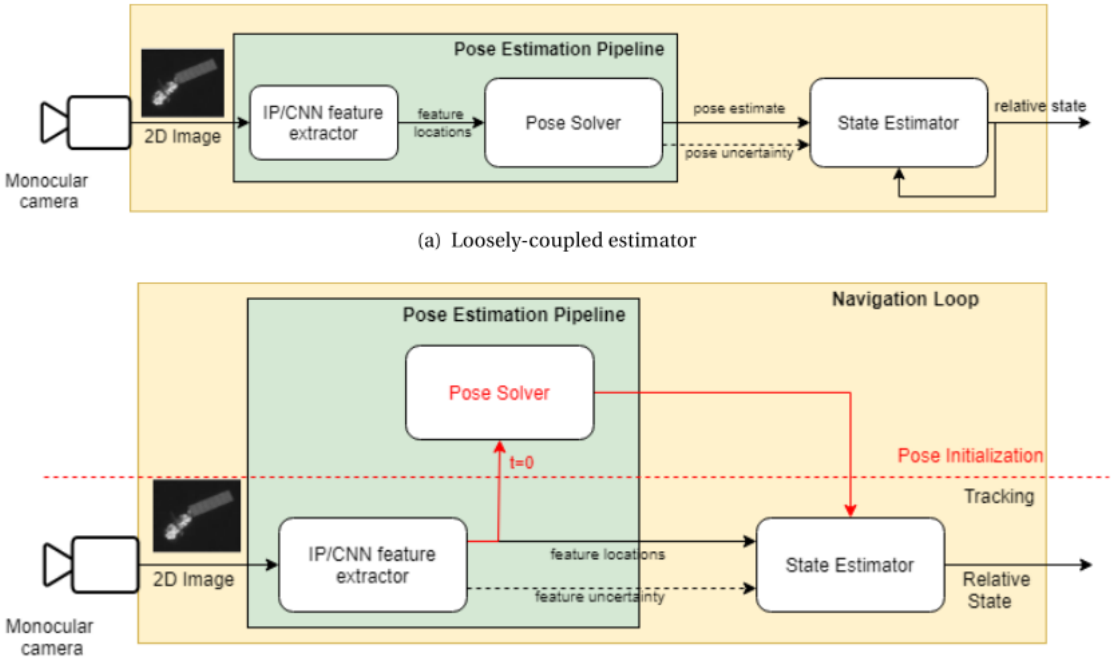
A state estimator is included in the VBN system to track the state of the target with respect to the chaser frame. It uses the pose estimation input and/or direct input from the vision subsystem. It combines knowledge of a dynamics model and the measurement data to determine a target's state, which typically is the relative position and velocity as well as the relative attitude and angular velocity. The state estimator is also used to provide the GNC loop with a state estimate at the desired frequency, which could be higher than the operating frequency of the pose estimation pipeline. The state estimation architectures can be divided into two groups: loosely and tightly coupled types. Flow diagrams of the two types are visualised in Figure 2.8.

Loosely-coupled:

If loosely-coupled state estimators are implemented, the VBN system blocks are linked in series in a navigation loop. The state estimator takes only the estimated pose from the pose estimation pipeline, and does not need additional input like the location of features etc. This is shown in Figure 2.8a [4].

Tightly-coupled

Tightly-coupled estimators use the pose solver only during pose initialisation in order to initialise the state filter. Once pose initialisation is achieved, the extracted features are directly inserted into the state estimator, bypassing the pose solver. This is illustrated in Figure 2.8b. The computational complexity of loosely-coupled state estimators is lower, since they only take 6 variables as input (3D position and attitude), compared to tightly-coupled estimators which require a lot more input data.



(a) Loosely-coupled estimator

(b) Tightly-coupled estimator

Figure 2.8: Vision-based navigation system flow for various estimator architectures [4]

2.5. Monocular-based Pose Estimation of Unknown Uncooperative Targets

Pose estimation where no apriori knowledge of the target is required is regarded as the ideal method to be implemented in ADR missions and asteroid exploration. However, this is also the most challenging type of pose estimation. Monocular based SLAM for pose estimation towards unknown targets has not yet been tested in space. Several novel architectures have been simulated or experimentally tested on ground. They are listed below.

Pose Estimation Approaches
In 2018, a study was performed in the application of ORB-SLAM –a feature-based SLAM– to spacecraft relative navigation [20]. The SLAM algorithm exploits ORB features, which are image features invariant to rotation, based on BRIEF binary descriptors. They have as advantage that they can be rapidly extracted, significantly decreasing the computational time. The algorithm incorporates robustness by initialising a great amount of keypoints, and discarding the weaker ones. ORB-SLAM also includes a bag-of-words method for place recognition, automating the loop closure step. So when a recognisable scene is viewed for a second time after many image frames have passed, it executes a full bundle adjustment. This reduces the error in the estimation by eliminating the accumulated drift between successive keyframes.
It is important to note that monocular-based pose estimation gives a scale-ambiguous pose when only a monocular camera is used. Also, in this study they assumed all detected features to be part of the rigid body, meaning they have a fixed position in the target frame. This has not been tested on images with the Earth in the background, for example. Hence its robustness to such images should be further tested. Also, it was observed that the algorithm had difficulties initialising the map when the target is far away, due to the relative small translational motion between frames. So for successful initialisation, large translational motion would benefit the initialisation process, since it induces a parallax in the image plan. Additionally, loop closure of the algorithm could not be achieved repeatedly in its experiment, where the chaser completed several revolutions around the target.

A direct-SLAM is a SLAM method that does not rely on features, but directly uses the intensity of the pixels for pose estimation. One of these methods is LSD-SLAM, which is also studied for relative navigation towards unknown uncooperative targets in 2019 [39]. The method is however only tested on an experiment at which the relative motion only had one degree of freedom (one rotation). It was found to be only accurate on the short-term and close range motion. Also for this algorithm, the loop closer could not be tested. Additionally, this has been simulated with just a plain black background only.

A filter-based feature SLAM for monocular pose estimation of unknown targets is investigated by Capuano et al. [13]. This algorithm, called Simultaneous Estimation of Pose and Shape (SEPS), is monocular based, but assumes a single beam LIDAR measurement to recover the scale of the restructured map, as this is not possible for solely monocular vision. SEPS architecture is shown in Figure 2.9.

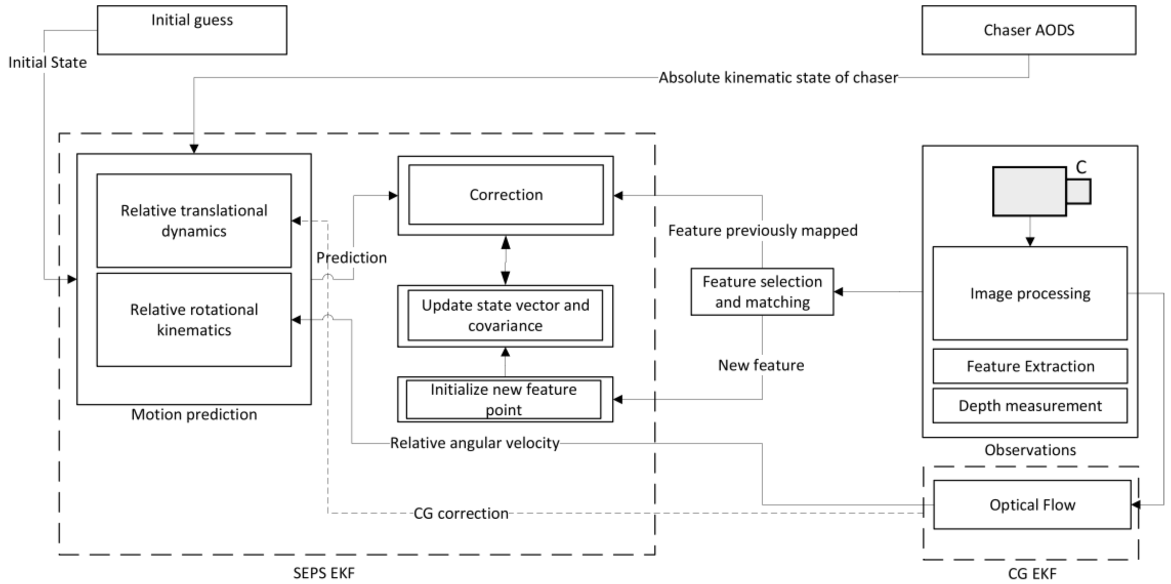

Initial guess
Initial State
Chaser AODS
Absolute kinematic state of chaser
Relative translational dynamics
Relative rotational kinematics
Motion prediction
Prediction
Correction
Update state vector and covariance
Initialize new feature point
Feature previously mapped
Feature selection and matching
New feature
C
Image processing
Feature Extraction
Depth measurement
Observations
Relative angular velocity
CG correction
Optical Flow
SEPS EKF
CG EKF

Figure 2.9: SEPS architecture

In SEPS, target features are extracted by means of a monocular camera, and the Harris corner detector algorithm. However, this paper only analyses the pose determination algorithm on synthetic images. The feature detection algorithm may not be effective for actual spacecraft images due to their high contrast, low resolution, and low signal-to-noise ratio.
The angular velocity is not included in the state vector, but is directly obtained from optical flow. As a result, the moment of inertia estimation is not needed in the state propagation, which decreases the uncertainty of the filter estimate. Also, it lowers the computational burden, due to the size reduction of the filter state.
Contrary to previous implementations of optical flow, which assume either a fixed camera or a fixed world, [13] break up the optical flow measurement into different inertial components that allow to separate the own motion of the camera from the rotation and relative velocity of the target.

Another filter-based algorithm is proposed by Jin et al. [33]. This algorithm decomposes the target state estimation into relative pose estimation and feature point position estimation based on relative pose. The relative pose of the target is estimated by Extended Kalman Particle Filter (EKPF), and Extended Kalman Filter (EKF) estimates the feature point position of the target. This algorithm is also based on a monocular camera, and a single-beam LIDAR measurement to obtain depth information. This algorithm is only tested on synthetic target images, generated by OpenGL. The relative pose consists of the attitude, position, angular velocity and velocity.

A filter-based algorithm is also described in [32]. This algorithm had some restrictions. It can only recognise spacecraft if the image is not saturated by sunlight and if the background is dark. Furthermore, temporary occlusion of reference features can occur during motion, which affects the accuracy. The provided algorithm has not been tested thoroughly, as it has only been able to track for 3 seconds.

An EKF-SLAM approach has been discussed in [67]. This architecture analyses two cases: one where the target has a constant rotational velocity around one principal axis, and one where the target motion is nutational, but the Moment of Inertia (MOI) distribution is known. The tests demonstrated the working principle on synthetic images, however no verification has been performed on real spacecraft images.

2.6. Image Feature Detection, Description, and Matching

The process of feature detection, description, and matching plays a significant role in many SLAM algorithms. Therefore, it is necessary to obtain a good understanding of the working principle of this process. Firstly, the definition of features is described, after which each step in the process is discussed.

2.6.1. Features

Fundamentally, a feature is a piece of information which is relevant for solving the computational task related to a specific application. With respect to images and computer vision, these features represent and define an image. Features can be categorised in global and local features.

Global features
When images are represented by global features, the image is described by a single multi-dimensional vector, describing the information of the whole image. This vector contains values that relate to information in the whole image, like colour, texture, or shape. In practice, two images are matched by comparing their feature vector. For illustration, when one wants to compare an image of the sea (blue), and an image of a forest (green), the colour element of the feature vector of both images would be significantly different. Hence, global features can be interpreted as a particular property of an image involving all pixels. This property can be colour, histograms, texture, edges etc. The advantage of global features is that they are much faster to process than local features. They are easy to compute and require small amounts of memory [3]. However, they do suffer from transformation variance, and are sensitive to clutter and occlusion.

Local features
On the other hand, the main goal of local feature representation is to distinctively represent the image based on some salient regions while remaining invariant to viewpoint and illumination changes. This means that, in contrast to global features, the image is represented based on its local structures by a set of local feature descriptors extracted from a set of interest regions in the image. Due to the fact that local structures are more distinctive and stable, these features are more suitable for image matching and object recognition. Since images generally have hundreds of local features, they do require more computational time than global features. An ideal feature detection algorithm relying on local features has the following properties [3]:

- Robustness, it should be able to detect the same feature locations independent of scaling, rotation, shifting, photo-metric deformations, compression artefacts, and noise.
- Repeatability, the algorithm should be able to detect the same features of the same scene or object repeatedly under a variety of viewing conditions.
- Accuracy, it should accurately localise the image features (same pixel locations), especially for image matching tasks, where precise correspondences are needed to estimate the epipolar geometry.
- Generality, the feature detection algorithm should be able to detect features that can be used in different applications.
- Efficiency, the feature detection algorithm should be able to detect features in new images quickly to support real-time applications.
- Quantity, the feature detection algorithm should be able to detect all or most of the features in the image. Where the density of detected features should reflect the information content of the image for providing a compact image representation.

The general difference between both types of features is represented in Figure 2.10. The type of features to use greatly depends on the application.

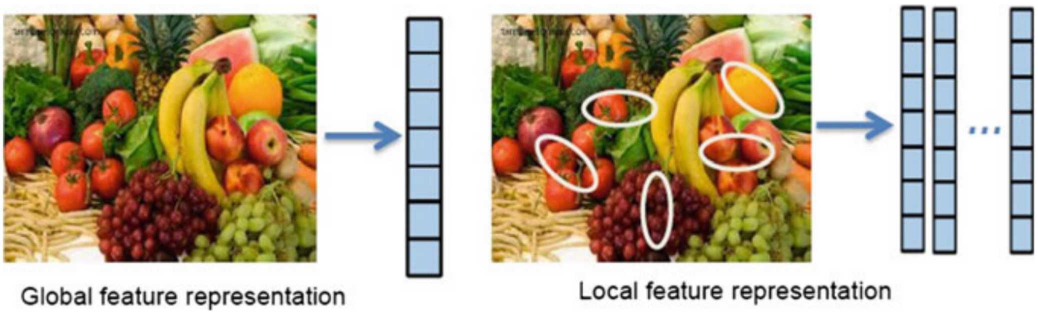


Figure 2.10: Global and local image features representation [3]

2.6.2. Feature Detection

For the application of SLAM to spacecraft relative navigation, local feature detection has the advantage. Therefore, this will be the focus of the upcoming sections. Feature detectors can be classified into three categories: single-scale, multi-scale, and affine invariant detectors.

Single-scale Detectors
Single-scale detectors only use one representation for the features or the object contours using the detector's internal parameters. They are invariant to rotations and translations, changing illumination, and noise. They however cannot cope with the scaling problem or affine transformations. Examples of single scale detectors are:

- Moravec's Detector
- Harris Detector
- SUSAN Detector
- FAST Detector
- Hessian Detector

Multi-scale Detectors
Multi-scale detectors are single-scale detectors with the ability to detect features invariant to scale. The most notable algorithms are:

- Laplacian of Gaussian (LoG)

- Difference of Gaussian (DoG)
- Harris-Laplace
- Hessian-Laplace
- Gabor-Wavelet detector

Affine Invariant Detectors
The feature detectors discussed above exhibit invariance to translations, rotations, and uniform scaling; assuming that the localisation and scale are not affected by an affine transformation of the local image structures. Thus, they partially handle the challenging problem of affine invariance, keeping in mind that the scale can be different in each direction rather than uniform scaling. That in turn makes the scale invariant detectors fail in the case of significant affine transformations. Therefore, building a detector robust to perspective transformations necessitates invariance to affine transformations. An affine invariant detector can be seen as a generalised version of a scale invariant detector. Some feature detectors have been extended to include affine transformation invariance:

- Harris-Laplace detector
- Mikolajczyk and Schmid

2.6.3. Feature Description

Once a set of interest points has been detected from an image at a location $p(x, y)$, scale s, and orientation θ, their content or image structure in the neighbourhood of p needs to be encoded in a suitable descriptor that describes the feature in a unique and efficient way. Apart from the image structure around p, the descriptor should also encode the orientation and scale. There are a large number of image feature descriptors, the most notable ones are listed:

- Scale Invariant Feature Transform (SIFT)
- Speeded Up Robust Features (SURF)
- Oriented FAST and Rotated BRIEF (ORB)
- Accelerated-KAZE (AKAZE)
- Binary Robust Invariant Scalable Keypoints (BRISK)

As stated, there are many more, each with its own benefits, drawbacks and suitability for certain applications. An appropriate trade-off should be made when choosing the type of descriptor.

2.6.4. Feature Matching

Feature matching is the task of establishing correspondences between two images of the same scene/object. A common approach to image matching consists of detecting a set of interest points, each associated with image descriptors from image data. Once the features and their descriptors have been extracted from two or more images, the next step is to establish some preliminary feature matches between these images. This is illustrated in Figure 2.11.

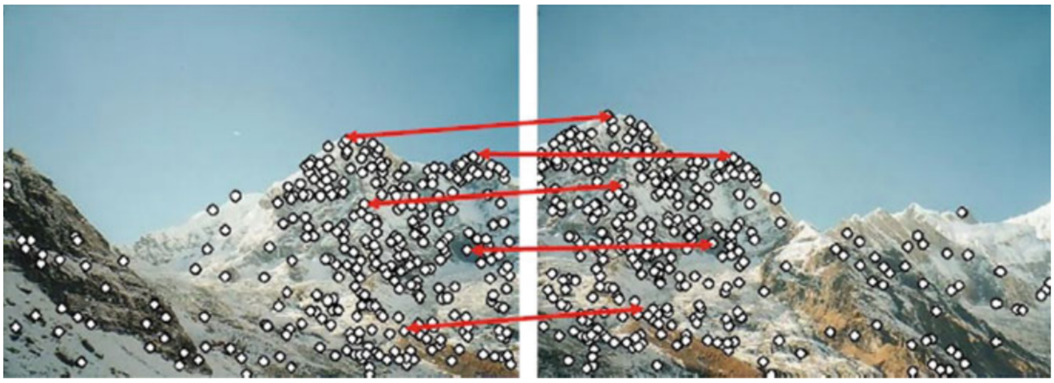


Figure 2.11: Matching image regions based on their local feature descriptors [3]

Without losing the generality, the problem of image matching can be formulated as follows [3], suppose that p is a point detected by a detector in an image associated with a descriptor:

$$\Phi(p) = \{\phi_k(P) | k = 1, 2, ..., K\} \tag{2.1}$$

where, for all K, the feature vector provided by the k-th descriptor is:

$$\phi_k(p) = (f_{1p}^k, f_{2p}^k, ..., f_{np}^k) \tag{2.2}$$

The aim is to find the best correspondence q in another image from the set of N interest points $Q = \{q_1, q_2, ..., q_N\}$ by comparing the feature vector $\phi_k(p)$ with those of the points in the set Q. To this end, a distance measure between the two interest points descriptors $\phi_k(p)$ and $\phi_k(q)$ can be defined as:

$$d_k(p,q) = |\phi_k(p) - \phi_k(q)| \tag{2.3}$$

Based on the distance d_k, the points of Q are sorted in ascending order independently for each descriptor creating the sets:

$$\Psi(p,Q) = \{\psi_k(p,Q) | k = 1, 2, ..., k\} \tag{2.4}$$

Such that:

$$\psi_k(p,Q) = \{(\psi_k^1, \psi_k^2, ..., \psi_k^N) \in Q | d_k(p, \psi_k^i) \leq d_k(p, \psi_k^j), \forall i \geq j\} \tag{2.5}$$

A match between the pair of interest points (p,q) is accepted only if:

1. p is the best match for q in relation to all the other points in the first image
2. q is the best match for p in relation to all the other points in the second image.

In the context of real-time applications, it is important to devise an efficient algorithm to perform this matching process as quickly as possible. The nearest-neighbour matching in the feature space of the image descriptors in Euclidean norm can be used for matching vector-based features. In practice, the optimal feature matching algorithm and its parameters depend on the dataset characteristics. Furthermore, to eliminate matching candidates for which the correspondence may be regarded as ambiguous, the ratio between the distances to the nearest and the next nearest image descriptor is required to be less than a specified threshold. Binary feature description is one of the most computational efficient methods to describe features. Binary features are compared using the Hamming distance calculated by performing a bitwise XOR (Exclusive OR) operation followed by a bit count on the result. This involves only bit manipulation operations that can be performed quickly. The typical solution in the case of matching large datasets is to replace the linear search with an approximate matching algorithm that can offer speedups of several orders of magnitude compared to the linear search. This is at the cost that some of the nearest neighbours returned are approximate neighbours, but usually close in distance to the exact neighbours.

2.7. Relative Navigation Facilities

This section provides an overview of several existing facilities where relative navigation operations are simulated and tested. The purpose of this overview is to get a general indication on how their lay-out is and what components they use.

- **ASTROS**
  The ASTROS is an experimental facility for research in space proximity operations [72]. It is a facility of the Georgia Institute of Technology. It is designed and built to fulfil the ability to test new relative navigation and guidance algorithms, grasping algorithms, vision processing and perception algorithms that can support proximity operations in orbit. The facility is shown in Figure 2.12

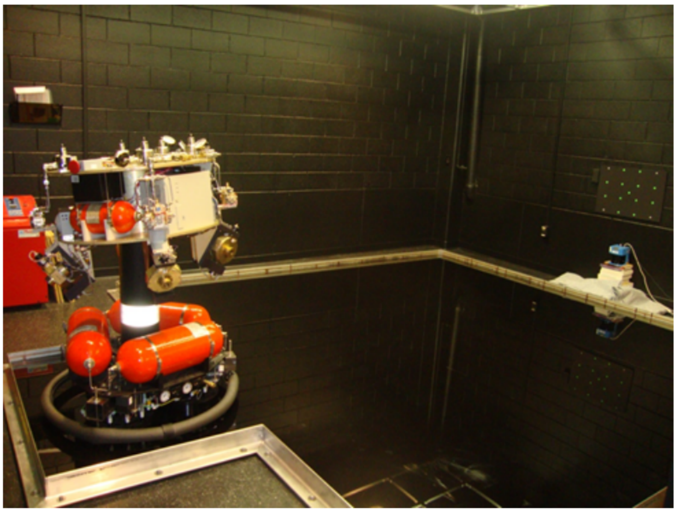


Figure 2.12: ASTROS facility [72]

- **TRON**
  The TRON facility is at the Space Rendezvous Laboratory of Stanford university [55]. It consists of two 6-DOF robot arms and a set of Vicon motion track cameras to reconfigure an arbitrary relative pose between a camera and a target mock-up model. The facility is shown in Figure 2.13.

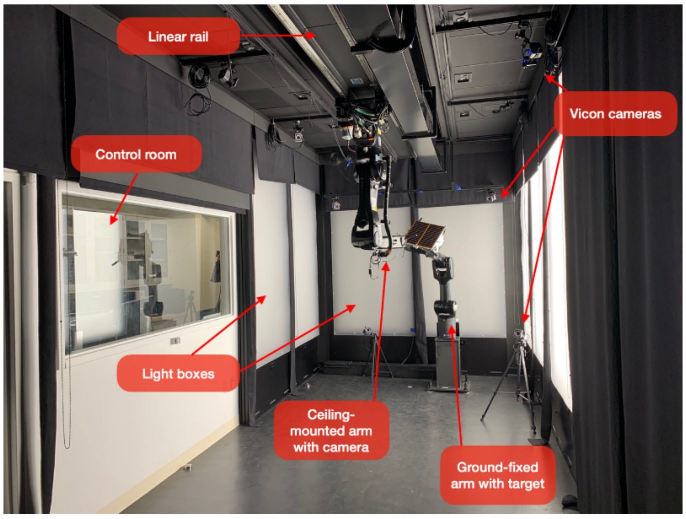


Figure 2.13: TRON facility [55]

- **GRALS**
  The GRALS testbed is a testbed by ESA for spacecraft proximity operation tests. The system is shown in Figure 2.14.

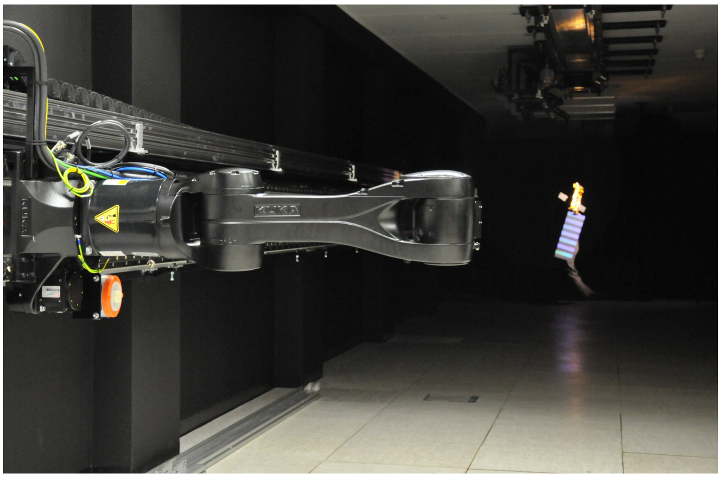


Figure 2.14: GRALS facility

2.8. Conclusions

This chapter provided background knowledge to support the follow-up work of this thesis. In this chapter an overview is given of typical missions that require VBN systems. Additionally, all aspects of monocular-based pose estimation of unknown targets are discussed. Furthermore, an introduction to image feature detection, description, and matching is provided, after which the various relative navigation facilities are outlined.

II

Research Framework Design

3
Framework Overview

In this chapter, an overview is established of the technical research framework. Section 3.1 describes the architectural design of the framework. Then, the novelty of the framework is discussed in section 3.2. Subsequently, the method of accuracy determination is described in section 3.3. Finally, the chapter is concluded in section 3.4.

3.1. Architecture Design

In this work, a technical research framework is designed. This framework produces results from which conclusions can be drawn with respect to the research questions. The framework is illustrated in Figure 3.1.

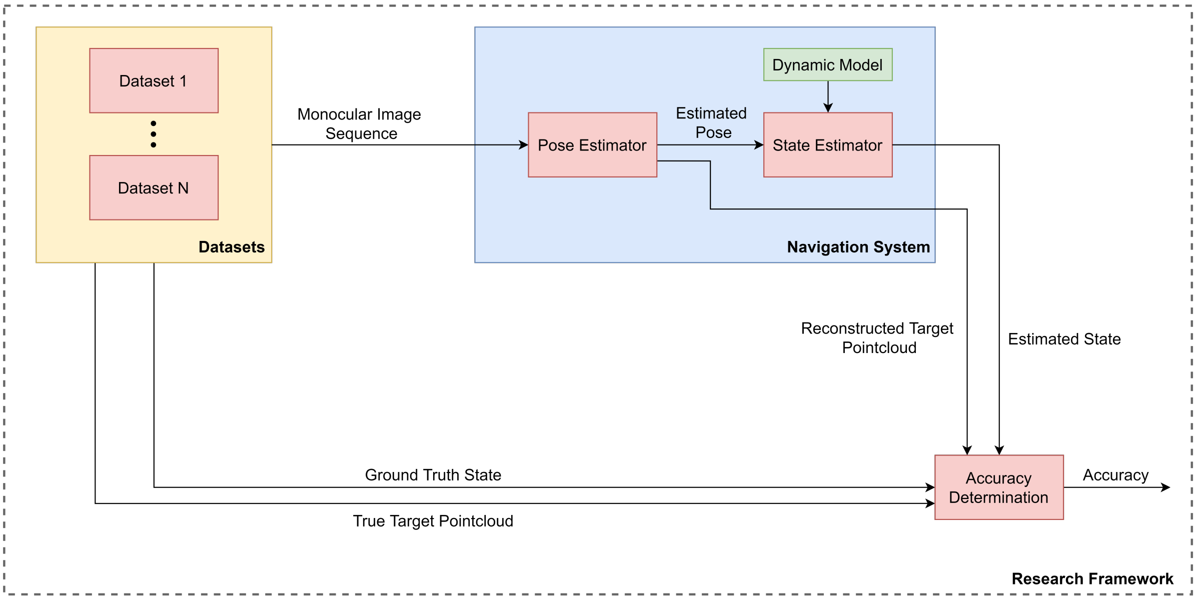


Figure 3.1: Technical research framework

The framework is based on the objective of the research as stated in section 1.3. In order to thoroughly assess the suitability of SLAM algorithms for spacecraft relative navigation, it is of great importance that the limitations of SLAM under orbital conditions become clear. Therefore, a significant variety of input data should be obtained or generated. For monocular-based SLAM, the input data consists of a sequence of monocular images that capture a target in relative motion to the camera frame. For the purpose of researching the accuracy, robustness and suitability of the to-be-designed navigation system, the ground-truth values of each frame and the target 3D model are required. The suitability of the designed navigation system is determined based on its accuracy and robustness performance under various realistic orbital scenarios, and by using similar hardware that is currently exploited in spacecraft. Since there is a general unavailability of real space imagery capturing relevant close-proximity operations including ground-truth data, it has been determined that synthetically generated data is required in order to perform a valid assessment of the suitability of SLAM for spacecraft relative navigation. However, as synthetically generated data is never as representative as real data, and as it is difficult to simulate artefacts present in real imagery, it is also desired to obtain lab-generated data. Hence, both types of datasets are included in the research framework.

The generated datasets are used as input in a navigation system. The navigation system consists of the pose estimator and the state estimator. As the objective of the thesis is to assess the pose estimator, it has been decided that a loosely-coupled approach (an approach where the state estimator only uses input from the pose estimator and not directly from the monocular image sequence) is used in the navigation system. The state estimator is simply designed for proof of integration of both pose and state estimator. Typically, (Kalman) filters are used to provide state estimates in spacecraft. Hence, it has been determined that the state estimator will be a simple EKF that makes use of a dynamic model to propagate the state, and position and attitude measurements to update the filter parameters. In mission mode, this navigation system should output the filtered estimated relative state at GNC loop frequency. However, in this case, this system will also output the raw pose measurements for evaluation and assessment of the pose estimator.

The navigation system does not have any prior knowledge of the target. Hence, it will create a target frame that does not coincide with the actual centre of gravity and basis vector definitions of the actual target. So, a transformation between the actual target frame and generated target frame will have to be obtained to assess the accuracy. The actual target frame is the target frame from which the ground-truth values are created. The generated target frame by the SLAM algorithm is the (pseudo-)randomly initialised target frame by the algorithm. Furthermore, monocular-only SLAM will provide a scale ambiguous pose estimate, since the actual scale of the environment cannot be extracted from single monocular images. The scaling parameter will also have to be determined in order to assess the accuracy of the pose estimation system. For the purpose of this research, it is assumed that the scale is perfectly known by the navigation system and the scale recovery method will not be included and assessed in the evaluation of the pose estimator performance. The main reason for this is that a different secondary sensor will be required, and the integration and data fusion of this sensor with the monocular-only pose estimator is outside the scope of the thesis. However, as a scale-ambiguous navigation system is not of much use, various scale recovery methods are further discussed in chapter 4.

For each test scenario, the navigation system will be evaluated on two factors. The first one is the stability performance of the algorithm during the simulation. If unstable pose estimates are output at every time step, it means that the pose estimator reached its performance limits. Unstable pose estimates are defined as estimates which do not logically make sense relative to nominal performance, and which are significantly inaccurate. The second factor is the accuracy. The accuracy will be evaluated when the performance of the algorithm on the dataset is stable. Further clarifications and illustrations of stable versus unstable performance are provided in subsection 6.7.1.

3.2. Novelty of the Framework

For every research it is important to highlight the novelties and the contribution of the research to the respective field. Therefore, this research is compared to the current state-of-the-art of monocular pose estimation of unknown targets in space. This is visualised in Table 3.1.

Table 3.1: Comparison of the state-of-the-art researches

Reference	Robustness Tests			Dataset type	State Estimator Integration
	Background	Relative Motion	Hardware		
This work	**yes**	**yes**	**yes**	Lab-generated, Synthetic	**yes**
Dor and Tsiotras [20]	no	no	no	Lab-generated, Real	no
Lei et al. [39]	no	**yes**	no	Lab-generated, Synthetic	no
Capuano et al. [13]	no	no	no	Synthetic	**yes**
Jin et al. [33]	no	no	no	Synthetic	**yes**
Ivanov et al. [32]	no	no	no	Synthetic, Real	no

The main novelties include:

- robustness to background assessment
- robustness to various relative motion assessment
- robustness to target visibility
- hardware specifications analysis

The current state-of-the-art performs single scenario analysis, meaning they only measure the performance of the designed algorithm in a single case. This shows the proof of concept of the algorithm to be applied to spacecraft relative navigation. But for relative navigation of uncooperative targets, the state of the target is unknown. So it is essential that the algorithm works under various scenarios of relative motion. By the extent of the author's knowledge, this has only been tested in the work of Lei et al. [39]. In that work, the relative motion between the camera and the target had one degree of freedom (rotation of the target around its z-axis). They performed a robustness test by changing the angular rate of the target satellite around that axis. Robustness tests have not been performed in other work. In this research, robustness will be tested to variations in relative motion, to objects in the background, and to variations in hardware parameters to get a clear picture of what hardware would be required aboard chaser spacecraft. Additionally, so far only Capuano et al. [13] and Jin et al. [33] have been able to show achievement of integration with a state estimator. As only a pose estimator on-board of a satellite would not be of much use, it is imperative that the feasiblity of the designed pose estimation integration with the state estimator is demonstrated. This is particularly important since no prior knowledge of the target is known, meaning that target properties like moments of inertia are unknown, which could induce large uncertainties in the dynamic model of a state filter.

3.3. Accuracy Determination

The ground-truth values of the pose, are measured in a different frame of reference than the frame of reference that ORB-SLAM uses. Therefore, a transformation is required between both frames to measure the accuracy of the pose estimation algorithm. This section describes the method that is conducted to achieve this.

3.3.1. Pose Comparison

SLAM puts the origin of the map (target) frame at the location of the camera once the map is initialised. This means that if at t=0 the map gets initialised, the target frame and the camera frame exactly coincide. From each time step onward, the pose estimator describes the camera pose relative to this initial map frame. Since the camera frame, indicated by C, never coincides with the actual geometric target frame, from now on indicated by T, a transformation is required between the actual target frame T, and the SLAM-generated target frame N. The pointcloud/map of the target that is built in the SLAM algorithm, also originates from frame N. The ground truth data is measured in their respective frame of reference, T. To find the transformation $A^{N}_{T/N}$, the Coherent Point Drift (CPD) algorithm will be exploited. This algorithm compares two pointclouds and performs rigid registration to find the transformation matrix between the two pointclouds. Also, it gives an estimate of the scale difference between the point clouds which is necessary to obtain a scale-unambiguous pose estimate. In order to do this, the pointcloud in frame N is extracted from SLAM, and the pointcloud in frame T is generated from the CAD model.

Taking it from a global perspective, the objective of vision-based relative navigation is to find the rigid-body transformation $A^S_{T/S}$, where S is the body-fixed frame of the chaser/servicer satellite. In monocular-only mode, the pose estimator will produce a scale ambiguous version of the transformation $A^N_{C/N}$. If C' is the camera frame estimated by the pose estimator, then $t^{C'}_{N/C'}$ is the scale ambiguous position vector of the frame N with regards to the camera frame C' as expressed in frame C' coordinates. If the unknown scale is defined as λ, then it means that $t^C_{N/C} = \lambda t^{C'}_{N/C'}$. From this it follows that:

$$A^C_{N/C} = \begin{bmatrix} R_{N/C'} & \lambda t^{C'}_{N/C'} \\ 0 & 1 \end{bmatrix} \tag{3.1}$$

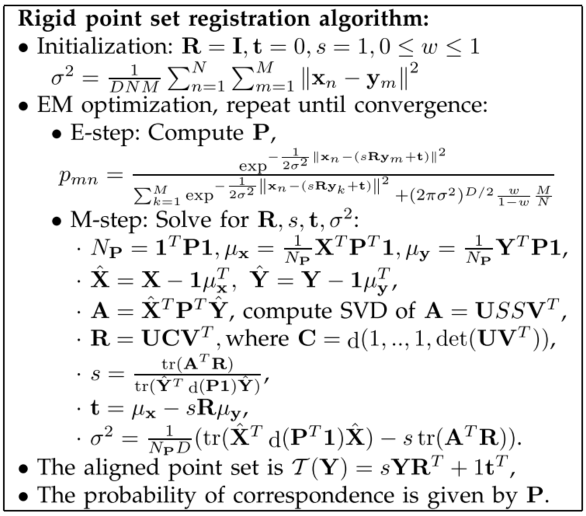
The above transformation matrix only holds when all detected features on the target frame are fixed with respect to the target body. Since the designed pose estimator discards background features, this assumption is valid. The desired transformation $A^S_{T/S}$ is then obtained by cascading transformations:

$$A^S_{T/S} = A^S_{C/S} A^C_{N/C} A^N_{T/N} \tag{3.2}$$

Where, to summarise, $A^S_{C/S}$ is known by the designers of the chaser spacecraft, $A^C_{N/C}$ is obtained from the pose estimator, and $A^N_{T/N}$ is obtained from the CPD algorithm by transforming the point cloud of the reconstructed target to that of the target 3D model.

3.3.2. Coherent Point Drift Algorithm

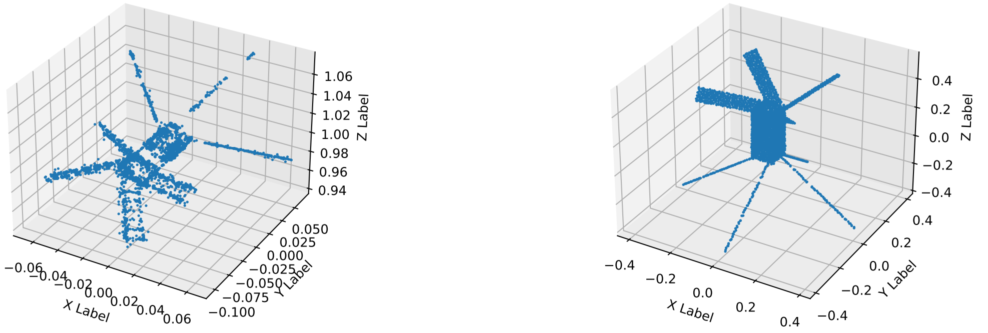
The CPD algorithm [47] is a point set registration method developed by Myronenko and Song. A point set registration problem assigns correspondences between two sets of points, and performs the transformation that maps one point set to the other. In the application of this work, the rigid registration method is used, as the rigid transformation of $A^N_{T/N}$ is desired. Mathematically, the algorithm is described in Figure 3.2.



```
Rigid point set registration algorithm:
• Initialization: $\mathbf{R} = \mathbf{I}, \mathbf{t} = 0, s = 1, 0 \leq w \leq 1$
  $\sigma^2 = \frac{1}{DNM} \sum_{n=1}^{N} \sum_{m=1}^{M} \|\mathbf{x}_n - \mathbf{y}_m\|^2$
• EM optimization, repeat until convergence:
    • E-step: Compute $\mathbf{P}$,
      $p_{mn} = \frac{\exp^{-\frac{1}{2\sigma^2}\|\mathbf{x}_n - (s\mathbf{R}\mathbf{y}_m + \mathbf{t})\|^2}}{\sum_{k=1}^{M} \exp^{-\frac{1}{2\sigma^2}\|\mathbf{x}_n - (s\mathbf{R}\mathbf{y}_k + \mathbf{t})\|^2} + (2\pi\sigma^2)^{D/2} \frac{w}{1-w} \frac{M}{N}}$
    • M-step: Solve for $\mathbf{R}, s, \mathbf{t}, \sigma^2$:
        · $N_{\mathbf{P}} = \mathbf{1}^T\mathbf{P}\mathbf{1}, \mu_{\mathbf{x}} = \frac{1}{N_{\mathbf{P}}}\mathbf{X}^T\mathbf{P}^T\mathbf{1}, \mu_{\mathbf{y}} = \frac{1}{N_{\mathbf{P}}}\mathbf{Y}^T\mathbf{P}\mathbf{1}$,
        · $\hat{\mathbf{X}} = \mathbf{X} - \mathbf{1}\mu_{\mathbf{x}}^T,\ \hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{1}\mu_{\mathbf{y}}^T$,
        · $\mathbf{A} = \hat{\mathbf{X}}^T\mathbf{P}^T\hat{\mathbf{Y}}$, compute SVD of $\mathbf{A} = \mathbf{U}SS\mathbf{V}^T$,
        · $\mathbf{R} = \mathbf{U}\mathbf{C}\mathbf{V}^T$, where $\mathbf{C} = \mathrm{d}(1, .., 1, \det(\mathbf{U}\mathbf{V}^T))$,
        · $s = \frac{\mathrm{tr}(\mathbf{A}^T\mathbf{R})}{\mathrm{tr}(\hat{\mathbf{Y}}^T \mathrm{d}(\mathbf{P}\mathbf{1})\hat{\mathbf{Y}})}$,
        · $\mathbf{t} = \mu_{\mathbf{x}} - s\mathbf{R}\mu_{\mathbf{y}}$,
        · $\sigma^2 = \frac{1}{N_{\mathbf{P}}D}(\mathrm{tr}(\hat{\mathbf{X}}^T \mathrm{d}(\mathbf{P}^T\mathbf{1})\hat{\mathbf{X}}) - s\,\mathrm{tr}(\mathbf{A}^T\mathbf{R}))$.
• The aligned point set is $\mathcal{T}(\mathbf{Y}) = s\mathbf{Y}\mathbf{R}^T + 1\mathbf{t}^T$,
• The probability of correspondence is given by $\mathbf{P}$.
```

Figure 3.2: Rigid point set registration algorithm [47]

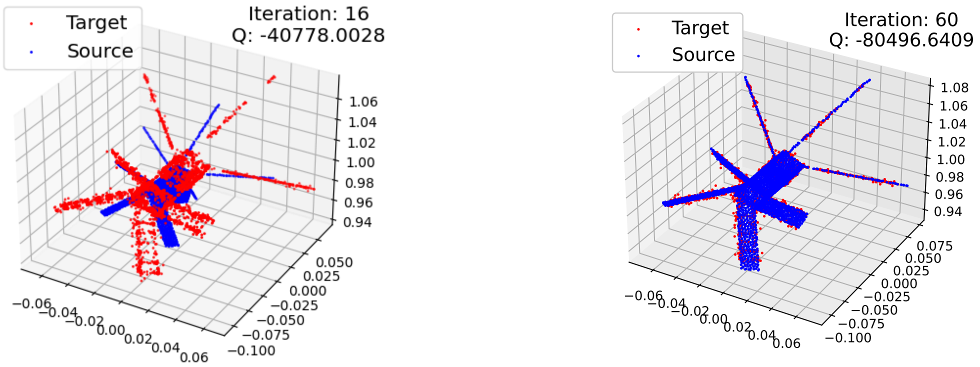
In words, the algorithm considers the alignment of two sets of points as a probability density estimation problem, where one point set represents the Gaussian Mixture Model (GMM) centroids, and the other one represents the data points. The GMM centroids are fitted to the data, by maximising the likelihood. At the optimum, alignment of the point sets is achieved, and the correspondence is obtained using the posterior probabilities of the GMM components. An example is illustrated where two sets of points, Figure 3.3a and Figure 3.3b, are used as input in the CPD algorithm. The intermediate process is shown in Figure 3.4a, while the final alignment is shown in Figure 3.4b.



(a) Constructed point cloud from the pose estimator

(b) True point cloud from the target 3D model

Figure 3.3: Generated pointclouds



(a) CPD algorithm in process of alignment

(b) CPD algorithm final alignment

Figure 3.4: Process of point cloud alignment using the CPD algorithm.

3.4. Conclusions

This chapter provided an overview of the developed technical research framework. This research framework is used to obtain answers to the research questions of this thesis. The framework mainly consists of three blocks: Datasets, Navigation System, and State comparison. The state comparison is performed by comparing the output of the SLAM algorithm with the ground truth. Since these are both measured in different frames, the frames need to be transformed. This transformation is obtained using the CPD-algorithm. The novelty of the research lays in the robustness analysis of relative motion, background objects, and hardware specifications.

4
Pose Estimator Design

The purpose of this chapter is to document the methodology that results in the selection of the most suitable pose estimation system for spacecraft relative navigation in unknown environments. An engineering approach is used to select the most suitable algorithm. In section 4.1, the selection criteria for the application of algorithms for spacecraft relative navigation to unknown targets are established. Subsequently, the design options for SLAM are evaluated in section 4.2. Afterwards, section 4.3 describes the final pose estimator design, and section 4.4 outlines the state estimator integration. Finally, the chapter is concluded in section 4.5.

4.1. Selection Criteria

Before any SLAM algorithm is analysed, it is of great essence that the selection criteria for the application of SLAM algorithms in unknown space environments are outlined. This will aid in eliminating and selecting the most suitable algorithms for spacecraft relative navigation purposes.
The main input for monocular-based SLAM algorithms are images taken in orbital conditions. Space images are subject to high noise, high contrast, and adverse illumination conditions. Also, on-board processors of spacecraft are typically limited in performance. Hence, computationally expensive algorithms are unfavourable. Furthermore, robustness to loss of pose tracking is crucial, as navigational losses in space can have great impact on the success of space missions. Therefore, the following selection criteria are established:

- robustness to images with high noise and high contrast;
- robustness to changing and adverse light conditions;
- robustness to dynamic environments;
- robustness to loss of pose tracking.

4.2. SLAM Design Options

Monocular-based pose estimation in unknown environments is achieved by means of SLAM algorithms, as outlined in chapter 2. Hence, a design option tree should be created to gain a comprehensive overview of all pose estimation options. This design option tree is illustrated in Figure 4.1. Since it is not possible to analyse all possible SLAM algorithms that exist, decisions on direction in each level of the tree are made, which will narrow down the amount of options. At every level in the tree, only one branch is further investigated, unless there is no clear superior choice. In that case, multiple branches will be further researched.

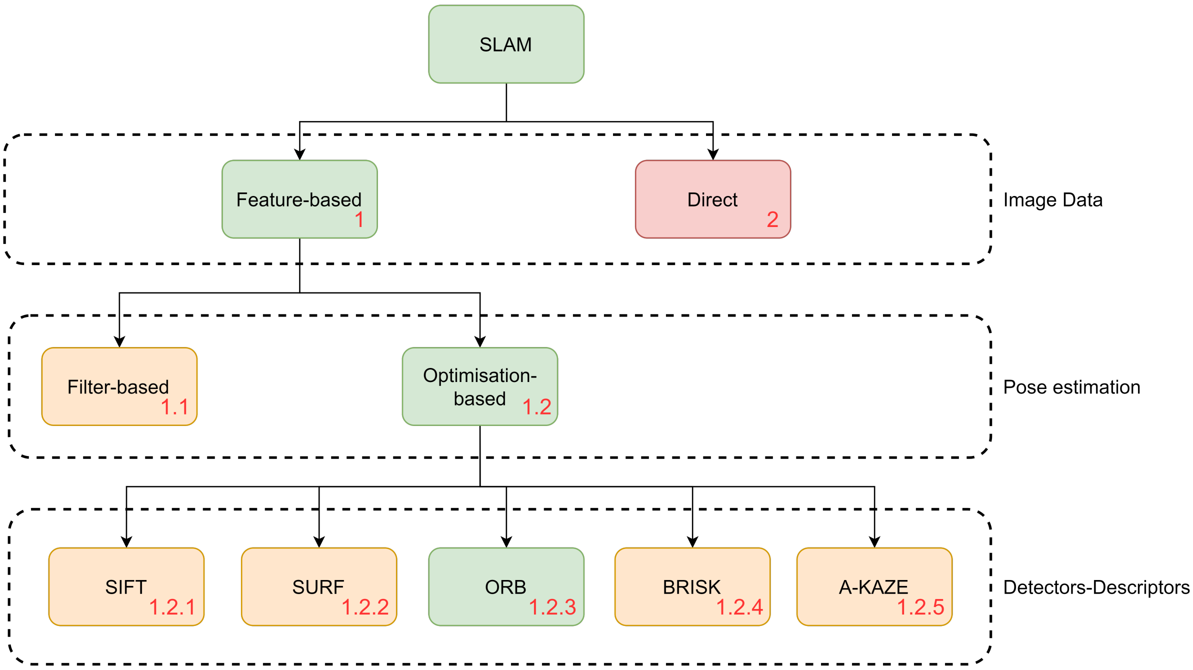


Figure 4.1: Design option tree for pose estimation. Red options are significantly inferior, green options are superior, yellow options are mildly inferior.

4.2.1. Feature-based vs Direct SLAM

The first division in the types of SLAM is feature-based and direct methods. The working principle of feature-based SLAM is based around the extraction and matching of a set of features (key points) across multiple frames [77]. Then the pose of the camera relative to the features is estimated by minimising the reprojection errors between feature pairs. Modern feature descriptors are, to a certain extent, invariant to illumination and view-point changes, making this method more robust to inconsistencies in brightness and large changes in view-point across consecutive frames. The main drawback of feature-based methods is the fact that the matching and extraction of features is computationally expensive, limiting the amount of features that can be extracted and used by the system. This creates relatively sparse point clouds of the generated map, which might bring limitations to certain applications of the algorithm.

On the other hand, direct methods use the intensity of the pixels to estimate the relative pose. Depending on the algorithm, either all pixels, all pixels with a sufficiently large intensity gradient, or sparsely selected pixels are used. Direct methods then minimise the photometric error that is obtained by direct image alignment on the selected pixels. This minimisation is performed by exploiting non-linear optimisation algorithms. The benefit of direct methods is that more image information can be used in contrast to feature-based methods, making them also more robust in low-texture environments. Additionally, the density of the reconstructed map is much higher. In correspondence to the direct image alignment, direct SLAM is very sensitive to un-modeled artefacts like rolling shutter effect, camera auto expose and gain control. Also, the assumption of constant brightness does not always hold, which drastically decreases the performance in environments with changing lighting conditions.

This last point severely limits the application of direct-SLAM methods in space. That, in combination with the superior robustness of certain modern feature descriptors in adverse lighting conditions, eliminates direct-SLAM in the design option tree.

4.2.2. Filter-based vs Optimisation-based SLAM

Further classification of feature-based SLAM can be categorised into two distinct approaches, filtering and optimisation (Bundle Adjustment (BA)). The properties and qualities of both approaches are highlighted in this section.

The general problem of SLAM can be posed in terms of inference on a graph [70], as illustrated by the Markov Random Field in Figure 4.2. The variables T_i represent the position of the camera at time i and the variables x_j

represent the position of a feature j, static in the target frame. They are linked by image feature measurements z_{ij}: the position of feature x_j from pose T_i. These links are the edges in the graph. In real-time SLAM this network will expand continuously as new poses and measurements are added at each time step, and new features are added whenever unexplored parts of the scene are exposed to the camera. Hence, finding the full maximum likelihood solution to the graph from scratch at every time step would get computationally more expensive with every step, and quickly impossible. To avoid this computational explosion, two key possibilities exist in real-time implementation of SLAM: filtering and BA, as illustrated in Figure 4.2b and 4.2c respectively.

Filtering makes sure that all poses other than the current one are marginalised out after every frame. However, features are usually kept, since they can occur again in a future frame. This results in a more or less steady, compact graph, that is perfectly steady during repeated movements in an enclosed area. The drawback of this is that a fully interconnected graph will be obtained fast. This is because eliminations of past pose variable cause fill-in with new connections between each pair of feature variables to which it was linked. Joint potentials over all of these mutually interconnected variables must therefore be stored and updated. Propagation of joint distributions gets increasingly more computationally expensive with the amount of variables, which leads to the main drawback of filter-based SLAM: the number of features in the graph is highly limited.

BA approaches solve the graph from scratch at each time step as it expands, but the graph is sparsified by eliminating all but a small subset of past poses. This subset of past poses are called *keyframes* and are chosen intelligently or heuristically. The elimination of the remaining poses is in contrast to filtering approaches, where they are marginalised instead of discarded. Consequently, BA methods produce graphs containing more elements, but the lack of marginalisation means it will stay sparsely inter-connected, which is important for inference. Hence, BA optimisation remains efficient, even when the amount of features and measurements from the keyframes are very high. The incorporation of more feature measurements compensates for the information lost from the discarded frames.

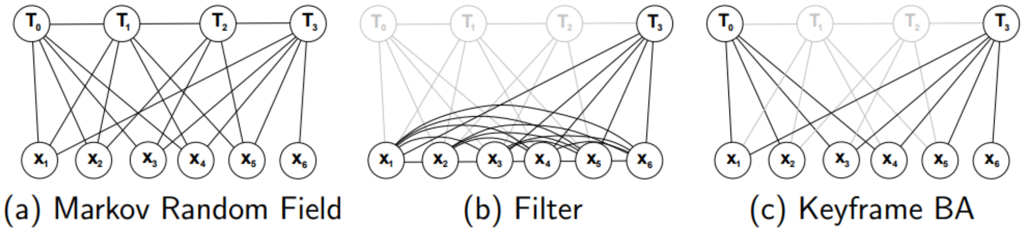


Figure 4.2: (a) SLAM/Structure from Motion (SFM) as markov random field without representing the measurements explicitly. (b) and (c) visualise how inference progressed in a filter and with keyframe based optimisation [70]

Strasdat et al. [70] demonstrated that optimisation-based approaches have better performance in terms of accuracy for the same computational cost. Hence, it has been decided to eliminate filter-based approaches and select optimisation-based methods for further study.

4.2.3. Feature Detectors-descriptors

As a further subdivision, the feature detectors and descriptor algorithms are to be discussed. Since there are many detectors and descriptor methods, it is deemed impossible to analyse all combinations of them. Therefore, it has been decided that only the most popular detector-descriptor combination algorithms are further regarded. This is deemed a valid decision as those are likely the most optimal in terms of performance. SIFT, SURF, AKAZE, ORB, and BRISK are considered as the fundamental and most renowned feature detection-description algorithms [71] and will therefore be analysed in more detail in this section.

In general, features are detected in the form of edges, corners, lines, junctions, blobs etc. These features are then described and identified logically in different ways on the basis of unique patterns possessed by their neighbouring pixels [71]. This is called feature description. In the design option tree, the most renowned feature-detection-description algorithms are evaluated that are invariant to scale, translation, orientation and illumination, as these properties are crucial for spacecraft relative navigation.

SIFT

SIFT was introduced in 2004 by D. G. Lowe [43]. The detection algorithm is based on the Difference-of-

Gaussians (DoG) operator, an approximation of Laplacian-of-Gaussian (LoG). The DoG algorithm enhances features by subtracting a Gaussian blurred version of the original image, from another less blurred version of the original image. This is mathematically shown in Equation 4.1, where G is the Gaussian function, I is the image with x and y representing the 2D location of a pixel in the image. σ is the standard deviation of Gaussian blurring, and K a number greater than 1. The feature-points are subsequently extracted by observing local maxima at various scales of the images.
The descriptor in the algorithm extracts a 16x16 neighbourhood around each detected feature and further segments the region into sub-blocks, resulting in a total of 128 bin values for description. SIFT is robustly invariant to image rotations, scale, and illumination changes. It is a widely used feature detector-descriptor algorithm in SLAM, but due to the computation complexity often not sufficient for real-time application [28].

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) \cdot I(x, y) \tag{4.1}$$

SURF
In 2008, SURF was introduced [5]. It is based on the SIFT algorithm, but with the aim to improve computational efficiency. Feature detection uses the determinant of Hessian Matrix and integral images to improve feature detection speed. Features are then described by a 64 bin descriptor (this can be extended to 128 bin values when dealing with larger viewpoint changes) using a distribution of Haar wavelet responses within a certain neighbourhood. By making use of integral images, the computational efficiency is increased, while maintaining similar invariance to scale, orientation and illumination as the SIFT algorithm [28]. Equation 4.2 represents the Hessian Matrix in point $x = (x, y)$ at scale σ, where $L_{xx}(x, y)$, $L_{xy}(x, y)$, $L_{yy}(x, y)$ are the convolutions of Gaussian second order derivative with the image in point x.

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \tag{4.2}$$

ORB
In 2011, ORB detection and description was developed [61]. The algorithm is a combination of modified Features from Accelerated Segment Test (FAST) detection methods and direction-normalised Binary Robust Independent Elementary Features (BRIEF) methods for description. FAST features are computationally efficient, but they do not have an orientation component, making them not invariant to rotation. FAST detects features based on the intensity threshold between a centre pixel and a circular ring about the centre. Since FAST does not provide a measure of cornerness, ORB employs a Harris corner measure [27] to order the FAST key-points. For a target number of extracted keypoints N, the detection threshold is set low enough such that more than N keypoints are extracted, then the Harris measure orders the points by means of their cornerness score, out of which the top N keypoints are extracted. Neither does FAST generate multi-scale features. Therefore, ORB employs a scale pyramid of the image. At each level of the scale pyramid, Harris-filtered FAST keypoints are extracted. To incorporate orientation in the FAST keypoints, ORB makes use of the intensity centroid [60]. The intensity centroid assumes that the intensity of a corner is offset from its centre, and the resulting vector can be used to impute an orientation. This implementation gives a uniformly good orientation, even under large image noise [61]

The description is based on BRIEF. It uses simple binary tests between pixels in a smoothed image patch. The performance of BRIEF is similar to SIFT in terms of robustness to blur, lighting and perspective distortion [61]. It is however very sensitive to in-plane rotation. Therefore, ORB implements a rotation-aware BRIEF. In order to elaborate on the rotation-aware version of BRIEF, the basic implementation of BRIEF will be outlined briefly. For a smoothed image patch p, A binary test τ is defined by

$$\tau(p; x, y) := \begin{cases} 1 & : p(x) < p(y) \\ 0 & : p(x) \geq p(y) \end{cases} \tag{4.3}$$

where $p(x)$ is the intensity of p at a point x. The feature is described as a vector of n binary tests:

$$f_n(p) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x_i, y_i) \tag{4.4}$$

The distribution of tests is a Gaussian distribution around the centre of the patch with vector length $n = 256$. The smoothing of the image is obtained using an integral image where each test point is a 5x5 subwindow of a 31x31 pixel patch. Now, in order to make BRIEF invariant to in-plane rotation in an inexpensive way, BRIEF is steered according to the orientation of keypoints. For any set of n binary tests at location (x_i, y_i), define the $2 \times n$ matrix

$$S = \begin{bmatrix} x_1, ..., x_n \\ y_1, ..., y_n \end{bmatrix} \tag{4.5}$$

Using the patch orientation θ and the corresponding rotation matrix R_θ, a steered version S_θ of S can be created:

$$S_\theta = R_\theta S \tag{4.6}$$

Then the steered BRIEF operator is expressed as

$$g_n(p,\theta) := f_n(p) | (x_i, y_i) \in S_\theta \tag{4.7}$$

The angle is discretised in increments of 12 degrees and a look-up table is constructed of precomputed BRIEF patterns. This implementation makes ORB features invariant to scale, translation, orientation and illumination.

BRISK
BRISK was developed around the same time as ORB and also build upon the BRIEF concept [40]. The detection algorithm is the Adaptive and Generic Corner Detection Based on the Accelerated Segment Test (AGAST) algorithm, which is an improvement compared to FAST by choosing a more robust method of estimating the decision tree [28]. BRISK exploits a refined method for estimating keypoints at various scales. The descriptor, containing 512 elements, differs from BRIEF as intensity tests are performed on a regular circular pattern around the keypoint using each location multiple times, thereby improving the look-up performance. Furthermore, smoothing of the intensity values is conducted, depending on the distance to the keypoint location. Additionally, the keypoint orientation is determined based on long-distance pairs. Also these features are invariant to scale, rotation and illumination.

AKAZE
The AKAZE algorithm was developed in 2013 [2]. It is based on non-linear diffusion filtering with its non-linear scale spaces constructed using a framework called Fast Explicit Diffusion (FED). The detector is based on the determinant of Hessian Matrix. The quality of rotation invariance is improved using Scharr filters. Maxima of the detector responses in spatial locations are picked up as feature-points. The descriptor of AKAZE is based on Modified Local Difference Binary (MLDB) algorithm. Due to the exploitation of non-linear scale spaces, AKAZE features have more distinctiveness to varying scales.

Comparison
All of the above described detection and description algorithms are invariant to scale, rotation, and illumination, meaning they are suitable for SLAM. When comparing them, Tareen and Saleem[71] conclude based on numerous experiments that ORB

- can detect the highest number of features,
- is the most efficient feature-detector-descriptor with least computational cost,
- has the least feature-matching cost,
- takes the least time for outlier-rejection and homography fitting,
- provides the fastest image matching.

However, it has a lower overall accuracy than SIFT, which is found to be the most accurate feature-detector descriptor. Nonetheless, SIFT has low computational efficiency making it less suitable for real-time applications, as well as for spacecraft applications [13]. Hence, as this is an important requirement for online navigation purposes, it has been concluded that SLAM algorithms based on the ORB feature detector-descriptor algorithm are most suitable for relative navigation in an unknown environment. Thus, this will be used in the pose estimation system.

4.3. Pose Estimator

This section represents the final pose estimation algorithm that is to be used in the research framework. Firstly, the baseline algorithm will be discussed, after which the modifications are introduced to optimise the algorithm for spacecraft relative navigation. Subsequently, a high-level comparison of the algorithm to the state-of-the-art is shown and scale recovery methods are analysed.

4.3.1. Baseline Algorithm

Recently, ORB-SLAM3 [12] has been developed as an open-source SLAM algorithm based on the work of previous versions of ORB-SLAM [46] [45]. The novelties of this work lay in the improvement of the accuracy, the implementation of a multi-map system which allows for map merging when tracking gets lost, and the improvement of loop closing, mitigating drift. In the field of robotics and computer vision, this algorithm is deemed the new reference, being as robust as the current state-of-the-art, while also being more accurate [12]. This algorithm will be the main component of the pose estimator of the real-time vision-based navigation system. A description of the working principles is provided in this section.

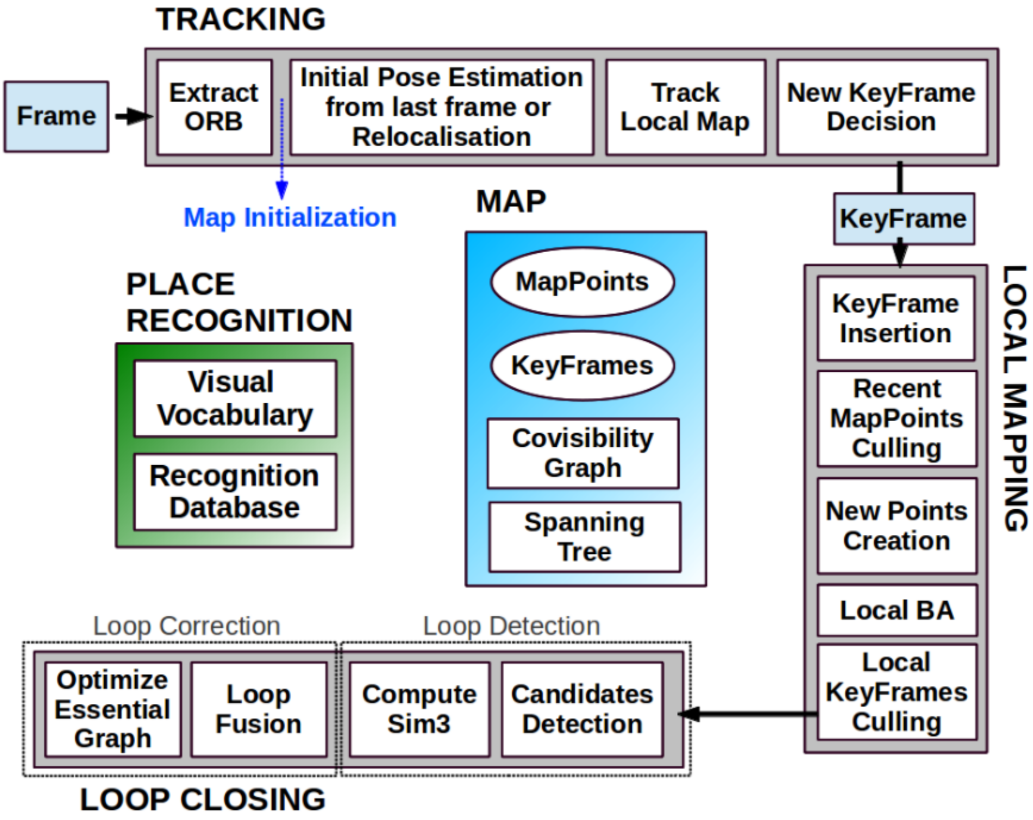


Figure 4.3: ORB-SLAM System Overview [46]

An overview of the ORB-slam system is depicted in Figure 4.3. The system incorporates three threads that run in parallel: tracking, local mapping and loop closing.

Tracking

The tracking thread is responsible for localising the camera with every frame and deciding when to insert a new keyframe. The thread extracts ORB features from each frame. Subsequently, map points of the last key frame are searched in the current keyframe by using a constant velocity motion model to predict the pose of the camera. If the motion model is not accurate, i.e. if not enough matches are found, a wider search of the map points around the position in the last frame is adapted, after which the pose is optimised with the found correspondences. Once the map and pose are initialised, the local map is tracked by projecting the stored map points in the current frame based on the estimated pose to identify and search for correspondences. Based on the correspondences, the camera pose is optimised again. The last step in this thread is to decide whether the current frame qualifies as a keyframe. This decision is based on the following conditions which have to be met:

1. More than 20 frames must have passed from the last global relocalisation or map initialisation.
2. More than 20 frames have passed from last keyframe insertion.
3. Current frame tracks at least 50 features.

4. Current frame tracks less than 90% features that are present in any other keyframe.

The robustness of this algorithm compared to similar SLAM algorithms is deduced from the fact that this algorithm implements a tolerable keyframe insertion policy. This means that keyframes are *incautiously* inserted, after which the redundant keyframes are culled. This yields high robustness in difficult scenarios and hard exploration conditions, while maintaining computational efficiency.

Local Mapping
After a new keyframe is inserted, it brings additional map points (features) to the map as well. In order to ensure that these new map points are trackable and not wrongly triangulated, they must pass a restrictive test during the first three keyframes after creation. They must meet the following two conditions:

1. The tracking must find the point in more than 25% of the frames in which it is predicted to be visible.

2. If more than one keyframe has passed from map point creation, it must be observed from at least three keyframes.

Once a point has successfully passed this test, it can only be removed from the constructed map if at any time it is observed from less than three keyframes. This can occur when keyframes are culled or when local bundle adjustment discards outlier observations. This results in very few outliers in the constructed map.
When new map points are inserted, they are triangulated. To accept the new points, numerous checks are performed: parallax, reprojection error, and scale consistency. As a map point is initially observed from two keyframes only, the point is projected onto connected keyframes as well to search for correspondences, creating a robust map. Once optimisation of the current keyframe has been performed, redundant keyframes are culled to obtain an efficiency algorithm. Keyframes are discarded when 90% of their visible map points are also included in other keyframes that are in the same scale.

Loop Closing
This thread takes the recent keyframe processed by the local mapping thread and tries to detect and close loops. Loop closing is performed to eliminate accumulated drift in the estimated trajectory of the camera. There are seven ways the trajectory can drift: three translations, three rotations and in scale. Loop candidates are detected by computing the similarity between the keyframe and its neighbouring keyframes in the covisiblity graph. To obtain information about the error due to accumulated drift, a transformation is required between the two keyframe candidates. To fuse the loop, the pose of the current keyframe is corrected using the similarity transformation, after which the correction is propagated to the other neighbouring keyframes. The loop is closed by optimising each map point using the similarity transformation.

4.3.2. Algorithm Optimisation

With respect to the standard algorithm, some modifications have been made which improve the robustness for spacecraft relative navigation. The baseline algorithm is tested on benchmark datasets containing imagery taken on Earth. This means that the algorithm is not optimised for spacecraft relative navigation. This section will optimise the algorithm based on the main challenges found. There are three main challenges that are to be solved: robustness to background objects (i), pose initialisation with low parallax due to the typical low relative velocity of the target and camera (ii), and feature matching due to high noise in the images, making the algorithm unstable (iii).

Robustness To Background Objects
SLAM is generally applied to static environments, meaning all features are static and stay fixed with respect to each other. The application of SLAM to spacecraft relative navigation requires robustness in dynamic environments as well. When an object is in the background of the frame, it does not have the same motion as the target satellite. When there are multiple objects in the frame in relative motion to each other, the environment changes to dynamic. An example is given when the Earth is in the background. The Earth does not have the same motion as the target satellite in the camera frame, resulting in an inaccurate and likely unstable SLAM algorithm. To mitigate any influence of these background features on the performance of the algorithm, it would be ideal if those features are not taken into account. Therefore, the algorithm has been modified by having a *far points threshold.* The algorithm measures the distance of the features and only stores the features that are below the distance threshold for pose estimation. The other features are discarded. Therefore, features detected on Earth could be filtered out by setting the distance threshold at a distance much further than the expected distance of the target. As monocular-only SLAM requires an estimate of the

scale in order to reveal the true relative distance of the feature points, input from the scale estimator of the system is required. The general methods for scale recovery are discussed in subsection 4.3.4.

Algorithm 1: Background Feature Filtering Pseudocode

```
for $f \in F$ do
    if $(dist_{f,c_{KF1}} < dist_{thres}) \wedge (dist_{f,c_{KF2}} < dist_{thres})$ then
        store f
    else
        discard f
    end
end
```

The pseudo-code of the algorithm is described in algorithm 1, where $dist_{f,c_{KFi}}$ and $dist_{thres}$ are the distance of feature f to the camera c in keyframe (KF) i, and the threshold distance respectively. This algorithm simply says that if the distance of a new feature to the camera in two separate keyframes is smaller than the set threshold, it will be used for pose estimation. Otherwise it will be discarded. The distance is measured from two keyframes where it is present, for the purpose of robustness and consistency. Obviously, looping through every feature at each timestep would not be computationally efficient. Hence, this is only analysed for new features which are not yet stored in the map.

Robustness to Pose Initialisation with Low Parallax

In order to match features between different frames, sufficient parallax is needed between the frames. Parallax is the apparent displacement of an object due to the change of view of the observer. Sufficient parallax is needed in order to obtain a valid initial estimate of the relative pose. This is created by tracking features across multiple sequential frames. Spacecraft relative navigation is typically characterised by high relative distance between the chaser and target, and low relative motion. Hence, pose initialisation is difficult to obtain as demonstrated in this paper [20], where they tried pose initialisation of SLAM on the Hubble telescope during relative navigation operations of the space shuttle. The amount of neighbouring frames to take into account for sufficient parallax is a trade-off between computational time and initialisation robustness. For low parallax, this parameter is optimised when it is high as possible, while still allowing to process the frames in real-time. As sufficient parallax is needed for an accurate and stable initial pose estimate, it would not be beneficial to decrease the parallax threshold too much. The optimum and default parallax threshold for this algorithm is set at a ratio of 0.01 (1%) between the distance between two keyframes and the median depth of the scene, as measured from the second keyframe. This is illustrated in Figure 4.4.

Algorithm 2: Initialisation procedure in pseudocode

```
STATE = Initialise ;
for $KF2 \in NeighKFs$ do
    baselineratio = $dist_{KF1-KF2}/MapMedianDepth_{KF2}$ ;
    if baselineratio > 0.01 then
        Search Matches ;
        Triangulate Matches ;
        if Reprojection Error < Threshold then
            add matched map points to map;
            estimate pose ;
            STATE = Track ;
            exit loop ;
        end
    end
end
```

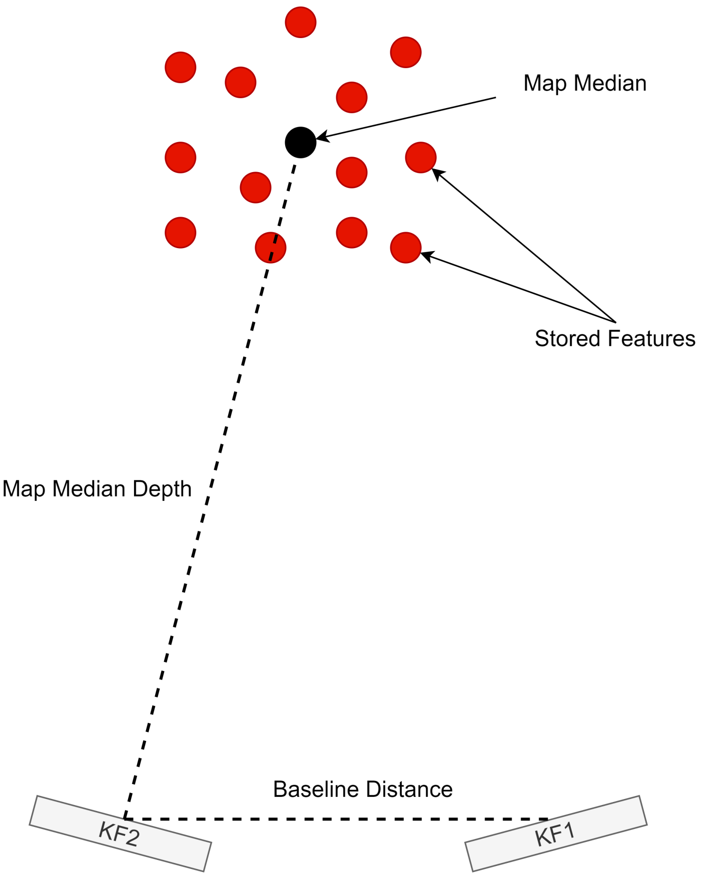


Figure 4.4: Illustration of the parallax threshold: the ratio between the baseline distance and the median depth of the map.

The algorithm is visualised in algorithm 2. From the algorithm it can be seen that by increasing the number of neighbouring keyframes, a higher chance of sufficient parallax is obtained, as keyframes that are further away are included as well. The loop might get computationally more expensive with many neighbouring keyframes, and thus real-time performance might suffer. Therefore, the parameters should be rightly set.

Other ideas to improve the initialisation phase are considered as well. One of those ideas is to only insert keyframes when sufficient parallax is achieved. However, that would reduce the performance dramatically as far fewer keyframes will be generated. Alternatively, the reprojection error threshold could be reduced to allow for faster matching of features, thereby decreasing the initial pose accuracy but allowing for faster initialisation. However, as the reprojection error threshold is directly related to the robustness of feature matching, this will provide unstable pose estimates. Consequently, this idea has also been discarded.

Space Imagery Optimisation

Due to the high noise in spacecraft images, features are not as clear as during the benchmark datasets that the algorithm has been tested on. Hence, this is not optimised for spacecraft relative navigation. The relative descriptor distance between features in different frames is a measure of similarity between features. A distance threshold decides if features are similar enough to be classified as a match. If the threshold goes down, fewer features are matched, but the certainty that the features are correctly matched goes up. This is a trade-off between robustness and accuracy of the pose. Therefore, in higher quality images the threshold can go down, but with lower resolution and high noise images the threshold might be too strict, making feature tracking more difficult. Hence, the threshold can be tuned to sacrifice accuracy for better robustness, and the other way around. The pseudocode is described in algorithm 3, where f is a feature. It should be noted that there are other conditions for a feature match as well, but not discussed here to keep it uncluttered and clear.

Algorithm 3: Feature matching procedure in pseudocode

```
for all f in 2 KFs do
    find dist between each combination of $f$;
    if $dist(f_1, f_2) < dist_{thres}$ then
        $f_1, f_2$ is a match (if other conditions are true as well)
    end
end
```

4.3.3. Comparison State-of-the-art

It is important that the decisions made in the selection and design of this pose estimation algorithm are compared on a high level to the current state-of-the-art to pinpoint odd selection choices. If the resulting algorithm is significantly different from the state-of-the-art, there is likely a wrong assumption in the methodology and criteria that lead to the decision of this algorithm. It can be noticed from Table 4.1 that the majority of the algorithms rely on feature-based methods. There is a single paper that describes a direct method (Lei et al.). They motivate their choice for a direct method by stating that direct methods have strong robustness under occlusion and weak texture scenes. However, they do not state that direct methods are very unstable under illumination changes, which is required for spacecraft relative navigation. On top of that, they do not take this into account in their simulations and experiments, which is a critical aspect of direct methods. Hence, questioning the actual validity of their algorithm for space applications. The fact that the majority of the state-of-the-art relies on feature-based methods is an assurance that the right method has been selected.

The general approach to pose estimation is split. Three papers use filtering methods. The trade-off that led to the decision of using filter-based approaches is not stated clearly in their work. Capuano et al. [13] and Ivanov et al.[32] do not mention optimisation-based methods at all, while Jin et al. [33] describe their choice for filtering due to the defect of large amount of calculation for optimisation-based methods. They state that optimisation-based approaches use data collected at all times to perform pose estimation. However, that is not exactly true, since in optimisation-based approaches data is only stored if a frame is a keyframe, discarding the other data. These keyframes are intelligently chosen [70], making the approach efficient, especially under large number of features. Based on the analysis performed in the previous section, it is deemed still the best solution to use an optimisation-based approach. However, a recommendation of further study could benefit from a comparison between a filter-based approach and optimisation-based approach, making use of the same feature detector-descriptor method. Dor and Tsiotras[20] state the benefit of using optimisation-based approaches using ORB-SLAM as it contains full BA when loop closing is achieved, eliminating the accumulated drift over time. This is another advantage of optimisation-based methods for spacecraft applications over filter-based methods.

Additionally, the same paper also states their motivation for ORB features. They can be rapidly extracted when compared to other algorithms like SIFT, increasing the total number of features that can be extracted in real-time. In the other papers, Feature detectors are based on the Harris corner detector algorithm and a self-constructed one [32]. The self-constructed one is too simple to really be implemented in real life. It only works when the relative velocity is low, when there are well-recognisable features, and when illumination conditions are not too harsh. The other papers use the Harris corner detection algorithm. They motivate this by mentioning that their main focus is not robustness to space imagery, but they are focusing on the novelty of their pose estimation filter algorithm. The synthetic images that they use as input for the algorithm are not representative as space imagery, since they do not contain high noise and contrast. This confirms that their focus point is indeed not on feature extraction. Since the one paper that tested their algorithm successfully on real space imagery also uses ORB features, it is confirmed that the right choice of feature detection-description algorithm has been made.

Table 4.1: Comparison of pose estimation methods in the current state-of-the-art

Reference	Image Data	Pose estimation	Detector Descriptor
This work	Features	Optimisation	ORB
Dor and Tsiotras [20]	Features	Optimisation	ORB
Lei et al. [39]	Direct	-	-
Capuano et al. [13]	Features	Filtering	Harris
Jin et al. [33]	Features	Filtering	Harris
Ivanov et al. [32]	Features	Filtering	Self-constructed

4.3.4. Scale Recovery

The focus of this work is on the monocular SLAM estimation algorithm. Therefore, the scale of the environment is assumed to be known exactly. To obtain an estimate of the scale, a secondary sensor is required to be on-board. The data fusion between the two sensors and the analysis of the resulting accuracy are deemed outside the scope of this work. The scale estimate collected from a real sensor is not necessary to assess the performance of the designed pose estimator in this work. In the current state-of-the-art, the only two papers that highlight the aspect of scale recovery are Capuano et al. [13] and Jin et al.[33]. They both assume a low-power single beam LIDAR measurement for a depth estimation. As they do not analyse the sensor fusion and integration in the pose estimation, it is assumed that in their work also a perfectly known scale is used. This section however analyses the various options to recover the scale.

Depth Sensors

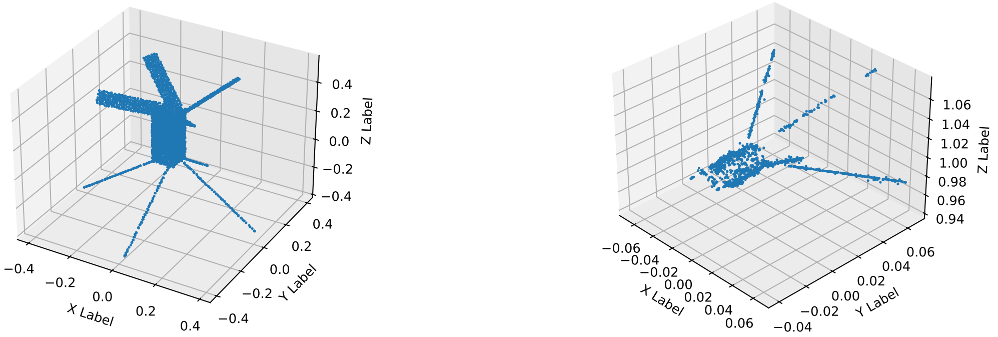
When there is no information about the target, the scale of the target cannot be recovered using only a single camera. Additionally, inertial sensors like Inertial Measurement Unit (IMU)s, which are typically used in satellites, cannot be used when the target in the camera is not inertially fixed. This is because the inertial sensors measure the motion of the chaser satellite with respect to inertial space, but the camera measures the motion of the chaser satellite with respect to a non-inertial target during relative navigation operations. Since the motion of the target with respect to inertial space is not known, IMUs cannot be used. Therefore, depth sensors are required that allow to measure the distance between a feature point and the sensor itself. The uncertainty in the depth measurement would be directly related to the uncertainty in the relative position according to the transformation matrix $A^C_{T/C}$, with scale λ.

$$A^C_{T/C} = \begin{bmatrix} R_{T/C} & \lambda t^C_{T/C} \\ 0 & 1 \end{bmatrix} \tag{4.8}$$

Where T is the target frame, C is the camera frame, R the rotation matrix, t the translation vector. When no information is available, depth measurements are seen as the only option for scale recovery.

Target Apriori Information

When there is partial information about the target available, it can be used to obtain scale estimates. An example is given when a target has been subjected to damage, and OOS or ADR operations are to be performed. In that case, the reconstructed visibly damaged target can be compared to a 3D model of the target to obtain a scale estimate. This process does not have to be performed online, but can be done offline in parallel to tracking of the scale-ambiguous target. An algorithm which could do that is the CPD algorithm, which is explained in more detail in section 3.3. CPD does not require an exact match of the two pointclouds as is tested by using a pointcloud generated in ORB-SLAM of the Delfi-n3Xt and a pointcloud of the 3D model. This is illustrated in Figure 4.5a to Figure 4.6. The illustrations show that CPD can accurately perform a transformation between half a reconstructed satellite by SLAM, and a 3D pointcloud of the model.



(a) Pointcloud of the 3D model

(b) Pointcloud of the constructed target with damage

Figure 4.5: Pointclouds of a full 3D model and partially reconstructed target

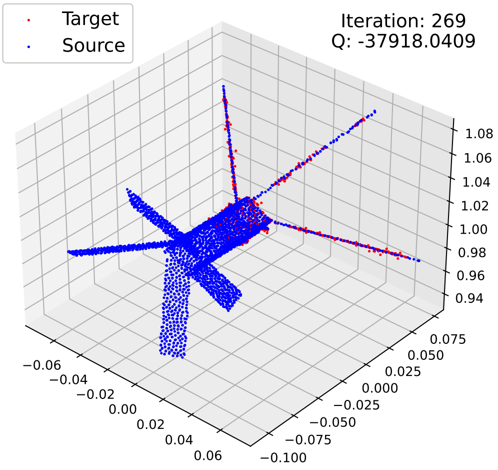


Figure 4.6: Accurate rigid registration of half a constructed target and a full 3D model pointcloud

4.4. State Estimator Integration

Spacecraft require constant estimation of its state at a certain frequency which may not be the frequency the pose estimator can deliver pose measurements. Furthermore, pose measurements might not always be available, for instance during eclipses or any other losses of tracking. Therefore, filter estimates are required which could update the state of the system based on a dynamic model. The purpose of this section is to prove the feasibility of the designed pose estimator with a state estimator. Typical Kalman filters use a dynamic model that can propagate the state of the target, where the state is defined as the pose and its derivatives. The problem for unknown uncooperative target pose estimation is that properties of the target are not available. Typical attitude propagation models require the mass moment of inertia of the object. But since this is not available, large uncertainties would be imposed on the state estimation when propagating the state. Therefore in this approach, the state vector is defined as the relative pose and the relative translational velocity only. The relative angular velocity comes from a measurement of the pose estimator. It is assumed for simplicity that the relative velocity is constant between pose measurements. Hence, The relative angular velocity is the difference between two attitude measurements over time.

More sophisticated approaches can include the measurement of angular velocity of the target from optical flow of the feature points. A novel approach is presented in [13]. They use direct optical flow measurements in combination with a star tracker to obtain an estimation of the relative velocity of the target and the location of the centre of gravity, by breaking up the optical flow measurement in different inertial components in order to separate the camera's own motion from the target's rotation and relative velocity. This way, the angular velocity of the target relative to inertial space is determined. Also, if the target's rotation axis is a principal axis, the rotational axis will be obtained on which the centre of gravity is located. If the rotation axis is a non-principal axis, then a different axis of rotation can be determined in a subsequent frame and by crossing both axes the exact location of the centre of gravity is obtained, from which the advantage can be taken that the relative position can be measured from the actual centre of gravity instead of the initialised centre of the target. The proof of concept of this method is demonstrated in [13]. Since this is not the novelty of this work, such sophisticated approaches are not implemented here. Merely a simple Kalman filter integration will be demonstrated that uses direct measurements from the pose estimator.

4.4.1. Dynamics of Relative Motion

In order to propagate the state of the previous time step to the current time step, a dynamic model is needed that can accurately describe the relative motion between two frames of reference. The dynamics of relative motion can be split up in the translational and rotational dynamics.

Translational Dynamics

The translational dynamics for relative orbital motion can be described by the Clohessy-Wiltshire (C-W) equations [16]. Assuming a mean circular orbit for both satellites, they are stated as:

$$\begin{aligned} 0 &= \delta\ddot{x} - 3n\delta x - 2n\delta\dot{y} \\ 0 &= \delta\ddot{y} + 2n\delta\dot{x} \\ 0 &= \delta\ddot{z} + n^2\delta z \end{aligned} \tag{4.9}$$

Analytically solving these equations result in the closed-form solution as a function of time:

$$\begin{bmatrix} \mathbf{r}(t) \\ \mathbf{v}(t) \end{bmatrix} = \Phi_{rv}(t) \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{v}_0 \end{bmatrix} \tag{4.10}$$

where **r** and **v** are respectively the relative position and velocity, and where Φ_{rv} is the state transition matrix as expressed in Equation 4.11 and Equation 4.12.

$$\Phi_{rv}(t) = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} \tag{4.11}$$

where,

$$\begin{aligned} \Phi_{11} &= \begin{bmatrix} 4-3\cos nt & 0 & 0 \\ 6(\sin nt - nt) & 1 & 0 \\ 0 & 0 & \cos nt \end{bmatrix} \\ \Phi_{12} &= \begin{bmatrix} \frac{1}{n}\sin nt & \frac{2}{n}(1-\cos nt) & 0 \\ \frac{2}{n}(\cos nt - 1) & \frac{1}{n}(4\sin nt - 3nt) & 0 \\ 0 & 0 & \frac{1}{n}\sin nt \end{bmatrix} \\ \Phi_{21} &= \begin{bmatrix} 3n\sin nt & 0 & 0 \\ 6n(\cos nt - 1) & 0 & 0 \\ 0 & 0 & -n\sin nt \end{bmatrix} \\ \Phi_{22} &= \begin{bmatrix} \cos nt & 2\sin nt & 0 \\ -2\sin nt & 4\cos nt - 3 & 0 \\ 0 & 0 & \cos nt \end{bmatrix} \end{aligned} \tag{4.12}$$

In these equations, n is the mean orbital motion and t the time since their initial position and velocity were measured. It is important to note that the camera frame of reference should be transformed to the frame of reference of the C-W equations. Furthermore, the C-W equations are merely used for preliminary analysis of the integration of the state estimator. These equations do not accurately model perturbations and other disturbances in orbit. For the scope of this work, the use of the C-W equations is deemed sufficiently accurate.

Rotational Dynamics

The rotational dynamics are generally described by a kinematics equation for the attitude representation, and a dynamics equation for the relative angular velocity. However, as observed in Euler's equation of rotational motion (Equation 4.13), the moment of inertia of the target is required. Since no information about the target is available, this would impose a large uncertainty on the filter estimates. Consequently, the risk of divergence in the filter is significantly increased if that uncertainty is not accurately modelled. Therefore, the angular velocity measurement is directly taken from the pose estimator, creating lower accuracy, but higher robustness in the filter output.

$$\tau = \mathbf{I}\dot{\omega} + \omega \times (\mathbf{I}\omega) \tag{4.13}$$

In contrast to the mentioned dynamics equation of rotational motion, the attitude kinematics are modelled in the EKF. The attitude is represented by Modified Rodrigues Parameters (MRPs), since they are highly linear, efficient, and have great simplicity [14]. The differential equations of the kinematics of MRPs are stated as:

$$\dot{\sigma} = \frac{1+\|\sigma\|^2}{4}\left(I_3 + 2\frac{[\sigma\times]^2 + [\sigma\times]}{1+\|\sigma\|^\mathbf{2}}\right)\omega \tag{4.14}$$

Where σ is the orientation in MRPs, I_3 the 3x3 identity matrix, ω the angular velocity, and where $[\sigma\times]$ is the cross product matrix defined as:

$$[\sigma\times] = \begin{bmatrix} 0 & -\sigma_3 & \sigma_2 \\ \sigma_3 & 0 & -\sigma_1 \\ -\sigma_2 & \sigma_1 & 0 \end{bmatrix} \tag{4.15}$$

4.4.2. Multiplicative Extended Kalman Filter

This section outlines the implementation of the multiplicative Extended Kalman Filter that will be exploited. As with all Kalman filters, there exists a propagation step and a measurement update step. The system is initialised with a state vector and state covariance matrix:

$$\hat{\mathbf{x}}_0^+ = E[\mathbf{x}_0] \tag{4.16}$$

$$\mathbf{P}_\mathbf{0}^+ = E[(\mathbf{x_0} - \hat{\mathbf{x}_0})(\mathbf{x_0} - \hat{\mathbf{x}}_0)^T] \tag{4.17}$$

Propagation Step

For a linearisable dynamical system, the system dynamics and a measurement model in discrete time can be written in discrete form as:

$$\mathbf{x}_k^- = \Phi_k \mathbf{x}^+{}_{k-1} + \mathbf{w}_k \tag{4.18}$$

where Φ_k is the state transition matrix and $\mathbf{x}_k$ is the state vector at discrete time intervals $k = 1, 2, ..., n$. The process noise in state propagation $\mathbf{w}_k$ is assumed to be zero-mean Gaussian noise vectors with covariance Q_k such that:

$$\mathbf{w}_k \sim N(0, Q_k) \tag{4.19}$$

Appropriate values for Q_k should be determined for a robust and accurate filter. The state vector is mathematically expressed as:

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{r}_k \\ \boldsymbol{v}_k \\ \boldsymbol{\sigma}_k \end{bmatrix} \tag{4.20}$$

And the state transition matrix is:

$$\mathbf{\Phi} = \begin{bmatrix} \Phi_{rv} & 0 \\ 0 & \Phi_\sigma \end{bmatrix} \tag{4.21}$$

Where Φ_{rv} is described in Equation 4.11, and Φ_σ is obtained by neglecting the higher orders in Equation 4.14 and assuming constant angular velocity between time steps:

$$\Phi_\sigma = \left[-\Delta t[\omega\times]\right] \tag{4.22}$$

With $[\omega\times]$ being the cross product matrix of ω.

Next to the state vector, the state covariance matrix $\boldsymbol{P}$ also needs to be propagated over time. This propagation is performed using the equation:

$$\boldsymbol{P}_k^- = \Phi_k \boldsymbol{P^+}_{k-1} \Phi_k + Q_k \tag{4.23}$$

Update Step

The update step is implemented to update the filter parameters with new measurements. This step is only performed once new measurements are available. It is governed by the following equation.

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{4.24}$$

where $\mathbf{H}_k$ is a matrix relating the the state vector $\mathbf{x}_k$ to the measurement vector $\mathbf{z}_k$ at discrete time intervals $k = 1, 2, ..., n$. The measurement noise $\mathbf{v}_k$ is assumed to be zero-mean Gaussian noise a vector with covariance R_k such that:

$$\mathbf{v}_k \sim N(0, R_k) \tag{4.25}$$

Subsequently, The state estimate is updated with the Kalman gain:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + R_k)^{-1} \tag{4.26}$$

The state error estimation is defined as:

$$\delta \boldsymbol{x}_k = \boldsymbol{K}_k (\boldsymbol{z}_k - \boldsymbol{H}_k \boldsymbol{x}_k) \tag{4.27}$$

and the state estimation is then:

$$\begin{gathered} \boldsymbol{x}_{rv_k}^+ = \boldsymbol{x}_{rv_k}^- + \delta \boldsymbol{x}_{rv_k} \\ \boldsymbol{x}_{\sigma_k}^+ = \delta \boldsymbol{\sigma}_k \otimes \boldsymbol{\sigma}_k \end{gathered} \tag{4.28}$$

Where $\boldsymbol{\sigma} \otimes \boldsymbol{\sigma}$ is:

$$\sigma = \delta\sigma \times \hat{\sigma} \tag{4.29}$$

$$\frac{(1 - |\hat{\sigma}|^2 \delta\sigma + (1 - |\delta\sigma|^2)\hat{\sigma} - 2\delta\sigma \times \hat{\sigma}}{1 + |\delta\sigma|^2 |\hat{\sigma}|^2 - 2\delta\sigma^T \hat{\sigma}} \tag{4.30}$$

The state covariance matrix is updated as:

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K_k H_k})^T + \mathbf{K_k} R_k \mathbf{K_k^T} \tag{4.31}$$

4.4.3. Navigation System Architecture

A high-level overview of the navigation system architecture is provided in Figure 4.7. As the pose estimator is relatively complex, only the main function blocks are illustrated. It should be noted that for mission mode of this navigation system, the filter will provide constant state output at the desired frequency by the GNC loop, regardless if pose measurements are available. The update step in the filter is only performed when it receives a pose and angular velocity measurement from the pose estimator. The loosely-coupled approach in the navigation system is clearly visible, and is used to analyse the performance of the pose estimation system. Furthermore, for the scale recovery, the pose estimator would need an additional input from a secondary depth sensor. But as already discussed in this chapter, the scale is assumed to be known.

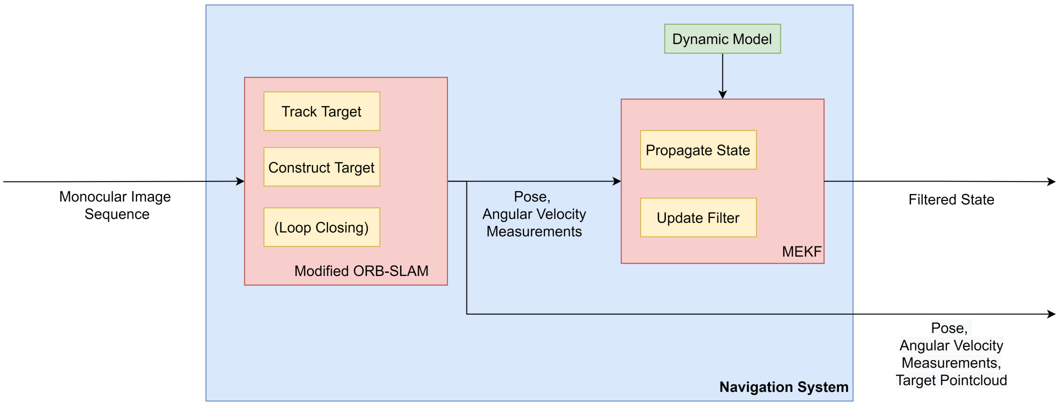


Figure 4.7: High-Level Navigation System Architecture

4.5. Conclusions

This chapter documented the design process of the pose estimator to be used in the analysis. The resulting pose estimator is based on ORB-SLAM [12]. It is modified such that the performance for spacecraft relative navigation is improved. The modifications include a background feature elimination system, a robustness to low-parallax implementation, and a modification on the feature distance threshold, such that the accuracy is improved. The pose estimator is integrated in a state estimator in a loosely-coupled approach. The state estimator is an EKF with improved robustness by not including the angular velocity in the state vector, but by including the angular velocity in the measurement vector. This results in a more robust state estimator, since large uncertainties due to the unknown moment of inertia of the target are removed.

5
Datasets

In order to analyse the performance and suitability of monocular SLAM algorithms for space applications, a research framework is designed around the SLAM algorithm. An integrate part of this framework are the datasets that are fed to the pose estimator. This chapter is divided into four sections. The first section, section 5.1, gives an overview of the dataset objectives. Then, both the synthetic and lab-generated datasets are described in section 5.2 and section 5.3, respectively. Finally, the chapter is concluded in section 5.4.

5.1. Dataset Objectives

It is of great importance that the design of the datasets makes it achievable to answer the research questions of this thesis. Therefore, the datasets should adhere to certain guidelines:

- The datasets need to be characterised by high contrast, and high noise to represent real space imagery.
- The datasets need to contain various orbital scenarios so that robustness tests to relative motion can be performed.
- The datasets need to contain background objects for robustness assessment.
- The images in the datasets need to be captured using various hardware settings for hardware limitation assessment.
- The datasets need to contain ground-truth data for accuracy assessment.

Various methods have been examined to create datasets. Ideally, existing space footage of relative navigation manoeuvres would be used. But as this is scarce, no real robustness assessment can be performed. Furthermore, absolute ground-truth data can often not be retrieved. Instead, estimated ground truth data of real space imagery is generally retrieved from sensors on both the chaser and target satellite during demonstrations. It should be emphasised that this is no absolute ground-truth, but can give an approximate performance assessment. In the context of this work, another drawback is found in the necessity for a 3D model of the target satellite. Accurate 3D CAD models of spacecraft are typically not publicly available. This 3D model would be required for accuracy assessment by means of the CPD algorithm explained in section 3.3.

A second method to obtain realistic datasets is to use *professional* and accurate close proximity simulators or existing spacecraft pose estimation datasets. A benchmark example for pose estimation dataset is the SPEED dataset [36]. This dataset is used mainly for learning-based algorithms, but is unsuitable for SLAM since it does not contain a chronological sequence of images. Acquisition of any software or datasets has not been achieved for this work, therefore it has been determined that the results in this work will be based on self-generated datasets.

The self-generated datasets will be a combination of synthetically generated images, and lab-generated images. The primary objectives of the synthetic datasets are listed as

1. Provision of spacecraft image sequences for hardware limitation assessment.

2. Provision of spacecraft image sequences for relative motion robustness assessment.
3. Provision of spacecraft image sequences for background objects robustness assessment.

Ideally, lab-generated datasets are created using machines/robotic arms that can accurately position the camera. However, that is not available for this thesis. Instead, the lab-generated dataset is captured on a hand-held gimbal. This makes the testing of relative motion influence on the performance of the pose estimation more difficult, as exact velocity and pose requirements can not be met. Therefore, it has been determined that the lab-generated dataset will be used for verification of the following points:

1. Verification of realistic quality for synthetic images.
2. Verification of feature detection and matching on real, space-represented imagery.

The first point is listed to confirm the quality of the results obtained from the synthetic dataset. It is important that the synthetic datasets are generated as realistic as possible in terms of appearance of the target. By making use of an actual solar simulator that can recreate the exact solar spectrum as present in orbit, and an actual target mock-up model, the realisticness of the synthetic target can be verified.
The second point will verify that the pose estimator does not only work on synthetic imagery, but can also detect and match features on real imagery. ORB-SLAM has already been tested on benchmark datasets, but the contribution of this point is to test it on imagery that simulates the space environment.

5.2. Synthetic Datasets

The synthetic datasets are created using the Blender software [11]. Blender is a user-friendly, free, and open-source 3D computer graphics software, and deemed suitable to learn in the timeframe of this work. This section describes the process of the dataset generation in Blender.

5.2.1. Environment Modelling

The environment is modelled by the background of the image, the lighting conditions, the target object, and the relative trajectory.

Background
The background is modelled as plain black, since this is most representative for real space imagery and is also used in the SPEED dataset [3]. In the datasets for background robustness tests, the Earth will be modelled. The Earth is modelled by overlaying three images on top of each other, after which the image is wrapped around a blue sphere. The three images are displayed in Figure 5.1 to Figure 5.3. The background has been made black for better showcasing in this report. Depending on the direction of the simulated sunlight in Blender, the Earth's night and day is also realistically modelled.

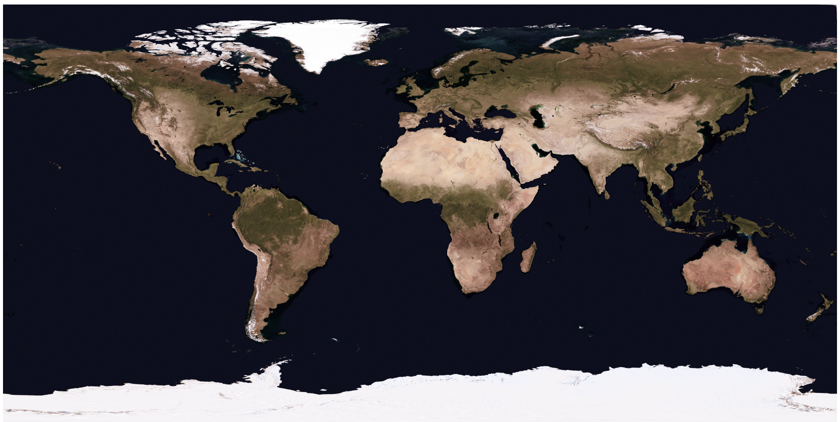


Figure 5.1: Image of Earth in full sunlight

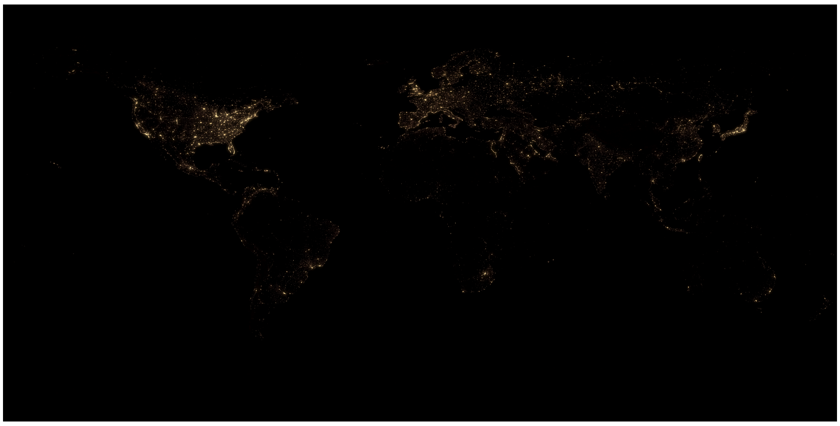


Figure 5.2: Image of Earth in no sunlight

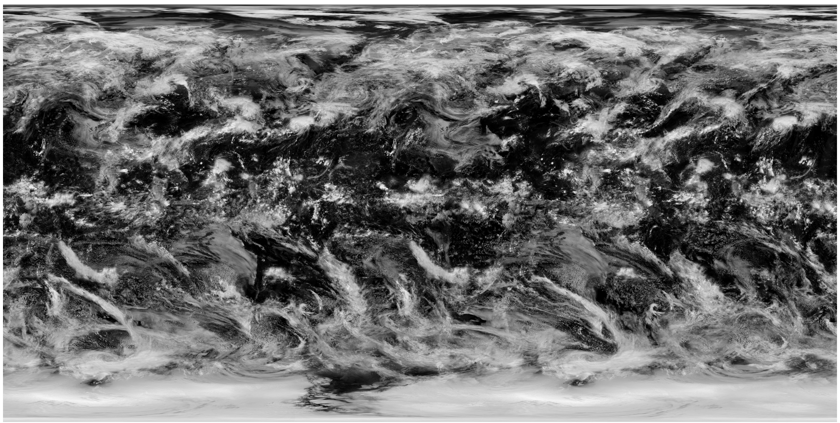
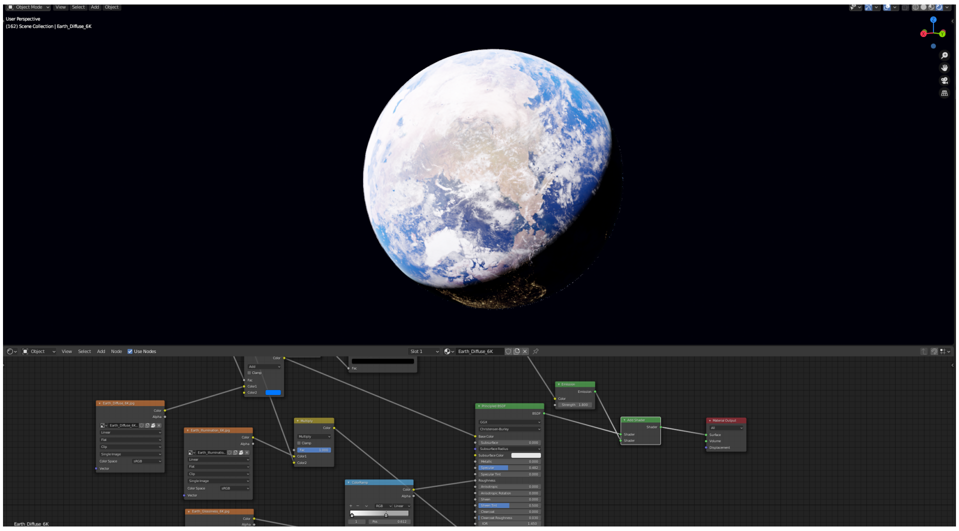


Figure 5.3: Image of Earth cloud field

The final result is visualised in Figure 5.4. The night/day line is clearly observed and shifts with the direction of the sunlight. Since the purpose of the Earth is to test the background feature elimination capabilities of the pose estimator, it is essential that features can be detected by the algorithm on the Earth. The capability of feature detection and matching with the Earth in the background has been tested on the default algorithm of ORB-SLAM. This is visualised in Figure 5.5. Hence, proving the functionality of the simulated Earth.



User Perspective
(162) Scene Collection | Earth_Diffuse_6K

Earth_Diffuse_6K.jpg: Earth_Diffuse_6K, Linear, Flat, Clip, Single Image, Color Space sRGB

Earth_Illumination_6K.jpg: Earth_Illumination, Linear, Flat, Clip, Single Image, Color Space sRGB

Earth_Glossiness_6K.jpg

Mix: Add, Clamp

Multiply: Multiply, Clamp, Fac 1.000

ColorRamp: RGB, Linear, 1, Pos 0.612

Emission: Strength 1.800

Principled BSDF: GGX, Christensen-Burley, Subsurface 0.000, Subsurface Radius, Metallic 0.000, Specular 0.482, Specular Tint 0.000, Anisotropic 0.000, Anisotropic Rotation 0.000, Sheen 0.000, Sheen Tint 0.500, Clearcoat 0.000, Clearcoat Roughness 0.030, IOR 1.450

Material Output: All

Slot 1, Earth_Diffuse_6K

Figure 5.4: Earth modelling in Blender

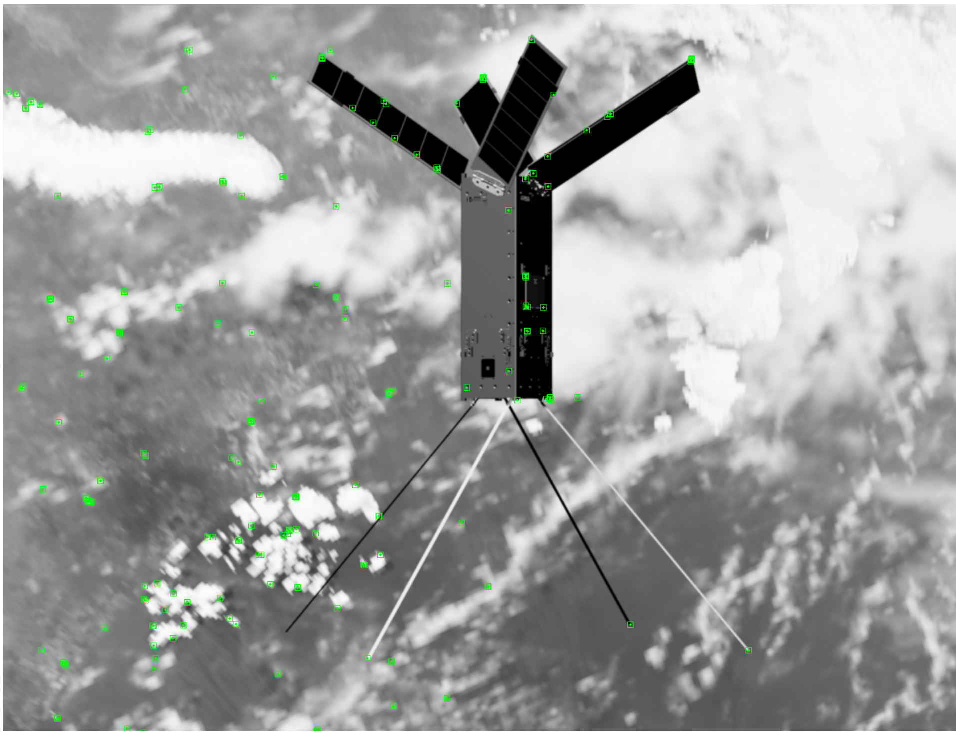


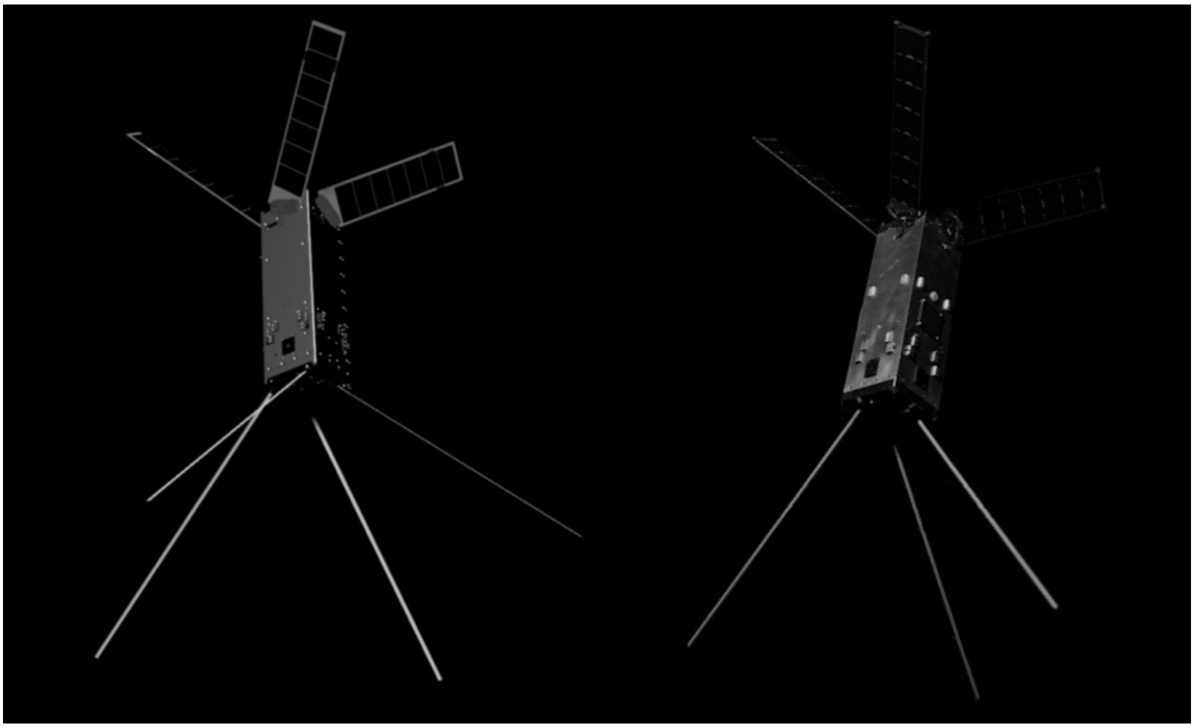
Figure 5.5: Clear features detected on the simulated Earth background

Lighting
The visible light in orbit comprises sunlight and albedo from celestial objects. In orbit, the majority of albedo comes from Earth's surface. The sunlight in Blender has been set to pure white, and the power has been adjusted such that the brightness of the target is similar to the SPEED dataset [3], meaning that the features are of similar visibility. Since the SPEED dataset is a benchmark dataset for pose estimation systems, this is deemed a good reference for the self-generated datasets. The appearance and visibility verification of the target are further described under the header **Target** hereafter. The Earth albedo is adjusted such that it is 30% of the sunlight's power [51].

Target
It is important that the appearance of the simulated target is as realistic as possible. Therefore, the Delfi-n3Xt materials have been simulated in Blender using the properties of the real materials. It is essential that the reflection of sunlight on the material is similar to real space imagery, since the visibility of features has significant influence on the performance of the pose estimator. The realistic quality of the target is verified by means of comparison to images generated in the lab, and to images in the SPEED dataset. This is verified using the following comparisons.

The comparison to the lab-generated image is displayed in Figure 5.6a and Figure 5.6b. Both images are captured under the same angle, and the approximate same direction of sunlight.
Apart from the lack of an antenna in the lab-generated satellite and the visible distortions of the real camera, it is noticed that the visibility and reflectivity of the components are relatively close. The main difference is the texture on the aluminium surface of the lab-generated satellite, which is lacking in the simulated one. Due to the lack of texture, it can be assumed that the performance on the synthetic images will be conservative, since the real imagery will thus contain more texture for features to be detected on. Also, it is noticeable that the solar panels of the CAD model are slightly different than the real mock-up. The solar cells do not cover as much surface in the CAD model, making the underlying panel itself more visible. The visible edge of the panel in the lab image is comparable in brightness. Hence, the reflective properties of the solar panels are considered rightly chosen.

The second comparison is with an image of the SPEED dataset, visualised in Figure 5.7. It should be noted that the target satellite is different. So a real comparison cannot be made. However, the visibility of the features can be compared and assessed. It is noticed that the shading on the different materials in the SPEED image is



(a) Delfi-n3Xt synthetic

(b) Delfi-n3Xt under real space lighting

Figure 5.6: Comparison synthetic and lab-generated target

also uniform, just like the self-created one. Furthermore, the spectrum of brightness, shading, and visibility of features is comparable in both images. Thereby verifying the quality of the self-generated datasets.

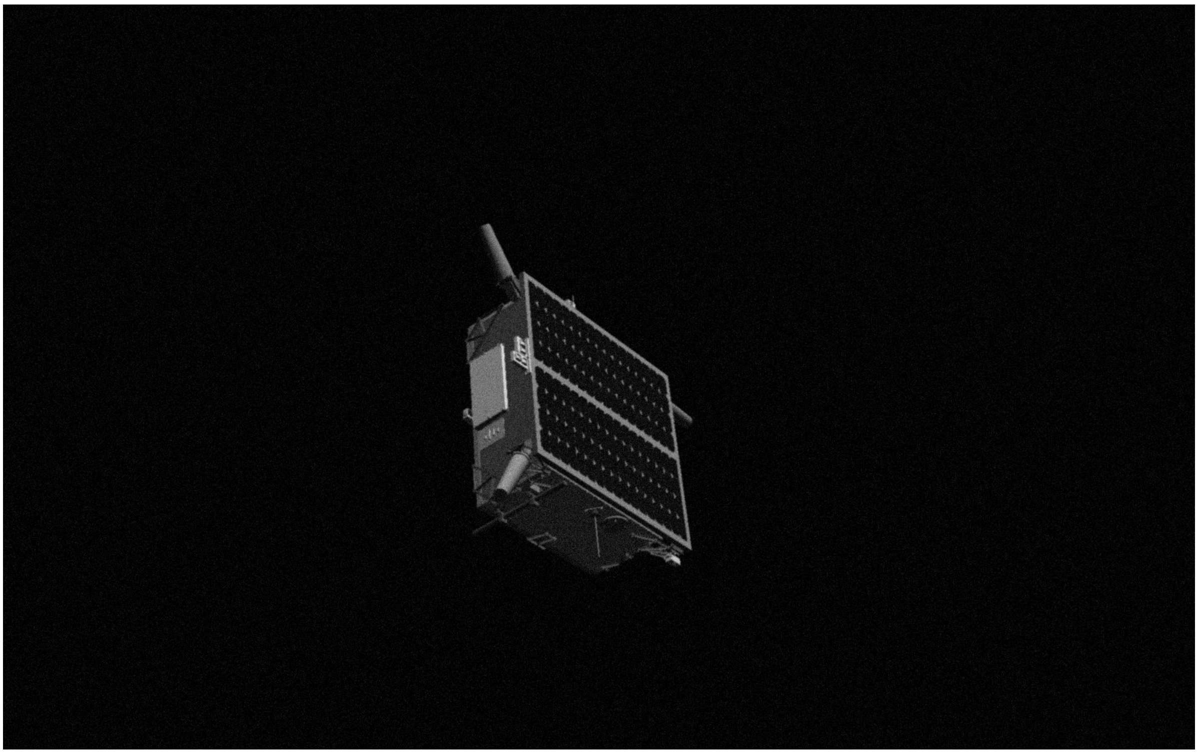


Figure 5.7: SPEED dataset imagery [3]

Trajectory

The trajectory has been defined as a circle around the target, as illustrated in Figure 5.8. This is chosen due to several reasons. First, a pointcloud of the target needs to be constructed by the pose estimator in order to obtain pose accuracy information, as is discussed in section 3.3. The more complete the pointcloud is, the greater the validity of the obtained results in this research. By travelling in full circles around the target, the target can be reconstructed from all sides, thereby creating a more complete pointcloud. Secondly, the influence of the lighting conditions on the pose estimator can be evaluated as the lighting direction remains fixed, but the camera will be rotating with respect to the light direction. Hence, it can be verified if tracking will become more difficult on the darker sides of the target. Thirdly, Dor and Tsiotras[20] showed that map initialisation is difficult for spacecraft relative navigation due to the low parallax between frames. By travelling in a circle around the target, parallax requirements for map initialisation can be obtained. The relative distance to the target is then not of influence, as the decisive factor for sufficient parallax is the angular velocity. Consequently, this will simplify the testing scenarios for relative motion as the map initialisation will be directly related to the relative angular velocity.

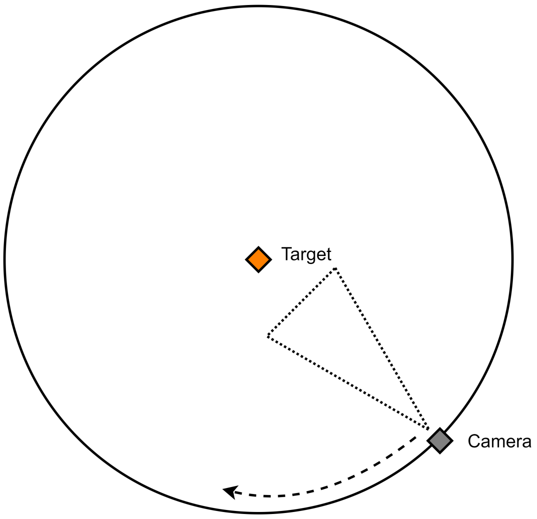


Figure 5.8: Illustration of camera trajectory

5.2.2. Image Post-processing

The camera that is simulated in Blender is a pinhole camera without any modelled distortions. Furthermore, rendering of the frames yields no visible noise or motion blur. As these phenomena are inherent to real-life cameras, especially in the field of spacecraft VBN, the rendered frames are post-processed. In the post processing step, Gaussian noise and motion blur are added to each frame. The motivation for these processing steps is based on the SPEED dataset. In this dataset, Gaussian white noise is added with $\sigma^2 = 0.0022$, and Gaussian blur with $\sigma = 1$ [3]. The same noise and blurring method, with the exact parameters, is adapted in this work.

5.2.3. Ground Truth

Ground truth values of each frame are extracted from Blender. At every frame, a transformation matrix is computed between the camera and target frame. Subsequently, the Blender camera frame basis vectors are transformed to the camera frame basis vectors of Open3D, which are used in the pose estimator. The open3D frame has the z-axis pointing along the boresight of the camera, the y-axis down, and the x-axis complements the right-handed coordinate system. This frame of reference is also assumed to coincide with the body-fixed frame of the chaser spacecraft.

5.2.4. Test Cases

An overview of the testing scenarios is described in Table 5.1. The motivation for each of the scenarios is outlined. It is further noted that these are the baseline datasets to answer the research questions. In case inconclusive results are obtained, additional datasets are generated in chapter 6.

Table 5.1: Overview of synthetically generated datasets

Scenario	**Ang Vel [deg/s]**	**Resolution [px]**	**Frame rate [Hz]**	**Brightness [%]**	**Background [-]**	**Revolutions [-]**
RMVELhigh	10	1920x1440	10	100	no	3
RMVELmed	5	1920x1440	10	100	no	3
RMVELlow	1	1920x1440	10	100	no	3
HWRESmed	5	1080x810	10	100	no	3
HWRESlow	5	860x645	10	100	no	3
HWFRmed	5	1920x1440	5	100	no	3
HWFRlow	5	1920x1440	1	100	no	3
EEBRhigh	5	1920x1440	10	150	no	3
EEBRlow	5	1920x1440	10	50	no	3
EEbackground	5	1920x1440	10	100	yes	3
EEeclipse	5	1920x1440	10	1	no	3

Relative Motion [RM]
The RM datasets test the robustness of the pose estimator to a range of relative angular velocities. As mentioned earlier, the angular velocity is the single important factor for the functionality of the pose estimator in terms of map initialisation. This is because the most challenging part of pose estimation using SLAM is the initialisation phase, especially for spacecraft applications as demonstrated in [20]. Since the motion is circular around the target, sufficient parallax between frames is determined by the relative angular velocity (and frame rate). A realistic range for the relative angular velocity for spacecraft relative navigation is determined to be between 1 to 10 deg/s. This is also the range that various other pose estimation algorithms for space applications have been tested on [4] [13]. As a result, the angular velocity for test cases RMVELhigh, RMVELmid, and RMVELlow are respectively 10, 5, and 1 deg/s, while keeping the other dataset parameters fixed. As a baseline for other datasets, 5 deg/s for the relative angular velocity has been chosen.

Image Resolution [HWRES]
For feature detection, the resolution of the images plays a significant role. When the object contains more pixels, feature detection will be easier. The maximum resolution is based on the resolution of the SPEED images, which have a resolution of 1920x1200 pixels [3]. The medium resolution is based on the images of the Hubble telescope by the space shuttle, which have a resolution of 1000x1000 pixels [20]. Since CubeSats generally have low processing power available, a lower resolution is also tested. This results in the following resolutions to be tested: 1920x1440, 1080x810, 860x635. All images have an aspect ratio of 4:3. As a baseline resolution, the maximum resolution of 1920x1440 will be used. This is motivated by the fact that it allows for finding the ultimate limitations of the pose estimator when it is tested on other parameters.

Frame Rate [HWFR]
The frame rate is of significant importance for multiple reasons. It is important for feature matching, but also for map initialisation in correlation with relative angular velocity. The higher the frame rate, the less change in between subsequent frames, meaning features can be more easily matched since the viewpoint difference is minimum. The upper frame rate is determined from running the algorithm in real-time on an Intel Core i7 9th gen processor. At an image resolution of 1920x1440px, the pose estimator can process frames at a rate of 0.066 seconds per frame. However, sometimes the algorithm required more time per frame, as is plotted in Figure C.14b. It should be noted that the performance of the processor has not been maximised during the execution of the algorithm. On average the algorithm consumed 25% of the CPU. As a result, the frame rate range has been set at 10, 5, and 1 frames per second.

Brightness [EEBR]
As brightness can influence the performance of the pose estimator, it is essential that this is correctly modelled. To take into account any uncertainty in the illlumination model, datasets are generated with 50% and 150% of the original lighting. The performance of the pose estimator is then compared to the original, after which conclusions can be drawn with respect to the validity of the brightness conditions. Finally, the lighting settings in Blender have been set to their minimum to simulate eclipse. Since no accurate verification method is obtained of lighting conditions in eclipse, real eclipse performance cannot be assessed. However, it is interesting to observe what brightness conditions are required for feature detection and matching.

Background [EEBG]
A dataset is created to test the robustness of the pose estimator to background objects. The background dataset consists of an Earth, as described in the previous section. The Earth comes in the scene at certain positions and orientations of the camera.

5.3. Lab-generated Dataset

This section describes the set-up and assumptions used that result in the generation of the lab-generated dataset.

5.3.1. Facility

The dataset is created in the CyberZoo at TU Delft. The Cyberzoo is shown in Figure 5.10. It is a 10x10 meter space packed with 12 infrared cameras for target tracking as visualised by the blue markers in Figure 5.9.

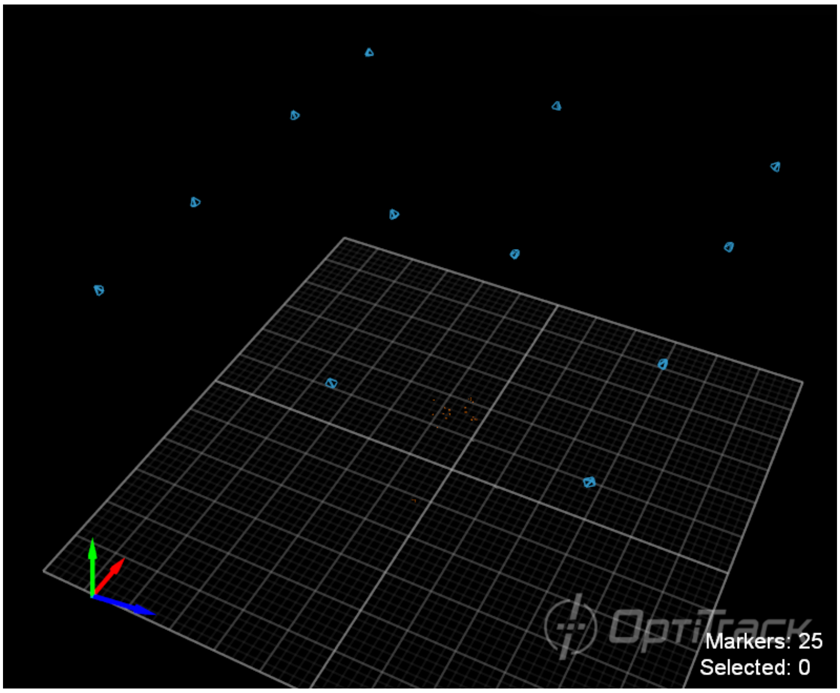


Figure 5.9: Graphical lay-out of the Cyberzoo. The blue markers represent the position of the IR cameras, the orange dots the reflective markers on the objects in the room

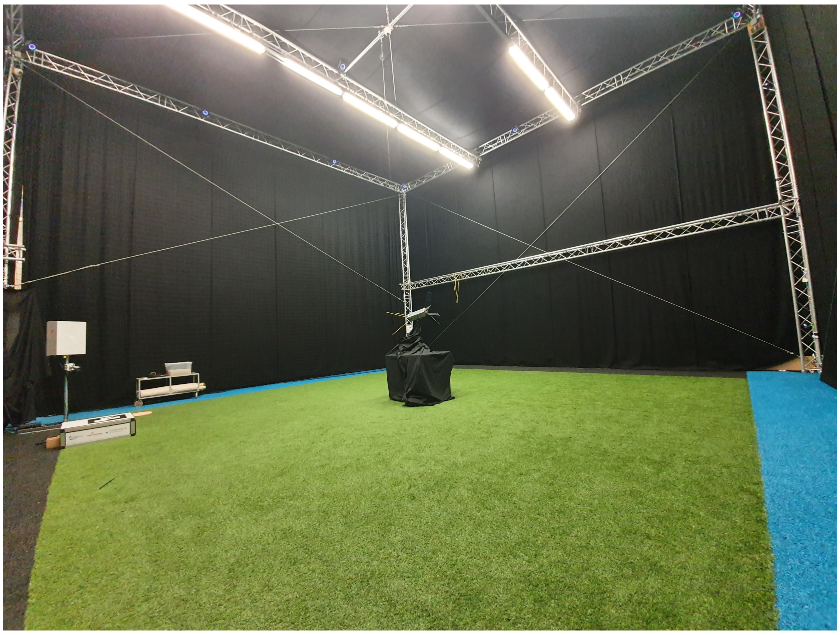


Figure 5.10: CyberZoo

The cameras send out infrared light, which is reflected by markers that are put on the tracked objects. The markers' positions are then tracked and stored at a set frequency. The accuracy of the ground truth is dependent on various factors, like the calibration, position of the markers, and software settings. The OptiTrack software that is used, states an accuracy of approximately 0.2 mm and 0.1 deg [54].

In order to simulate a space environment as accurate as possible, the lighting conditions have to be representative. In order to do this, a perfectly dark room would be required ideally. This room should contain lights that can represent the sunlight and earth albedo, as those are the main light sources for a satellite in orbit.

5.3.2. Set-up

The setup is described by the lay-out of the room, the light source, the target object, the camera, and trajectory.

Lay-out

The set-up is inspired by the facilities described in chapter 2. All the facilities are darkened by black paint or curtains. This is created in the CyberZoo by closing all curtains. However, not all objects in the room can be removed. These most notable objects are the metal bars that represent the frame of the room, as can be seen in Figure 5.10. Hence, no full control of lighting incident on the target spacecraft can be achieved. Therefore, it is assumed that the light reflected by these objects presents the Earth albedo, as the albedo of the Earth has a significant impact on the visibility of targets in orbit [51].

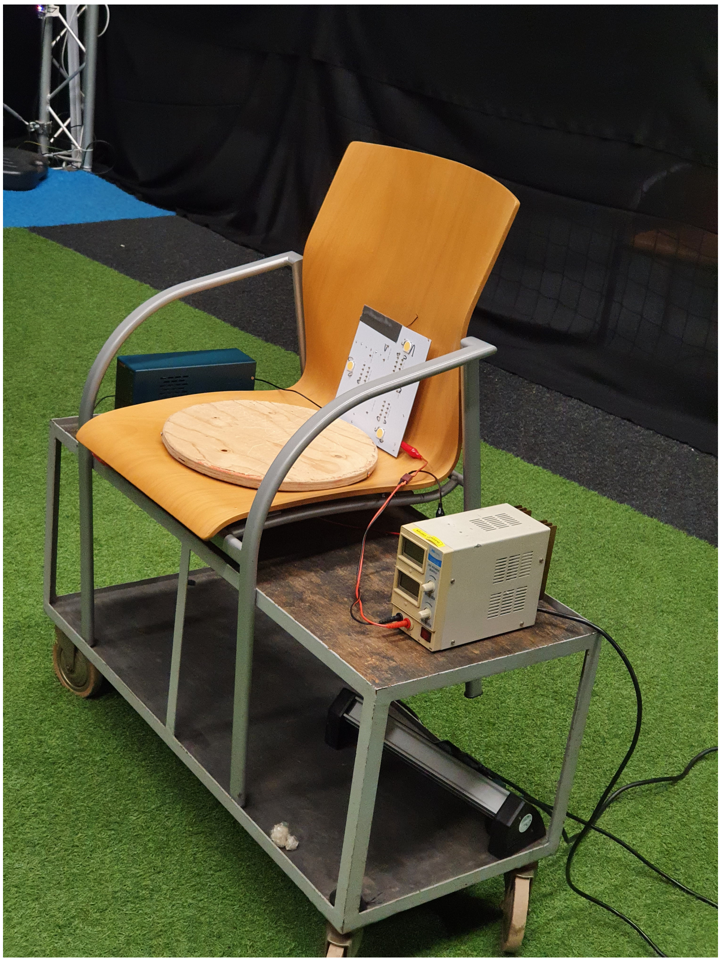


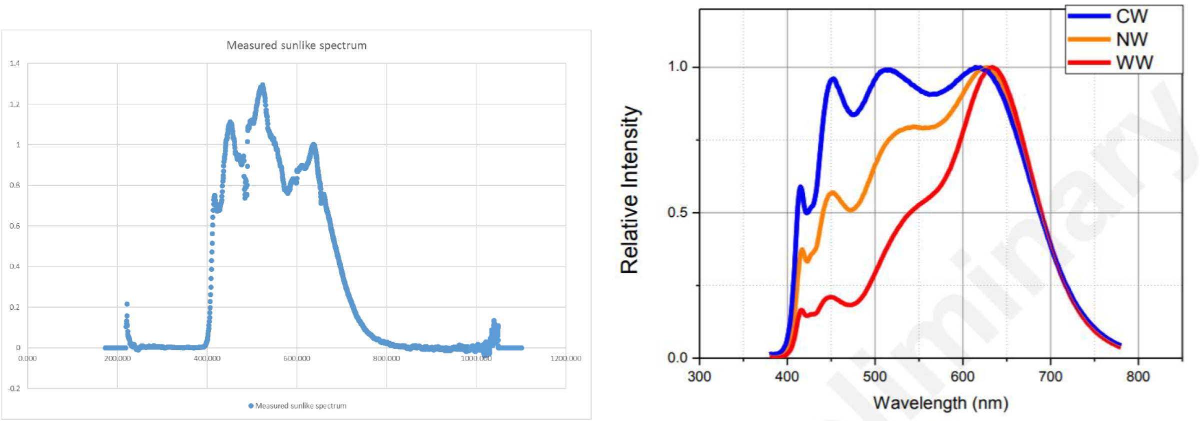
Figure 5.11: Solar Simulator Set-up.

Light Source

The light source is an in-house developed solar simulator that can exactly simulate the solar lighting conditions in space with the same intensity. The same simulator has been used to test the solar panels of the Delfi-PQ satellite. The radiation output of the solar simulator and the expected output are illustrated in Figure 5.12a and Figure 5.12b respectively [68].

Target

The target is a 1:1 mock-up of the Delfi-n3Xt. It is put on a stand, covered by black cloth to minimise the effect of feature detection on the stand.



(a) Measured spectrum of the solar simulator [68]

(b) Expected spectrum of the solar simulator [68]

Figure 5.12: Solar simulator spectrum

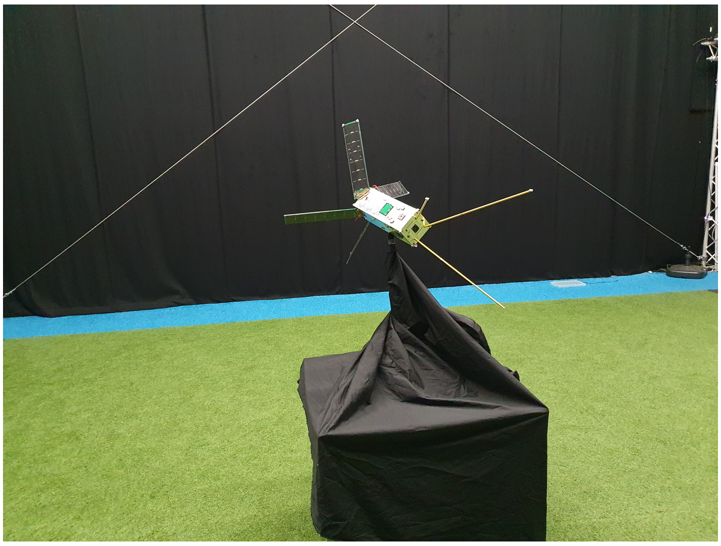


Figure 5.13: Delfi-n3Xt mock-up

Camera

The chaser spacecraft is represented by a phone camera on a stabilising gimbal. The gimbal is used to represent the relatively smooth motion of satellites in orbit. It is seen in the other facilities that the chasing system is also stabilised to mitigate shocks from movement. The camera used is a phone camera on the Samsung Galaxy S10. The specifications are depicted in Table 5.2. The lens is calibrated using chessboard calibration to obtain its distortion coefficients, defined as the intrinsic and extrinsic camera parameters. The resulting coefficients are shown in Table 5.3.

Table 5.2: Camera Settings

Resolution	960x540px
FOV	77 deg
Frame Rate	30 fps

Table 5.3: Extrinsic and intrinsic camera parameters

fx	752.3096133
fy	751.58763784
cx	476.49933235
cy	361.44730329
k1	3.48365615e-01
k2	-2.06872393e+00
p1	-7.74701027e-05
p2	1.34887826e-03

Trajectory

As this dataset functions as a complementary dataset to the synthetic ones, the trajectory does not have to be circular. In fact, it is desired to perform a different pseudo-random trajectory to test its robustness to unsteady motion. Initially, the trajectory will include a scanning motion around the target to obtain accurate pointcloud data. Afterwards, backward-forward motions are performed to simulate approaching scenarios where viewpoint changes are minimum.

5.3.3. Ground-Truth

Ground-truth data is obtained from the reflective markers placed on the camera and target. Markers are placed on all the corners of the target, after which the marker points are interpolated to create a dense pointcloud of the target. Additionally, three markers are placed on the camera, centred around the lens. The markers placed on the target and camera are seen from the software in Figure 5.14. The actual markers are visualised in Figure 5.15a and Figure 5.15b. The basis vectors of the camera and target frame are expressed in the cyberzoo frame, after which the required transformations are computed to obtain the ground truth pose for comparison.

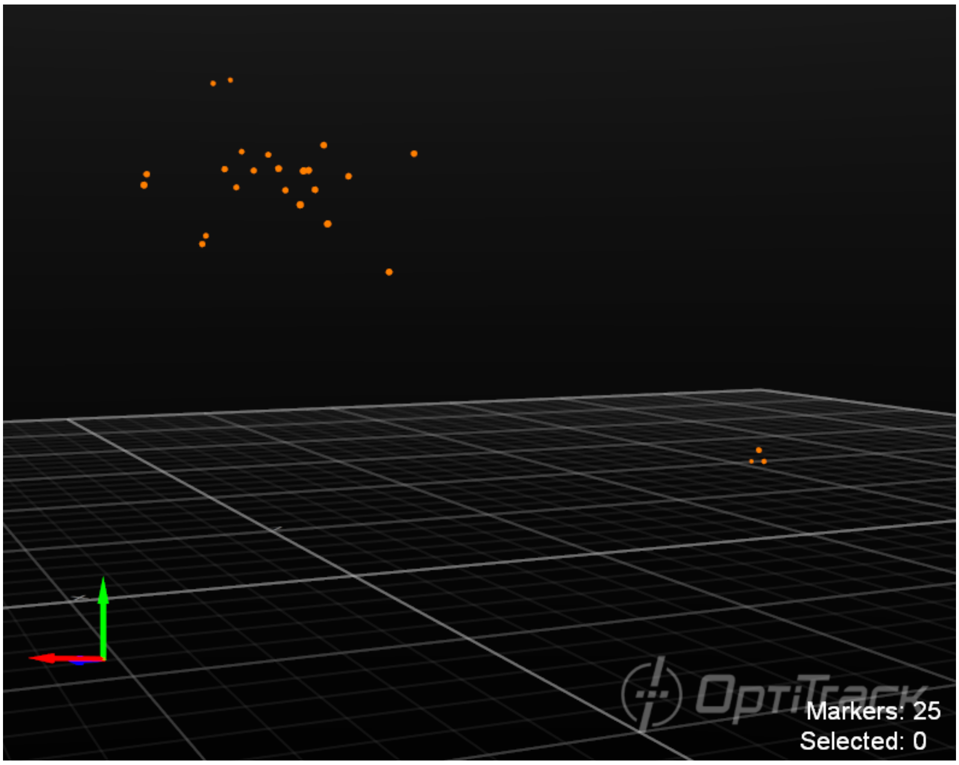
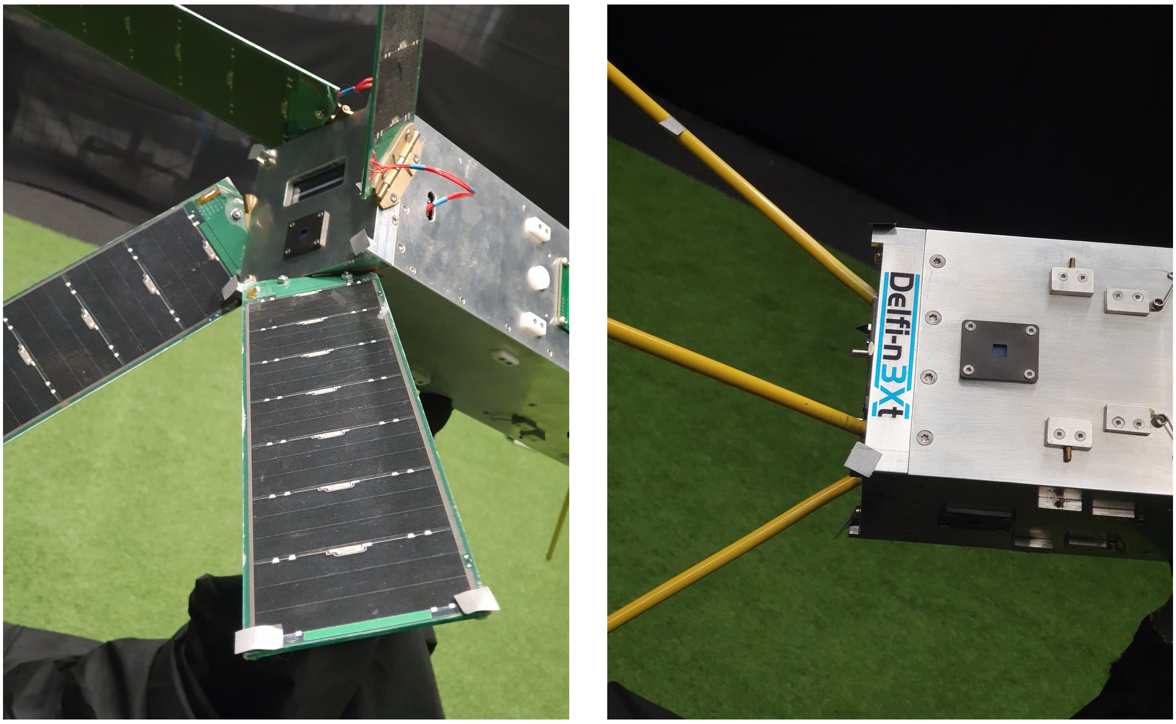


Figure 5.14: Markers visible by the software



(a) Markers on solar panel

(b) Markers on the central body

Figure 5.15: Reflective markers on the satellite

5.4. Conclusions

The datasets on which the designed pose estimation algorithm is assessed are created as part of the thesis. Both synthetically and lab-generated datasets are created. The synthetic datasets are made using Blender. The datasets capture various scenarios that will provide an analysis of the robustness of relative motion, background objects, illumination influence, and hardware limitations. The lab-generated dataset is used to demonstrate that feature matching and detection works on real, space-representative imagery. Furthermore, it provides a verification method for the realistic quality of the synthetically generated images.

III

Navigation System Evaluation

6

Results & Analyses

This chapter analyses the results obtained in this research. Firstly, an overview of the obtained results is outlined in section 6.1. Subsequently, section 6.2, section 6.3, and section 6.4 analyse the results with respect to the environmental influence, hardware limitations, and relative motion, respectively. Afterwards, the practical results, filter integration, and effects of the algorithm modifications are evaluated in section 6.5, section 6.6, and section 6.7, respectively. At the end of this chapter, a conclusion is drawn on the suitability of monocular SLAM for spacecraft relative navigation in section 6.8.

6.1. Overview of Results

This section provides an overview of the obtained results in each test scenario. The results are displayed in Table 6.1. The accuracy values are all root mean squared errors. This is deemed the most representable accuracy measurement over a period of time. Furthermore, the accuracy values are taken over the second half of the datapoints. This is because the initial pose estimate can be off significantly and converges over time to a stable constant accuracy.

Table 6.1: Results

Scenario	**Ang Vel**	**Resolution**	**Frame rate**	**Brightness**	**Background**	**Accuracy**			
	[deg/s]	**[px]**	**[Hz]**	**[%]**	**[-]**	**Position** **[m]**	**Orientation** **[deg]**	**Velocity** **[m/s]**	**Ang Vel** **[deg/s]**
RMVELhigh	10	1920x1440	10	100	no	0.014	0.298	0.002	0.082
RMVELmed	5	1920x1440	10	100	no	0.026	0.322	0.002	0.072
RMVELlow	1	1920x1440	10	100	no	0.046	1.221	0.002	0.494
HWRESmed	5	1080x810	10	100	no	0.062	0.265	0.004	0.203
HWRESlow	5	860x645	10	100	no	0.084	0.469	0.005	0.299
HWFRmed	5	1920x1440	5	100	no	0.023	0.442	0.002	0.086
HWFRlow	5	1920x1440	1	100	no	-	-	-	-
EEBRhigh	5	1920x1440	10	150	no	0.077	0.186	0.002	0.074
EEBRlow	5	1920x1440	10	50	no	0.033	0.196	0.002	0.077
EEbackground	5	1920x1440	10	100	yes	0.027	0.230	0.002	0.071
EEeclipse	5	1920x1440	10	1	no	0.301	0.483	0.002	0.087
VEL1_FR1	1	1920x1440	1	100	no	0.053	0.146	0.002	0.062

6.1.1. Accuracy Uncertainty

It should be noted that the accuracy might not be exact due to the method of accuracy determination. This is because there is an uncertainty in the scale and transformation matrix obtained from CPD. A slight adjustment in the scale significantly affects the accuracy. The CPD algorithm lines up the true and estimated pointcloud to its best capability, but a certain uncertainty will always be present. Therefore, slight changes in true position and orientation are expected among the various tests. As the velocity and angular velocity are directly obtained from the derivation of the pose and its preceding pose estimate, they are a good indication of the accuracy as well. This is because the propagation accuracy between the pose at subsequent frames is to a certain extent independent of the scaling. This is also seen in the results as position and orientation accuracy can be inconsistent, but the derivatives are relatively consistent throughout all scenarios. A good illustration of this is shown in Figure 6.1, where the position is overlaid. There is a translational offset in the z-axis, but the propagation of the pose looks very accurate. This can indicate that the transformation is not perfect. Hence in the analysis, the derivatives should also be taken into account due to the nature of their estimation method and to obtain valid conclusions.

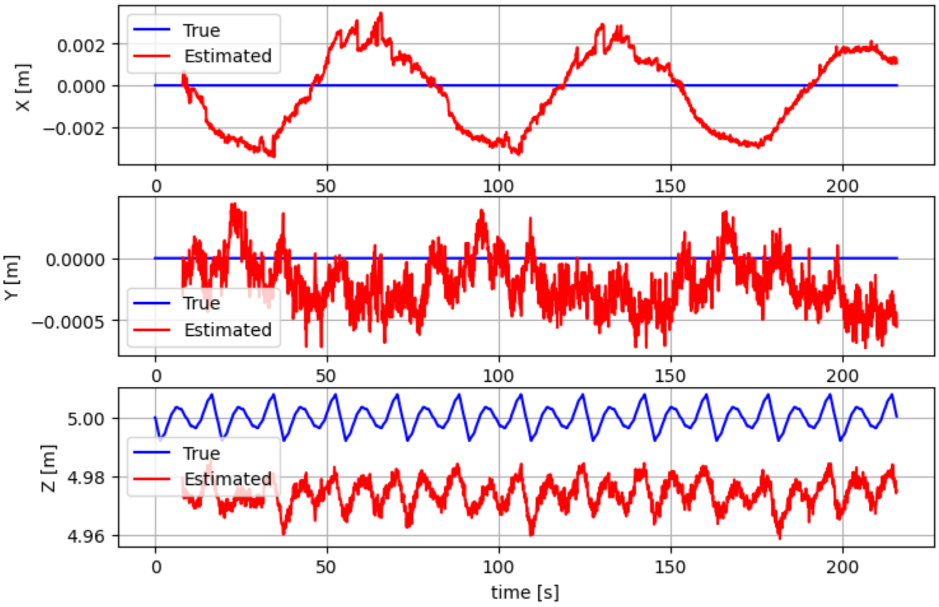


Figure 6.1: Pose propagation accuracy and absolute pose accuracy

6.1.2. Pose Results Characteristics

There are a few characteristics of the obtained plots in Appendix C that stand out. One can notice that the results are not smooth but noisy. This has to do with the fact that these are raw measurements. This is a natural phenomenon in unfiltered raw measurements, since, at each timestep, the pose is measured (semi-)independent of the previous measurements, which results in a non-smooth propagation. The size of the spikes is also an indication of the uncertainty in the pose estimate. This can be seen as the *noise* on the true accuracy. The larger the uncertainty, the greater the spikes are.

Furthermore, one can see some cyclic behaviour. This is caused by the fact that the relative circular trajectory is travelled three times by the chaser satellite. Another factor that causes this cyclic behaviour is related to the non-perfect transformation using the CPD algorithm. If the transformation is slightly inaccurate, the generated target frame might not be perfectly concentric with the focus point of the circle, inducing variable cyclic behaviour of the relative distance and orientation plots.

Finally, it is observed that the initial pose estimate error is relatively high, after which it converges to an approximately constant and steady error. This is related to the optimisation-based behaviour of the algorithm. With every new insertion of a keyframe, an optimisation of the pose is performed, resulting in convergence of the accuracy. At the initialisation of the pose, the pose is estimated based on very few keyframes. With every addition of a keyframe, the map and pose are optimised.

6.2. Environmental Influence

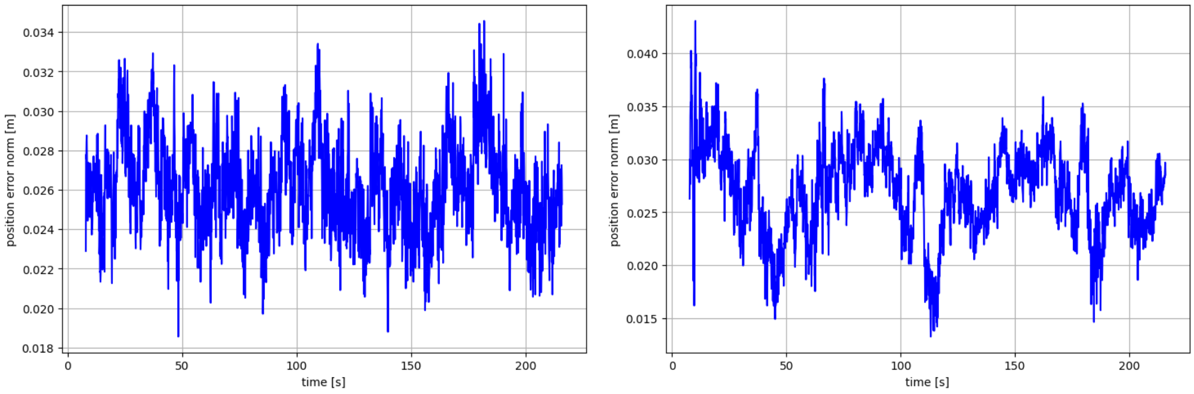
Regarding the influence of environmental factors like background objects and illumination conditions, one needs to consider scenarios RMVELmed, EEBRhigh, EEBRlow, EEbackground and EEeclipse. The results are summarised in Table 6.2.

Table 6.2: Results Environmental Influence

Scenario	**Accuracy**			
	Position [m]	**Orientation [deg]**	**Velocity [m/s]**	**Ang Vel [deg/s]**
RMVELmed	0.026	0.322	0.002	0.072
EEBRhigh	0.077	0.186	0.002	0.074
EEBRlow	0.033	0.196	0.002	0.077
EEbackground	0.027	0.230	0.002	0.071
EEeclipse	0.301	0.483	0.002	0.087

6.2.1. Background

To be robust to background objects, EEbackground is a dataset containing a rotating Earth in the background which is dynamic with respect to the chaser and target frame. This is done to check the robustness to feature detection on the Earth. It is important that the Earth is dynamic with respect to the camera frame so that features are matched over subsequent frames. If the background would be static in the camera frame, no change in viewpoint is observed and thus no features can be stored. The purpose of this dataset is to test the background robustness of the algorithm. In this simulation, the threshold for far points has been set at 50 meters. This is a safe distance as all feature points of the target are well within that distance, while the Earth exceeds that distance. The position and orientation error of the simulations with and without background object are shown in Figure 6.2 and Figure 6.3.



(a) RMVEL5 position error norm

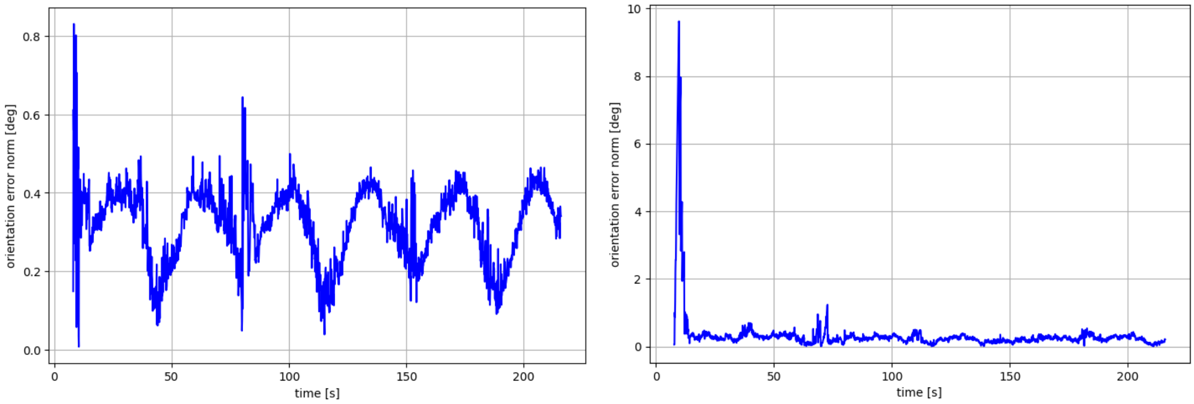
(b) EEbg position error norm

Figure 6.2: Position Error Norm comparison between RMVEL5 and EEbg

The Earth is in frame in the following periods. From 57 to 70 s, from 129 to 142 s, and from 201 to 214 seconds. It can be noticed that the initial pose estimate of the background scenario is not very accurate. However, this is unrelated to the fact that that dataset contains the Earth in the background, as the Earth does not come in the frame until 57 seconds into the simulation. This is just the exact reason why the accuracy is measured over the second part of the simulation, as pose initialisation needs time to converge to stable pose estimates. Apart from a little disruption in the background orientation graph at around 70 seconds, no abnormal behaviour is noted, meaning the background robustness implementation is shown to work. Another visualisation of the working of the background elimination algorithm is shown in Figure 6.4a and Figure 6.4b.

6.2.2. Illumination Conditions

When comparing the brightness scenarios, it is observed that the propagation of the pose between EEBRhigh, EEBRlow and RMVELmed (as baseline scenario) are almost identical. This shows on one hand the robustness of the algorithm to illumination variations, but on the other hand also that highly accurate illumination conditions in the input data do not significantly benefit the validity of the performance assessment. Of course,



(a) RMVEL5 orientation error norm

(b) EEbg orientation error norm

Figure 6.3: Orientation error norm comparison between RMVEL5 and EEbg

this is under the condition that the illumination conditions are approximately representative of the real illumination in space. The verification of the lighting conditions has been discussed in chapter 5. Acquiring the knowledge that the algorithm is robust to certain brightness variations, also contributes to the validity of the simulated results.

In eclipse, when the illumination is very low, significant differences can be observed in terms of pose. However, in the eclipse simulation the imagery has not been compared, meaning that the realistic quality of the orbital scenario is not assessed. This has not been done as no accurate comparison is available. The lighting conditions have just been set to minimum in Blender. The main conclusion that can be drawn from the eclipse scenario is that under very low lighting power, when the target is still visible, initialisation and tracking can be achieved.

6.3. Hardware Limitations

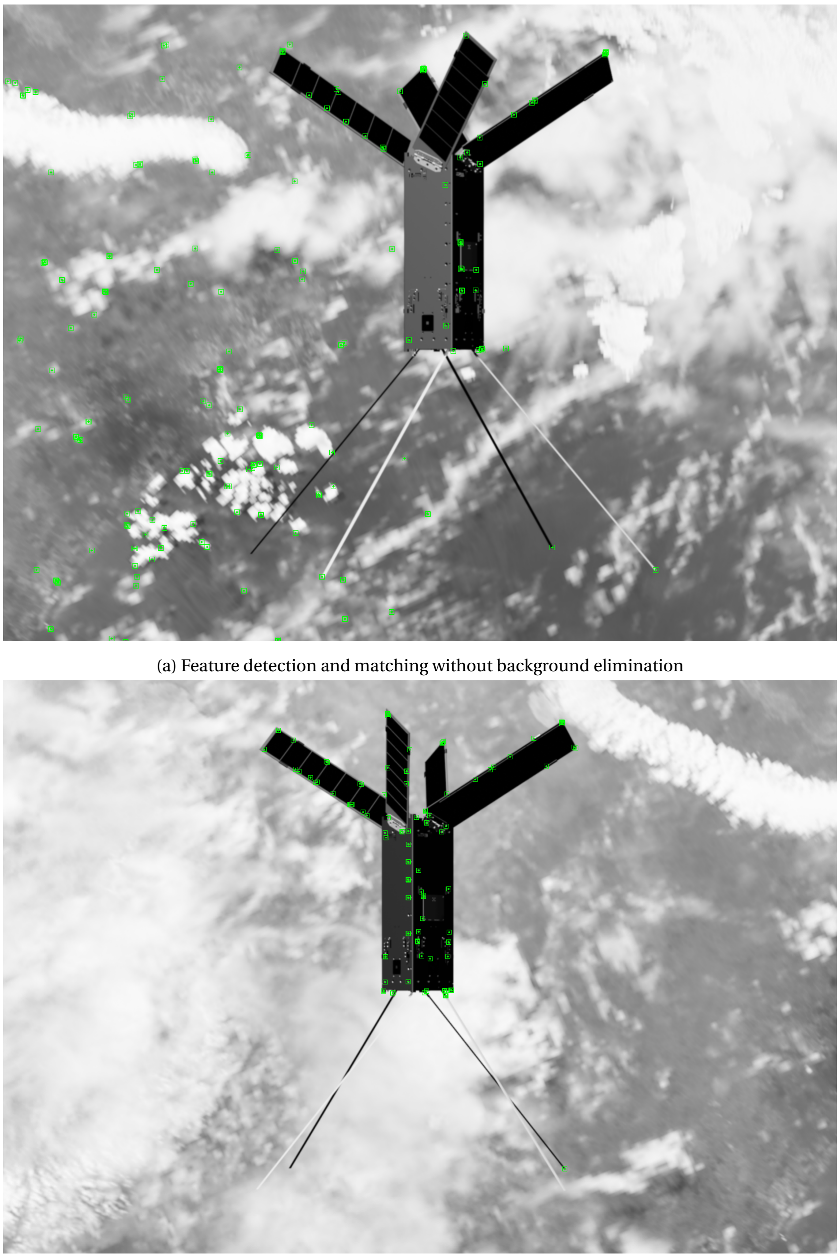
The hardware results are analysed in terms of frame rate and resolution. The results are shown in Table 6.3.

6.3.1. Resolution

From the simulations, it is noted that the resolution significantly affects the performance. When the resolution decreases, the accuracy drops as well. However, the accuracy is not dropped too significant, since the relative pose accuracy is still obtained within an accuracy of a few centimetres and sub-degree level. With the current noise model on the images and relative motion scenario, resolutions lower than 800x600 pixels yield no results due to instabilities of the pose. Various lower resolutions have been tried, but no stable results could be obtained. This shows that resolution is an important factor for pose estimation of noisy space imagery. Furthermore, that the accuracy does not drop significantly has to do with the nature of SLAM, SLAM needs a certain accuracy in their map, as map points are matched across frames by reprojecting them in the current frame. When this reprojection is highly inaccurate, matching becomes inaccurate as well, and thus unstable behaviour of the SLAM algorithm is observed. This indicates that the accuracy does not linearly drop with resolution, but a stable pose estimate simply cannot be obtained when the resolution is too low. It is likely that for detailed pose estimation design for certain missions and applications, stable pose estimates at lower resolutions could be achieved, but since this work is mainly a global research on the suitability of SLAM for space applications, the perfect optimisation of SLAM algorithms is deemed outside the scope of the research.

Table 6.3: Results Resolution Influence

Scenario	Accuracy			
	Position [m]	**Orientation [deg]**	**Velocity [m/s]**	**Ang Vel [deg/s]**
RMVELmed	0.026	0.322	0.002	0.072
HWRESmed	0.062	0.265	0.004	0.203
HWRESlow	0.084	0.469	0.005	0.299



(a) Feature detection and matching without background elimination

(b) Feature detection and matching with background elimination

Figure 6.4: Effect of background elimination

6.3.2. Frame Rate

Frame rate and relative motion are correlated. The higher the frame rate or the lower the angular velocity, the less the target displaces between frames. Hence, the frame rate has been analysed with a constant relative motion to visualise the effect more clearly. However, a combined conclusion should be drawn with the analysis of relative motion in the next section. It is observed that for the scenario of 1 frame per second at a relative angular velocity of 5 deg/s, no initialisation could be achieved. This has to do with the fact that the relative motion is too high (parallax is too big) for ORB features to be matched. ORB has a certain invariance to view point, but only up to a certain degree. It is clear that in this case, that was out of its limits. Therefore, it is interesting to investigate, in this application of spacecraft relative navigation, what the maximum degree of relative rotation of the target with respect to the camera per frame is. For RMVELhigh and HWFRmed, this is one degree per frame. To investigate if this is still valid for a frame rate of 1 frame per second, an extra dataset has been generated, VEL1FR1, where the relative velocity is 1 deg/s, and the frame rate is 1 fps. This simulation has also successfully shown to achieve stable pose estimates. The other test scenarios all implement rotation per frame of either 0.1 deg/frame in RMVELlow, or 0.5 deg/frame in the rest of the scenarios. Additional tries have been made to achieve a larger rotation than 1 deg/frame, but without success. Therefore, one can conclude that in order to apply monocular SLAM for spacecraft relative navigation, orientation propagation of the target between frames exceeding 1 deg/frame might result in unstable pose output. This is an important finding and indication of the limitations of monocular SLAM for spacecraft relative navigation, as the state of the target is typically unknown for uncooperative pose estimation.

Table 6.4: Results Frame Rate Influence

Scenario	**Accuracy**			
	Position [m]	**Orientation [deg]**	**Velocity [m/s]**	**Ang Vel [deg/s]**
RMVELmed	0.026	0.322	0.002	0.072
HWFRmed	0.023	0.442	0.002	0.086
HWFRlow	-	-	-	-
VEL1FR1	0.053	0.146	0.002	0.062

6.4. Robustness to Relative Motion

The general range in which the pose estimation should work has been tested in RMVELhigh, RMVELmed, and RMVELlow. It was able to achieve pose estimation on all simulations. Initially, initialisation could not be achieved for RMVELlow as the parallax between subsequent frames was too low, but after adaption of the algorithm to use more neighbouring keyframes for feature matching, this was achieved. Even though all scenarios produce stable pose estimates, thus demonstrating the robustness of the algorithm, it is observed that the accuracy drops with the relative angular velocity. This is strongly related to the achievement of parallax between frames. If there is not much movement between frames, the uncertainty in the pose estimate increases due to the relatively small viewpoint change. This shows that the algorithm performs better in the range of 0.5 to 1 deg/frame. Further conclusions of this section tie in to the frame rate conclusions, as a higher frame rate is required for higher relative angular velocities.

Table 6.5: Results relative motion influence

Scenario	**Accuracy**			
	Position [m]	**Orientation [deg]**	**Velocity [m/s]**	**Ang Vel [deg/s]**
RMVELhigh	0.014	0.298	0.002	0.082
RMVELmed	0.026	0.322	0.002	0.072
RMVELlow	0.046	1.221	0.002	0.494

6.5. Practical Experiment

The following objectives of the practical experiment were identified in chapter 5:

1. Verification of realistic quality for synthetic images.
2. Verification of feature detection and matching on real-life, space-like imagery

The first primary objective has already been provided in the previous chapter. Due to the setting in which the lab-generated images are taken, the use of realistic space illumination and a real mock-up satellite model, the realistic quality has been verified. The other objective is discussed in this section.

The 2D trajectory of the camera in the Cyberzoo reference frame as seen from above is visualised in Figure 6.5. The target is located in the origin and centre of the plot. As can be observed, first a half-circle is made around the target, after which back and forth motion is performed. It is noticed from Figure 6.6 that constant tracking of the target has been achieved, with stable pose estimates, since a pointcloud of the target could be constructed for rigid registration. This indicates that verification of feature detection and matching on real-life space-like imagery has been achieved. However, pose accuracy is significantly off. This could be explained by the difficulties experienced in obtaining ground-truth values.

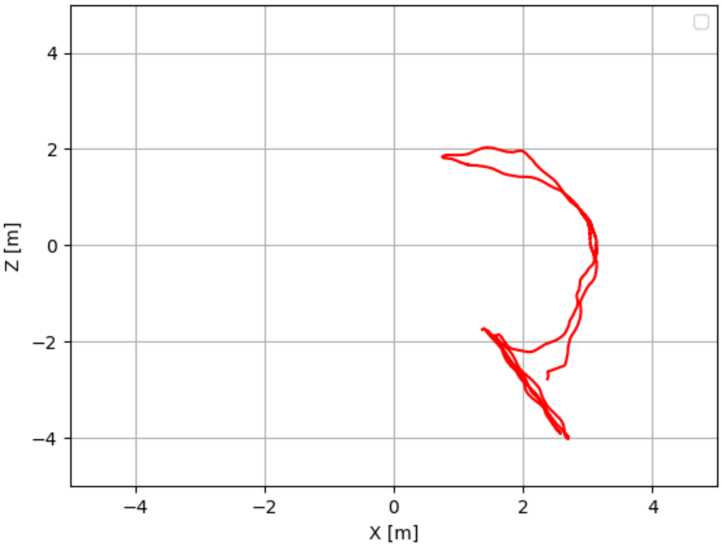


Figure 6.5: Trajectory of the camera with the target in the origin

Firstly, marker placement on the target induces inaccuracies. Since, the distance between markers on the camera is very low, a few millimetres offset in marker alignment already causes significant errors in true pose, especially orientation. Furthermore, it was found that markers have not been constantly tracked. This is caused by the fact that the tracking cameras are mounted on the ceiling, and the camera is pointed sideways, making it difficult for the markers to always be observable by the cameras. Furthermore, it was noted that markers on the camera were sometimes fused and seen as one marker. The data has been interpolated between gaps in the true pose measurements, but this also imposes inaccuracies. The most important conclusion with respect to the practical experiment is that verification of feature detection and matching has been achieved under all circumstances. The relative motion was varied, and constant-viewpoint manoeuvres have been performed, simulating spaceraft approach scenarios. Thereby proving the robustness of the algorithm under various relative motion.

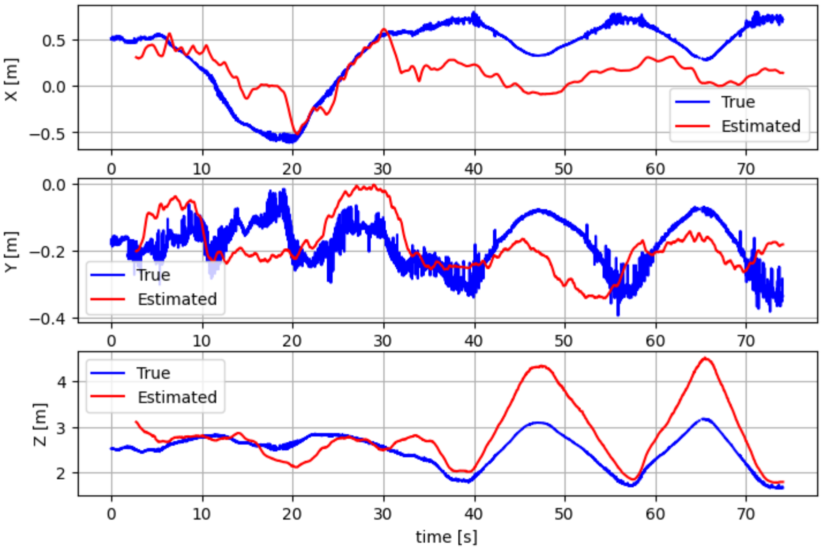


Figure 6.6: cyberzoo inaccurate position overlay

In the cyberzoo, a half-circular path was walked back and forth around the target, after which several back and forth motions to and from the target have been done, simulating v- and r-bar approaches. During all these scenarios, target tracking has not been lost. Also, from the figure it is noticed that there are no hick-ups in the pose estimation, indicating a smooth output and thus stable estimation. Therefore, feature tracking under nonconstant motion has also been achieved, which is important for robustness when the relative motion is not constant in real scenarios. Furthermore, as there are still detectable objects in the background, the background robustness algorithm can be tested and verified again. Since no instabilities are observed, this has been deemed achieved.

6.6. Filter Integration

This section demonstrates the feasibility of the integration of the pose estimator with a filter. By comparing the integration with the filter and solely pose estimator output, one can see that the integration is done successfully. The translational aspect of the state significantly decreases in error, while the orientation and its derivative stay the same. The angular velocity measurements from the pose estimator are just used as measurements, and not propagated by the filter. The angular velocity measurements are defined as the difference in orientation between two subsequent frames divided by the time difference the frames were taken. Since the state filter propagates the orientation in the exact same way, the same orientation error is achieved. It should be highlighted that the method of angular velocity determination is a simple approach and merely implemented for proof of integration of a state filter with the pose estimator. More sophisticated approaches have been demonstrated in literature, like deriving the angular velocity from optical flow [13]. In any case, the designed filter can output state estimates at the frequency desired by the GNC system, even when no pose measurements are available.

Table 6.6: Results Filter Integration

Scenario	**Accuracy**			
	Position [m]	**Orientation [deg]**	**Velocity [m/s]**	**Ang Vel [deg/s]**
RMVELmed	0.026	0.322	0.002	0.072
with filter	0.004	0.322	0.001	0.072

6.7. Effects of Algorithm Optimisation

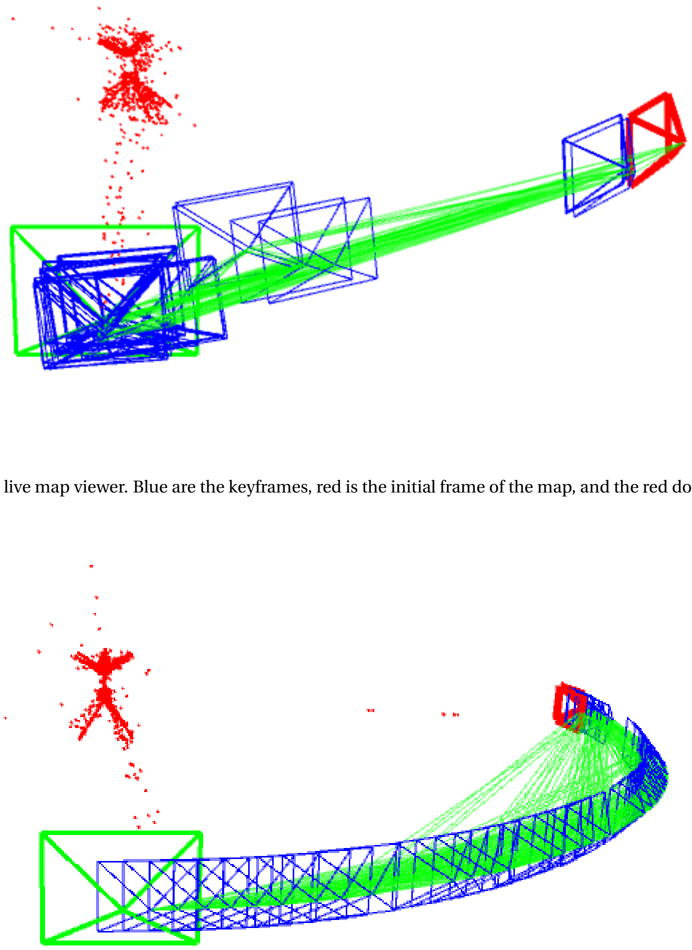
It is important that the effect of the modifications on the algorithm is highlighted, which is done in this section.

6.7.1. Robustness to Background Objects

The effect of the background elimination system algorithm is visualised in Figure 6.4a, Figure 6.4b, as well as Figure 6.7a and Figure 6.7b. It was noted that when the Earth came in frame, unstable behaviour of the pose estimator was observed. This resulted in an inaccurate map of the target, for which no accuracy comparison could be made due to the inaccuracy of the map points when used in CPD. Hence, the effect is best observed in the live map viewer during running of the algorithm, visualised in Figure 6.7a and Figure 6.7b. As the camera runs in a circle relative to the target, the key frames should be lined up along that circle, as is shown in Figure 6.7b. Unstable behaviour is observed when these keyframes are not lined up in order any more but cluttered up on top of each other. This is the result of dynamic environments in the camera frame.

6.7.2. Robustness to Low Parallax Pose Initialisation

VEL1FR10 did not initialise since not enough parallax was achieved between frames. Consequently, the neighbouring keyframes number has been increased to extend the parallax. With such high frame rate and low relative velocity, keyframes are very close to each other. Hence for initialisation, keyframes further away are desired. By increasing the number of neighbouring keyframes, initialisation has been achieved. This is displayed in Figure 6.8.



(a) Unstable behaviour in the live map viewer. Blue are the keyframes, red is the initial frame of the map, and the red dots are the map points.

(b) Stable behaviour in the live map viewer. Blue are the keyframes, red is the initial frame of the map, and the red dots are the map points.

Figure 6.7: Unstable vs stable behaviour

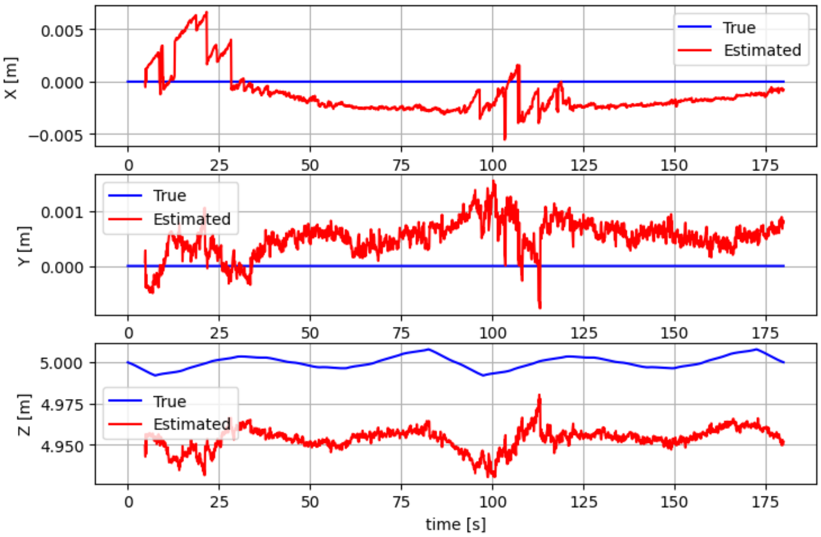
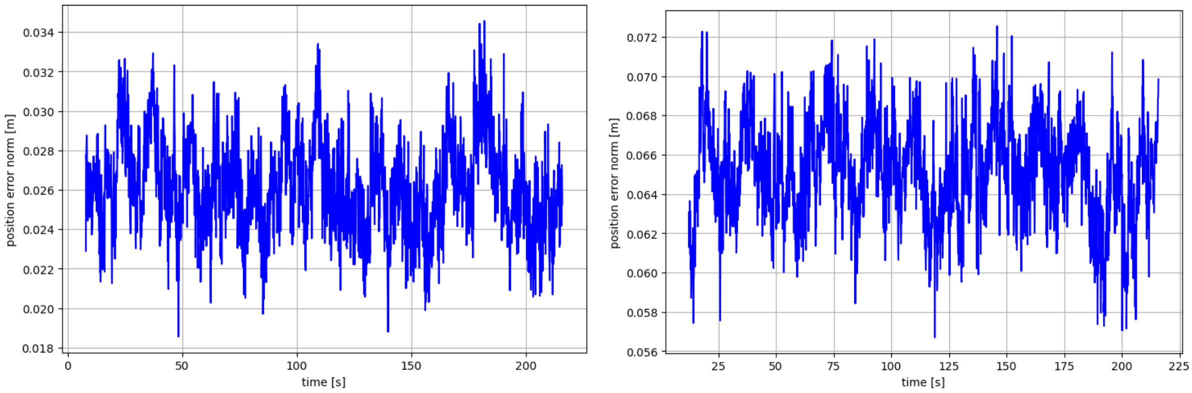


Figure 6.8: VElFR10 with initialisation successful just after 5 seconds

6.7.3. Space Imagery Optimisation

The difference in the effect of the feature descriptor matching threshold is visualised when running two different versions of the algorithm on the same dataset. This is shown in Figure 6.9a and Figure 6.9b. The plot in Figure 6.9a shows the result when the threshold is set at 40, the other when the threshold is set at 60. The difference is significant. The lower threshold version has an accuracy that oscillates around 0.026 m, the higher threshold version around 0.065 m. It was noted that a distance below 30 resulted in some occasions in a low amount of feature detection and high initialisation phase. Thus, 40 has been determined as optimum.



(a) RMVEL5 position error norm when descriptor distance threshold set at 40 (b) EEbg position error norm when descriptor distance threshold set at 60

Figure 6.9: Effect of descriptor distance threshold

6.8. Conclusions

With the results obtained in this section, conclusions need to be drawn with respect to the research questions.

From the results, it is concluded that resolution has a significant impact on robustness. Stable and accurate pose estimation has been achieved for resolutions higher than 800x600 pixels. However, lowering the resolution further resulted in unsuccessful pose estimation. Additionally, a limitation is observed related to relative motion and frame rate. When the orientation propagation of the target between frames exceeds 1 deg/frame, pose initialisation cannot be obtained. An ideal range in terms of accuracy is found for a propagation between 0.5 and 1 deg/frame. This is the range that allows for sufficient parallax between frames that significantly benefits the accuracy. Lower propagation levels also succeed, but it is noticed that the accuracy drops slightly. These results impose limitations on the space system encompassing the processor and monocular sensor. It is important that the on-board processor can process these resolutions at a sufficient frame rate.

Additionally, it is found that the brightness of the target does not significantly affect the performance of the algorithm. This demonstrates the robustness of the pose estimator to variations in illumination conditions, but also increases the validity of the results, since an extremely accurate illumination representation is not required for the dataset generation.

Furthermore, robustness to background objects has been achieved by implementing an algorithm that discards features further than a threshold distance. Also, it has been demonstrated that initialisation of very low parallax motion can be achieved.

Moreover, the lab-generated dataset showed the functionality of the algorithm on real space-representative imagery, under unsteady relative motion. The pose estimator has been successfully integrated in a state estimator that does not require any knowledge of the target. This is achieved by excluding the angular velocity from the state vector, and directly using the angular velocity as measurements for the filter algorithm.

On average this system achieved an accuracy of sub-degree level on orientation, and cm-level on relative position. In related literature, Capuano et al.[13] achieved an accuracy of sub 1 degree and 0.3 meters. Jin et al. [33] obtained accuracies of sub 1 degree as well and 0.1 meters. This means that the pose estimation system used in this work has similar, if not better, accuracy than the state-of-the-art.

The suitability of the pose estimator for spacecraft relative navigation is assessed on three factors. Firstly, achievement of robustness to orbital relative navigation conditions is required. This includes relative motion, background, space image quality, and illumination robustness. The robustness to these factors is demonstrated in this chapter. The second factor is based on the capability of this system to work in real-time on space hardware. In this work, it is found that a lower limit of 800x600 pixels on image resolution guarantees stable performance of the pose estimator. Additionally, The frame rate should be high enough such that the target's orientation propagation between frames does not exceed 1 deg/frame. When observing hardware used on VBN systems, it is noted that the RNS camera in [49] could capture resolutions of 1024x1024px at a frame rate of 3 Hz. Based on the fact that the design of that system was developed more than ten years ago, and the fact that technology advances significantly over the years, it is deemed feasible that the algorithm could be implemented on space hardware. The third and last point is related to the achievable accuracy. On average, this system achieved an accuracy of sub-degree level in orientation, and centimetre-level in relative position. For particular mission phases of RPO, strict accuracy requirements need to be met. A reference example of requirements set for the e.Deorbit mission is found in [59]. They state accuracies for the closing phase (50-100m) of sub-5 deg and sub-10m, and sub-2 deg and sub-0.05 m for the sync and capture phase (0-50m). Additionally, during Servicing Mission 4 of the Hubble telescope [49], accuracies were required of sub 1 deg and sub 0.1m for the most critical phase (below 5 meters). Since the required accuracies of both the e.Deorbit mission and Servicing Mission 4 have been achieved in this work, the designed pose estimation system is deemed suitable for spacecraft relative navigation.

IV

Closure

7

Conclusions & Recommendations

7.1. Conclusions

This work researched the suitability of monocular-SLAM for spacecraft relative navigation to unknown targets. The motivation for this work is based on the rapidly increasing objects in orbit, of which the majority is classified as debris or non-operational satellites. To sustain a healthy space environment, feasible and cost-effective ADR and OOS solutions are required that can be implemented on a wide scale. Ideal ADR and OOS solutions require no prior information of the target. This research aims to support development in that field by investigating the use of monocular-SLAM for pose estimation around unknown targets. By using monocular cameras for relative navigation, cost-effective ADR and OOS can be achieved on a global scale. Monocular cameras have the advantage that they have low cost, mass, power and hardware complexity, compared to active sensors like LIDARs. The challenge of monocular cameras is that space imagery is characterised by adverse illumination conditions, high noise and contrast.

This thesis is centred around the research question:

How can monocular-based pose estimation be achieved to enable relative navigation towards an unknown target in space?

This question has been divided into sub-questions to structurally guide the thesis. The conclusions to each subquestion are summarised here.

RQ-1: What monocular-based pose estimation algorithm is most suitable for spacecraft relative navigation towards unknown targets ?

RQ-1a: What criteria define the suitability of a pose estimation system for space applications?
Due to the inherent complexity and high cost of space missions, robustness of space systems is of major importance to achieve mission success. Hence, the suitability of a pose estimation system is strongly related to its robustness under various factors. A pose estimator is deemed suitable for spacecraft relative navigation if; it is robust to imagery with high noise and contrast (i), it is robust to adverse and changing illumination conditions (ii), in dynamic environments (iii), and to loss of pose tracking (iv).

RQ-1b: What are the current approaches to enable pose estimation in unknown environments?
First and foremost, it should be noted that the area of research is still of considerable novelty, meaning that not much related works are available. The general approach to pose estimation in unknown environments is centred around SLAM algorithms. Within the field of SLAM, various methods are used in the state-of-the-art. The majority makes use of feature-based algorithms, which provide robustness and invariance to illumination, scale, and view-point changes. The pose estimation is then generally performed by filter-based or optimisation-based algorithms. Both types have been used in literature.

RQ-1c: How can the scale of the environment be recovered?
Since monocular-only SLAM produces a scale-ambiguous pose estimate, a secondary sensor is needed. In the case that the target is fully unknown, solely depth sensors are suitable. In the literature, this secondary sensor is assumed to be a low-power single-beam LIDAR. If the target is partially known, various methods can be used to obtain scale estimates. An example that is demonstrated in this work is based on the CPD algorithm.

RQ-1d: How can the pose estimation pipeline be integrated in a state estimator for target state tracking?
In this work, a loosely-coupled approach is used to integrate the pose estimator with a state estimator. Conventionally, state estimating is done by (Kalman) filters. In this research, a simple EKF is implemented. Kalman filters make use of a dynamic model to propagate the state to the new time step. However, this dynamic model requires information about the target. The unknown properties of the target then impose large uncertainties on the filter estimate. To improve robustness and prevent divergence of the filter, the angular velocity is not included in the state vector, but directly used as a measurement from the pose estimator.

With the support of the above sub-questions, a pose estimator is designed based on the ORB-SLAM algorithm developed by Campos Martínez et al.[12]. The algorithm is adapted to improve the suitability for spacecraft relative navigation. Background robustness has been achieved by defining a distance threshold for feature storing. Features that are located further than a predefined threshold will be discarded, thereby separating fore- and background features. Additionally, pose initialisation has been optimised by increasing the robustness under low-parallax conditions, while maintaining real-time performance. Finally, the feature descriptor distance threshold for feature matching has been tuned to improve the accuracy of the algorithm for spacecraft relative navigation purposes.

RQ-2: How does the designed pose estimation algorithm perform?

RQ-2a: What test scenarios are required to assess the suitability for spacecraft relative navigation?
To assess the suitability for spacecraft relative navigation, the performance of the algorithm is tested on datasets containing imagery characterised by high contrast and noise to represent real space imagery. Relative motion robustness, robustness to brightness and illumination changes, robustness to background objects, and robustness to hardware specifications are tested. These tests provide conclusive insights to analyse the suitability of the adapted algorithm.

RQ-2b: How can the tests be realised?
The datasets are self-generated. This omits the problem of general unavailability of real spacecraft relative navigation footage, and allows for full control of the test scenario parameters. Both synthetic and lab-generated datasets are created. The synthetic datasets, generated in Blender, are mainly exploited for analyses of the robustness and suitability of the pose estimator, since realistic relative motion can be captured and accurate ground-truth data is available. The lab-generated dataset provides a method of verification of the realistic quality of the synthetic datasets, and proves the functionality of the algorithm on real space-representative footage.

RQ-2c: How can the realistic quality of the tests be verified?
Since the generated datasets are not real space imagery, it is important that they are representative of space imagery. The realistic quality of the images is verified by comparison with the spacecraft pose estimation benchmark dataset SPEED, and by using the lab-generated dataset that is captured with a realistic solar simulator and spacecraft model.

RQ-2d: What hardware specifications would be required to guarantee performance?
The results and analyses in this thesis showed that a minimum image resolution of 800x600 pixels is necessary to guarantee stable performance on the designed datasets. Furthermore, a sufficiently large frame rate is required such that the orientation propagation of the target between frames does not exceed 1 deg/frame.

RQ-2e: What are the limiting environmental factors?
As the algorithm is feature-based, it is of great importance that the features are visible. The overall visibility of the features is varied from the minimum brightness settings of the used software, to 150% of the original brightness. The results showed that no instabilities occur during the simulations, demonstrating great robustness to illumination changes. Furthermore, robustness has been achieved to background objects. In the datasets, a rotating Earth has been simulated for this purpose.

RQ-2f: How does the designed pose estimation system perform compared to the state-of-the-art?
Due to the fact that there is currently not much research done in this field and that the state-of-the-art does not test on the same datasets, true comparison in terms of accuracy cannot be made with literature. However, on average this system achieved an accuracy of sub-degree level on orientation, and cm-level on relative position. In related literature, Capuano et al.[13] achieved an accuracy of sub 1 degree and 0.3 meters. Jin et al. [33] obtained accuracies of sub 1 degree as well and 0.1 meters. This means that the pose estimation system used in this work has similar, if not better, accuracy than the state-of-the-art.

Overall, the designed pose estimation algorithm showed great robustness to changes in illumination, background objects, and various relative angular velocities. The algorithm is found to be limited by image resolution, and frame rate. In this work, it was observed that a minimum image resolution of 800x600 pixels is required to guarantee stable performance. Additionally, the frame rate should be high enough such that the attitude propagation of the target between frames does not exceed 1 deg/frame.

To answer the main research question, the suitability of the pose estimator for spacecraft relative navigation is assessed on three factors. Firstly, achievement of robustness to orbital relative navigation conditions is required. This includes relative motion, background, space image quality, and illumination robustness. The robustness to these factors is demonstrated in this report. The second factor is based on the capability of this system to work in real-time on space hardware. In this work, it is found that a lower limit of 800x600 pixels on image resolution guarantees stable performance of the pose estimator. Additionally, the frame rate should be high enough such that the target's orientation propagation between frames does not exceed 1 deg/frame. When observing hardware used on VBN systems, it is noted that the RNS camera in [49] could capture resolutions of 1024x1024px at a frame rate of 3 Hz. Based on the fact that the design of that system was developed more than ten years ago, and the fact that technology advances significantly over the years, it is deemed feasible that the algorithm could be implemented on space hardware. The third and last point is related to the achievable accuracy. On average, this system achieved an accuracy of sub-degree level in orientation, and centimetre-level in relative position. For particular mission phases of RPO, strict accuracy requirements need to be met. A reference example of requirements set for the e.Deorbit mission is found in [59]. They state accuracies for the closing phase (50-100m) of sub-5 deg and sub-10m, and sub-2 deg and sub-0.05 m for the sync and capture phase (0-50m). Additionally, during Servicing Mission 4 of the Hubble telescope [49], accuracies were required of sub 1 deg and sub 0.1m for the most critical phase (below 5 meters). Since the required accuracies of both the e.Deorbit mission and Servicing Mission 4 have been achieved in this work, the designed pose estimation system is deemed suitable for spacecraft relative navigation.

The novelties of this work related to robustness analysis of the adapted SLAM algorithm to spacecraft relative motion, the capability of handling background objects in the frame, and the analysis of hardware requirements, provide a relevant contribution to the work already performed in this field. Thereby achieving the research objective: To support technology development in the field of spacecraft RPO by researching the suitability of monocular SLAM for spacecraft relative navigation around an unknown target.

7.2. Recommendations

The limitations and assumptions made in this work, allow for recommendations for future work. These recommendations are described in this section.

Validate on real relative navigation footage

The results in this work are based on synthetically and lab-generated datasets. For real validation of the pose estimator, it should be tested on real space imagery capturing a relative navigation scenario. Such footage has not been obtained for this work, but is definitely an important factor to increase the validity of the work. Hence, this is seen as an important recommendation for future work.

Test in relative navigation testing facilities

An additional step to obtain accurate ground-truth data from lab-generated tests is made by testing in a relative navigation facility where the motion is computer-controlled. This will allow for more accurate relative motion scenarios and better acquisition of ground truth data.

Test the real-time capabilities on space hardware

This work demonstrated that hardware can severely limit the performance of the designed pose estimation system. Hence, to analyse the hardware limitations in more detail, it is recommended to test the algorithm performance on space hardware.

Implement scale recovery methods

The research performed in this work is based on the assumption that the scale of the generated map in monocular-SLAM is known. However, for actual scale recovery a secondary sensor is required. The sensor fusion and thorough analysis of the effects of this are deemed outside the scope of this work, but are categorised as a recommendation for future work.

Test on other targets
The pose estimation system is only assessed on one target. Since the performance might be different on another target, further research should be done on the real model-agnostic performance of the pose estimator.

Test other algorithms
Based on the selection criteria and motivation described in this chapter, the pose estimation system has been based on ORB-SLAM, which is an optimisation-based SLAM algorithm. However, some related work in the state-of-the-art uses filter-based SLAM. In order to properly compare the two distinct methods, the analysis of this work could be performed on filter-based SLAM in future studies to check if the conclusions of this thesis also hold for those types of SLAM.

Bibliography

[1] European Space Agency. Active debris removal. https://www.esa.int/Safety_Security/Space_Debris/Active_debris_removal, 2021.

[2] Pablo F. Alcantarilla, Jesús Nuevo, and Adrien Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *BMVC 2013 - Electronic Proceedings of the British Machine Vision Conference*, 2013. doi: 10.5244/C.27.13.

[3] Ali Ismail Awad and Mahmoud Hassaballah. *Image Feature Detectors and Descriptors*, volume 630. Springer International Publishing, 2016. ISBN 9783319288543. doi: 10.1007/978-3-319-28854-3.

[4] Kuldeep Rambhai Barad. Robust Navigation Framework for Proximity Operations around Uncooperative Spacecraft (MSc Thesis). Master's thesis, Delft University of Technology, 2020. URL http://repository.tudelft.nl/.

[5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. ISSN 10773142. doi: 10.1016/j.cviu.2007.09.014.

[6] Christopher Becker, Richard Howard, and John Rakoczy. Smartphone Video Guidance Sensor for Small Satellites. *27th Annual AIAA/USU Conference on Small Satellites*, pages 1–8, 2013.

[7] J.-A. Beraldin. Eye-safe digital 3-D sensing for space applications. *Optical Engineering*, 39(1):196, 2000. ISSN 0091-3286. doi: 10.1117/1.602352.

[8] J. A. Beraldin, Francois Blais, Marc Rioux, Luc Cournoyer, Denis G. Laurin, and Steve G. MacLean. Short and medium range 3D sensing for space applications. *Visual Information Processing VI*, 3074(July):29–46, 1997. doi: 10.1117/12.280635.

[9] Edward Beshore, Dante Lauretta, William Boynton, Christopher Shinohara, Brian Sutter, David Everett, Jonathan Gal-Edd, Ronald Mink, Michael Moreau, and Jason Dworkin. The OSIRIS-REx asteroid sample return mission. *IEEE Aerospace Conference Proceedings*, 2015-June(520), 2015. ISSN 1095323X. doi: 10.1109/AERO.2015.7118989.

[10] Chris Blackerby, Akira Okamoto, Kohei Fujimoto, Nobu Okada, Jason L Forshaw, and John Auburn. Elsa-D: an in-Orbit End-of-Life Demonstration Mission. *Astroscale*, 6:43644, 2018. URL www.astroscale.com.

[11] Blender. Blender website. https://www.blender.org/, October 2021.

[12] Carlos Campos Martínez, Richard Elvira, Juan J. Gómez Rodríguez, José M.M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate Open-source library for visual, Visual-inertial and Multi-map SLAM, 2020. ISSN 23318422.

[13] Vincenzo Capuano, Kyunam Kim, Alexei Harvard, and Soon Jo Chung. Monocular-based pose determination of uncooperative space objects. *Acta Astronautica*, 166(September 2019):493–506, 2020. ISSN 00945765. doi: 10.1016/j.actaastro.2019.09.027. URL https://doi.org/10.1016/j.actaastro.2019.09.027.

[14] Jizheng Chen, Jianping Yuan, and Qun Fang. Flight vehicle attitude determination using the modified rodrigues parameters. *Chinese Journal of Aeronautics*, 21(5):433–440, 2008. ISSN 10009361. doi: 10.1016/S1000-9361(08)60056-4. URL http://dx.doi.org/10.1016/S1000-9361(08)60056-4.

[15] John A. Christian, Mogi Patangan, Heather Hinkel, Keiko Chevray, and Jack Brazzel. Comparison of orion vision navigation sensor performance from STS-134 and the space operations simulation center. *AIAA Guidance, Navigation, and Control Conference 2012*, pages 1–19, 2012. doi: 10.2514/6.2012-5035.

[16] W. Clohessy and R Wiltshire. Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, 27(9):653–658, 1960.

[17] W. D. Compton and C. D. Benson. *Living and working in space. A history of Skylab.* NASA, 1983.

[18] A Cropp, P Palmer, and C I Underwood. Pose Estimation of Target Satellite for Proximity Operations. *Proceedings of the 14th Annual AIAA/USU Conference on Small Satellites,*, 44(0):1–7, 2000.

[19] Simone D', N.A. Amico, Mathias Benn, and John L. Jørgensen. Pose estimation of an uncooperative spacecraft from actual space imagery. *International Journal of Space Science and Engineering*, 2(2):171, 2014. ISSN 2048-8459. doi: 10.1504/ijspacese.2014.060600.

[20] Mehregan Dor and Panagiotis Tsiotras. ORB-SLAM applied to spacecraft non-cooperative rendezvous. *Space Flight Mechanics Meeting, 2018*, 2018. doi: 10.2514/6.2018-1963.

[21] Xiaodong Du, Bin Liang, Wenfu Xu, and Yue Qiu. Pose measurement of large non-cooperative satellite based on collaborative cameras. *Acta Astronautica*, 68(11-12):2047–2065, 2011. ISSN 00945765. doi: 10.1016/j.actaastro.2010.10.021. URL `http://dx.doi.org/10.1016/j.actaastro.2010.10.021`.

[22] Chad English, Sean Zhu, Christine Smith, Stephane Ruel, and Iain Christie. TriDAR: A hybrid sensor for exploiting the complementary nature of triangulation and LIDAR technologies. *European Space Agency, (Special Publication) ESA SP*, pages 599–607, 2005. ISSN 03796566.

[23] ESA Space Debris Office. ESA ' s Annual Space Environment Report. Technical Report September, ESA, 2020.

[24] Jason L. Forshaw, Guglielmo S. Aglietti, Simon Fellowes, Thierry Salmon, Ingo Retat, Alexander Hall, Thomas Chabot, Aurélien Pisseloup, Daniel Tye, Cesar Bernal, François Chaumette, Alexandre Pollini, and Willem H. Steyn. The active space debris removal mission RemoveDebris. Part 1: From concept to launch. *Acta Astronautica*, 168(September 2019):293–309, 2020. ISSN 00945765. doi: 10.1016/j.actaastro.2019.09.002. URL `https://doi.org/10.1016/j.actaastro.2019.09.002`.

[25] Robert B. Friend. Orbital Express program summary and mission overview. *Sensors and Systems for Space Applications II*, 6958(April 2008):695803, 2008. ISSN 0277786X. doi: 10.1117/12.783792.

[26] Lance B. Gatrell, William A. Hoff, and Cheryl W. Sklair. Robust image features: concentric contrasting circles and their image extraction. *Cooperative Intelligent Robotics in Space II*, 1612(March):235–244, 1992. doi: 10.1117/12.56761.

[27] C Harris and M Stephens. A combined corner and edge detector. *Alvey Vision Conference*, pages 147–151, 1988.

[28] Jan Hartmann, Jan Helge Klussendorff, and Erik Maehle. A comparison of feature descriptors for visual SLAM. *2013 European Conference on Mobile Robots, ECMR 2013 - Conference Proceedings*, pages 56–61, 2013. doi: 10.1109/ECMR.2013.6698820.

[29] Tatsuaki Hashimoto, Takashi Kubota, Jun'ichiro Kawaguchi, Masashi Uo, Kenichi Shirakawa, Takashi Kominato, and Hideo Morita. *Vision-based guidance, navigation, and control of Hayabusa spacecraft - Lessons learned from real operation*, volume 18. IFAC, 2010. ISBN 9783902661968. doi: 10.3182/20100906-5-jp-2022.00045. URL `http://dx.doi.org/10.3182/20100906-5-JP-2022.00045`.

[30] Richard T. Howard, Andrew F. Heaton, Robin M. Pinson, Connie L. Carrington, James E. Lee, Thomas C. Bryan, Bryan A. Robertson, Susan H. Spencer, and Jimmie E. Johnson. The advanced video guidance sensor: Orbital Express and the next generation. *AIP Conference Proceedings*, 969(January 2008):717–724, 2008. ISSN 0094243X. doi: 10.1063/1.2845036.

[31] Haidong Hu, Dayi Wang, Hao Gao, Chunling Wei, and Yingzi He. Vision-based position and pose determination of non-cooperative target for on-orbit servicing. *Multimedia Tools and Applications*, 79(21-22): 14405–14418, 2020. ISSN 15737721. doi: 10.1007/s11042-018-6696-8.

[32] Danil Ivanov, Mikhail Ovchinnikov, and Marianna Sakovich. Relative pose and inertia determination of unknown satellite using monocular vision. *International Journal of Aerospace Engineering*, 2018, 2018. ISSN 16875974. doi: 10.1155/2018/9731512.

[33] Zeming Jin, Ling Wang, Hanhan Liu, Ronghua Du, and Xiang Zhang. Monocular-Based Pose Estimation of Non-Cooperative Space Targets Using EKF and EKPF. *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 46–51, 2020. doi: 10.1109/cisp-bmei51763.2020.9263615.

[34] Keyvan Kanani, Antoine Petit, Eric Marchand, Thomas Chabot, and Bernard Gerber. Vision based navigation for debris removal missions. *Proceedings of the International Astronautical Congress, IAC*, 4:2674–2681, 2012. ISSN 00741795.

[35] T S Kelso. Analysis of the Iridium 33-Cosmos 2251 collision. In *Advances in the Astronautical Sciences*, volume 135, pages 1099–1112, 2010. ISBN 9780877035572.

[36] Mate Kisantal, Sumant Sharma, Tae Ha Park, Dario Izzo, Marcus Märtens, and Simone D'Amico. Satellite pose estimation challenge: Dataset, competition design, and results. *IEEE Transactions on Aerospace and Electronic Systems*, 56(5):4083–4098, 2020. ISSN 15579603. doi: 10.1109/TAES.2020.2989063.

[37] Richard P. Kornfeld, Robert L. Bunker, Gordy C. Cucullu, John C. Essmiller, Fred Y. Hadaegh, C. Christian Liebe, Curtis W. Padgett, Edward C. Wong, Sanjeev Seereeram, and Raman K. Mehra. New millennium ST6 autonomous rendezvous experiment (ARX). *IEEE Aerospace Conference Proceedings*, 1: 369–380, 2003. ISSN 1095323X. doi: 10.1109/AERO.2003.1235067.

[38] Denis G. Laurin, J. A. Beraldin, Francois Blais, Marc Rioux, and Luc Cournoyer. Three-dimensional tracking and imaging laser scanner for space operations. *Laser Radar Technology and Applications IV*, 3707 (May):278, 1999. ISSN 1996756X. doi: 10.1117/12.351351.

[39] Ting Lei, Xiao Feng Liu, Guo Ping Cai, Yun Meng Liu, and Pan Liu. Pose estimation of a noncooperative target based on monocular visual SLAM. *International Journal of Aerospace Engineering*, 2019, 2019. ISSN 16875974. doi: 10.1155/2019/9086891.

[40] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. BRISK: Binary Robust invariant scalable keypoints. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2548–2555, 2011. doi: 10.1109/ICCV.2011.6126542.

[41] Carl Christian Liebe, Alex Abramovici, Randy K. Bartman, Robert L. Bunker, Jacob Chapsky, Cheng Chih Chu, Daniel Clouse, James W. Dillon, Bob Hausmann, Hamid Hemmati, Richard P. Kornfeld, Clint Kwa, Sohrab Mobasser, Michael Newell, Curtis Padgett, W. Thomas Roberts, Gary Spiers, Zachary Warfield, and Malcolm Wright. Laser radar for spacecraft guidance applications. *IEEE Aerospace Conference Proceedings*, 6(February):2647–2662, 2003. ISSN 1095323X. doi: 10.1109/AERO.2003.1235190.

[42] Chang Liu and Weiduo Hu. Relative pose estimation for cylinder-shaped spacecrafts using single image. *IEEE Transactions on Aerospace and Electronic Systems*, 50(4):3036–3056, 2014. ISSN 00189251. doi: 10.1109/TAES.2014.120757.

[43] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. ISSN 09205691. doi: 10.1023/B:VISI.0000029664.99615.94.

[44] Masaaki Mokuno and Isao Kawano. In-orbit demonstration of an optical navigation system for autonomous rendezvous docking. *Journal of Spacecraft and Rockets*, 48(6):1046–1054, 2011. ISSN 15336794. doi: 10.2514/1.52193.

[45] Raul Mur-Artal and Juan D. Tardos. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. ISSN 15523098. doi: 10.1109/TRO.2017.2705103.

[46] Raul Mur-Artal, J. M.M. Montiel, and Juan D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. ISSN 15523098. doi: 10.1109/TRO.2015.2463671.

[47] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drifts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010. ISSN 01628828. doi: 10.1109/TPAMI.2010.46.

[48] Bo J. Naasz, Richard D. Bums, Steven Z. Queen, John Van Eepoel, Joel Hannah, and Eugene Skelton. The HST SM4 relative navigation sensor system: Overview and preliminary testing results from the flight robotics lab. *Journal of the Astronautical Sciences*, 57(1-2):457–483, 2009. ISSN 00219142. doi: 10.1007/BF03321512.

[49] Bo J. Naasz, John Van Eepoel, Steven Z. Queen, C. Michael Southward, and Joel Hannah. Flight results from the HST SM4 Relative Navigation Sensor system. *Advances in the Astronautical Sciences*, 137:723–744, 2010. ISSN 00653438.

[50] NASA. Overview of the DART Mishap Investigation Results. *Defense*, pages 1–10, 2005.

[51] NASA. Measuring earth's albedo. `https://earthobservatory.nasa.gov/images/84499/measuring-earths-albedo`, October 2021.

[52] Cornelius J Dennehy Nesc and J Russell Carpenter. A Summary of the Rendezvous , Proximity Operations , Docking , and Undocking (RPODU) Lessons Learned from the Defense Advanced Research Project Agency (DARPA) Orbital Express (OE) Demonstration System Mission. *Russell The Journal Of The Bertrand Russell Archives*, 2011.

[53] Roberto Opromolla, Giancarmine Fasano, Giancarlo Rufino, and Michele Grassi. A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations. *Progress in Aerospace Sciences*, 93(June):53–72, 2017. ISSN 03760421. doi: 10.1016/j.paerosci.2017.07.001.

[54] OptiTrack. Quick start guide: Precision capture. `https://v22.wiki.optitrack.com/index.php?title=Quick_Start_Guide:_Precision_Capture`, October 2021.

[55] Tae Ha Park, Juergen Bosse, and Simone D'Amico. Robotic Testbed for Rendezvous and Optical Navigation: Multi-Source Calibration and Machine Learning Use Cases. *CoRR*, pages 1–20, 2021. URL `http://arxiv.org/abs/2108.05529`.

[56] Staffan Persson, Per Bodin, Eberhard Gill, Jon Harr, and John Jörgensen. PRISMA - An autonomous formation flying mission. In *European Space Agency, (Special Publication) ESA SP*, volume 625 SP, pages 25–29, 2006. ISBN 9290929367.

[57] Marco Pertile, Mattia Mazzucato, Luca Bottaro, Sebastiano Chiodini, Stefano Debei, and Enrico Lorenzini. Uncertainty evaluation of a vision system for pose measurement of a spacecraft with fiducial markers. *2nd IEEE International Workshop on Metrology for Aerospace, MetroAeroSpace 2015 - Proceedings*, pages 283–288, 2015. doi: 10.1109/MetroAeroSpace.2015.7180669.

[58] Antoine Petit, Keyvan Kanani, and Eric Marchand. Vision-based Detection and Tracking for Space Navigation in a Rendezvous Context. *Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS˜2012*, 2012.

[59] Wolfgang Rackl, Marco De Stefano, Roberto Lampariello, Nuno Santos, Pedro Serra, Marco Canetri, Finn Ankersen, and Jesus Gil-fernandez. GNC architecture for the e . Deorbit mission. *7th European Conference for Aeronautics and Space Sciences (EUCASS)*, 2017. doi: 10.13009/EUCASS2017-317.

[60] Paul L. Rosin. Measuring Corner Properties. *Computer Vision and Image Understanding*, 73(2):291–307, 1999. ISSN 10773142. doi: 10.1006/cviu.1998.0719.

[61] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2564–2571, 2011. doi: 10.1109/ICCV.2011.6126544.

[62] S. Ruel, C. English, M. Anctil, and P. Church. 3DLASSO: Real-time pose estimation from 3D data for autonomous satellite servicing. *European Space Agency, (Special Publication) ESA SP*, pages 591–598, 2005. ISSN 03796566.

[63] Timothy E. Rumford. Demonstration of Autonomous Rendezvous Technology (DART) project summary. *Space Systems Technology and Operations*, 5088(August 2003):10–19, 2003. doi: 10.1117/12.498811.

[64] Claire Samson, Chad English, Adam Deslauriers, Iain Christie, François Blais, and Frank Ferrie. The Neptec three-dimensional laser camera system: From space mission STS-105 to terrestrial applications. *Canadian Aeronautics and Space Journal*, 50(2):115–123, 2004. ISSN 00082821. doi: 10.5589/q04-009.

[65] Sumant Sharma and Simone D'Amico. Comparative assessment of techniques for initial pose estimation using monocular vision. *Acta Astronautica*, 123:435–445, 2016. ISSN 00945765. doi: 10.1016/j.actaastro.2015.12.032. URL `http://dx.doi.org/10.1016/j.actaastro.2015.12.032`.

[66] Colin L. Smithpeter, Robert O. Nellums, Steve M. Lebien, George Studor, and George H. James III. Ladar measurements of the International Space Station. *Laser Radar Technology and Applications VI*, 4377 (September):65–72, 2001. doi: 10.1117/12.440123.

[67] Arne Sonnenburg, Marcel Tkocz, and Klaus Janschek. *EKF-SLAM based approach for spacecraft rendezvous navigation with unknown target spacecraft*, volume 18. IFAC, 2010. ISBN 9783902661968. doi: 10.3182/20100906-5-jp-2022.00058. URL `http://dx.doi.org/10.3182/20100906-5-JP-2022.00058`.

[68] J. Spaander, T. Nachtergaele, and Rijlaarsdam D. Design, construction, verification and validation of a led solar simulator. Technical report, Delft University of Technology, 2019.

[69] Spaceflight101. Osiris-rex mission & trajectory design. `https://spaceflight101.com/osiris-rex/osiris-rex-mission-profile/`, 2021.

[70] Hauke Strasdat, J. M.M. Montiel, and Andrew J. Davison. Real-time monocular SLAM: Why filter? *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2657–2664, 2010. ISSN 10504729. doi: 10.1109/ROBOT.2010.5509636.

[71] Shaharyar Ahmed Khan Tareen and Zahra Saleem. A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. *2018 International Conference on Computing, Mathematics and Engineering Technologies: Invent, Innovate and Integrate for Socioeconomic Development, iCoMET 2018 - Proceedings*, January:1–10, 2018. doi: 10.1109/ICOMET.2018.8346440.

[72] Panagiotis Tsiotras. ASTROS: A 5DOF experimental facility for research in space proximity operations. *Advances in the Astronautical Sciences*, 151(January):717–730, 2014. ISSN 00653438.

[73] Brent E. Tweddle and Alvar Saenz-Otero. Relative computer vision-based navigation for small inspection spacecraft. *Journal of Guidance, Control, and Dynamics*, 38(5):969–977, 2015. ISSN 15333884. doi: 10.2514/1.G000687.

[74] Markus Wilde, Jan Harder, and Enrico Stoll. On-Orbit Servicing and Active Debris Removal: Enabling a Paradigm Shift in Spaceflight. *Frontiers in Robotics and AI*, 6:136, 2019. ISSN 15336794. doi: 10.2514/1.19653.

[75] B. Williams, P. Antreasian, E. Carranza, C. Jackman, J. Leonard, D. Nelson, B. Page, D. Stanbridge, D. Wibben, K. Williams, M. Moreau, K. Berry, K. Getzandanner, A. Liounis, A. Mashiku, D. Highsmith, B. Sutter, and D. S. Lauretta. *OSIRIS-REx Flight Dynamics and Navigation Design*, volume 214. Springer Science+Business Media B.V., part of Springer Nature, 2018. ISBN 1121401805. doi: 10.1007/s11214-018-0501-x. URL `http://dx.doi.org/10.1007/s11214-018-0501-x`.

[76] Nan Yang, Rui Wang, Xiang Gao, and Daniel Cremers. Feature-based or Direct: An Evaluation of Monocular Visual Odometry. *IEEE Robotics and Automation Letters*, 3(4):2878–2885, 2018. ISSN 23773766. doi: 10.1109/LRA.2018.2846813.

[77] Nan Yang, Rui Wang, Xiang Gao, and Daniel Cremers. Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect. *IEEE Robotics and Automation Letters*, 3(4): 2878–2885, 2018. ISSN 23773766. doi: 10.1109/LRA.2018.2846813.

[78] Feng Yu, Zhen He, Bing Qiao, and Xiaoting Yu. Stereo-vision-based relative pose estimation for the rendezvous and docking of noncooperative satellites. *Mathematical Problems in Engineering*, 2014, 2014. ISSN 15635147. doi: 10.1155/2014/461283.

V

Appendices

A
Literature Study

The purpose of this literature review is to obtain an overview of the field of spacecraft VBN and to identify *knowledge gaps* which can potentially be explored in a thesis research. In order to obtain the gaps in the literature, the following procedure has been adapted for the study. The initial step is to obtain familiarity with the field. Hence, typical missions and VBN are analysed in a global sense. Subsequently, each main aspect of a VBN system will be analysed (sensors, targets, pose determination techniques etc.). In order to structure the literature research, each section is started with research questions which are to be answered in the particular section.

A.1. Missions Overview

The purpose of this section is gain an overview how a generic mission that exploits VBN in RPO looks like. The following research questions are formulated to guide the analysis in this section:

1. *What missions have been performed in the field of RPO?*
2. *How is a typical RPO mission profiled?*
3. *What ADR missions have been performed?*
4. *What OOS missions have been performed?*
5. *What asteroid exploration missions have been performed?*

In order to answer these research questions, this section starts with a brief overview of the history of RPO. Subsequently a discussion of ADR missions is presented, including insight in a typical ADR mission. Afterwards, the history of OOS is provided. Finally, asteroid exploration missions are outlined.

A.1.1. Proximity Operations and Rendezvous

Proximity operations date back to the start of the space age, with its roots at the Gimini program in the 1960s. The spacecraft mission involving Gemini VI and Gemini VII was the first mission to perform rendezvous and proximity operations. The operations were performed manually by astronauts by visual monitoring with the support of ground station. This lead to the first mission to utilise proximity operations in order to meet operational requirements: the Apollo 9 program. Around the same time (1967), the first fully autonomous RPO and docking was executed by two soviet spacecraft, Kosmos-186 and Kosmos-188, by means of an active inter-spacecraft communication link. From then onward, the advancements in proximity navigation grew. This led to the initiation of orbiting space stations, of which the ISS is the prime example.

A.1.2. Active Debris Removal

In order to stabilise the growth of space debris, ADR missions are necessary. Studies by ESA and NASA show that the number of debris objects in orbit would continue to grow even under idealised circumstances, where there are no launches, no debris release, and no explosions [1]. This is caused by an assumed collision rate between objects of once every 10 years. Consequently, only debris mitigation measures are not enough to obtain a sustainable, healthy orbital population. This stresses the importance of ADR missions, that reduce the number of large and massive objects in space.

RemoveDEBRIS successfully performed key ADR technology demonstrations, after it was launched in 2018 from the ISS. It demonstrated the use of both a net and harpoon for capturing, a dragsail deployment for accelerated deorbital rate of itself, and its VBN system [24]. Its VBN comprises a combination of LIDAR, IR, and optical cameras. It has been tested by ejecting a second CubeSat at very low velocity (2 cm/s) out of orbit plane, to comply with safety constraints and VBN demonstration needs (lightning, background, range).

Future ADR missions include the ELSE-D mission [10] and the ClearSpace-1 mission.

ADR Mission profile

In order to get better insight into how and when vision-based navigation is performed in ADR and OOS missions, it is essential to look into detail of a generic ADR mission. The e.Deorbit mission will be used as a baseline for this. This mission had as main objective to remove Envisat, a former remote sensing satellite, safely from orbit [59]. Its various mission phases are illustrated in Figure A.1. The chaser spacecraft is injected into a relative orbit, after which it performs several manoeuvres to reach the REG, about 8 km away from its target, by means of absolute navigation. From that point, the rendezvous phase is initiated, and thus the relative navigation system. It uses LOS measurements from a narrow-angle camera, and LOS and range measurements from a LIDAR sensor, to reduce the distance from 8 km to about 100 meters, when it reaches its Parking Hold Point. At this relative distance, the target is sufficiently large for 6-DOF pose estimation to be initialised. Once the pose estimate has been initialised, it performs state tracking on the target along the V-bar, and executes a target inspection fly-around. It then tries to synchronise its motion with the target, until it is fully synchronised and close enough to capture the target. Once captured, the coupled satellites are stabilised and disposed.

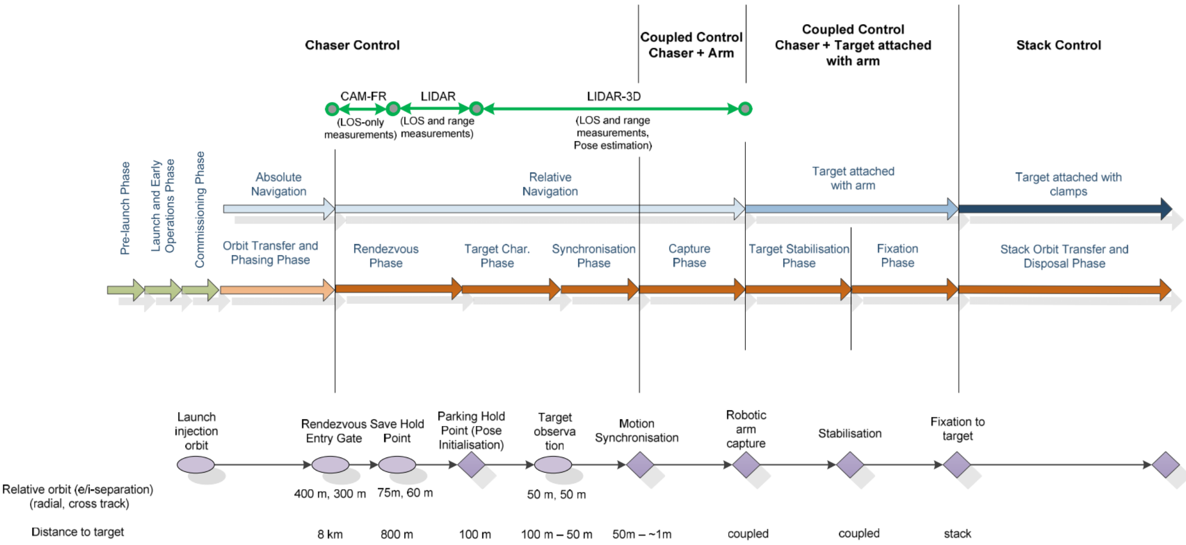


Figure A.1: e.Deorbit mission phases [59]

The state estimation accuracy requirements during the e.Deorbit mission are listed in Table A.1. It is however needless to mention that this is very mission dependent.

Table A.1: e.Deorbit RPO performance requirements in close proximity [59]

Performance Parameter (per axis)	Rendezvous, closing (100m)	Rendezvous, Sync. Capture (50m)
Relative Position [m]	10	0.05
Relative Velocity [m/s]	0.1	0.01
Relative Attitude [deg]	5	2
Relative Angular rate [deg/s]	0.5	0.5

A.1.3. On Orbit Servicing

The purpose of OOS is to perform maintenance or repair to extend the lifetime of spacecraft. This benefits the health in the space environment as it reduces the amount of debris in orbit. The repair of a jammed solar panel on Skylab-2 was the first real OOS operation. The repair was done by a crew using ground support [17]. The following milestone was achieved by on orbit servicing of the HST by the space shuttle, which performed semi-autonomous rendezvous and docking using its on-board relative navigation system. The close-proximity and docking were however performed manually by the astronauts. Advancements in the field of GNC resulted in the first fully autonomous OOS missions, like the DARPA's OE mission [52]. This mission executed autonomous GNC from a relative range of 200 km until docking and subsequent servicing functions like fuel transfer, computer and battery replacement. It used laser-based tracking and a monocular sensor for close-proximity navigation. Recently, the Northrop Grumman MEV-1 achieved the first commercial servicing mission on a target not designed for docking. The MEV-1 extended the life of the target spacecraft by five years.

A.1.4. Asteroid Exploration

Asteroid exploration is typically performed to research its materials. This material can be explored to obtain information about the asteroid's composition. This gives knowledge about the formation of the solar system, but this can also be exploited for further exploration and colonisation of the solar system, by using the asteroid's materials for in orbit manufacturing or refueling.

The first mission in which a spacecraft performed a fly-by of an asteroid was NASA's Galileo mission, which the spacecraft flew past asteroid Gaspara in 1991 and Ida in 1993. From then onward, there were numerous missions that studied asteroids. The NASA's NEAR-Shoemaker mission studied asteroids Mathilde and Eros; the Rosetta mission observed Steins and Lutetia, Deep Space 1 and Stardust both had close encounters with asteroids, and NASA's Dawn spacecraft orbited asteroid Vesta for more than a year.

Active asteroid exploration missions include OSIRIS-REx, which is an asteroid study and sample-return mission. It was launched in 2016, after which it rendezvoused with Bennu in 2018. It spent the next months studying the surface in order to find a suitable location from which to extract a sample. It successfully collected a sample in 2020. It is expected to be back at Earth in 2023. It's VBN system uses both star fields and asteroid landmakrs for navigation, captured by its optical cameras [75].
The Japanese Hayabusa returned asteroid samples from Itokawa in 2010, after it was launched in 2003. It's vision-based navigation systems was ground-based, meaning the pictures taken from the asteroids were downlinked and the relative pose of the spacecraft was determined on ground [29]. It had multiple visual cameras, as well as LIDARs on board. However, its landing process was autonomous, due to the communication delay of 30 min. From an altitude of 40 m above the surface, a target marker was released, which is used as relative navigation target. This reduces the computational load and increases the image processing certainty.
The successor of Hayabusa, Hayabusa2 was launched in 2014. After launch, it rendesvoused with asteroid Ryugu in 2018 and returned the asteroid samples to Earth in 2020. The relative navigation system is based on the one of Hayabusa, where they released target markers on the asteroid.

Mission profile
Asteroid exploration missions generally have slight differences compared to ADR and OOS missions. Therefore, a mission profile of a typical asteroid exploration mission will be described: the OSIRIS-REx mission, in particularly its proximity operations.

After launch and outbound cruise, it enters the approach phase. This begins when the asteroid is just bright enough to be detected by the VBN system. It then approaches the asteroid gently in order to optically acquire the asteroid with its on-board cameras [69], and also observing the vicinity of the asteroid for any potential hazards. When close enough, it takes images from the asteroid to start shape model development of the asteroid. The shape modelling is done using two independent strategies, one concerning stereo cameras, and one based on range images. Finally, it selects two target sites for touch down and obtain samples, which are reviewed for suitability.

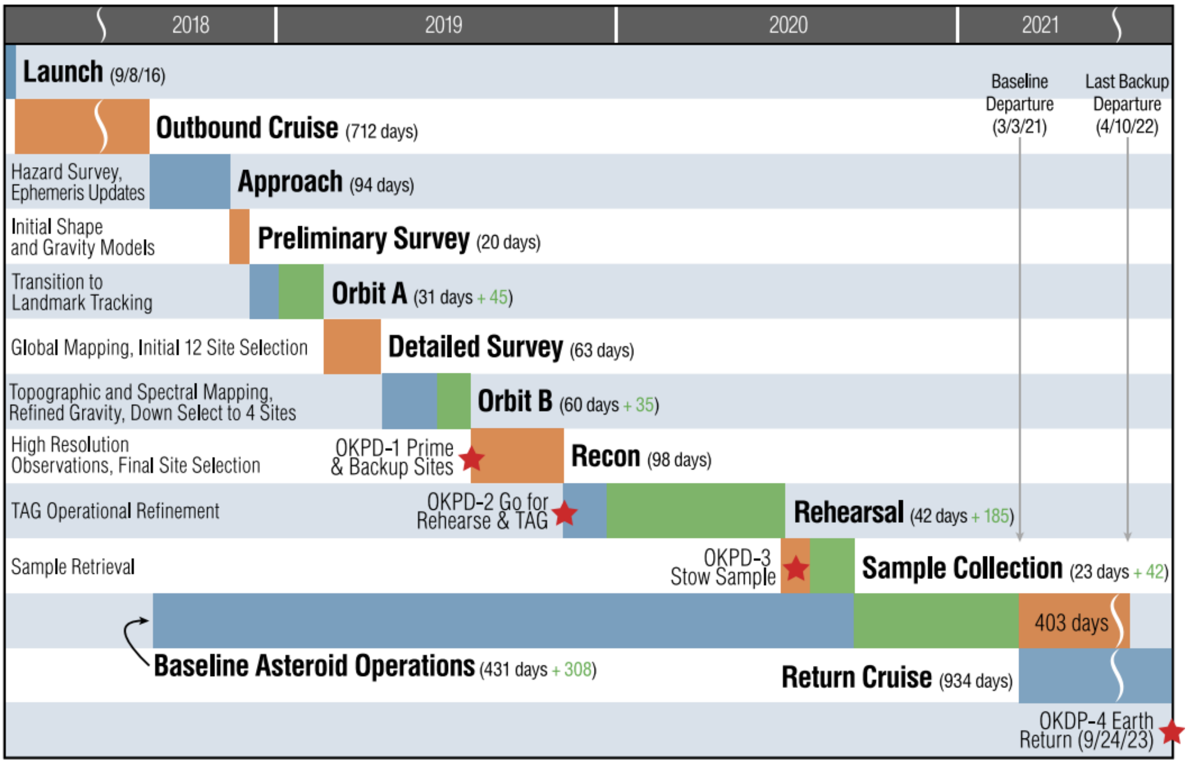


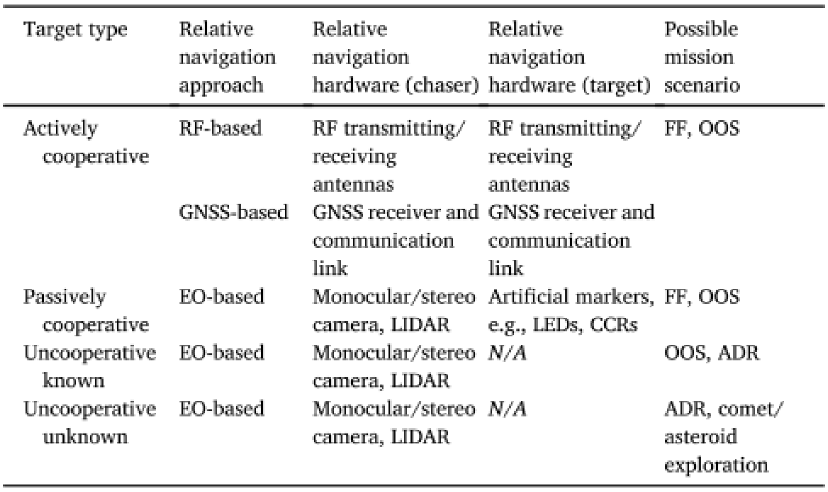
Figure A.2: Mission outline OSIRIS-REx [69]

A.2. Vision-Based Navigation

This section gives a general overview of relative navigation in close-proximity of various targets. The following questions will be researched in this section:

1. *How is close-proximity defined?*
2. *How is VBN performed?*
3. *What are the main components in a VBN system?*

Close-proximity is typically a distance less than 200 meters between the chaser and target, such that the target is large enough for pose estimation, while maintaining a safe distance [4]. In this range, EO sensors are the best option to ensure pose estimation, due to the limitations of RF-based and GNSS-based technologies [53]. The taxonomy of spaceborne relative navigation approaches and scenarios is illustrated in Figure A.3. Targets for relative navigation in close-proximity are generally classified as cooperative or uncooperative. Cooperative targets can either be passively or actively cooperative. A passively cooperative target contains artificial markers which can be detected and recognised by the chaser satellite. The markers itself can both be passive (e.g. CCRs) and active (e.g. LEDs) as well. Relative navigation to an uncooperative target is more difficult as there is a lack of cooperative information from the target. Uncooperative targets are further subdivided into known and unknown targets. Known targets are defined as targets from which the geometry/model is known.



Target type	Relative navigation approach	Relative navigation hardware (chaser)	Relative navigation hardware (target)	Possible mission scenario
Actively cooperative	RF-based	RF transmitting/ receiving antennas	RF transmitting/ receiving antennas	FF, OOS
	GNSS-based	GNSS receiver and communication link	GNSS receiver and communication link	
Passively cooperative	EO-based	Monocular/stereo camera, LIDAR	Artificial markers, e.g., LEDs, CCRs	FF, OOS
Uncooperative known	EO-based	Monocular/stereo camera, LIDAR	*N/A*	OOS, ADR
Uncooperative unknown	EO-based	Monocular/stereo camera, LIDAR	*N/A*	ADR, comet/ asteroid exploration

Figure A.3: Overview of relative navigation scenarios [53]

A VBN system determines the relative state from its sensor measurements. In general, it does so by first determining its pose from the sensor data, after which the relative pose, and potentially features from the sensor data itself are inserted in the state estimator. The pose estimation pipeline generally contains image processing algorithms in order to extract certain features from the images. The basic components are visualised in Figure A.4. These blocks will be further elaborated on in the next sections.

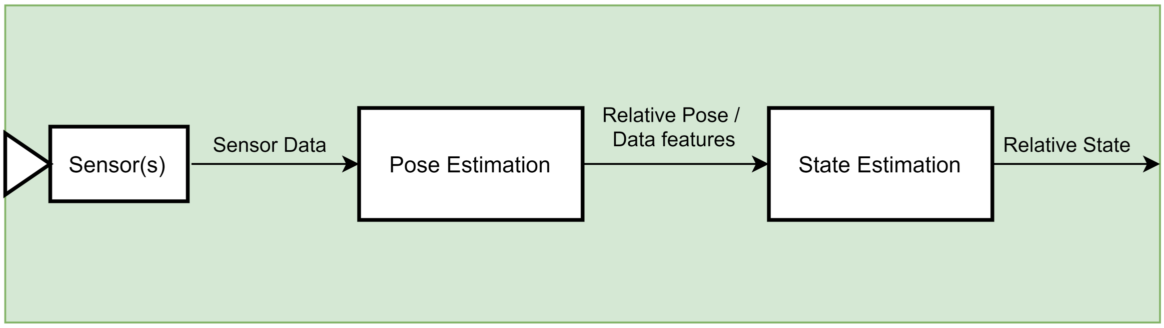


Figure A.4: High-level overview of a generic VBN system

A.3. Electro-optical sensors

This section discusses the types of sensors used in VBN systems. The following research questions provide a guide in the literature study of sensors for VBN systems:

1. *What type of electro-optical sensors are used in VBN systems?*
2. *How do they compare to each other?*

The taxonomy of EO sensors for spacecraft applications is illustrated in Figure A.5. It is noticed that only LIDAR sensors, as well as monocular and stereo cameras in the visible spectrum, can estimate the full pose of its target. These three will be further reviewed in this section. In this study, monocular and stereo cameras are referred for cameras in the visible spectrum, unless specifically mentioned otherwise.

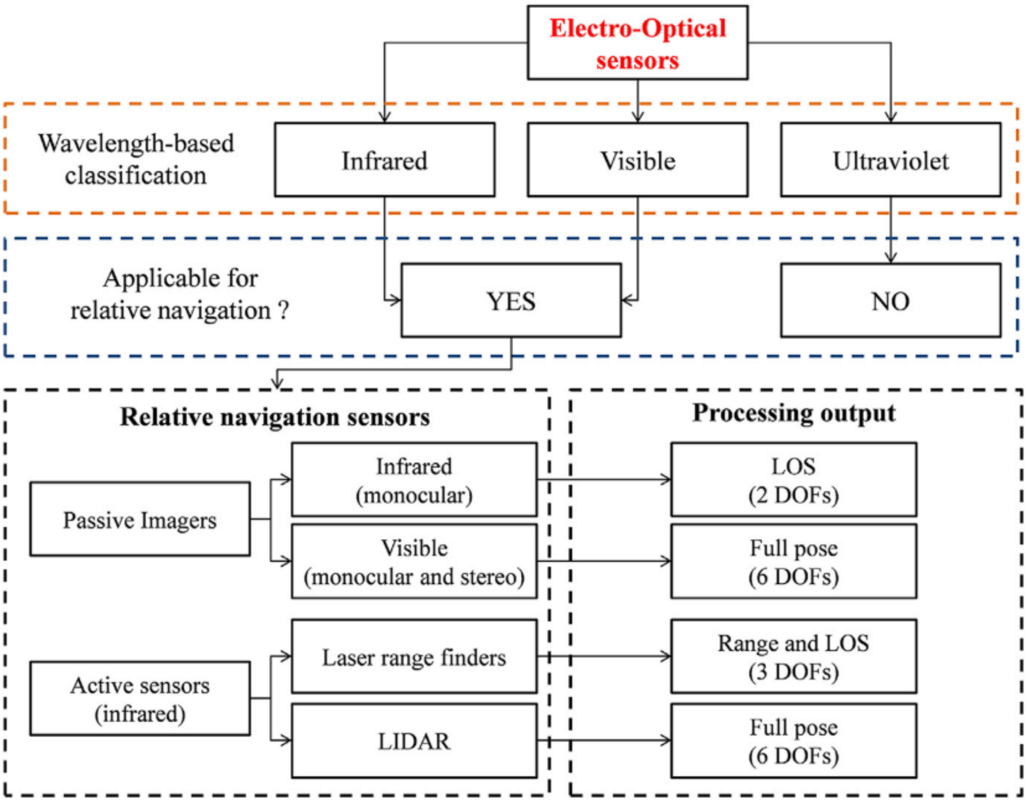


Figure A.5: Overview of EO sensors for relative navigation [53]

A.3.1. LIDAR

LIDAR sensors are able to measure distances by illuminating a target with an infrared laser source and then analysing the back-scattered radiation. The collected echoes are used to produce 3D data, called point clouds. The relative pose of the target can subsequently be determined from these point clouds. LIDARs are active sensors, which means they are characterised by higher hardware complexity, cost, and power consumption than passive sensors [53]. However, LIDARs performance is robust in poor illumination conditions and the fact that this is laser-based allows the target to be easily distinguished from the background. Another benefit of LIDARs compared to stereo cameras is that they can ensure larger operational ranges with constant level of accuracy. These sensors are very beneficial to be used on bigger spacecraft with a large enough power budget, but typically consume too much power for nanosatellites.

Types of LIDARs

There exist 3 types of LIDARs, illustrated in Figure A.6 under the *Technical solution* block. The first type are scanning devices. Scanning LIDARs generally contain one detector, and they work by directing a single narrow laser beam by means mirrors, lenses or other devices moved by high-speed and high-precision galvanometers. The area can be scanned on various patterns, like the raster scan, Lissajous, or spirals. Scanning LIDARs are relatively simple to calibrate as one only needs to be concerned about the light sensitivity and the timing for one detector. There are however a couple of drawbacks. Since the sensor comprises moving hardware, there are many potential points of mechanical failure. Additionally, the maximum frame rate is limited by the mechanism that performs the scan of the FOV. Finally, the relative motion between the sensor and target during scan time affects the accuracy, as it causes motion blur effects.

The second LIDAR type, scannerless sensors based on detector arrays, use a single broad laser beam to illuminate the full FOV. An array of detectors analyses the back-scattered radiation. They have as benefit that they do not have any moving parts, and they are not significantly affected by motion blur. They are however more difficult to calibrate, due to the amount of detectors. Additionally, they have poor spatial resolution, as the fabrication limit on the size of the detector arrays restrains the size of the point cloud.

The final type of LIDARs, classified as spatial light modulators, exploit compressed sensing algorithms to process the returned echo. The echo is generated by illuminating the scene in a particular, pre-defined pattern.

These are however still under development and far from to be applied in space.

LIDARs can further be classified into Continuous-wave (CW) and pulsed systems. The first type computes distances by observing the phase difference between an emitted signal (Amplitude-modulated (AM) or Frequency-modulated (FM)) and the reflected echo. They are however constrained to close-range applications, as their accuracy reduces with the inverse of the square root of the distance. Hence, they are generally used in a relative range of within 15 metres. CW systems are able to utilise the triangulation method as measurement principle, which can endure millimetre or sub-millimetre error [53].

Pulsed LIDARs work by measuring the time delay between transmitted and received laser pulses, thereby computing the relative distance. By using an incoherent laser source, they can provide a constant accuracy in the range measurement as the target distance varies. As a result, they can operate in a very large relative range (from metres to kilometres). The achievable time resolution between the transmitted and received pulses limits the maximum achievable accuracy of this system. As an example, a time resolution of tens of picoseconds is needed to get centimetre level accuracy, which requires complicated and expensive electronics [53].

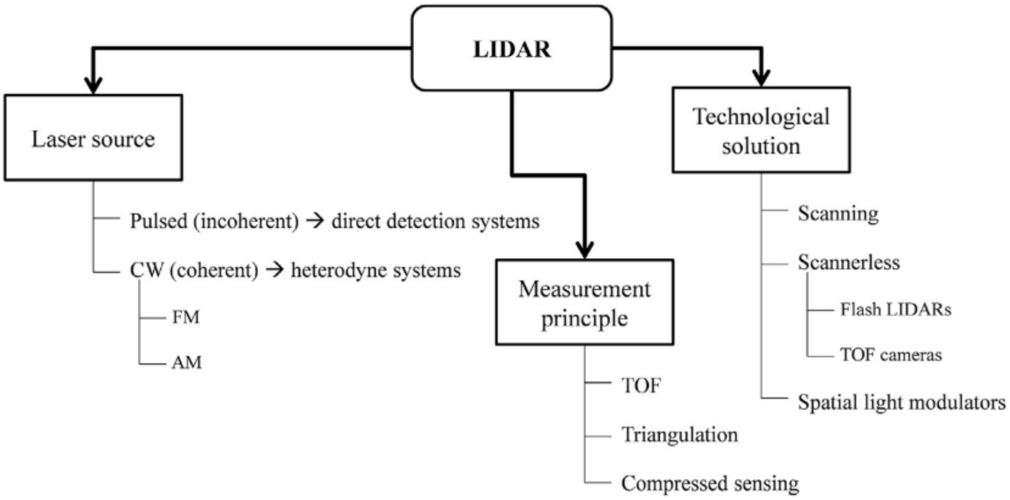


Figure A.6: Overview of LIDARs [53]

Spaceborne LIDARs
An overview of LIDARs for space applications is given in Table A.2. Each of them will be discussed individually.

The Canadian Space Agency (CSA) developed the Laser Range Scanner (LARS), a scanning type LIDAR. It can be used both for surface imaging and ranging & tracking of cooperative targets [7, 8, 38]. During the space shuttle STS-52 mission, its capabilities were tested by 3D inspecting and monitoring of the ISS' surface-reflectance. It can both use a CW laser source (with triangulation principle) for short range, as well as a pulsed laser for the larger ranges. This LIDAR system has great suitability for space applications due to its wide dynamic range of intensities.

The Laser Dynamic Range Imager (LDRI) is a space-qualified scannerless LIDAR. It can be used for characterisation of structures as well as proximity navigation. It was developed by the Sandia National Laboratories (SNL) and had been used for measuring vibrations, modal frequencies and amplitudes of the ISS structure during STS-97 [66]. The LIDAR exploits a CW-AM diffuse source to illuminate its FOV.

Neptec's Laser Camera System (LCS) is a scanning LIDAR which had as purpose to perform orbit inspection and relative navigation [62, 64]. It has been used during the missions STS-118 and STS-122 to scan the thermal protection system of the space shuttle. During this scan, it had a stand-off range of 3 metres and was able to detect cracks of sub 1 mm.

Laser Mapper (LAMP) is a Time Of Flight (TOF) scanning LIDAR, developed by the Jet Propulsion Laboratory (JPL). It has been served for rendezvous and docking, and several other GNC applications [37, 41]. One of the missions was the demonstration missions XSS-11, which carried out the Autonomous Rendezvous Experiment (ARX). During this mission it served as the primary sensor for relative navigation towards the target spacecraft. The sensor can be exploited for relative navigation towards both cooperative and uncooperative targets up to relative distances of respectively 5 and 2.5 km.

The Rendezvous and Docking Sensor (RVS) is also a TOF scanning LIDAR, designed by Jena-Optronik for automated docking of unmanned transfer vehicles with the ISS. It's functionality had been demonstrated in space during docking of the Space Shuttle with the Mir station, during missions STS-84 and STS-86. After which it was used on the first Automatic Transfer Vehicle (ATV).Newer versions of the sensor have been recently developed, which are both lighter and less power consuming. The RVS-3000 3D brings more advanced capabilities as it can generate high resolution point clouds, allowing for autonomous target detection and identification.

Another system developed by Neptec is the Triangulation LIDAR sensor (TRIDAR). TRIDAR makes use of different sensors for the short and long range [22]. It contains a high precision, auto-synchronous triangulation sensor for the short range, and a TOF LIDAR for the medium to long range. The system is compact due to the fact that both sensors exploit the same optical path and control electronics. It has been used on several space shuttle missions: the STS-128, the STS-131, and the STS-135. The sensor is known for being the first space-tested active vision system that can detect and track an uncooperative target autonomously.

In contrast to the aforementioned sensors, several flash LIDARs have been developed as well. Advanced Scientific Concepts (ASC) developed the DragonEye and GoldenEye, while Ball Aerospace developed the Vision Navigation System (VNS). The DragonEye has been successfully tested for autonomous rendezvous and docking during missions STS-127 and STS-133. the GoldenEye sensor is used on the recent OSIRIS-REx mission [9]. The VNS is the main navigation sensor on the Orion Multi-Purpose Crew Vehicle [15].

Table A.2: Overview of spaceborne LIDARs [53]

System (developer)	Operational mode	Technology and measurement principle	Operational range (m)	Range accuracy
LARS (CSA)	cooperative	Scanning CW Triangulation Pulsed TOF	0.5-10 10-10000	sub-mm 3 cm
LCS (Neptec)	cooperative uncooperative	Scanning CW Triangulation	1-10	0.1 mm - 5 mm (1σ)
LAMP (JPL)	cooperative uncooperative	Scanning Pulsed TOF	<5000 <2500	10 cm (bias) 2.6 cm (3σ)
RVS (Jena-Optronik)	cooperative	Scanning Pulsed TOF	1-1500	0.01 m -0.5 m (bias) 0.01 m - 0.1 m (3σ)
RVS-3000 (Jena-Optronik)	cooperative uncooperative	Scanning Pulsed TOF	1-1500 1-100	N/F
TRIDAR (Neptec)	uncooperative	Scanning CW Triangulation Pulsed TOF	0.5-2000	N/F
LDRI (SNL)	uncooperative	Scannerless CW AM	<45	0.25 cm
DragonEye (ASC Inc)	uncooperative	Scannerless Pulsed TOF	<1500	10 cm (bias) 15 cm (3σ)
GoldenEye (ASC Inc)	uncooperative	Scannerless Pulsed TOF	<3000	10 cm (bias) 15 cm (3σ)
VNS (Ball Aerospace)	cooperative (potentially uncooperative)	Scannerless Pulsed TOF	<5000	10-20 cm (3σ at 10 m)

As a conclusion, LIDARs are utilised over large operating distances and accuracies. Typically, LIDARs can provide higher accuracy, or have a higher operational range, towards cooperative targets than towards uncooperative targets. Analysing the types of sensors, it is noticed that scanning LIDARs have a better range accuracy compared to flash LIDARs. The latter have as benefit that they are lighter and have a slightly lower power consumption. Regarding scannerless sensors, they can provide measurements at a higher frequency, but suffer from lower resolution. A solution to this is found when combing LIDARs with high resolution cameras. This combines both the benefits of 3D information and high-resolution 2D data. The sensors discussed operate around 1500 nm wavelength, which is both safe for the eye and corresponds to a fall in the level of the solar irradiance. Their FOV is similar in elevation and azimuth (about 30 to 40 degrees). The choice of this is based on the computation load of pose estimation algorithms. For bigger FOVs, the pose estimation algorithms have to process bigger data, which increases the computational load.

A.3.2. Monocular and Stereo Cameras

Both stereo and monocular cameras are generally passive, but can be made active by including light sources to illuminate the target under poor illumination conditions. This would increase the power budget, but that would typically remain below the power budget of LIDARs.

Over the last decades, there have been three various vision-based systems developed and tested in space which are able to determine a cooperative target's full pose. These systems had as purpose to demonstrate OOS and FF capabilities.

The first system was launched in 1999 as part of the ETS-VII, developed by the NASDA [44]. The target and chaser satellite where launched together as one satellite into a circular orbit at 550 km altitude. When the orbit was reached, the target satellite was released from the chaser until a distance of about 9 km was obtained between both satellites. Then, the chaser satellite started its rendezvous procedures after docking was successfully achieved. In the docking phase, the PXS was used as navigation sensor. The PXS estimated the relative position and attitude between the docking frames of the chaser and target. The chaser contained a single visible camera and a LED array, pulsing visible light, at a 640-nm wavelength, within a cone of 30°in order to illuminate the target under poor visibility conditions. On the target's docking interface, a set of seven non-coplanar, passive, round shaped markers were installed. The PXS could measure the relative position from a distance of 9 m from the target, and the relative attitude when it was within 3 m. Just before docking, the attitude control system of the target was disabled. Then the chaser approached the target with 6-DOF control at a velocity of 0.01 m/s to align the reference frames of both docking systems.
Two types of optical interference were taken into account in the design of the PXS: sunlight interference and intersensor interference. Sunlight interference occurs when the sunlight is in direct view of the sensor, or when it is reflected via the target's surface. Intersensor interference happens when the emission of light of the sensor interfered with the other sensors on-board, or when the sensor confused its own reflectors with others. The times the sunlight interferes with the sensor depends on the orbital position of ETS-VII. In its circular orbit (550 km altitude, 35 deg inclination), the angle between the orbit plane and direction of the sun (beta angle) changes over a period of approximately 50 days, as the J2 invariant rotates the orbital plane. Direct sunlight interference occurs at the local sun time of around 18:00, and the beta angle becomes less than the FOV of the navigation sensor. The reflected sunlight interference is more difficult to avoid, due to this type of interference occurring at various beta angles and local sun times. Hence, countermeasures were taken to avoid interference of reflected sunlight:

1. Use a bandpass filter in front of the optical sensor receiver, that allows light of the same wavelength as the optical navigation sensor to penetrate.

2. Eliminate the disturbance image by taking the LED on/off images and subtracting these images.

To avoid intersensor interference, the PXS used different wavelengths for their light emitting sensors on board.

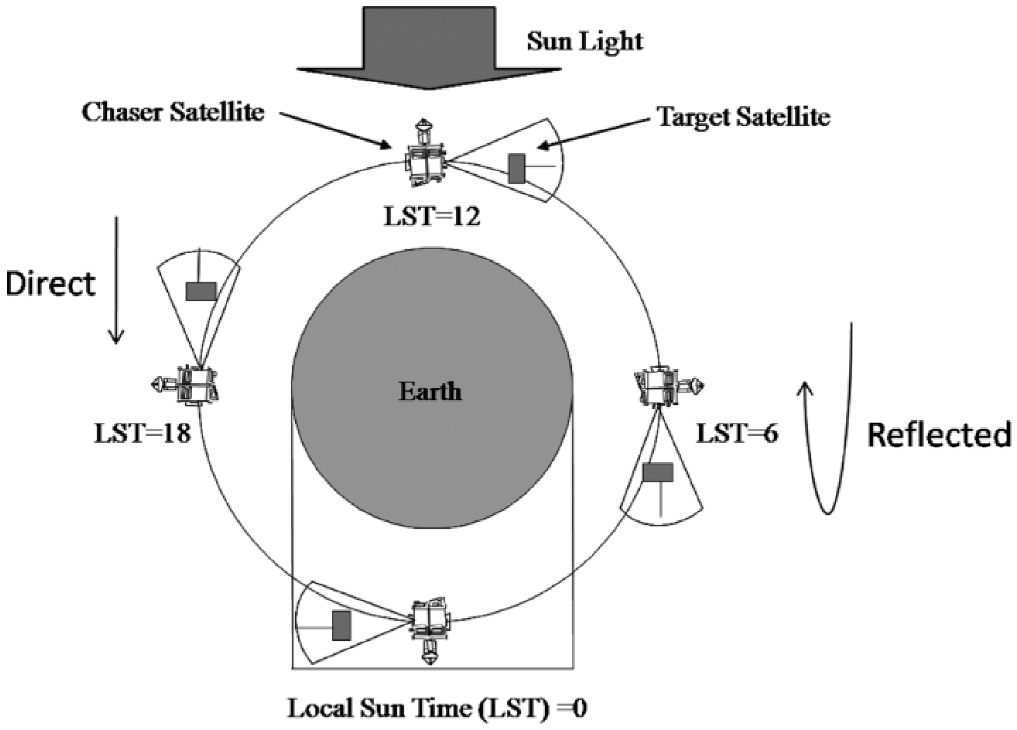


Figure A.7: Sunlight interference and local sun time [44]

The second system, the AVGS, was tested in 2005, during the DART mission, as one of the main relative navigation sensors, and in 2007, during the OE mission, as the primary docking sensor. The sensor was developed by the MSFC.
Apart from the necessary software and hardware to process the sensor images, the AVGS comprises two sets of laser diodes at different wavelengths (800 and 850 nm), and a camera that captures the return from the laser [30]. This sensor is designed to interact with a retro-reflective target, similar to the PXS. The two sets of four CCRs on the target have filters that allow one wavelength to pass through to be reflected, while blocking the other wavelength. The sensor captures a first image by firing one set of laser diodes, and then a second image by firing the other laser diodes. Then by subtracting the second image from the first, and including an intensity threshold, all background noise is eliminated. The left-over data is transformed into a set of spots which are compared to the target pattern, after which the relative position and attitude is determined between the chaser and target.
DART is launched in a circular parking orbit at 500 km. After it will perform several manoeuvres to place it 1 km behind its target. From that point, it will perform two sets of -V Bar and +R Bar manoeuvres to evaluate AVGS performance and its proximity operations algorithms [63]. the -V Bar manoeuvres include two approaches to 5 meters on the target vehicle docking axis, a simulated collision avoidance manoeuvre, and an evaluation of the maximum tracking range of the AVGS. The target vehicle of DART is MUBLCOM, an experimental communications satellite. It contains retroreflectors for use with the AVGS. During the mission, the target satellite is 3-axis stabilised in a fixed yaw position and its X-axis aligned with the velocity vector. This way the retroreflectors can be observed from the -X axis, which means the chaser can approach from behind the target.
The DART mission was not fully successful [50]. During the manoeuvres designed to evaluate AVGS performance, DART transitioned its navigational data source from GPS to AVGS as planned. During this transition, DART overshot a waypoint that would have triggered the final transition to full AVGS capability. Due to this, the range measurements supplied by the AVGS were not accurate. As a consequence, the chaser was able to steer towards its target, but eventually crashed into it as it was not able to accurately determine its distance.
In the other mission (OE mission), the AVGS was used as primary docking sensor between ASTRO, chaser satellite developed by Boeing, and NEXTSat, the target satellite provided by Ball Aerospace [25]. Both satellites were inserted in a circular orbit at 492 km as a physically linked pair. In orbit, the satellites undocked and redocked several times. First to a maximum distance of 10 m, which was successfully performed, after

which a 30 m separation was planned. This operation experienced a failure of the primary sensor computer, and both satellites ended up separately at a distance of 6 km. Eventually, the operation was recovered by sending new commands to the chaser satellite, and the AVGS successfully navigated and docked the chaser with its target. The next scenario was a fly-around of the NEXTSat at a range of 120 m and redocking. This was successfully performed and this marked the first time that a fly-around of another spacecraft was performed using primarily passive sensors, without any active exchange of relative navigation information, and without intervention or control from the ground station. The following scenario let the chaser satellite departure to 4km and return autonomously to redock. This was also was successfully achieved.

More recently, in 2010, the PRISMA in-orbit testbed was launched, which made use of the VBS developed by DTU. This system was one of the relative navigation technologies for FF applications aboard PRISMA, and is used for several experiments: Homing and rendezvous (relative distance 10 m to 100 km), proximity operations (5 to 100 m), and final approach and recede (0 to 5 m) [56]. VBS consists of two cameras on the chaser and a set of active (LED) markers on the target. The far-range camera is used for LOS-only navigation for a relative distance of 100 m up to 100 km. The short-range camera is able to carry out pose determination by imaging the markers on the target, the workable distance goes from tens of meters to a few centimetres. The visibility of the LEDs is enhanced by a band pass filter in front of the short-range camera. During this mission, the VBS performance was successful.

The characteristics of the above discussed systems are summarised in Table A.3. It shows that monocular-based sensors operating in the visible band are suitable for pose determination of cooperative targets in space in a sufficiently wide operational range.

Table A.3: Monocular systems for cooperative pose estimation [53]

System (year)	Mission (Developer)	Operational Range	FOV	Data rate	Weight	Power consumption
PXS (1997)	ETS-VII (NASDA)	below 3 m	16° x 12° (conic)	2 Hz	-	-
AVGS (2007)	OE (MSFC)	below 150 m	16° x 16° (conic)	5 Hz	9 kg	14 W stand-by 30 W tracking
VBS (2010)	PRISMA (DTU)	below 50 m	22° x 16° (rectangular)	1 Hz	0.425 kg	3 W

A main limitation of stereo cameras is that the baseline is optimised for a certain relative distance, making stereo cameras less flexible and convenient than monocular cameras with respect to operational distances, weight, power consumption and required processing power. They are however very useful for manoeuvres that involve very short relative distances between the chaser and target.

With regards to uncooperative spacecraft, VBN rely exclusively on sensors installed on the chaser spacecraft. The VBS camera described above, can also be used towards a known uncooperative target. Several uncooperative pose determination algorithms have been tested on the real images collected during the PRISMA mission [19]. Another sensor, the RNS, which consists of three monocular cameras operating at different ranges, was designed to be perform 6-DOF pose estimation of the Hubble telescope by means of vision-based algorithms only. It has been tested as an independent experiment on the space shuttle mission STS-125 [49]. Another camera has been tested in a simulation environment in a combination with uncooperative pose estimation algorithms [42]. The specifications of these cameras are listed in Table A.4

Table A.4: Examples of monocular cameras for uncooperative spacecraft pose estimation [53]

Monocular sensor	Pixel size (μm)		Pixel nr.		FOV (°)		IFOV (°)		Focal length (mm)
	hor.	ver.	hor.	ver.	hor.	ver.	hor.	ver.	
VBS camera	8.6	8.3	752	582	22.3	16.8	0.03	0.03	16.4
RNS3 camera	6.4	6.4	1024	1024	23	23	0.02	0.02	16
Simulated camera[42]	7.4	7.4	640	480	49.1	37.8	0.08	0.08	5.2

Two stereo camera systems have been tested using numerical simulations in [21] and [78]. The specifications of these cameras are outlined in Table A.5. From the camera specifications, it can be observed that a sensor must be carefully selected in order to obtain an angular resolution, like the IFOV, of a small fraction of a degree. This requirement is important in order to identify uncooperative features on the target body in the operational range of several tens of meters.

Table A.5: Examples of stereo cameras for uncooperative spacecraft pose estimation [53]

Stereo cameras	Pixel size (μm)		Pixel nr.		FOV (°)		IFOV (°)		Focal length	Stereo baseline
	hor.	ver.	hor.	ver.	hor.	ver.	hor.	ver.	(mm)	(m)
Simulated cameras [21]	27.5	27.5	512	512	22.3	16.8	0.11	0.11	13.5	0.4
Simulated cameras [78]	3.2	3.2	2048	2048	14.9	14.9	0.01	0.01	25	0.5

Monocular sensors are superior in terms mass, power budget, hardware complexity and cost. Hence, they have great suitability to be used on nanosatellites. With the increasing popularity and development of nanosatellites in the space industry, monocular sensors have the most interest for further research. Consequently, the focus of the remainder of the literature review will be on monocular-based navigation.

A.4. Monocular-based navigation

With the sensors discussed, this section seeks answers to the following questions related to monocular-based navigation:

1. *How is monocular pose estimation performed?*
2. *How is state estimation performed?*

A.4.1. Pose Estimation

Pose determination is the estimation of the relative position and attitude of a TRF with respect to a SRF. Pose estimators can be classified as model-dependent and model-agnostic architectures, depending on if the 3D model of the target is known. Model-dependent pose estimation required knowledge of the target's shape, while model-agnostic does not need any a priori information. An overview of the various architectures is illustrated in Figure A.8. The various architectures are discussed below.

Learning-based pose estimation

Learning-based pose estimation has recently obtained great popularity due to the introduction of CNNs. In general, learning based pose estimation is centred around a CNN model, which is trained for predicting outputs of a specific type, like keypoints or parameterised pose [4]. The CNN models produce a complex non-linear mapping between input images and the output information. This mapping is created and fine-tuned by training the model with a database of representative images, in this case of the target spacecraft or object. This training data is not available at the scale required for deep learning, due to the lack of representative spacecraft images. Hence, such models rely on synthetic images of the target spacecraft. Due to these factors, and the fact that learning-based pose estimation is widely and actively researched at the moment, this field would not be suitable for a thesis topic. Hence this will not be discussed to further detail in the rest of the literature study.

Model-dependent feature-based pose estimation

This architecture is based on extracting features from the 2D image, and match the features with the features in the 3D model of the target. These features can be for instance be corners or edges. Once the features are matched, they are inserted into a pose solver to obtain the relative pose. Consequently, this architecture comprises two tasks: feature detection and location, and pose solving.

IP-based feature detection is the conventional method to extract features from the image. The IP subsystems enhance or transform the initial image data and pixel information in order to expose features, like corners, edges and keypoints which can then be extracted. The extracted features are matched with known features in the 3D of the spacecraft, resulting in a matching matrix between features in the 2D input image and the 3D model. This is subsequently solved using a PnP or pose solver. Feature correspondence algorithms like RANSAC, Soft assign and feature groups, generally test multiple correspondence hypotheses when the pose is being initialised from a lost-in-space state. The computational complexity of this makes them very slow. Another challenge of IP systems are the quality of real space images, which are characterised by adverse illumination, high contrast and high image noise.

Model-dependent end-to-end pose estimation
Feature-based pose estimation described here fore consists of two distinct subsytems, feature extraction and pose solving. End-to-end pose estimation, in contrast to feature-based systems, is generally represented as one block whose input is a 2D image and output is the relative pose. This is not the case for feature-based systems, who interact with the input data at different stages in the process. If the learning-based methods are disregarded, the realisation of end-to-end pose estimation is solely obtained by exploiting TM.

Template matching is based on matching the input image with a template, stored in a database. This database is generated offline by sampling the six-dimensional pose space of the 3D model. The input image is then matched by finding correlations between the image and the stored templates by exploiting correlation techniques, for instance the sum of absolute difference or normalised cross-correlation. The estimated pose is hence based on the template that have the highest degree of similarity. The main drawbacks come from memory and computation inefficiency, by storing and cross-correlating high resolution pose samples.

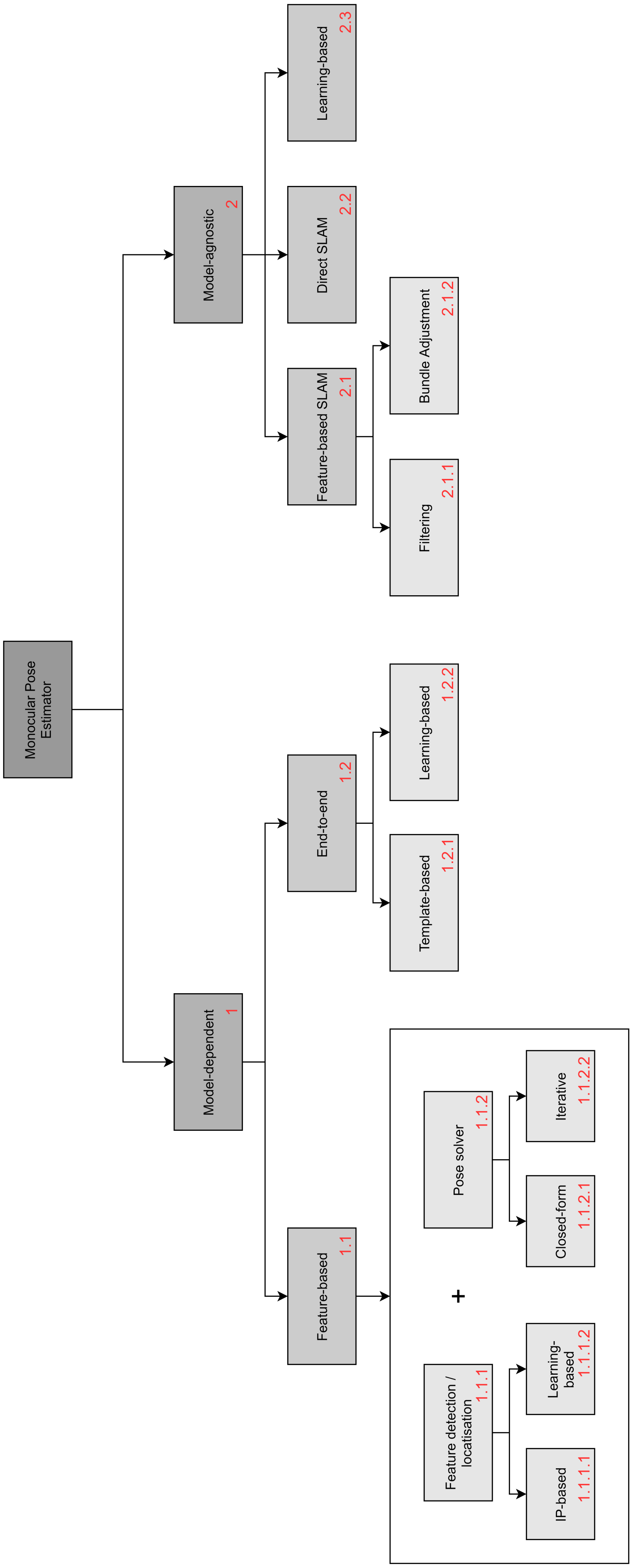


Figure A.8: Pose estimation architectures overview

Model-agnostic feature-based pose estimation

Model agnostic pose estimation is typically performed by exploiting SLAM algorithms. SLAM is the real-time process of an object moving through an unknown environment and simultaneously computing its trajectory (localisation) and building a map from the environment (mapping). In most cases however, the environment is static. This method can however also be applied in a non-static environment when it is composed of rigid objects. Hence this can be adopted to autonomous space applications involving a spacecraft flying in close-proximity of an unknown target.

Model-agnostic feature-based methods are based around the principle of extracting sparse sets of key points from input images and match them across multiple frames. By minimising the reprojection errors between these feature pairs, the relative pose between the features frame and the camera frame can be estimated. Due to the fact that modern feature descriptors are, to some extent, invariant to illumination and view-point changes, feature-based methods are more robust to inconsistencies in brightness and large view-point changes among consecutive frames, than direct methods which will be explained after [76]. On the contrary, the process of feature extraction and matching is computationally expensive. This highly limits the amount of features that can be stored and exploited by the algorithm. Consequently, the 3D reconstruction of the target is relatively sparse. Furthermore, lose of tracking can occur in low-texture environments. Hence, feature-based model-agnostic pose estimation methods have to be robust to tracking loss in order to be applied in space. A further classification of feature-based methods is found in filtering methods and methods that implement bundle adjustment.

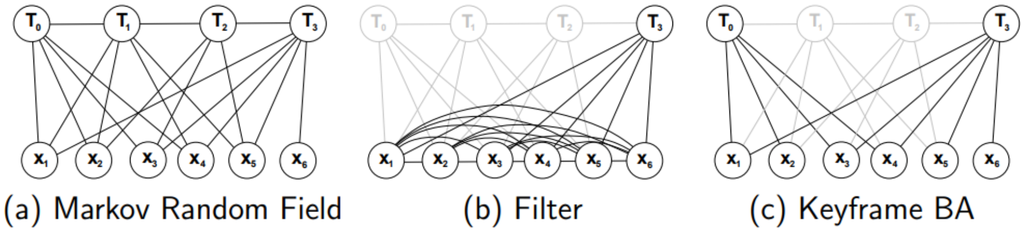


Figure A.9: (a) SLAM/SFM as markov random field without representing the measurements explicitly. (b) and (c) visualise how inference progressed in a filter and with keyframe based optimisation [70]

Filtering vs Bundle adjustment

The general problem of SLAM can be posed in terms of inference on a graph [70], as illustrated by the Markov Random Field in Figure A.9. The variables T_i represent the position of the camera at time i and the variables x_j represent the position of a feature j, static in the target frame. They are linked by image feature measurements z_{ij}: the position of feature x_j from pose T_i. These links are the edges in the graph. In real-time SLAM this network will expand continuously as new poses and measurements are added at each time step, and new features are added whenever unexplored parts of the scene are exposed to the camera. Hence, finding the full maximum likelihood solution to the graph from scratch at every time step would get computationally more expensive at every time step, and quickly impossible. To avoid this computational explosion, two key possibilities exist in real-time implementation of SLAM: filtering and BA, as illustrated in Figure A.9b and c respectively.

Filtering makes sure that all poses other than the current one are marginalised out after every frame. However, features are usually kept, since they can occur again in a future frame. This results in a more or less steady, compact graph, that is perfectly steady during repeated movements in an enclosed area. The drawback of this is that a fully interconnected graph will be obtained fast. This is because eliminations of past pose variable cause fill-in with new connections between each pair of feature variables to which is was linked. Joint potentials over all of these mutually interconnected variables must therefore be stored and updated. Propagation of joint distributions gets increasingly more computationally expensive with the amount of variables, which leads to the main drawback of filter-based SLAM: The number of features in the graph is highly limited.

BA approaches solve the graph from scratch at each time step as it expands, but the graph is sparsified by eliminating all but a small subset of past poses. This subset of past poses are called *keyframes* and are chosen intelligently or heuristically. The elimination of the remaining poses is in contrast to filtering approaches, where they are marginalised instead of discarded. Consequently, BA methods produce graphs containing more elements, but the lack of marginalisation means it will stay sparsely inter-connected, which is important for inference. Hence, BA optimisation remains efficient, even when the amount of features and measurements from the keyframes are very high. The incorporation of more feature measurements compromises the information lost from the discarded frames.

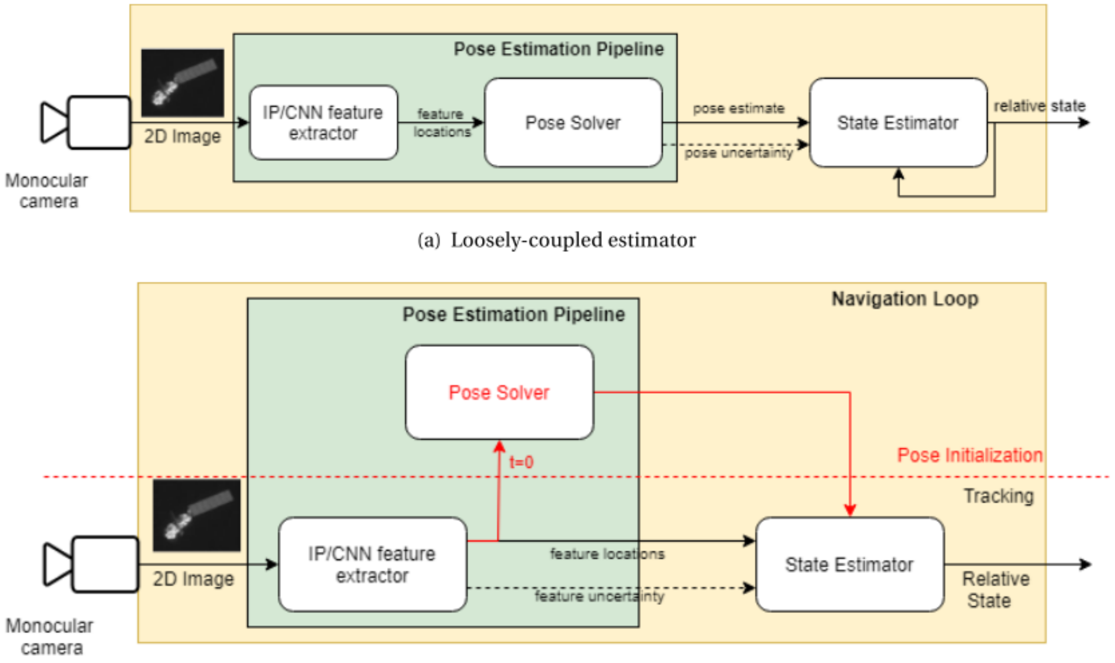
Model-agnostic direct pose estimation
In contrast to feature-based approaches, direct methods do not match image features and minimise reprojection errors. Direct methods use either all pixels (dense) of the input image, all pixels with sufficiently large intensity gradient (semi-dense), or sparsely selected pixels (sparse) [76]. Instead of minimising the projection error to determine poses, this method minimises a photometric error by making use of non-linear optimisation algorithms. This exploits all data in the input image, making it more robust in low-texture scenes. Additionally, it can generate a denser 3D reconstruction of the target. The drawbacks of direct SLAM is that they are highly sensitive to unmodelled artefacts like rolling shutter effect, camera auto exposure and gain control. Finally, direct methods under-perform in environments with rapidly changing lighting conditions.

A.4.2. State Estimation

A state estimator is included in the VBN system to track the state of the target with respect to the chaser frame. It uses the pose estimation input and/or direct input from the vision subsystem. It combines knowledge of a dynamics model and the measurement data to determine a target's state, which typically is the relative position and velocity as well as the relative attitude and angular velocity. The state estimator is also used to provide the GNC loop with a state estimate at the desired frequency, which could be higher than the operating frequency of the pose estimation pipeline. The state estimation architectures can be divided into two groups: loosely and tightly coupled types. Flow diagrams of the two types are visualised in Figure A.10.

Loosely-coupled:
If loosely-coupled state estimators are implemented, the VBN system blocks are linked in series in a navigation loop. The state estimator takes only the estimated pose from the pose estimation pipeline, and does not need additional input like the location of features etc. This is shown in Figure A.10a [4].

Tightly-coupled
Tightly-coupled estimators use the pose solver only during pose initialisation, in order to initialise the state filter. Once pose initialisation is achieved, the extracted features are directly inserted into the state estimator, bypassing the pose solver. This is illustrated in Figure A.10b. The computational complexity of loosely-coupled state estimators is lower, since they only take 6 variables as input (position and attitude), compared to tightly-coupled estimators which require a lot more input data.



(a) Loosely-coupled estimator

(b) Tightly-coupled estimator

Figure A.10: Vision-based navigation system flow for various estimator architectures [4]

A.5. Monocular Pose Estimation

The general problem of monocular-based navigation lies in the estimation of the target's pose. Hence, this section reviews the state-of-the-art of monocular pose estimation techniques, in order to find challenges suitable to further research in the thesis. The following research question will guide the review in this section:

1. *What are the challenges in the state-of-the-art of monocular pose estimation for spacecraft?*

The section is divided into pose estimation of cooperative targets, known uncooperative targets, and unknown uncooperative targets.

A.5.1. Cooperative Target

Cooperative pose determination in the field of VBN relies on artificial markers installed on the target satellite, which are either active or passive, depending on whether they emit radiation. The 3D position of these markers are known within the TRF, and they are more easily distinguished from natural features (like corners). As a result, the pose determination process includes the identification of the markers and determining its relative pose from this. The algorithms used for this depend on the selected EO sensors, which can either be monocular or 3D. The differences will be highlighted next.

For monocular pose determination, the 2D images are processed to extract the LOS of the individual centroids of the markers. This is typically done by standard IP algorithms incorporating image segmentation and feature detection. The most suitable algorithms are, of course, very dependent on the type of markers and configuration. Once the features are extracted, they are matched to the real markers by means of their individual shape or mutual geometry. As a result, they have n matches between real world and image points, or in other words, n 2D-to-3D points correspondences. From this point, the pose can be determined by solving the Perspective-n-Point (PnP) problem. It is of importance that these PnP-solutions are accurate, fast and robust, since outliers are no exception due to the various visibility conditions in space. As is noticed from Table A.6, a unique solution of the PnP problem can always be obtained from $n \geq 6$.

Table A.6: PnP problem characterisation as a function of the number of correspondences [53]

n	Upper bound of PnP ambiguous solutions	Notes
3	4	Explicit criteria can be assigned to obtain from one to four solutions
4	2	A unique solution can be analytically dervied if the four points are coplanar
5	2	A unique solution can be analytically derived unless the five points are arranged according to a set of well known critical configurations
≥6	1	A unique solution can always be analytically determined

3D pose determination algorithms extract the markers from 3D data obtained by the 3D sensor. Once the 3D positions of the markers are known in the SRF, there is not much difference in approach between monocular and 3D, as they both are given a set of n 3D-to-3D point correspondences. Their respective approaches are outlined in Figure A.11

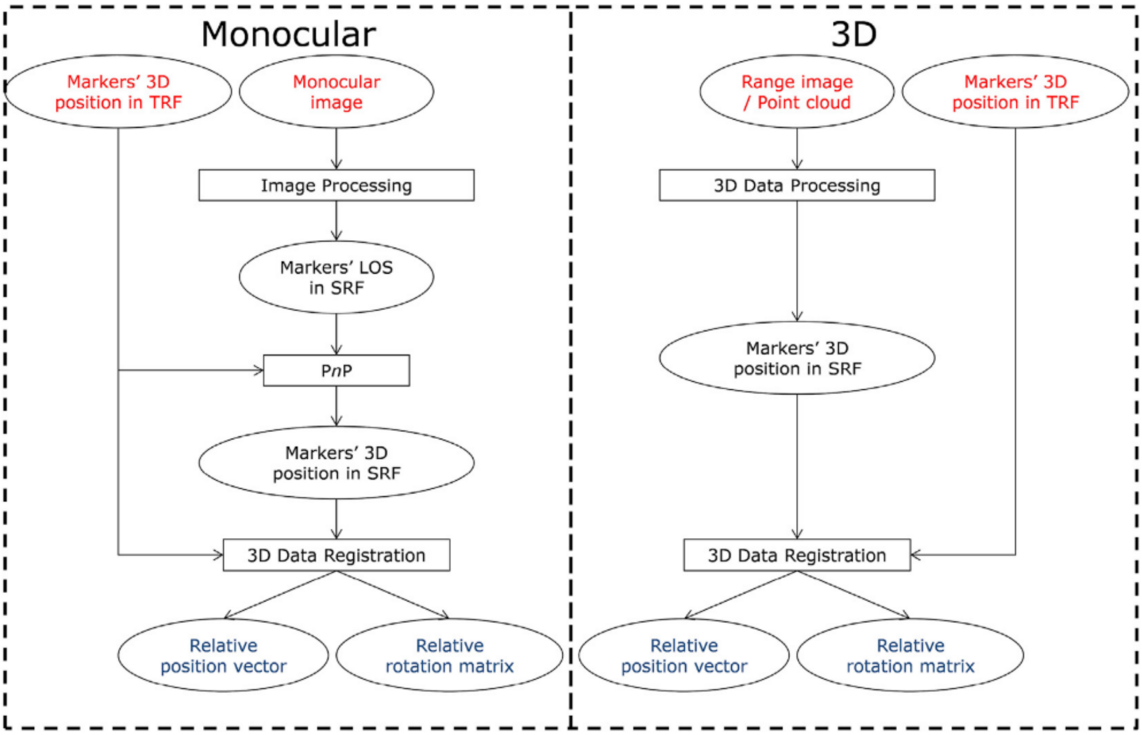


Figure A.11: Cooperative pose estimation architectures for monocular and 3D sensors [53]

Pose estimation approaches

The PXS, mentioned in subsection A.3.2, is able to easily detect the cooperative markers on the target, as it illuminates its target using active LEDs from a short distance. Pose estimation is performed using a set of coplanar markers and a non-coplanar marker. The position of non-coplanar central marker is analysed in order to compute the yaw and pitch rotation of the relative attitude. By observing the rotation of the baseline vector, formed by one of the three sets of symmetrical markers, the roll angle is determined, resulting in a full relative attitude estimation. In order to obtain the relative position, the variation of the baseline length (longitudinal axis) together with the displacements on the focal plane of the centre of the baseline (traversal axis) is measured. The performance achieved during the ETS-VII mission was centimetric error in relative position, and an error of tents of a degree in relative attitude [44].

The AVGS, the navigation system during the OE mission, exploits a background-subtraction method to detect the target's CCRs. The background subtraction works by subtracting two images, taken consecutively, where the target has been illuminated by two LEDs at different wavelengths (808-nm and 845-nm). Since the CCRs contains a filter for the 808-nm wavelength, the first image only comprises the background, while the subsequent image, illuminated by the 845-nm wavelength LED, includes potential valid locations of the CCRs. In order to distinguish the CCRs from other potential spots, the knowledge of their size and mutual geometry is taken into consideration. The pose is then finally determined by use of the Inverse Perspective algorithm. The target was equipped with two sets of CCRs, the short-range (for estimation below 10m) CCRs, and the long-range (30-150 m). In between those ranges, it used both sets of CCRs [30].

The above-mentioned systems included passive markers (CCRs). However, during the PRISMA mission LEDs were mounted on the target. This made it less prone to outliers, as LEDs are easier distinguished (due to their brightness) from the rest of the scene. The pose is solved using the P4P problem if at least four of the five LEDs are extracted. When more than than four LEDs are detected, all four-marker combinations are used to obtain pose solutions. By selecting the best fitting between the real markers and the image projection of the markers, the ambiguity of the problem is solved. The system was demonstrated to be robust against low visibility conditions, and in presence of bright objects in the scene. At 10 meter range it provided an accuracy of 1 cm, while sub-milliter and sub-degree error was accomplished in laboratory conditions at equal range [53].

Furthermore, a Smartphone Video Guidance Sensor has been designed [6]. It has as purpose to provide proximity operations and FF capabilities for small spacecraft platforms. It is based on an Android smartphone camera, flash and processor, and it can perform pose estimation of cooperative spacecraft, containing four CCRs in a predefined pattern. Considering CubeSats, a mounting possibility would be to place three CCRs coplanar on one of its surfaces, and a non-coplanar CCR placed offset in the normal direction of the surface. By implementing a brightness threshold on the pixels of the 2D image, a binary image can be created. This binary image comprises several potential spots for the CCRs. By finding the 4 spots that represent the best match to the real pattern, the CCRs can be identified. Once the CCRs are identified, the pose is estimated by exploiting a custom implementation of the collinearity equations. This has as limitation that it relies on the prior knowledge of a tentative solution of the pose. The system has been tested on the ground, by making use of a Galaxy-Nexus. The test range was from 5 to 30 meters and achieved a worst-case error of 1m and 1°for the relative position and attitude, respectively.

An OOS tailored approach for relative navigation towards small satellites in given in [73]. It uses specific markers for the target, which are designed to minimise their size and power (passive markers). These are the concentric contrasting circles [26]. Another advantage is that the centroids of the circles remain co-located under rotations and translations, hence enhancing the performance of finding image to world correspondences. Moreover, the ratio of the area between the outer and inner ring remains constant. The P4P problem is solved by exploiting a globally convergent, non-linear, iterative algorithm. It accomplished centimetre and degree level accuracy during experimental tests on ground, using the SPHERES satellites as target and chaser.

Another proposed VBN system for pose estimation of cooperative spacecraft is discussed in [57]. The markers on the target vehicle are square shaped and include each individually a predefined number of internal dots. The centroid of each marker is determined from the corners of the squares. The internal dots determine the correct matches between the centroids of the 3D markers on the image plane and their respective position in the TRF. The relative pose of the target is determined via a solution to the P3P problem. However, as can be seen in Table A.6, this would result in 4 ambiguous solutions. To solve this, a fourth marker is projected onto the image plane based on the four pose solutions, and then selecting the relative pose for which the reprojection error with respect to another marker-centroid is minimum. To remove false matches and outliers, a RANSAC algorithm is used to iteratively recompute the obtained pose. A 2-DOF experimental test was performed. the first DOF is the distance to the target, while the second is the the rotation of the target around a vertical axis. The accuracy achieved was around 1 cm in relative position, and 1°in relative attitude.

A.5.2. Known Uncooperative Target

Pose estimation to uncooperative targets is generally split up into pose initialisation/acquisition and pose tracking. During the initialisation process, no a priori knowledge of the target is used to estimate the pose/state. Once the first pose is obtained, previous pose and state estimates can be used for tracking. Often, an uncooperative pose determination process may include pose refinement as an extra step to improve accuracy and computational efficiency of the algorithm. Generally, three various ways of pose refinement can be included in the algorithm, as depicted in Figure A.12.

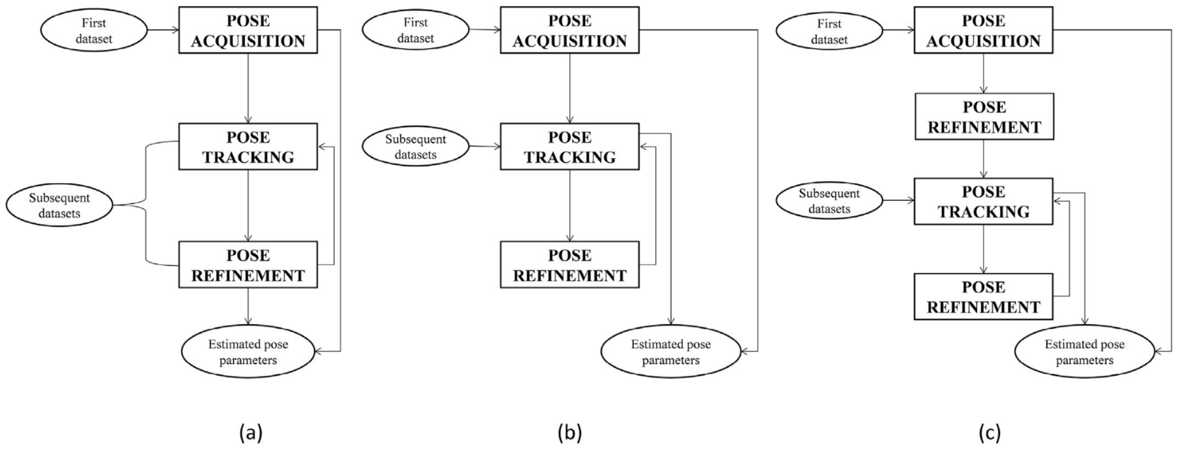


Figure A.12: Logical schemes for different uncooperative pose estimation architectures [53]

Scheme (a) in Figure A.12 uses pose refinement to update the final pose estimate, as well as to re-initialise the pose tracking algorithm for the subsequent measurement time. Scheme (b) only uses pose refinement to re-initialise the tracking algorithm. The last scheme uses two different refinement procedures, once after pose initialisation, and once after tracking.

When the target is known, it means that detailed geometrical knowledge of the target is available, or at least a simplified geometric model. Consequently, this can be handled by model-based algorithms. Model-based algorithms match data from the sensors with the same information from the target model stored on board. Features are extracted from the sensor measurements, like corners, lines, curves, contours, as well as 2D/3D point descriptors.

To acquire the pose via monocular approaches, feature-based algorithms rely on either PnP solvers, or TM. Algorithms that incorporate PnP solvers are extremely challenging due to the lack of artificial markers on the target surface. TM is the process of searching a monocular image for certain features/sections that can be matched with an assigned template. The template database is generated by sampling the 6-DOF pose space. Then the degree of similarity is computed between each template and the obtained image from the sensor. This is schematically illustrated in Figure A.13.

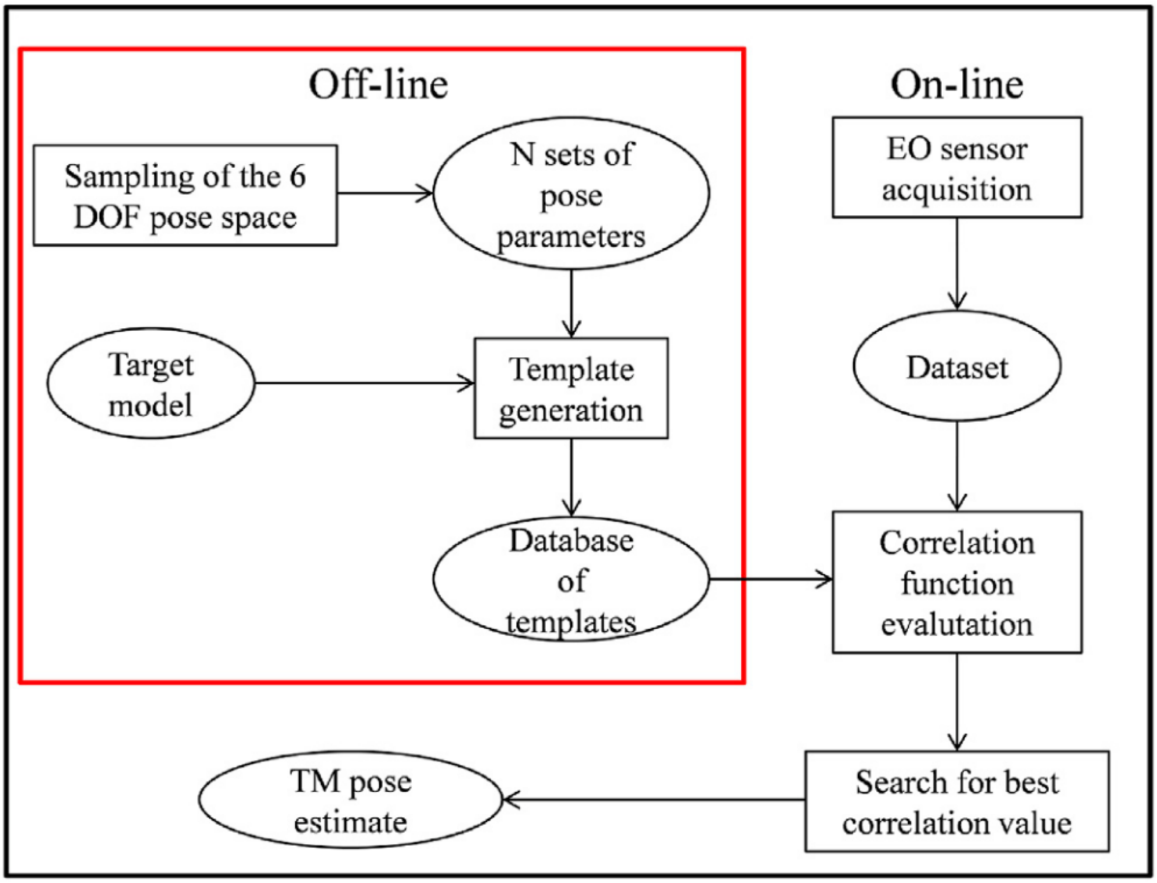

Off-line
On-line
Sampling of the 6 DOF pose space
N sets of pose parameters
EO sensor acquisition
Target model
Template generation
Dataset
Database of templates
Correlation function evalutation
TM pose estimate
Search for best correlation value

Figure A.13: Logical scheme of the classical application of the TM concept to pose estimation tasks [53]

Once the initial pose is acquired, the tracking can be achieved via the following procedure:

1. The 3D model of the target is represented as a set of features.
2. The same features are extracted in the 2D image
3. The 3D model is projected onto the image plane based on the initial pose.
4. The actual pose is obtained as the transformation that provides the best fit between the projected and image features. This latter step is typically performed by optimising a purposely defined cost function.

As an alternative to feature-based approaches for monocular sensors, appearance-based methods can be used. These methods analyse the shape and texture of the acquired datasets by making use of either Active Appearance Models (AAM) or Principle Component Analysis (PCA). However, these techniques are not very suitable for space applications due to the poorly textured scenes. Additionally, They always require an initial estimate of the pose as well as a pre-processing stage in order to create a database from a series of sample images.

With regards to 3D sensors, pose acquisition can be performed by means of global or local 3D feature descriptors. These descriptors allow matching the 3D sensor data with a consistent 3D representation of the target. An alternative to feature-based approaches are point-based techniques which can directly process 3D raw data like range images of point clouds. Since no feature extraction is required, these solutions can be computational time saving. Additionally, They are generally more robust to noise, distortions, and outliers.

Pose tracking can be performed by exploiting 3D data registration methods. These algorithms minimise a certain cost function in order to find the best rigid transformation between two datasets.

Pose determination of unknown cooperative targets is a challenging task, due to the lack of prior knowledge. Stereo cameras provide finer spatial resolutions and higher frame rate than LIDARs [53], but have a much more limited operational range.

Pose acquisition is the most critical task due to the necessity to search for an initial guess in entire 6-DOF pose space. The main problems relate to the computationally heavy algorithms, this occurs to be especially a problem for fast target-chaser relative dynamics. Additionally, the robustness to variability of pose conditions is an issue, due to space objects having typically a symmetric or nearly-symmetric shape. This results in the proneness to produce ambiguous measurements.

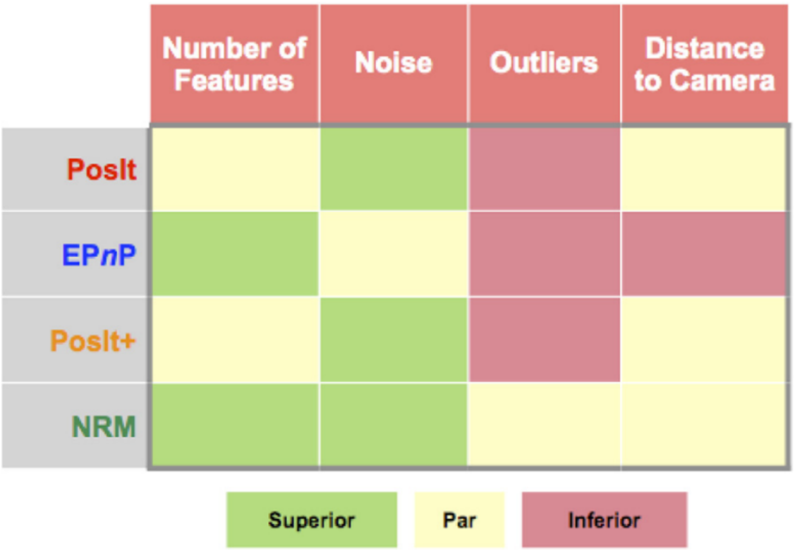
Pose estimation approaches

A fully monocular pose estimation architecture has been developed by Astrium Satellites as part of the Debritor programme [34]. The pose intialisation has been performed in two steps. A silhouette TM algorithm has been applied to a 2DOF database, which has been built offline and is organised as a hierarchical view graph, estimating two relative attitude parameters. The rest of the unknown parameters, the horizontal and vertical offsets, the scale factor, and the rotation around sensor boresight is estimated by a particle filtering algorithm. The applied strategy is characterised by an increased risk of tracking loss of the target, due to the fact that the convergence of the algorithm is split up over multiple frames. A modified edge tracking method is applied in order to perform pose tracking. This modification ensures improved robustness to outliers. The performance is assessed by means of both synthetic and real images. The real images are from the Soyuz TMA-12 spacecraft, when it was executing rendezvous with the ISS. The relative distance to the simulated targets varied from 10m to 100m, and the target's tumbling rate is set in the interval of 0°/s to 2 °/s. The reason behind this is that this is consistent with an ADR mission's final phase. Real-time performance is only to be achieved on ground. The pose estimate error was around 1°and 1% of the true relative range for the relative attitude and position respectively [58].

An approach to pose determination that makes use of perceptual grouping is found in [19]. Perceptual groups are combinations of lines and points that satisfy specific geometric constraints. This has increasing robustness compared to individual features. Especially for space applications, it is critical that unambiguous matching is performed, as spacecraft generally have a high-level of symmetry. Perceptual grouping improves this. In this approach, pose initialisation is obtained by analysing the best alignment between the perceptual groups extracted from the model and image. After this, the pose is refined by means of a multi-dimensional Newton-Raphson method. Pose tracking is done by a weighted iterative batch least-squares estimator. Images of the TANGO spacecraft, taken during the PRISMA mission, are used to assess the performance of this approach. Under the condition that the initial relative attitude is kept below 40°, the system obtained an accuracy of around 10 cm in relative position, and 10 °in relative attitude. The authors mention that the limiting factor in terms of both accuracy and reliability lays in the image processing, not the pose solving.

The RNS camera, aforementioned, provided monocular images which are exploited to test two other pose estimation algorithms: The Goddard Natural Feature Image Recognition (GNFIR) and the ULTOR Passive Pose and Position Engine [48, 49]. ULTOR is capable of doing both pose initialisation and pose tracking. Simulated target images are stored in a database, which is utilised for natural geometric structures identification on the surface of the target. It does this by means of correlation. Consequently, the pose is solved by the PnP problem. In contrast, GNFIR can only do pose tracking and tracks edges of the target, there being an edge-tracking algorithm. It creates an edge image by exploiting an edge-enhancing filter. The edge image, together with an initial estimate of the pose. the pose vector is updated. The algorithm can be run in real time, up to 3Hz. If no initial pose estimate is provided, the algorithm is capable of initialisation by predicting the motion of both the chaser and target. The algorithm re-initialises if the ratio between detected image edges and model edges gets below a threshold. The performance of both algorithms have been evaluated both in orbit and on ground. It turned out that the main problem of both algorithms is the pose initialisation. The pose initialisation was not robust enough to keep track of the target, especially for fast relative dynamics between the chaser and target. Furthermore, pose initialisation time of ULTOR was not fast enough. It took 12 to 15 seconds, this was particularly a problem due to the fast approach of the target during rendezvous manoeuvres.

From the examples of pose estimation algorithms towards known uncooperative targets, it is clear that the main challenge is the pose initialisation step. Hence, [65] provides a comparison between various PnP solvers: POSIT, coplanar POSIT, EPnP, and Newton Raphson. The simulation environment has been designed to reproduce the algorithm's input data in a realistic way. the target was the TANGO spacecraft. In these simulations, the effects of the number of point correspondences, the percentage of outlier matches, the level of pixel noise in the simulated detected features, and the inter-satellite distance have been researched. The performance of the algorithms have been assessed on computational run-time, 3D points reprojection error on the image plane, and the accuracy of the estimated pose. In Figure A.14, a qualitative illustration of the results in shown. From this it can be observed that the Newton-Raphson method (NRM) has the overall best performance. The computational load required for reaching convergence to a satisfying pose solution is not yet enough for space application. The conclusion of the paper suggests that an optimal pose acquisition solution by means of a PnP mode switcher, which autonomously combines the PnP solvers in order to accelerate the iteration process.



	Number of Features	Noise	Outliers	Distance to Camera
PosIt	🟨	🟩	🟥	🟨
EP*n*P	🟩	🟨	🟥	🟥
PosIt+	🟨	🟩	🟥	🟨
NRM	🟩	🟩	🟨	🟨

🟩 Superior 🟨 Par 🟥 Inferior

Figure A.14: Comparative assessment results for simulations from various PnP solvers [65]

The choice of the pose estimation algorithms should include the target's geometry, like shape and size. An algorithm for pose initialisation tailored for cylinder shaped spacecraft is given in [42]. By knowledge of the geometry, a course estimate of the symmetric axis direction and the relative translation vector, which has 4 DOF, can be achieved. The remaining DOF are determined using the non-symmetric components of the spacecraft, like antennas and solar arrays. After testing the performance of the algorithm on real images and synthetic images, a considerable problem was observed. The relative position accuracy reduced as the rotation of the cylinder with respect to its traversal axes increased. Finally, this algorithm suffered from partial surface occlusion, which could prevent shapes to be detected.

An additional method that takes benefit from knowing the precise target geometry is proposed in [18]. By detecting image lines with sub-pixel accuracy, heuristics generate a list of correspondences with the lines of the 3D model of the target. The relative pose is then generated by minimising two least-square error functions, satisfying specific geometric constraints. The pose is hereafter refined by a Gauss-Newton minimisation. The performance evaluation resulted in an average rotation error under 5°, and an translation error of approximately 2% of the target range with a maximum of 10m. It should however be mentioned that the algorithm required a large computational time (1 min), as well as an initial guess for the relative pose.

A.5.3. Unknown Uncooperative Target

Pose estimation where no a priori knowledge of the target is required, is regarded as the ideal method to be implemented on ADR missions and asteroid exploration. However, this is also the most challenging type of pose estimation. Monocular based SLAM for pose estimation towards unknown targets have not yet been tested in space. Several novel architectures have been simulated or experimentally tested on ground. They are listed below.

Pose estimation approaches

In 2018, A study was performed in the application of ORB-SLAM, a feature-based SLAM, to spacecraft relative navigation [20]. The SLAM algorithm exploits ORB features, which are image features invariant to rotation, based on BRIEF binary descriptors. They have as advantage that they can be rapidly extracted, significantly

decreasing the computational time. The algorithm incorporates robustness by initialising a great amount of keypoints, and discarding the weaker keypoints. ORB-SLAM also includes a bag-of-words method for place recognition, automating the loop closure step. So when a recognisable scene is viewed for a second time after many image frames have passed, it executes a full bundle adjustment. This reduces the error in the estimation by eliminating the accumulated drift between successive keyframes.
It is important to note that monocular-based pose estimation gives a scale-ambiguous pose when only a monocular camera is used. Also, in this study they assumed all detected features to be part of the rigid body, meaning they have a fixed position in the target frame. This has not been tested on images with the Earth in the background for example, hence its robustness to such images should be further tested. Also it was observed that the algorithm had difficulties initialising the map when the target is far away, due to the relative small translational motion between frames. So for successful initialisation, large translational motion would benefit the initialisation process, since this induces a parallax in the image plan. Additionally, loop closure of the algorithm could not be achieved repeatedly in its experiment, where the chaser completed several revolutions around the target.

A direct-SLAM is a SLAM method that does not rely on features, but directly uses the intensity of the pixels for pose estimation. One of these methods is LSD-SLAM, which is also studied for relative navigation towards unknown uncooperative targets in 2019 [39]. The method is however only tested on an experiment at which the relative motion only had one degree of freedom (one rotation). It was found to be only accurate on the short-term and close range motion. Also for this algorithm, loop closer was not been able to be tested. Additionally, this has been tested with just a plain black background only.

A filter-based feature SLAM for monocular pose estimation of unknown targets is investigated in [13]. This algorithm, called SEPS, is monocular based, but assumes a single beam LIDAR measurement to recover the scale of the restructured map, as this is not possible for solely monocular vision. SEPS architecture is shown in Figure A.15

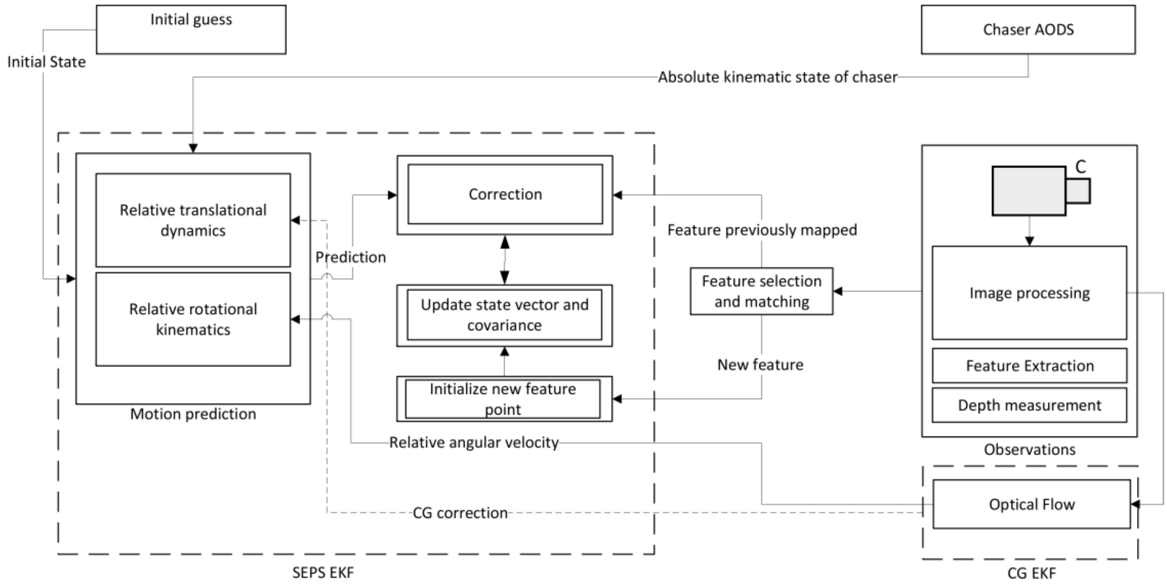

Initial guess
Initial State
Chaser AODS
Absolute kinematic state of chaser
Relative translational dynamics
Relative rotational kinematics
Motion prediction
Prediction
Correction
Update state vector and covariance
Initialize new feature point
Feature previously mapped
Feature selection and matching
New feature
C
Image processing
Feature Extraction
Depth measurement
Observations
Relative angular velocity
Optical Flow
CG correction
SEPS EKF
CG EKF

Figure A.15: SEPS architecture

Target features are extracted by means of a monocular camera, and the Harris corner detector algorithm. This paper however only reviews to the pose determination algorithm on synthetic images. the feature detection algorithm may not be effective for actual spacecraft images due to its high contrast, low resolution, and low signal-to-noise ratio.
The angular velocity is not included in the state vector, but is directly obtained from optical flow. As a result, they do not need the moment of inertia estimation in the filter state, which lowers the uncertainty of the filter estimate. Also, it lowers the computational burden, due to the size reduction of the filter state. In their paper, they use the EKF as a nonlinear filter.
In contrary to previous implementations of optical flow, which assume either a fixed camera or a fixed world,

they break up the optical flow measurement into different inertial components that allow to separate out the own motion of the camera from the rotation and relative velocity of the target.

Another filter-based algorithm is proposed in [33]. This algorithm decomposes the target state estimation into relative pose estimation and feature point position estimation based on relative pose. The relative pose of the target is estimated by EKPF, and EKF estimates the feature point position of the target. This algorithm is also based on a monocular camera, and a single-beam LIDAR measurement to obtain depth information. Also this algorithm is only tested on synthetic target images, generated by OpenGL. The relative pose consists of the attitude, position, angular velocity and velocity.

A filter-based algorithm is also described in [32]. This algorithm had some restrictions. It can only recognise spacecraft if the image is not saturated by sunlight and if the background is dark. Also, temporary occlusion of reference features can occur during motion, which affects the accuracy. The provided algorithm has not been tested successfully thoroughly, as it has only been able to track for 3 seconds.

An EKF-SLAM approach has been discussed in [67]. This architecture analyses two cases: one where the target has a constant rotational velocity around one principal axis, and one where the target motion is nutational, but the MOI distribution is known. The tests demonstrated the working principle on synthetic images, however no verification has been performed on real spacecraft images.

A.6. Knowledge Gaps Identification

The purpose of the literature study is to identify knowledge gaps suitable for a thesis research. The literature study started off with a global overview of the missions related to proximity operations, and an outline of VBN systems. Afterwards, typical sensors for VBN systems were analysed. From this it was deduced that monocular cameras have many advantages over 3D sensors, like lower cost, mass, power consumption, and hardware complexity, yet monocular-based pose estimation is the most challenging due to the lack of 3D data. Consequently, this has been explored to identify further challenges. It is concluded that a combination of monocular vision and pose estimation of unknown targets is a relatively novel field of research and brings the most challenges. With the recent advancements in monocular-based SLAM, like the development of ORB-SLAM3 [12], monocular-based pose estimation towards unknown targets is deemed an interesting and suitable field of research for the thesis. In this field, the following knowledge gaps were identified:

KG-1 Analysis of unknown target pose estimation, based on a monocular camera and an IMU
In the current state-of-the-art, monocular cameras have been used in combination with a LIDAR sensor in order to obtain a scale-unambiguous pose estimate of the target. This has been done in [13] and [33] for instance. However, LIDAR sensors require a relatively large amount of power, have a high mass, and a high hardware complexity, which makes them unsuitable for nano-satellites. Hence, it would be interesting to look at alternatives. An appealing alternative is the combination of a monocular camera and an IMU. This combination has not been analysed before for pose estimation towards unknown targets. Moreover, ORB-SLAM3 [12] proofs the feasibility of a monocular camera supported by an IMU.

KG-2 Analysis of unknown target pose estimation based on real space imagery (including background objects like Earth, stars, and other satellites)
Real space imagery is the ultimate ground-based simulation to verify the working principle of the algorithm. The current state-of-the-art test the performance of the pose estimators based on synthetic images. There are, however, large differences between synthetic and real space imagery. Real space images are characterised by high contrast, high noise, and low resolution. Additionally, objects might be in the background of real space images. This could for instance be the Earth, the stars or other satellites. Without depth sensors, it is challenging to distinguish background objects from the target. As an example, in feature-based pose estimation, features might be detected which do not belong to the target but to the Earth. Consequently, it is essential that pose estimators are also tested on real space imagery, including background objects.

KG-3 Analysis of pose estimation of multiple unknown targets/in different scenarios
From the found literature, no architectures have been tested on multiple targets or scenarios to verify its robustness. This is an important aspect of the performance, as the pose estimation accuracy can be dependent on the target's appearance, the relative distance, illumination conditions etc. It is beneficial to know what influences the performance and under what conditions the pose estimator might not work accordingly.

KG-4 Achievement of robustness to loss of pose tracking
Tracking loss has been an overall problem in the field of uncooperative target pose estimation. This is for instance observed in [20]. Features are lost when the target is viewed from different angles. This could potentially result in tracking loss, if not enough new features are tracked. As a result, robustness to loss of pose tracking is of significant importance to create a reliable pose estimator.

KG-5 Analysis of complete state estimation
Many of the current state-of-the-art pose estimators are not implemented in a state estimator. [20], [39], and [32] are examples of pose estimation algorithms that are not implemented in a full state estimation system. The full state of a target is defined as the three-dimensional relative position, attitude, translational velocity, and angular velocity. Generally, full state tracking is desired or even required. Therefore, it would be of use to analyse what the most optimal way would be to incorporate the pose estimator in a state estimator. Additionally, the state estimator could set some accuracy requirements on the pose estimator, in order for it to work. This could influence the design of the pose estimator.

KG-6 Analysis of hardware specifications for pose estimation
Hardware requirements are not discussed thoroughly in literature. The type of processor is generally specified on which the simulations are performed, but no additional computational load is mentioned. Also, camera or IMU/LIDAR specifications are not discussed anywhere. Since the accuracy of the sensors influences the performance of the pose estimator, it is necessary to obtain a general understanding of the relation between sensor specifications and pose estimation performance. This could set hardware requirements for missions that require a certain pose estimation accuracy. Analysing the hardware requirements is of specific importance for nanosatellites, which typically have limited volume, mass, and power budget.

B

Software Workflow

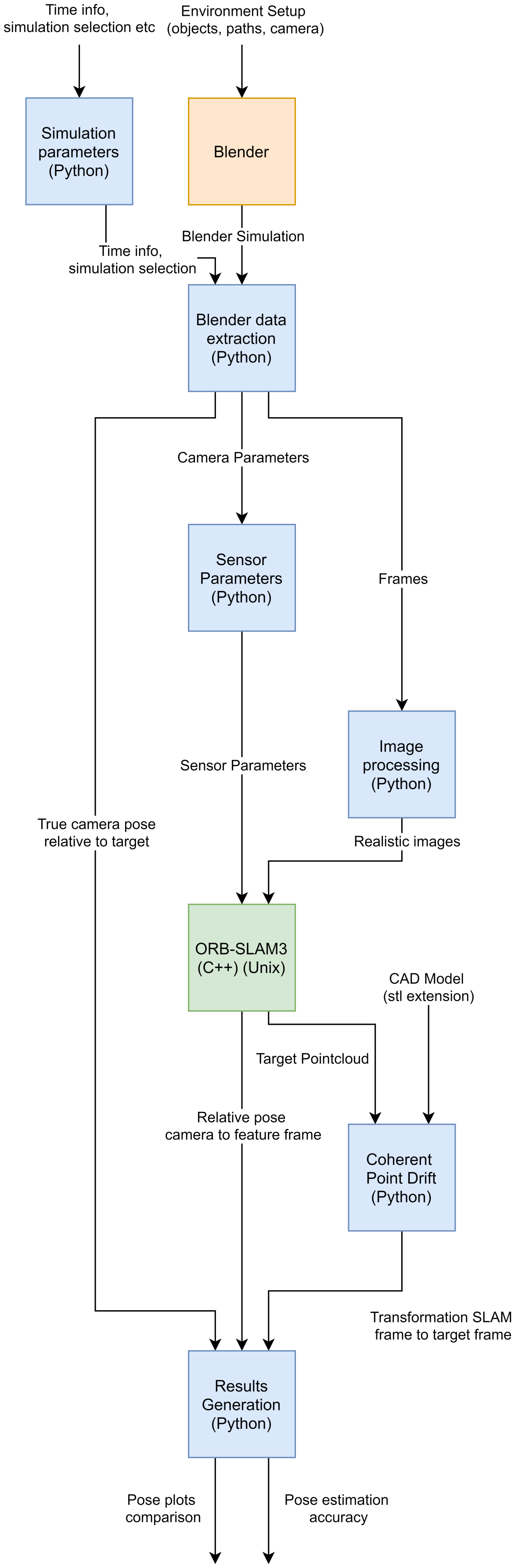


Figure B.1: Software Work Flow

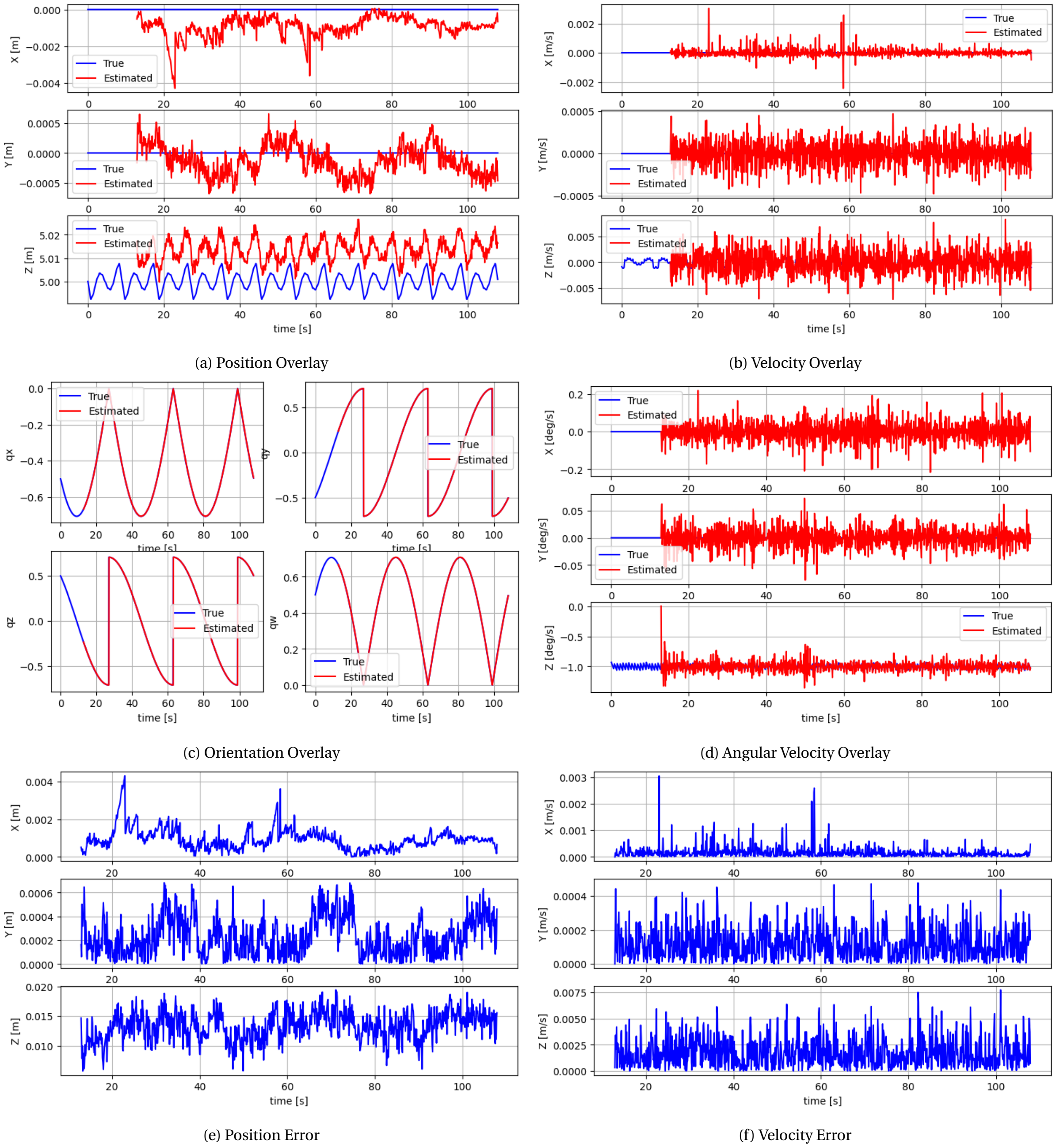
C

Additional Results

C.1. RM VEL 10

Table C.1: rM VEL 10 Results

Error	Value	Unit
Position	0.014094	[m]
Orientation	0.297616	[deg]
Velocity	0.002229	[m/s]
Omega	0.081873	[deg/s]



(a) Position Overlay

(b) Velocity Overlay

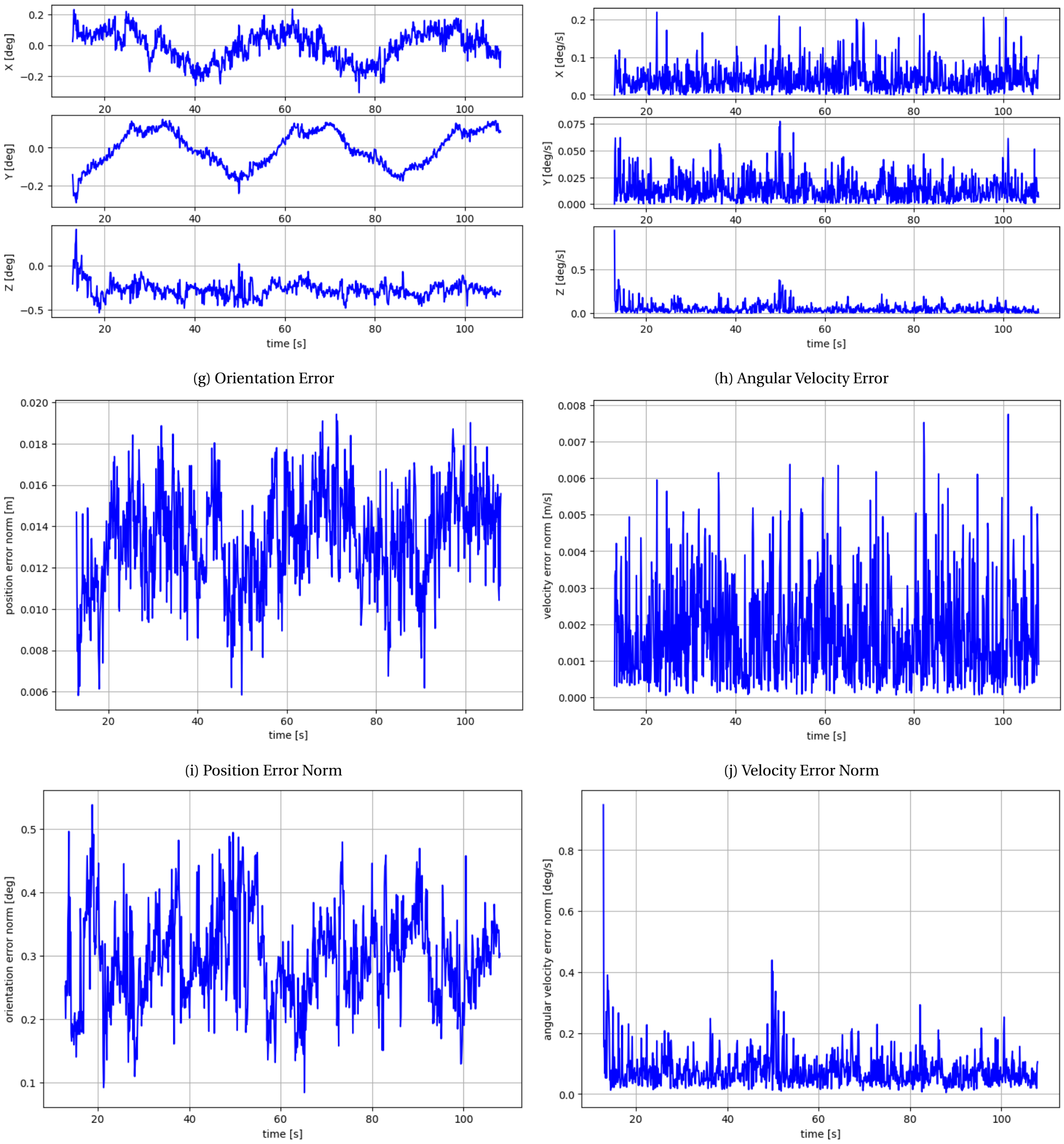
(c) Orientation Overlay

(d) Angular Velocity Overlay

(e) Position Error

(f) Velocity Error

Figure C.1: RMVEL10 results



(g) Orientation Error

(h) Angular Velocity Error

(i) Position Error Norm

(j) Velocity Error Norm

(k) Orientation Error Norm

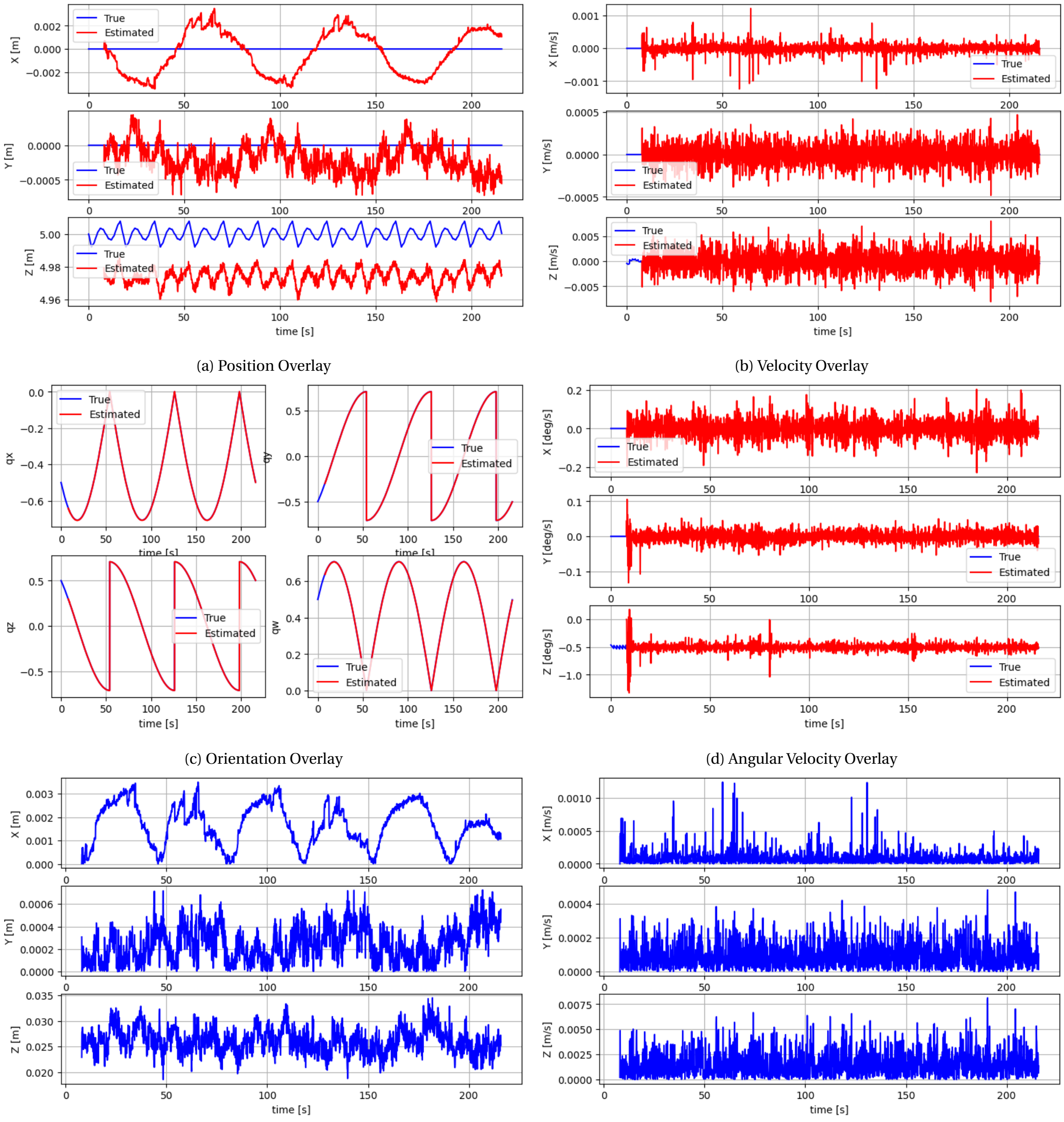
(l) Omega Error Norm

Figure C.1: RMVEL10 results

C.2. RM VEL 5

Table C.2: RM VEL 5 results

Error	Value	Unit
Position	0.025919	[m]
Orientation	0.321867	[deg]
Velocity	0.002064	[m/s]
Omega	0.072107	[deg/s]



(a) Position Overlay

(b) Velocity Overlay

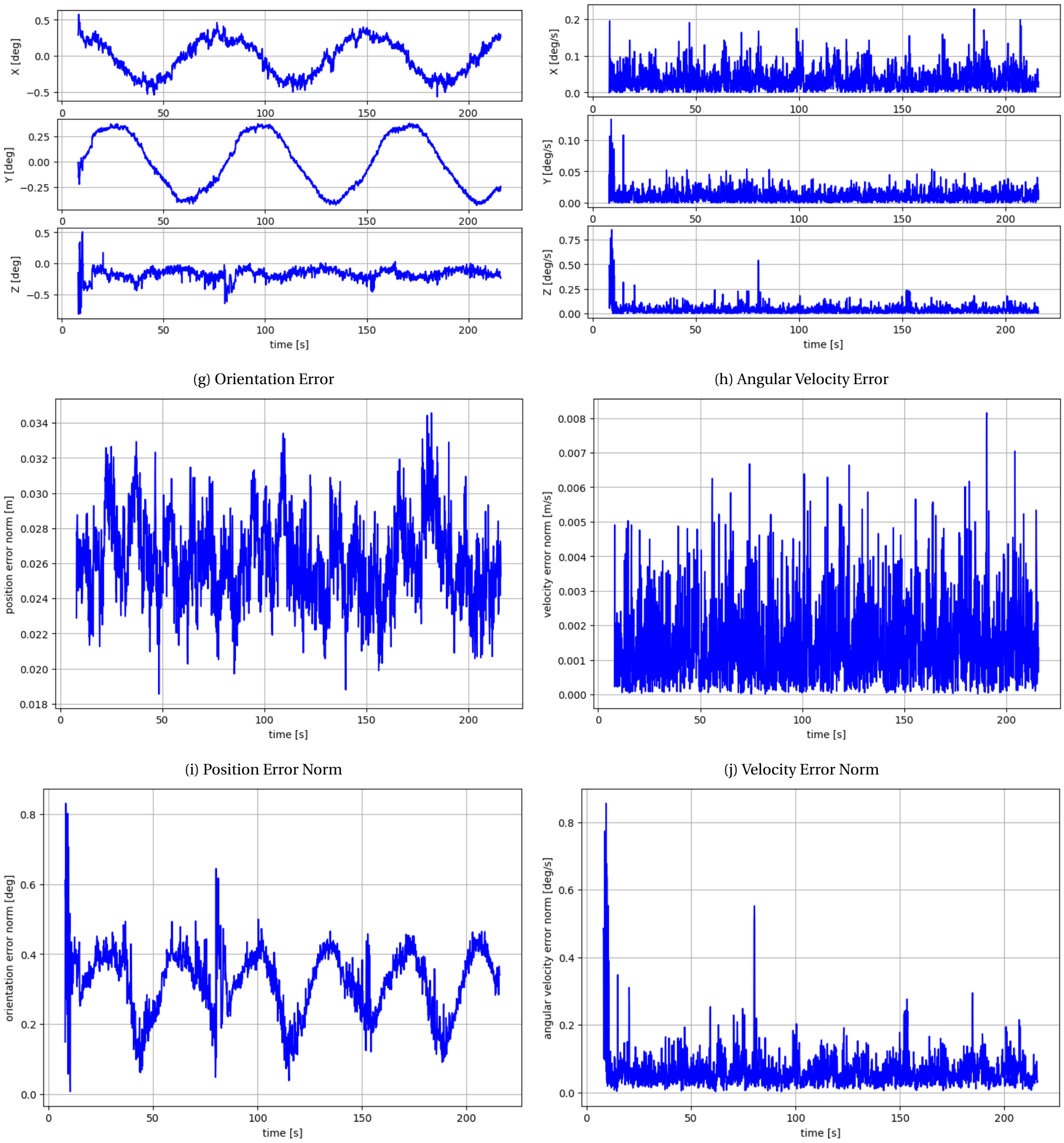
(c) Orientation Overlay

(d) Angular Velocity Overlay

(e) Position Error

(f) Velocity Error

Figure C.2: RMVEL5 results



(g) Orientation Error

(h) Angular Velocity Error

(i) Position Error Norm

(j) Velocity Error Norm

(k) Orientation Error Norm

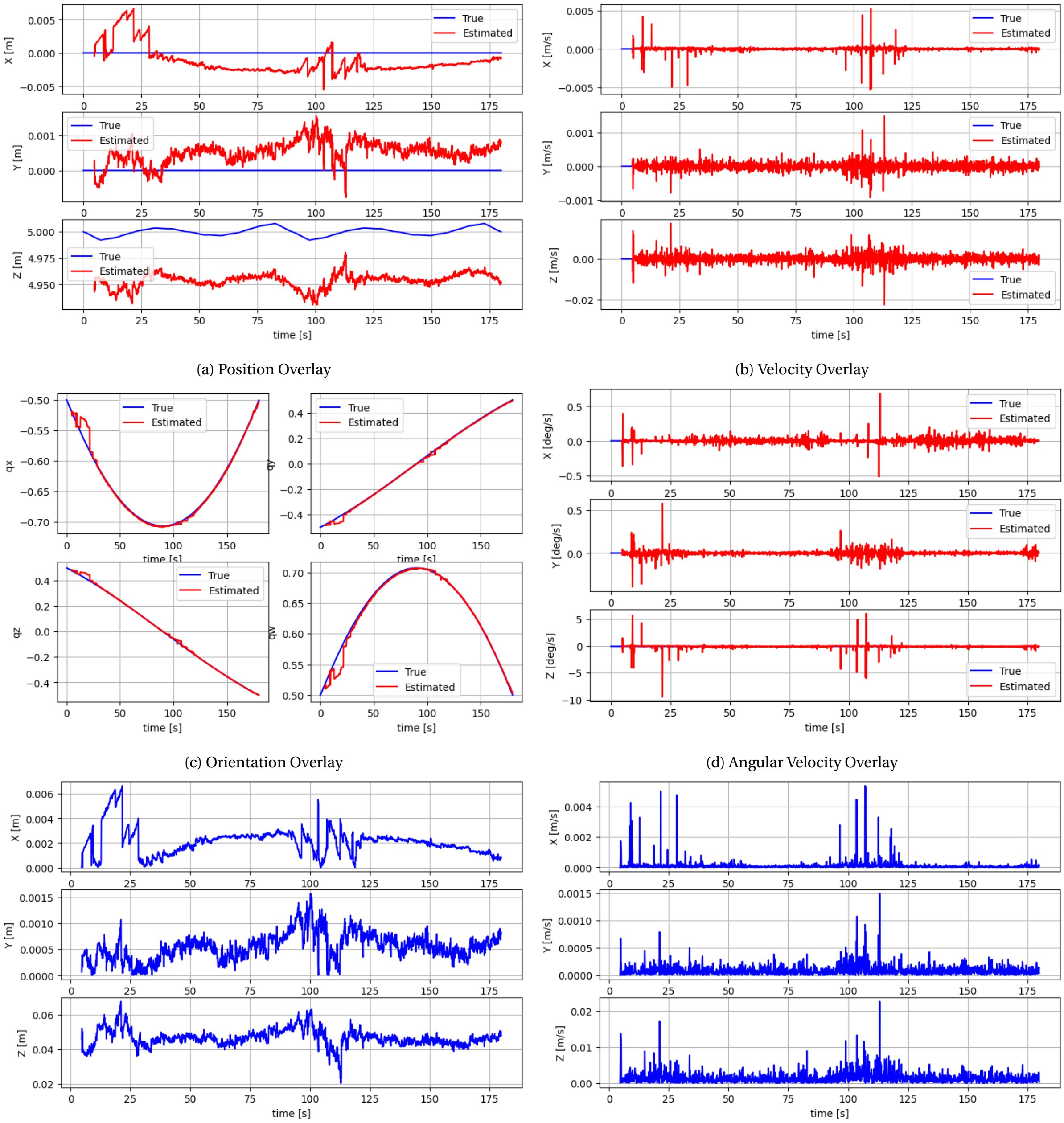
(l) Omega Error Norm

Figure C.2: RMVEL5 results

C.3. RM VEL 1

Table C.3: RM VEL 1 Results

Error	Value	Unit
Position	0.046171	[m]
Orientation	1.221062	[deg]
Velocity	0.002383	[m/s]
Omega	0.493522	[deg/s]



(a) Position Overlay

(b) Velocity Overlay

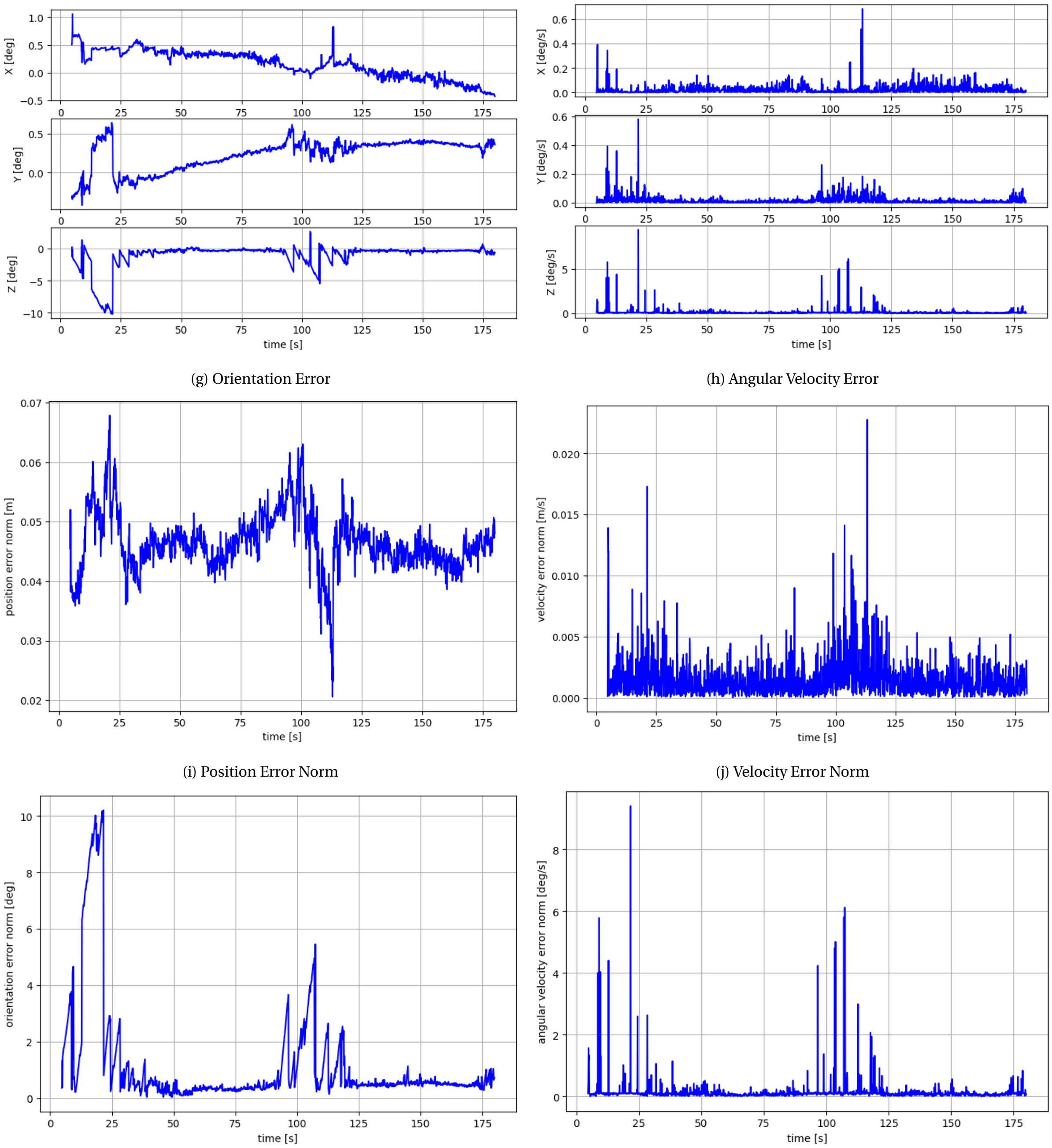
(c) Orientation Overlay

(d) Angular Velocity Overlay

(e) Position Error

(f) Velocity Error

Figure C.3: RMVEL1 results



(g) Orientation Error

(h) Angular Velocity Error

(i) Position Error Norm

(j) Velocity Error Norm

(k) Orientation Error Norm

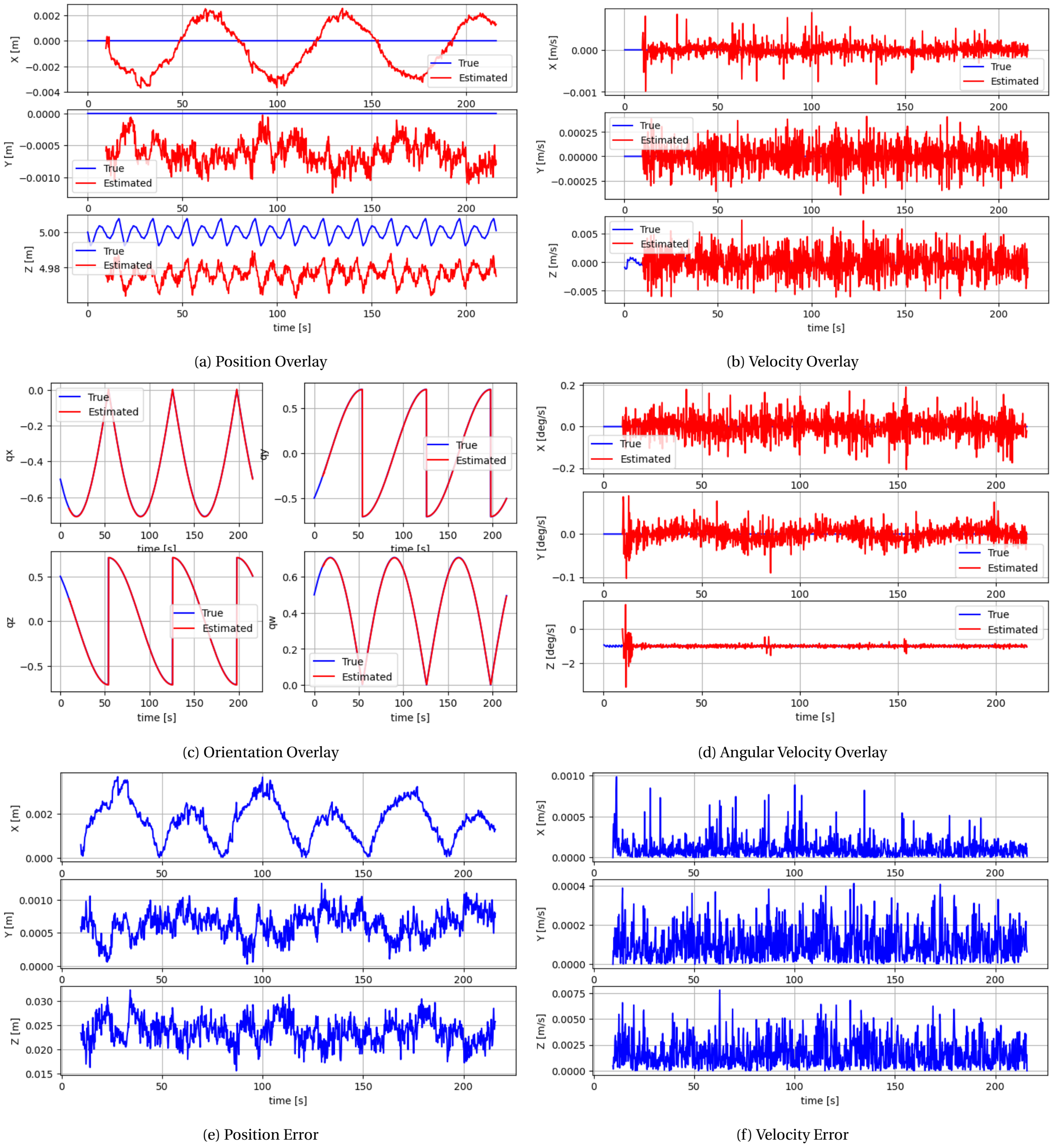
(l) Omega Error Norm

Figure C.3: RMVEL1 results

C.4. HW FR 5

Table C.4: HW FR 5 Results

Error	Value	Unit
Position	0.023468	[m]
Orientation	0.442494	[deg]
Velocity	0.002246	[m/s]
Omega	0.085677	[deg/s]



(a) Position Overlay

(b) Velocity Overlay

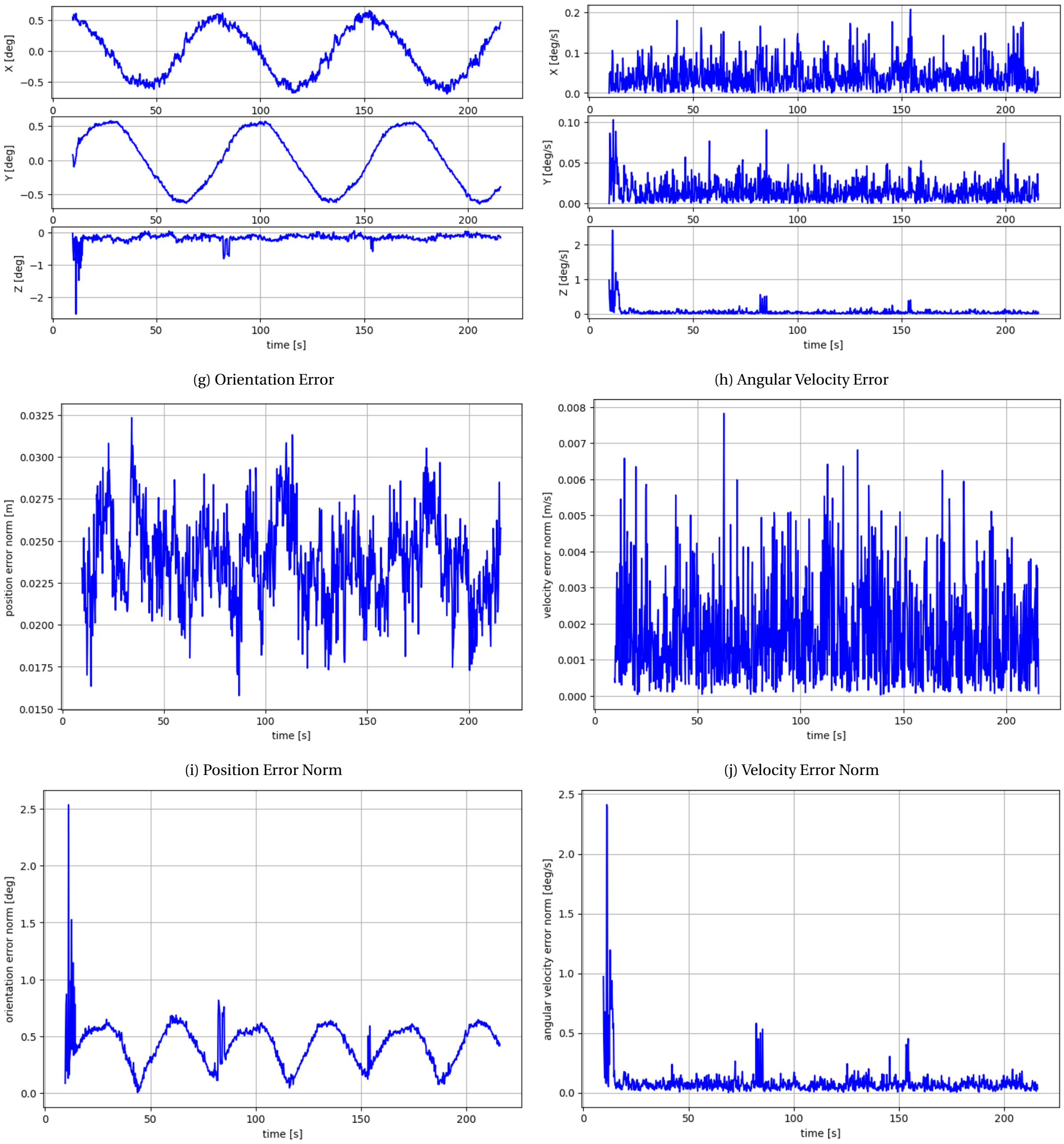
(c) Orientation Overlay

(d) Angular Velocity Overlay

(e) Position Error

(f) Velocity Error

Figure C.4: HW FR 5 results



(g) Orientation Error

(h) Angular Velocity Error

(i) Position Error Norm

(j) Velocity Error Norm

(k) Orientation Error Norm

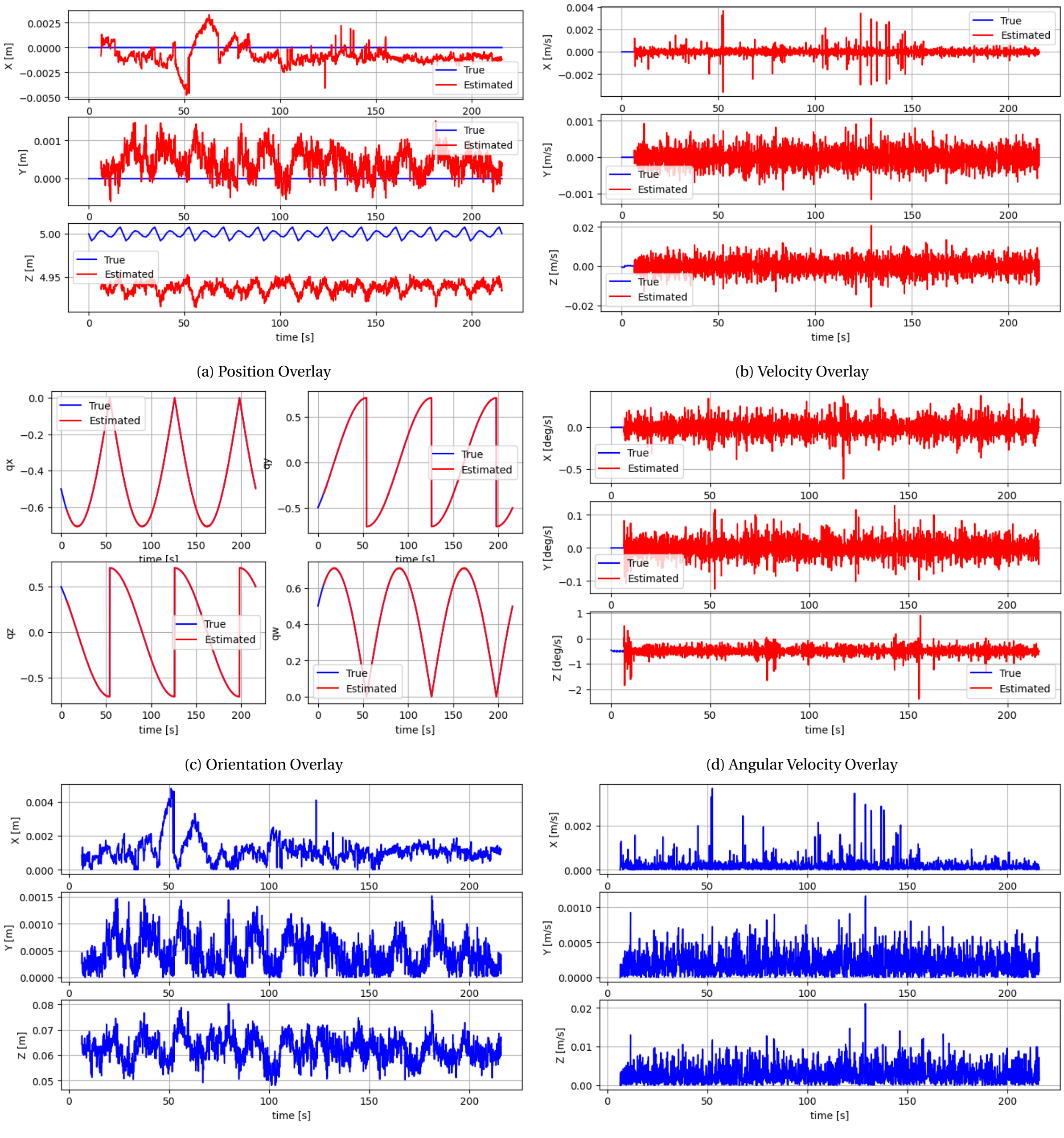
(l) Omega Error Norm

Figure C.4: HW FR 5 results

C.5. HW RES mid

Table C.5: HW RES mid Results

Error	Value	Unit
Position	0.061903	[m]
Orientation	0.265365	[deg]
Velocity	0.004093	[m/s]
Omega	0.202945	[deg/s]



(a) Position Overlay

(b) Velocity Overlay

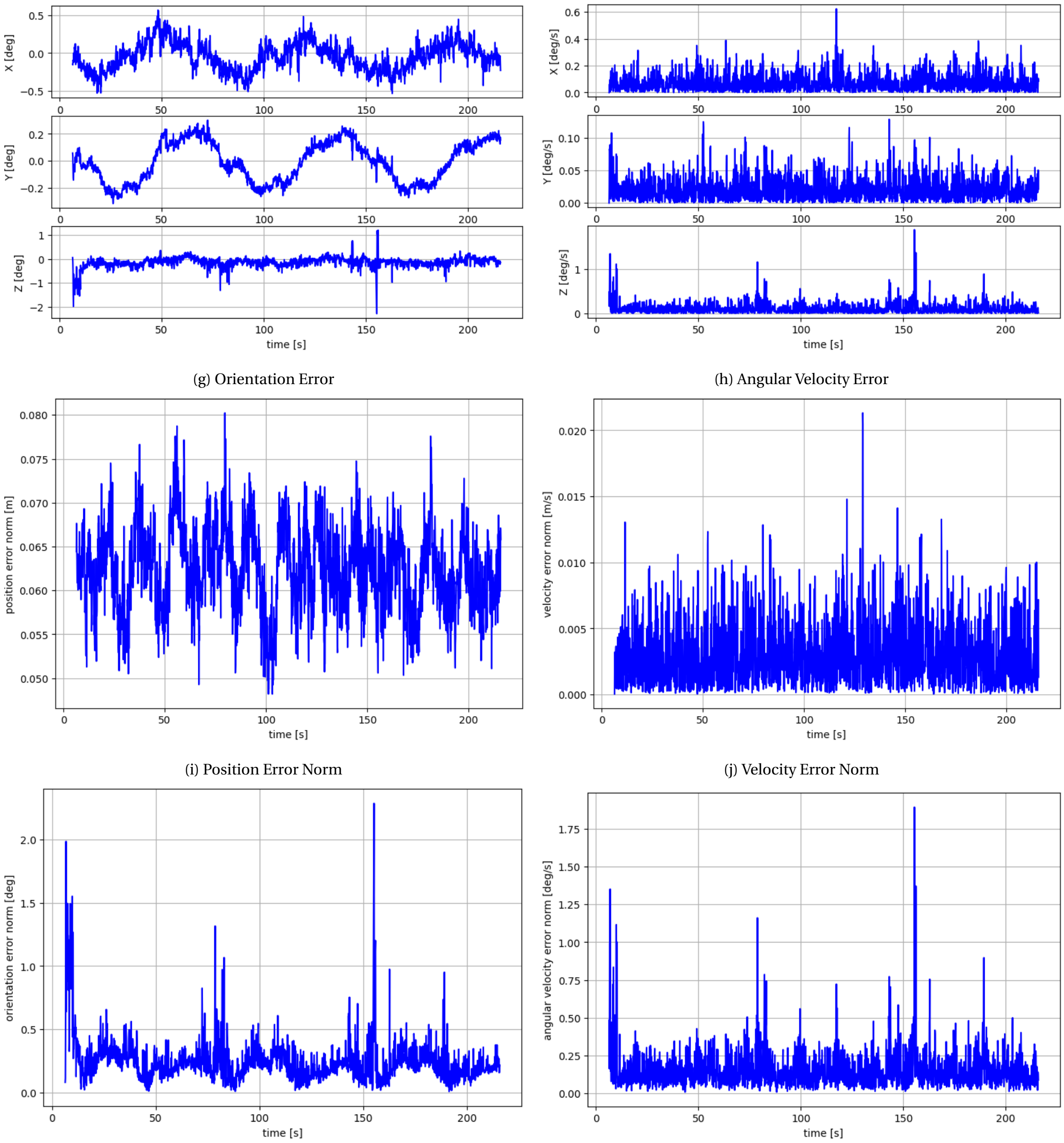
(c) Orientation Overlay

(d) Angular Velocity Overlay

(e) Position Error

(f) Velocity Error

Figure C.5: HWRESmid results



(g) Orientation Error

(h) Angular Velocity Error

(i) Position Error Norm

(j) Velocity Error Norm

(k) Orientation Error Norm

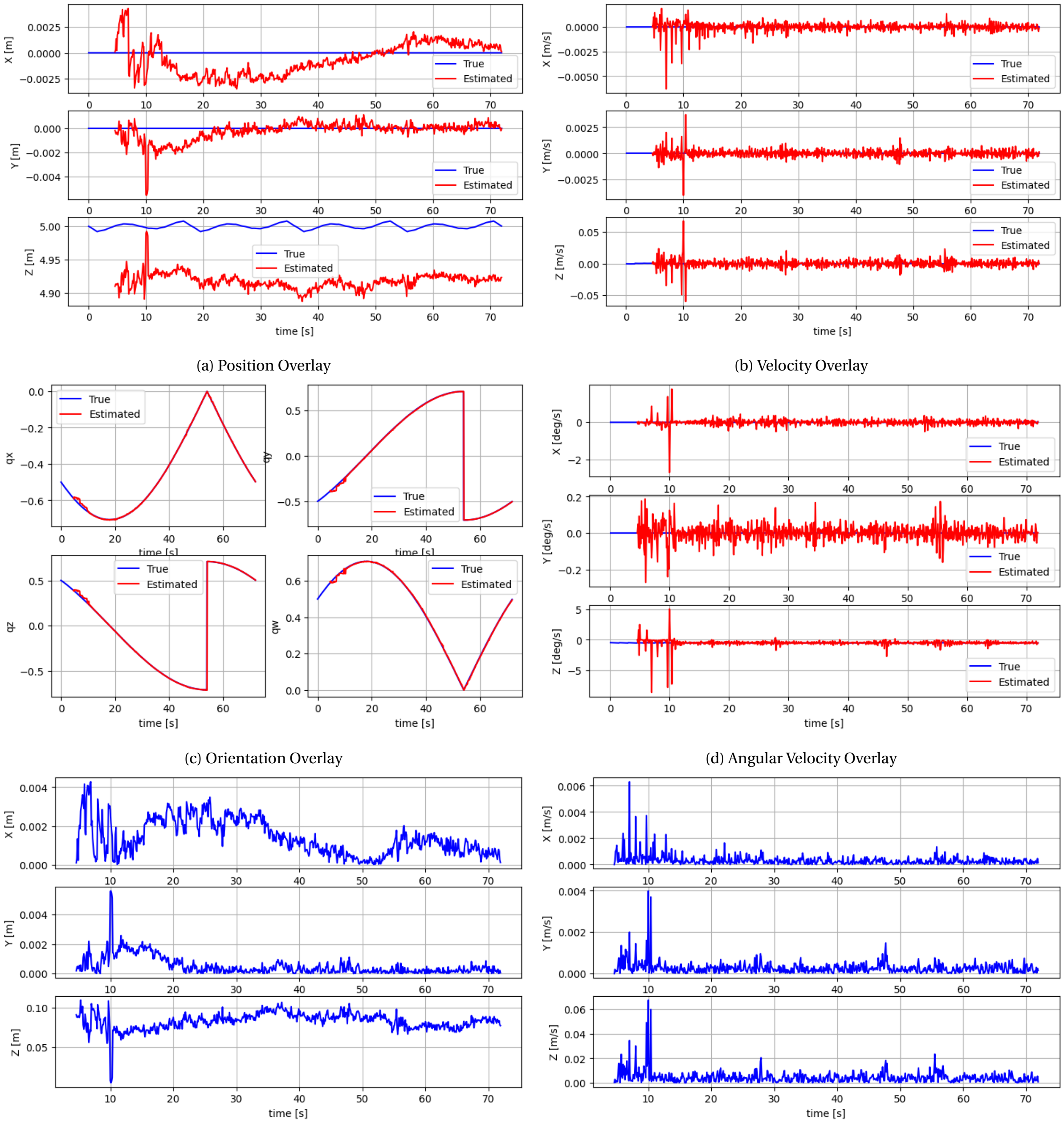
(l) Omega Error Norm

Figure C.5: HWRESmid results

C.6. HW RES low

Table C.6: HW RES low Results

Error	Value	Unit
Position	0.083513	[m]
Orientation	0.468512	[deg]
Velocity	0.005159	[m/s]
Omega	0.298647	[deg/s]



(a) Position Overlay

(b) Velocity Overlay

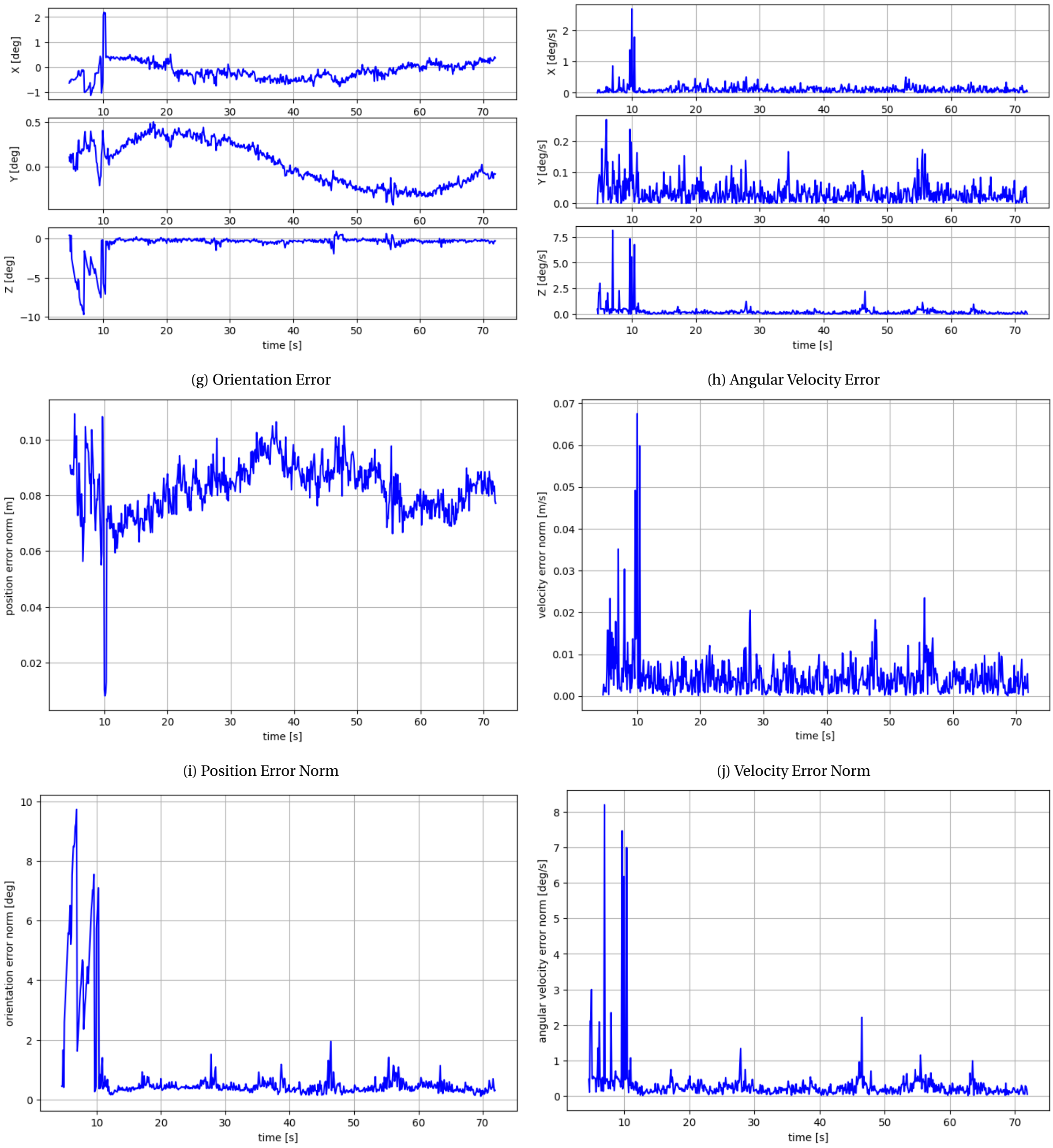
(c) Orientation Overlay

(d) Angular Velocity Overlay

(e) Position Error

(f) Velocity Error

Figure C.6: HWRESlow results



(g) Orientation Error

(h) Angular Velocity Error

(i) Position Error Norm

(j) Velocity Error Norm

(k) Orientation Error Norm

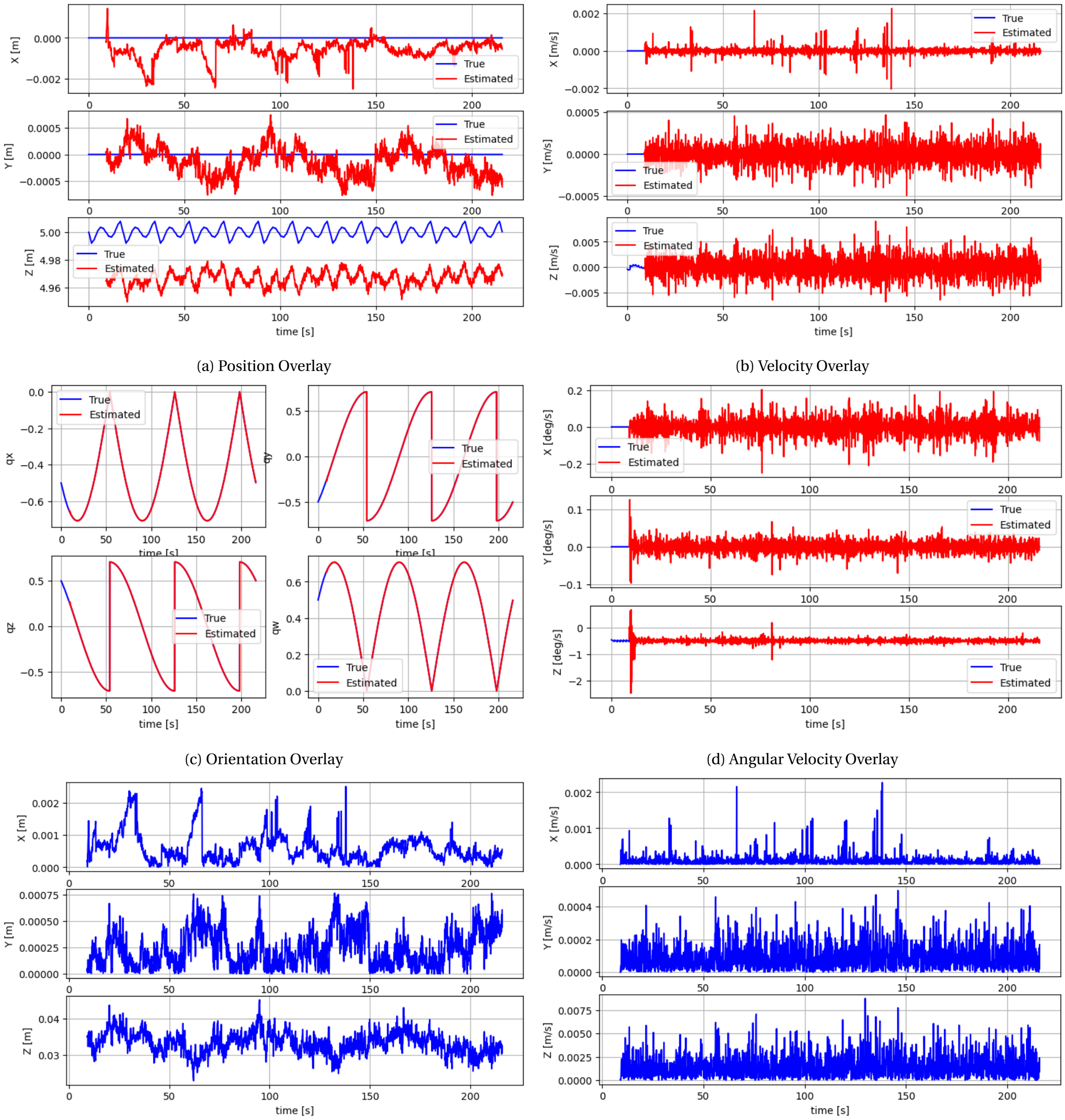
(l) Omega Error Norm

Figure C.6: HWRESlow results

C.7. EE 50%

Table C.7: EE 50% Results

Error	Value	Unit
Position	0.032703	[m]
Orientation	0.196261	[deg]
Velocity	0.00206	[m/s]
Omega	0.07661	[deg/s]



(a) Position Overlay

(b) Velocity Overlay

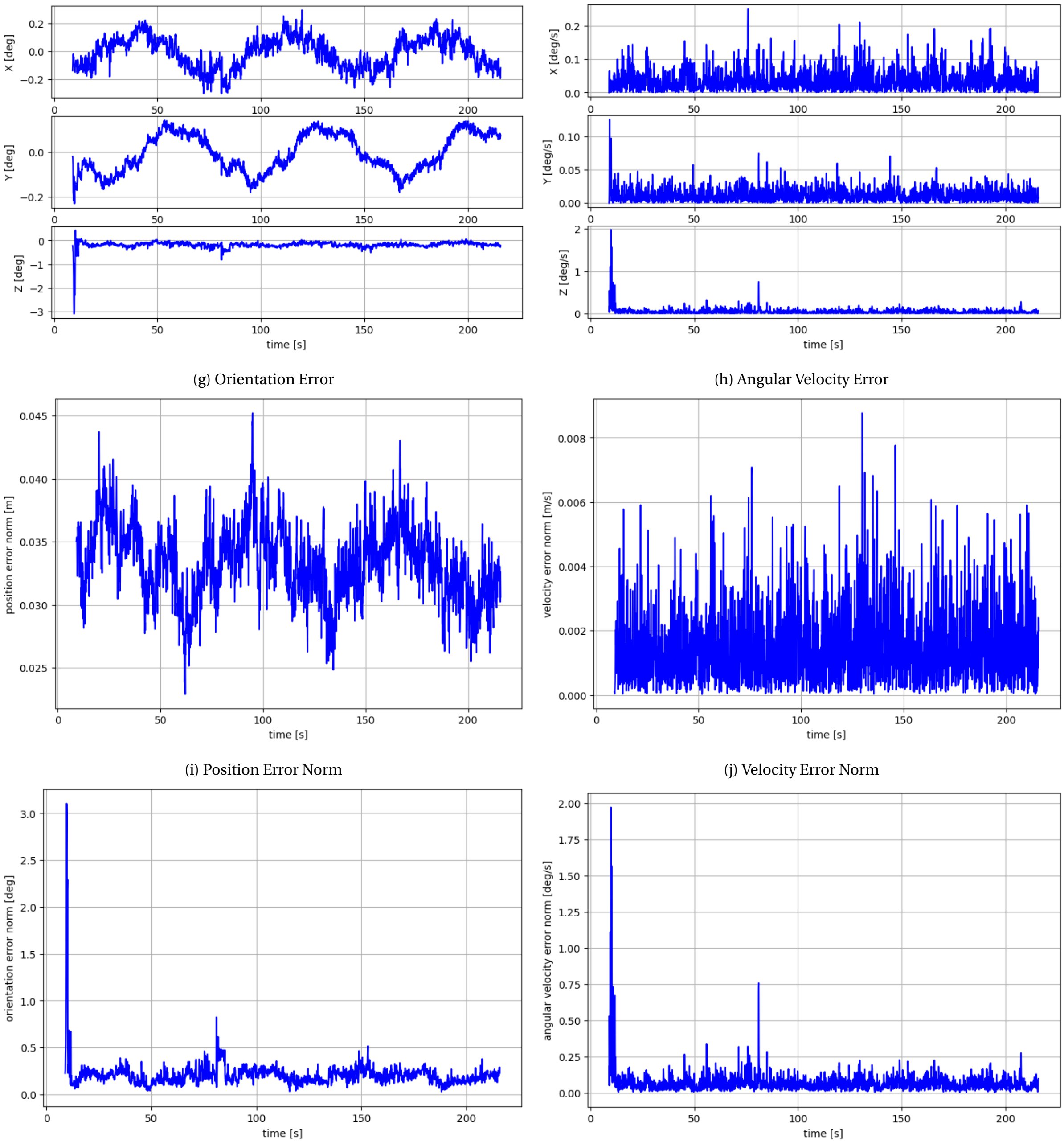
(c) Orientation Overlay

(d) Angular Velocity Overlay

(e) Position Error

(f) Velocity Error

Figure C.7: EE 50% results



(g) Orientation Error

(h) Angular Velocity Error

(i) Position Error Norm

(j) Velocity Error Norm

(k) Orientation Error Norm

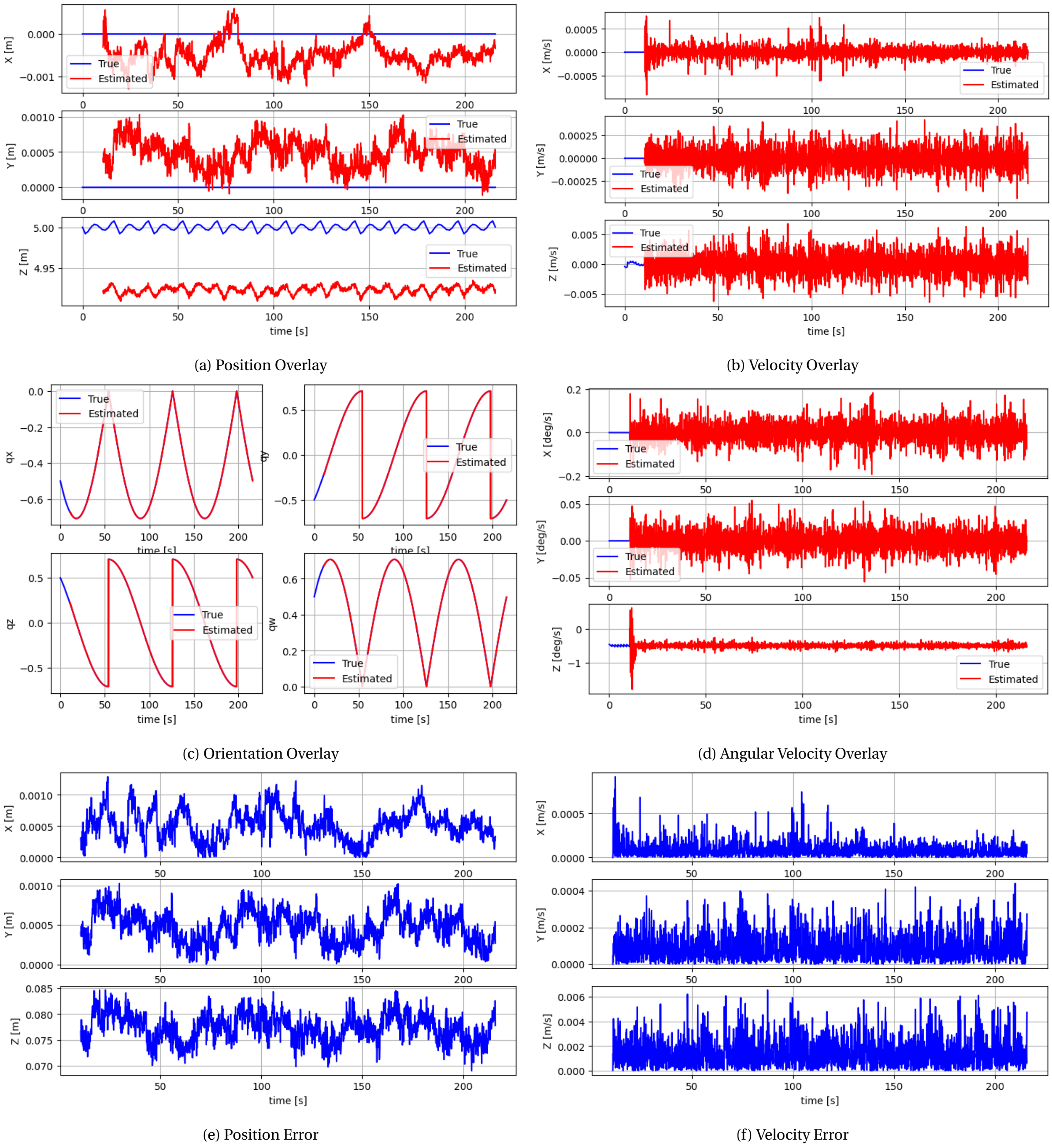
(l) Omega Error Norm

Figure C.7: EE 50% results

C.8. EE 150%

Table C.8: EE 150% Results

Error	Value	Unit
Position	0.076993	[m]
Orientation	0.186406	[deg]
Velocity	0.001936	[m/s]
Omega	0.074368	[deg/s]



(a) Position Overlay

(b) Velocity Overlay

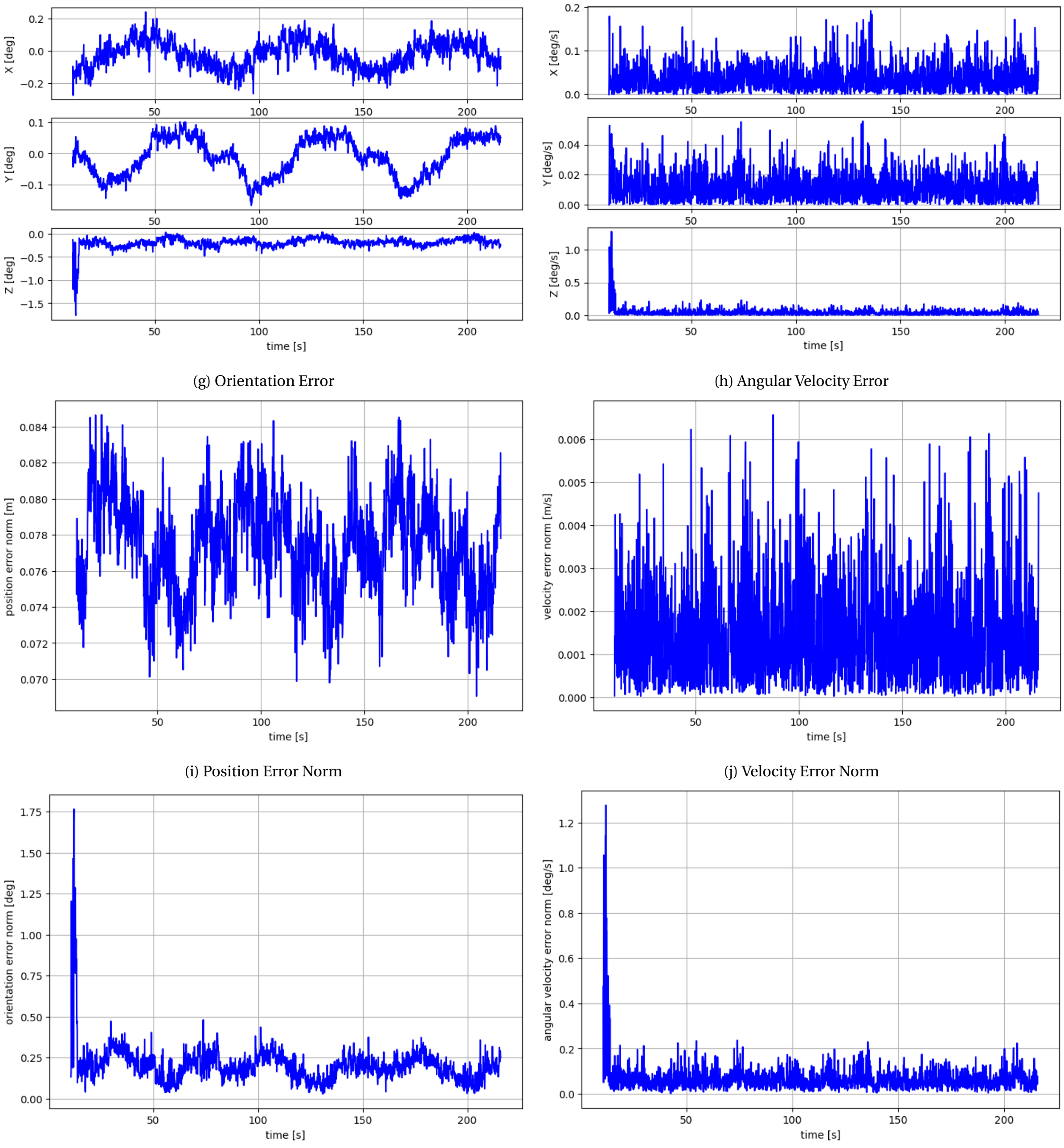
(c) Orientation Overlay

(d) Angular Velocity Overlay

(e) Position Error

(f) Velocity Error

Figure C.8: EE 150% results



(g) Orientation Error

(h) Angular Velocity Error

(i) Position Error Norm

(j) Velocity Error Norm

(k) Orientation Error Norm

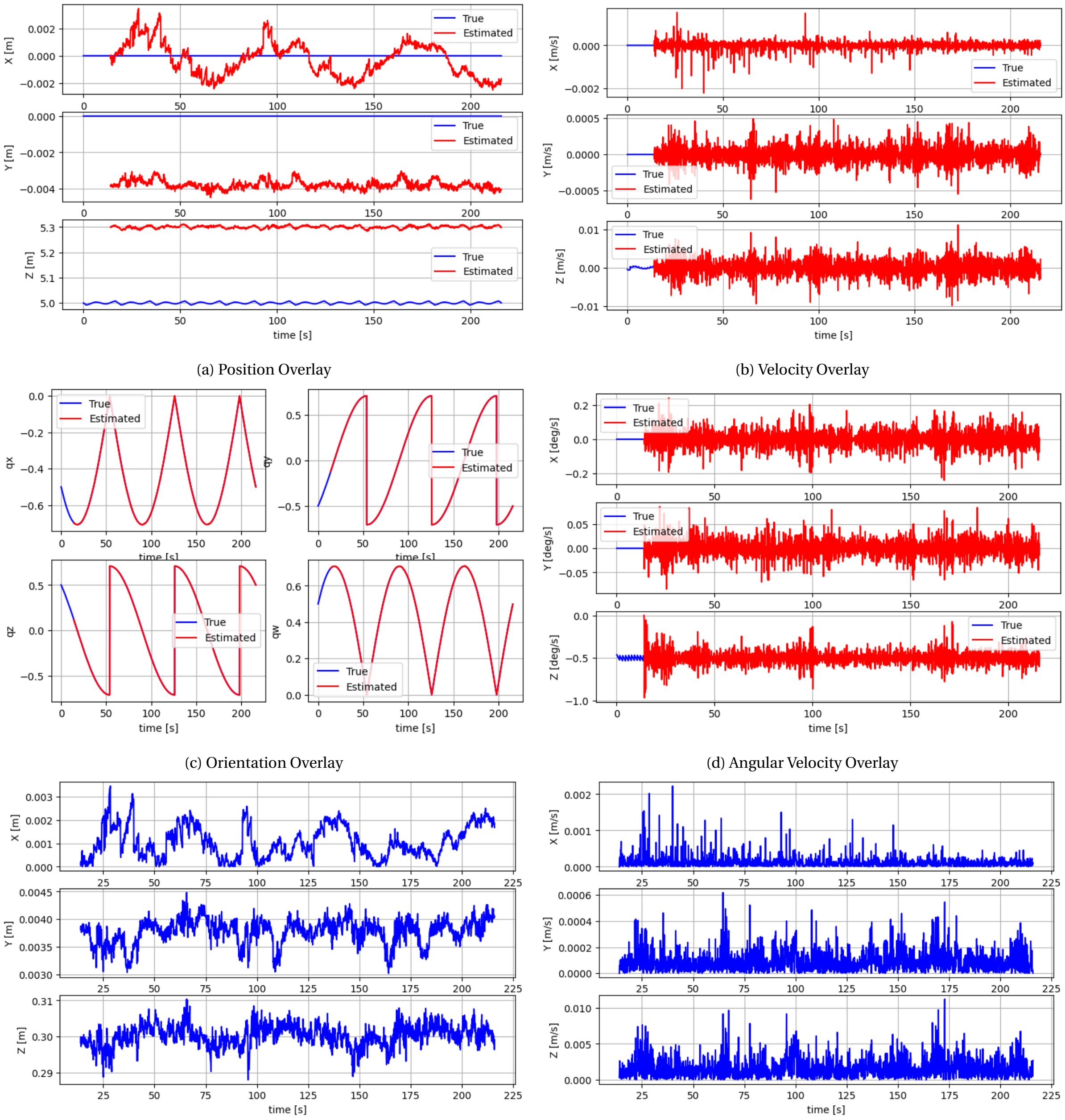
(l) Omega Error Norm

Figure C.8: EE 150% results

C.9. EE eclipse

Table C.9: EE eclipse Results

Error	Value	Unit
Position	0.300556	[m]
Orientation	0.483104	[deg]
Velocity	0.002183	[m/s]
Omega	0.0871	[deg/s]



(a) Position Overlay

(b) Velocity Overlay

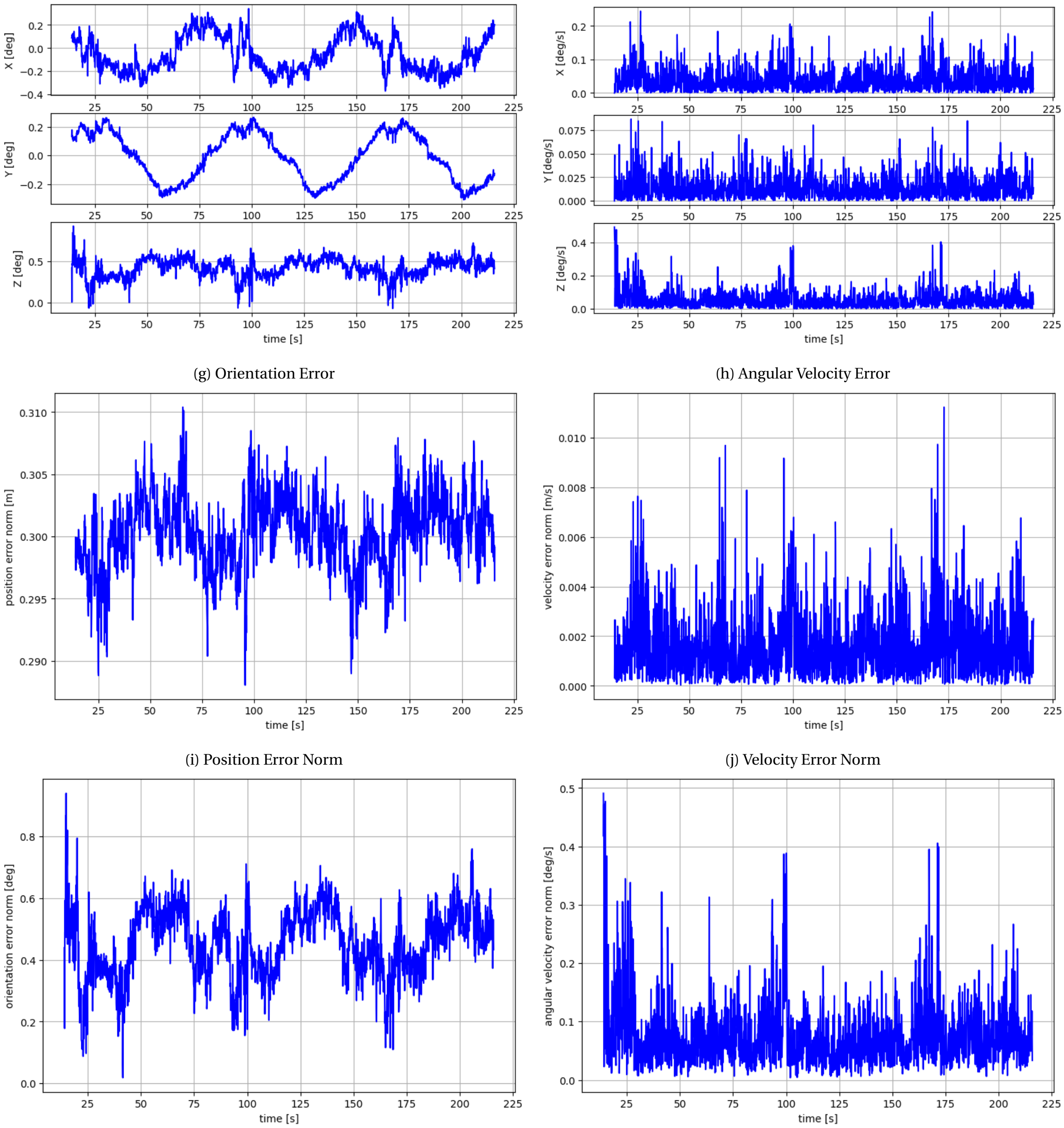
(c) Orientation Overlay

(d) Angular Velocity Overlay

(e) Position Error

(f) Velocity Error

Figure C.9: EE Eclipse results



(g) Orientation Error

(h) Angular Velocity Error

(i) Position Error Norm

(j) Velocity Error Norm

(k) Orientation Error Norm

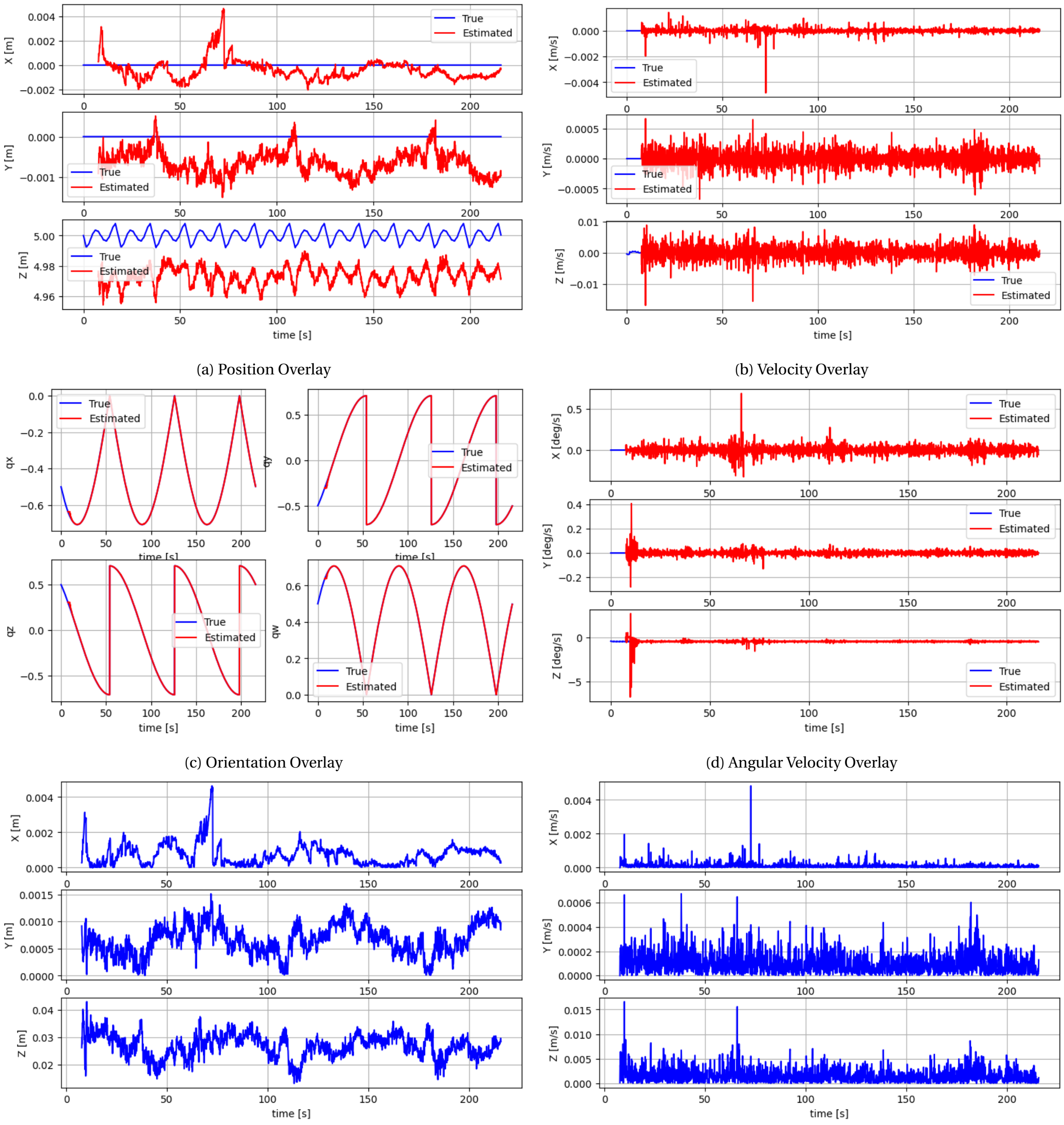
(l) Omega Error Norm

Figure C.9: EE Eclipse results

C.10. EE background

Table C.10: EE background Results

Error	Value	Unit
Position	0.026515	[m]
Orientation	0.230478	[deg]
Velocity	0.001923	[m/s]
Omega	0.071056	[deg/s]



(a) Position Overlay

(b) Velocity Overlay

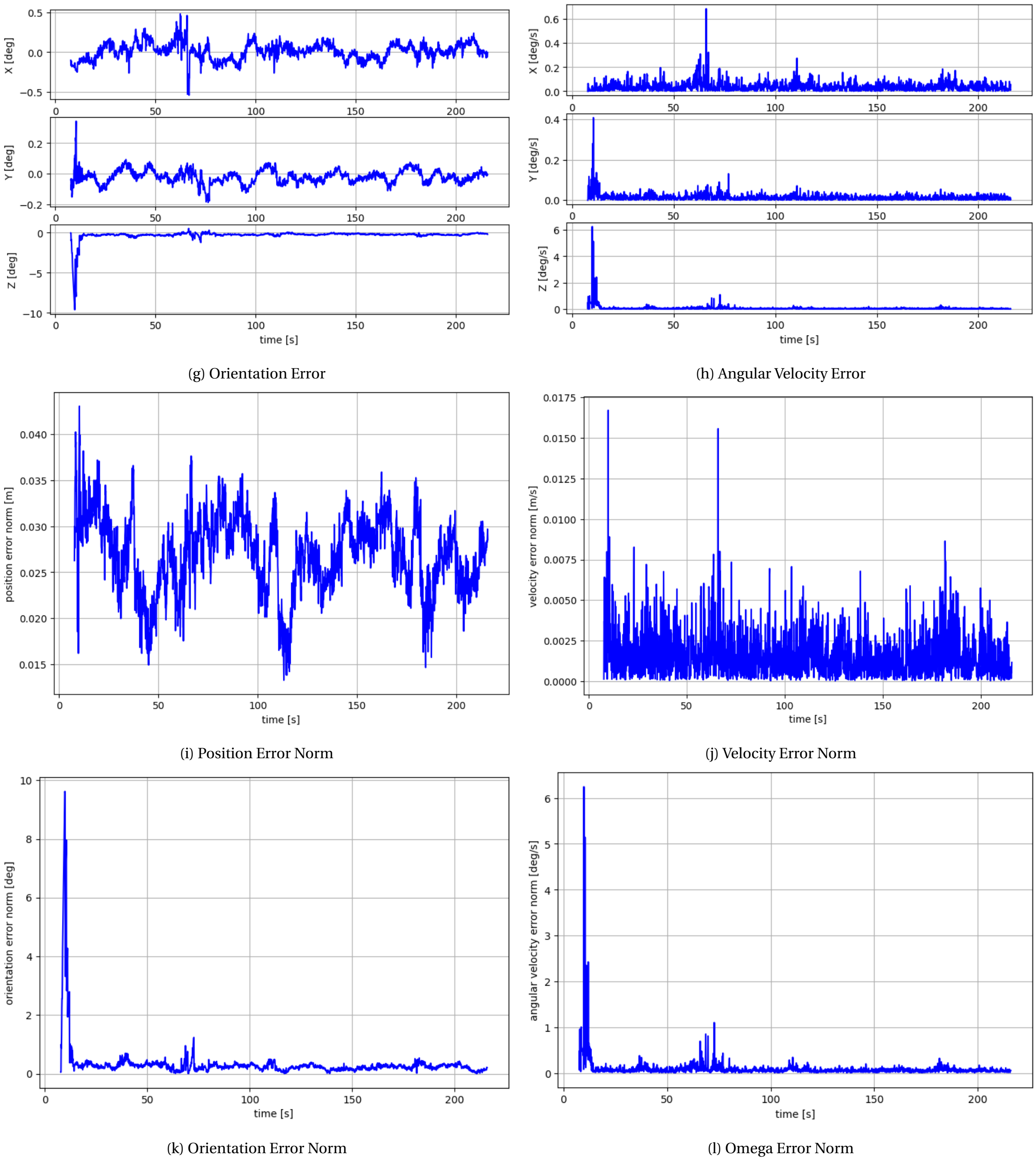
(c) Orientation Overlay

(d) Angular Velocity Overlay

(e) Position Error

(f) Velocity Error

Figure C.10: EE background results



(g) Orientation Error

(h) Angular Velocity Error

(i) Position Error Norm

(j) Velocity Error Norm

(k) Orientation Error Norm

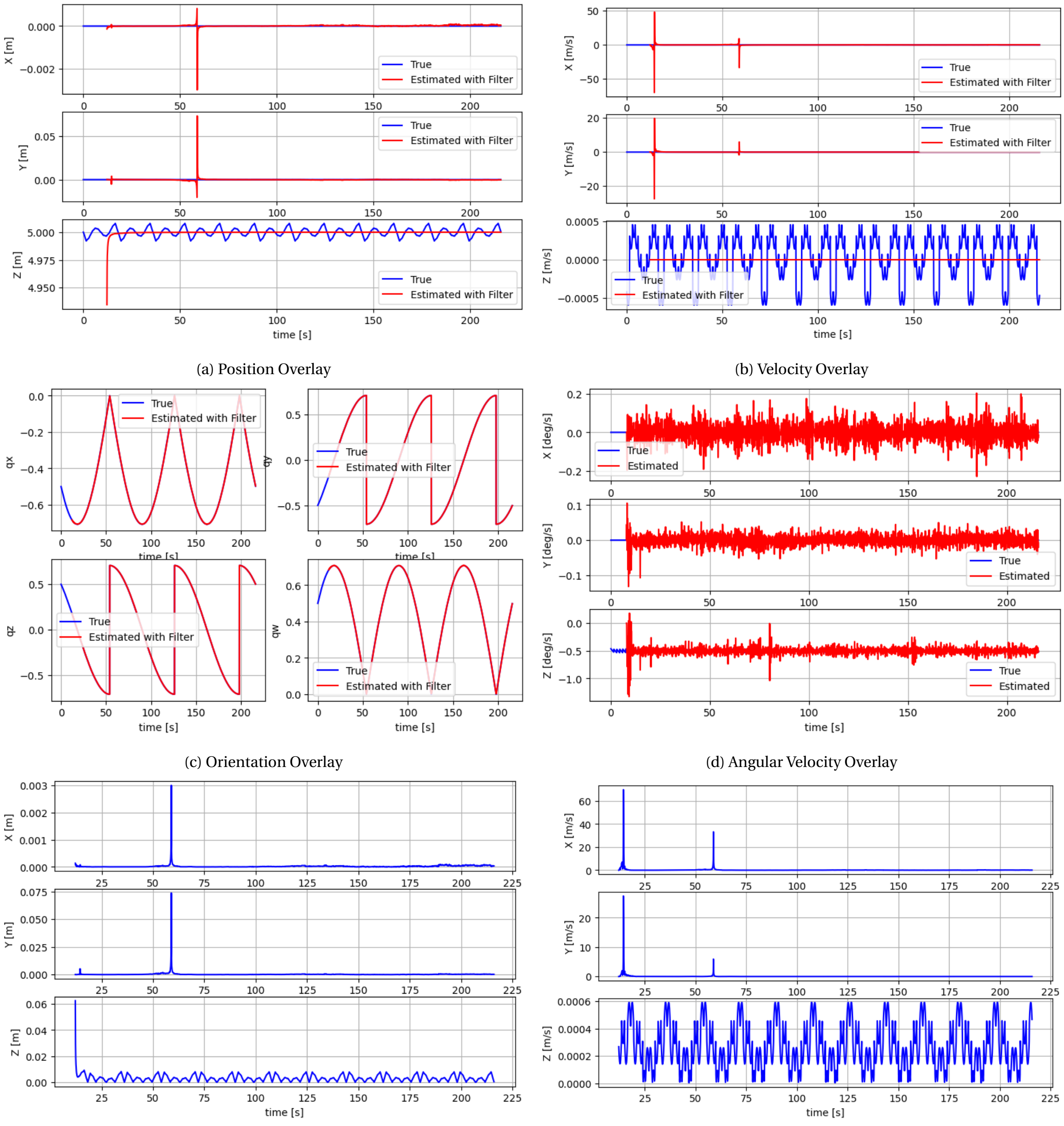
(l) Omega Error Norm

Figure C.10: EE background results

C.11. Filter Integration

Table C.11: Filter Results

Error	Value	Unit
Position	0.003815	[m]
Orientation	0.194048	[deg]
Velocity	0.00115149	[m/s]
Omega	0.072107	[deg/s]



(a) Position Overlay

(b) Velocity Overlay

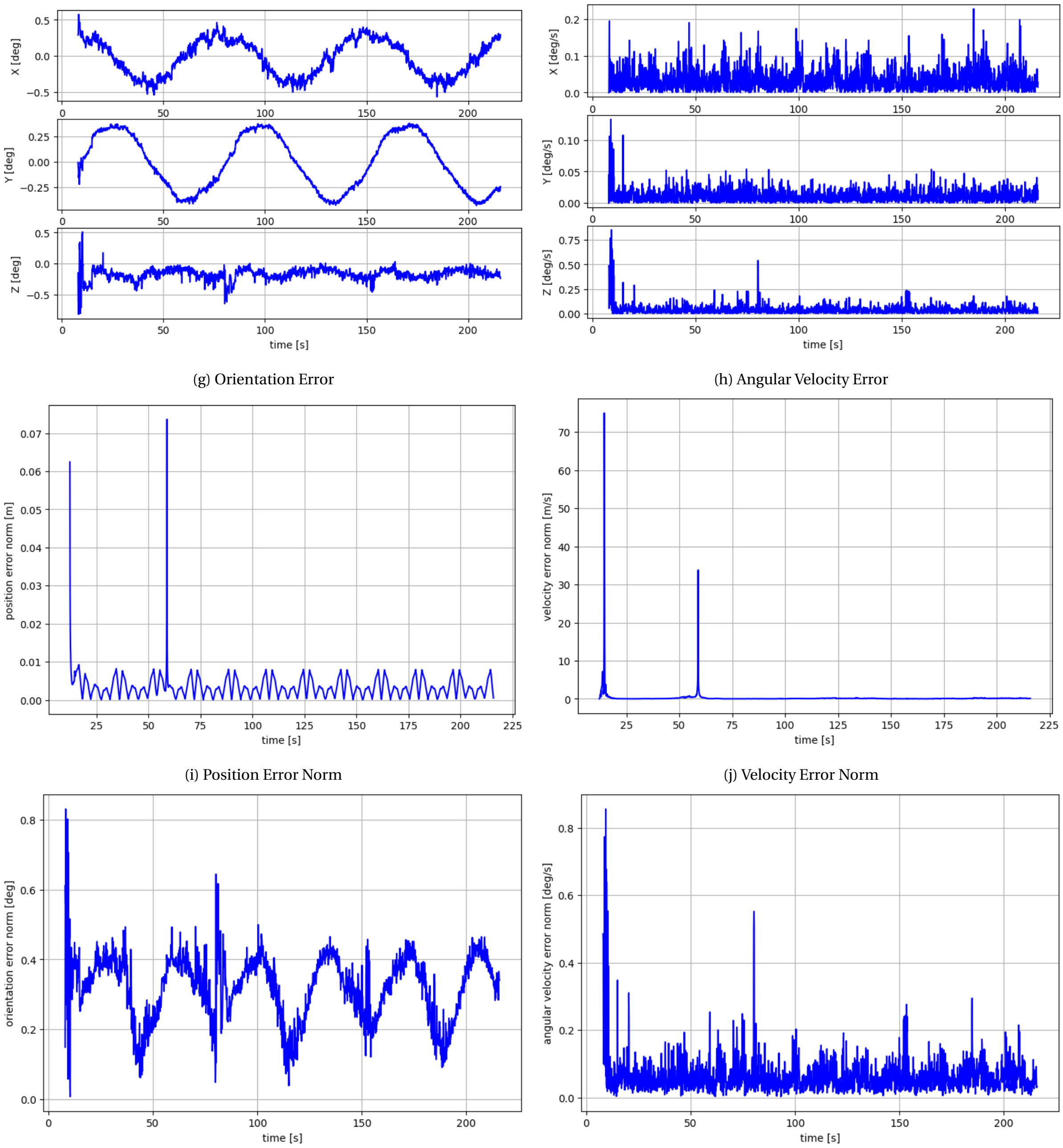
(c) Orientation Overlay

(d) Angular Velocity Overlay

(e) Position Error

(f) Velocity Error

Figure C.11: State Filter results



(g) Orientation Error

(h) Angular Velocity Error

(i) Position Error Norm

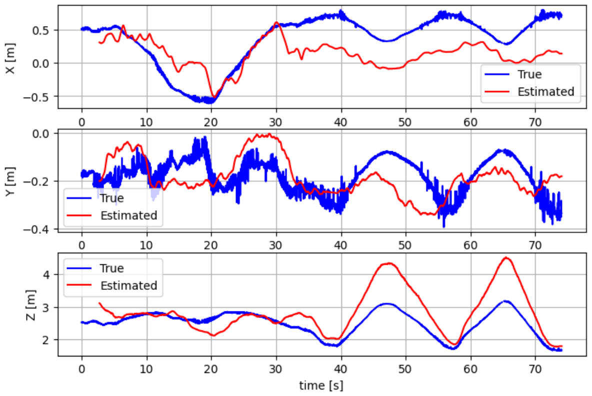
(j) Velocity Error Norm

(k) Orientation Error Norm

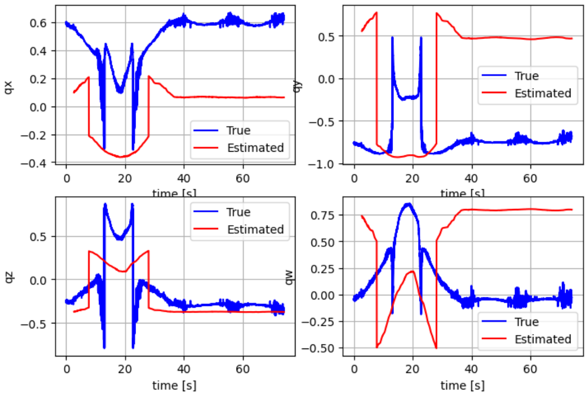
(l) Omega Error Norm

Figure C.11: State filter results

C.12. Practical Experiment



(a) Position Overlay

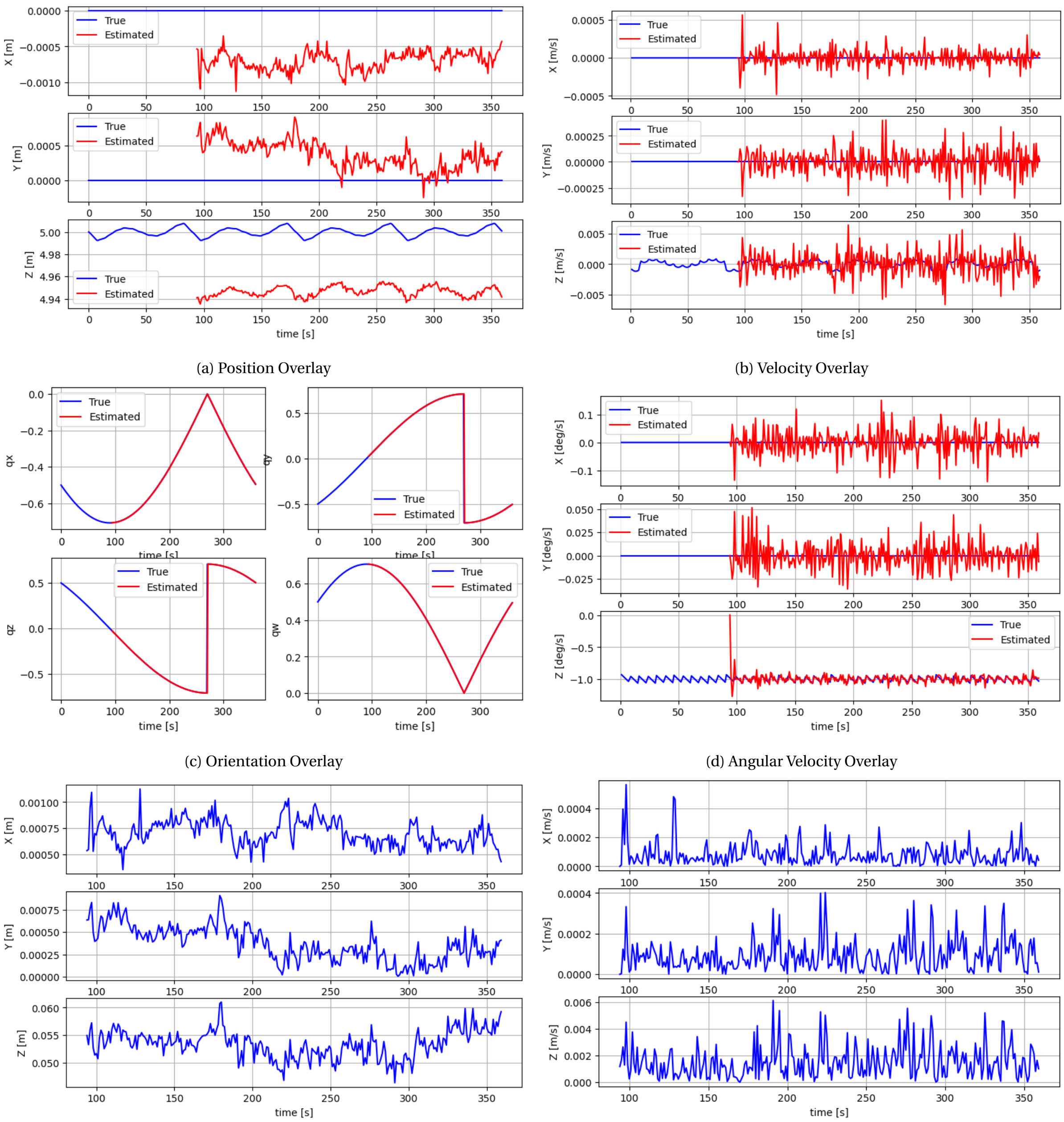


(b) Orientation Overlay

C.13. VEL1FR1

Table C.12: VEL1 FR1 Results

Error	Value	Unit
Position	0.053167	[m]
Orientation	0.146444	[deg]
Velocity	0.002051	[m/s]
Omega	0.061796	[deg/s]



(a) Position Overlay

(b) Velocity Overlay

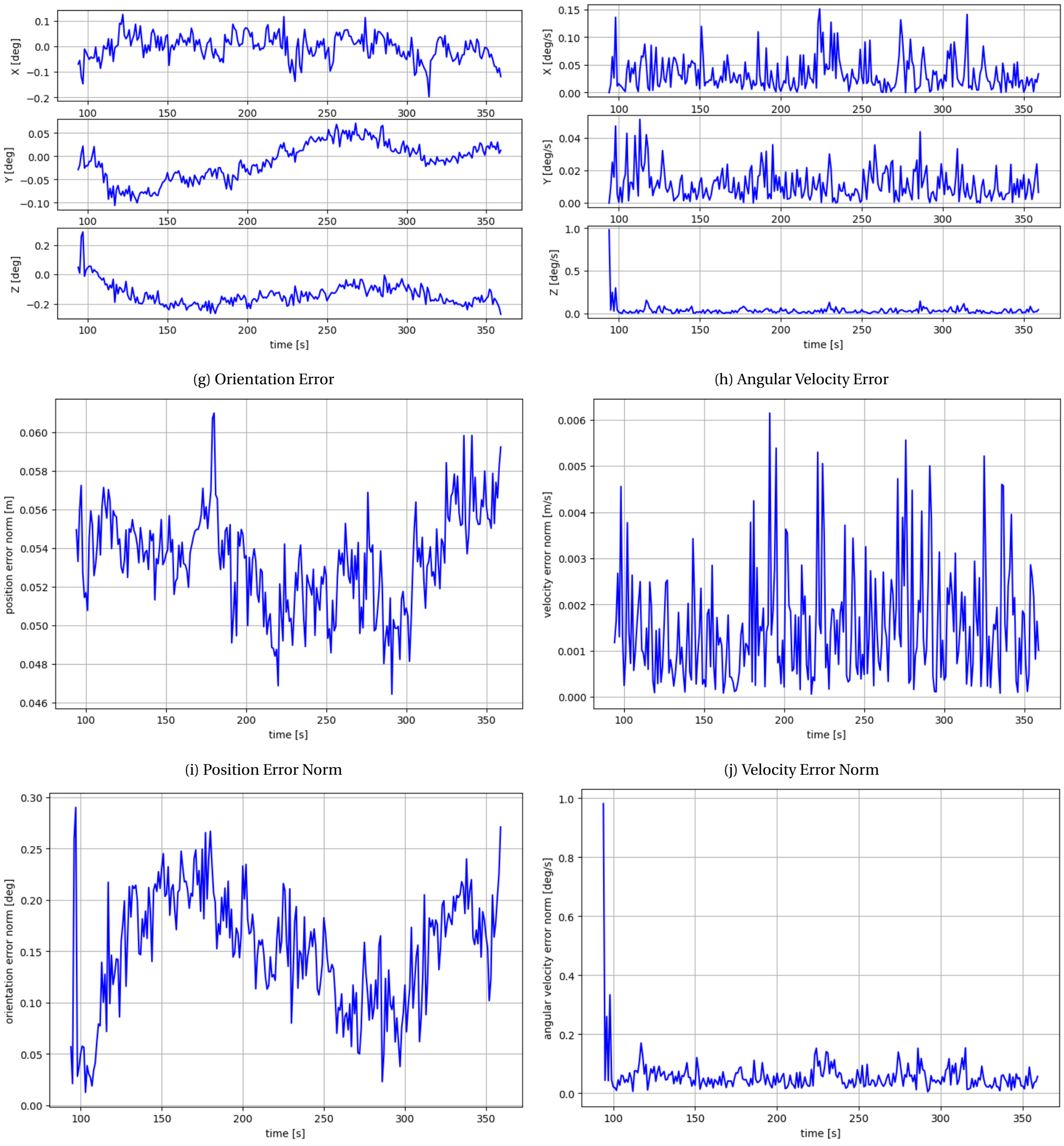
(c) Orientation Overlay

(d) Angular Velocity Overlay

(e) Position Error

(f) Velocity Error

Figure C.13: VEL1FR1 results



(g) Orientation Error

(h) Angular Velocity Error

(i) Position Error Norm

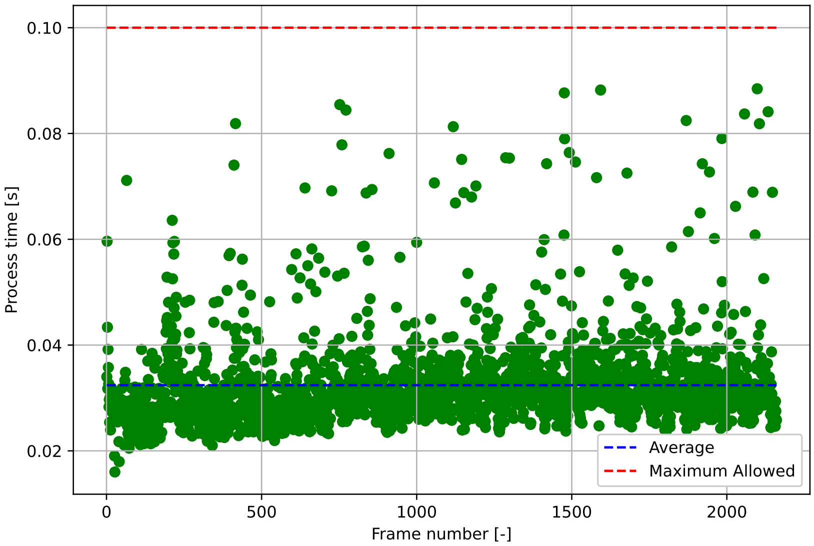
(j) Velocity Error Norm

(k) Orientation Error Norm

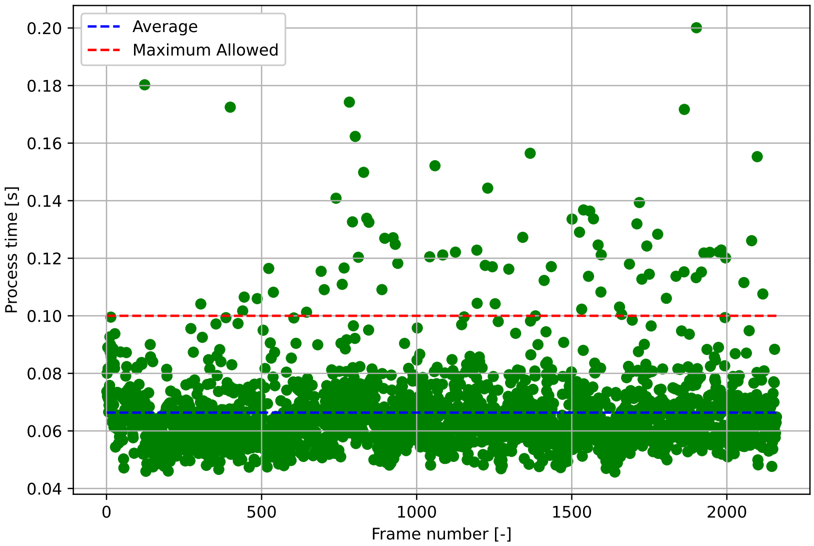
(l) Omega Error Norm

Figure C.13: VEL1FR1 results

C.14. Real Time Performance



(a) Average time required for resolution of 1080px is 0.032 seconds per frame



(b) Average time required for resolution of 1920px is 0.066 seconds per frame

Figure C.14: Real time performance

D

Scientific Paper

A Suitability Analysis of ORB-SLAM for Spacecraft Relative Navigation

Bart Kevers

Abstract— **For most of the history of spaceflight, orbits around Earth have been treated as an infinite resource and satellites as disposable articles. With the growing number of objects in orbit, and the space infrastructure shifting to *mega constellations*, the orbital environment is at increased risk. To sustain a healthy space environment, feasible and cost-effective Active Debris Removal and On-Orbit Servicing solutions are required that can be implemented on a wide scale. This thesis aims to support development in that field by investigating the use of monocular-based Simultaneous Localisation And Mapping (SLAM) for model-agnostic pose estimation around uncooperative spacecraft. This work establishes the suitability of real-time monocular ORB-SLAM for spacecraft relative navigation by performing robustness assessments and accuracy analyses. The assessment is based on synthetic datasets generated to analyse the robustness to relative motion, target visibility. background objects, and hardware specifications.**

I. Introduction

Ever since the inception of the space age, there has been more space debris in orbit than operational satellites [7]. With the current commercialisation of spaceflight, the amount of objects in space rises significantly, resulting in an increasing risk of involuntary collisions between operational payloads and space debris. Over the last two decades, on average 12 accidental fragmentations occur in space each year [7]. To improve the health of the space environment, feasible and cost-effective Active Debris Removal (ADR) and On Orbit Servicing (OOS) solutions are required. The realisation of OOS and ADR systems is an active topic of research, as the main technical challenges include hardware complexity, flexibility and autonomy, in particular when the target to be serviced is uncooperative.

Both OOS and ADR fall under the branch of spacecraft Rendezvous and Proximity Operations (RPO). This branch of space engineering has been critical in enabling challenging and ambitious space programmes and will play a major role in sustaining a healthy space environment around Earth. One of the greatest challenges in spacecraft RPO lies in the problem of relative navigation, particularly the relative pose estimation. The problem of relative navigation in the domain of spacecraft RPO concerns the estimation of the chaser's state relative to its target. In this context, the state is defined as the relative pose (position and orientation) and its derivatives (translational and angular relative velocity). Close-proximity navigation is typically done by means of Vision-Based Navigation (VBN) systems, which rely on optical sensors to provide state estimates relative to the target. A lot of research is focused around VBN with respect to known targets which would either be cooperative or uncooperative. However, VBN around completely unknown targets is still a relatively unexplored field of study. An unknown target is defined as a target object of which no information about their shape, appearance and mass properties is known. This work aims to contribute to technology developments within that field.

Inherent to the complexity of the mission domain, intricate and costly hardware is imposed to fulfil the mission objectives. In order to stimulate sustainable solutions that can be adapted on a wide scale, cost-effective technologies have to be researched and developed. Currently, relative navigation is mainly performed by means of Light Detection and Ranging (LIDAR) sensors. Those sensors are robust under various illumination conditions and have a high accuracy, but they introduce high cost, mass, power and hardware complexity. Recent advancements in the field of computer vision, robotics, and machine learning have allowed the utilisation of passive monocular cameras as main sensor for VBN. Monocular cameras are superior in terms of cost, mass, power and hardware complexity. It is however important to note that monocular-based pose estimation gives a scale-ambiguous pose when only a monocular camera is used. For pose estimation in fully unknown environments, additional sensors are required. Currently, various learning-based algorithms are being researched that implement monocular cameras for relative state estimation. However, due to the working principle of machine learning, those algorithms typically require a known model of its target and a large amount of datasets for training. Since real imagery of spacecraft in orbit is scarce, those datasets have to be created artificially, imposing an uncertainty on the functioning of the algorithm in real environmental conditions. For ideal ADR and OOS missions, no apriori information of the target should be required, meaning that the target is *unknown* by the navigation system. For this, Simultaneous Localisation And Mapping (SLAM) algorithms would be suitable, as they would reconstruct the target and measure its relative pose simultaneously. The research on the application of monocular SLAM algorithms for spacecraft relative navigation is still in early phases. Mainly simulation and laboratory tests have been performed yet. Due to the many unknowns in the suitability of monocular SLAM for the application of spacecraft relative navigation, this is considered the area of interest for this thesis.

A. Related Work

Various studies exist in literature that research monocular-SLAM for spacecraft relative navigation. The current state-of-the-art is listed here.

In 2018, a study was performed in the application of ORB-SLAM, a feature-based SLAM, to spacecraft relative navigation [6]. The SLAM algorithm exploits ORB features, which are image features invariant to rotation, illumination and scale. They have as advantage that they can be rapidly extracted, significantly decreasing the computational time. The algorithm incorporates robustness by initialising a great amount of keypoints, then subsequently discarding the weaker ones. ORB-SLAM also includes a bag-of-words method for place recognition, automating the loop closure step. This reduces the error in the estimation by eliminating the accumulated drift between successive keyframes. In this study, they tested the algorithm on footage captured during proximity operations of the NASA STS-125 Servicing Mission 4 to the Hubble Space Telescope, as well as on footage captured in a laboratory set-up. It was observed that the algorithm had difficulties initialising the map when the target is far away, due to the relative small translational motion between frames. Additionally, loop closure of the algorithm could not be achieved repeatedly in their experiment, where the chaser spacecraft completed several revolutions around the target.

A direct-SLAM approach is a SLAM method that does not rely on features, but directly uses the intensity of the pixels for pose estimation. One of these methods is LSD-SLAM, which is also studied for relative navigation towards unknown uncooperative targets in 2019 [10]. In their work, LSD-SLAM is tested on numerical simulations and a laboratory test. They test the algorithm by rotating a satellite model around one of its axis at a constant rate. It was found to be only accurate on the short-term and close range motion.

A filter-based feature SLAM for monocular pose estimation of unknown targets is investigated by Capuano et al. [5]. This algorithm, called Simultaneous Estimation of Pose and Shape (SEPS), is monocular based, but assumes a single beam LIDAR measurement to recover the scale of the reconstructed map. Target features are extracted by means of the Harris corner detector algorithm. They emphasise that the focus is on the novelty of the state estimation algorithm, and not on the feature extraction. Hence, they do not generate realistic space imagery. The algorithm is Extended Kalman Filter (EKF)-based and avoids the necessity of target information (for state propagation in the filter) by excluding the angular velocity from the state vector. Instead, the angular velocity is directly obtained from optical flow. This lowers the uncertainty of the filter estimate as well as the computational time, due to the size reduction of the state vector in the filter.

Another filter-based algorithm is proposed by Jin et al. [9]. This algorithm decomposes the target state estimation into relative pose estimation and feature point position estimation based on relative pose. The relative pose of the target is estimated by an Extended Kalman Particle Filter (EKPF), while an EKF estimates the feature point position of the target. Also this algorithm exploits both a monocular camera and a single-beam LIDAR measurement for depth information.

Ivanov, Ovchinnikov, and Sakovich [8] developed a pose and inertia determination method for unknown satellites in 2018. This algorithm is tested on the footage of the Chibis-M microsatellite separation. During this experiment, several limitations of the algorithm were observed.. It is only able to recognise spacecraft if the image is not saturated by sunlight and if the background is dark. Furthermore, it was found that temporary occlusion of reference features can occur during motion, which affects the accuracy. Lastly, it has only been able to track the target for 3 seconds in total.

B. Contribution

In the current state-of-the-art, the pose estimators have only been assessed on single-scenario tests. This demonstrates the functionality of the algorithm and gives an indication of the achievable performance. However, it is crucial that the robustness of the SLAM algorithms is tested as well. Robustness is a mission-critical quality of the pose estimator, as non-functional state estimation severely affects the success of spacecraft RPO missions. This work aims to perform robustness assessment on the following aspects:

- robustness to *background objects*
- robustness to *relative motion scenarios*
- robustness to *target visibility scenarios*
- robustness to *hardware specifications*

These aspects are deemed essential for assessing the suitability of SLAM for spacecraft relative navigation. During proximity operations, it is very plausible that objects, like the Earth, appear in the background of captured targets. It is important that these background objects do not result in instabilities in the pose estimation. Furthermore, it is essential that the pose estimator functions under various relative motion scenarios, since the relative state of the target is unknown. So, it is desired that the pose estimator is robust to initialisation under various relative motion scenarios. Additionally, analysing how the visibility of the target effects the algorithm's performance, will give insight in its robustness under certain illumination conditions. Finally, it is imperative to know how camera resolution and frame rate influences the performance of the algorithm, in order to assess whether the algorithm can be implemented on space hardware.

In this paper, first an overview of the designed framework will be given in section II. Then, the pose estimation system will be described in section III, after which the generation of the datasets is discussed in section IV. Subsequently, the results and discussion is presented in section V and section VI, respectively. Finally, the paper is concluded in section VII.

II. Research Framework

The exploited technical framework of this work is visualised in Figure 1. In order to thoroughly assess the suitability of SLAM algorithms for spacecraft relative navigation, it is of great importance that the limitations of SLAM under certain scenarios become visible. Therefore, a significant variety of input data should be obtained or generated. For each test scenario, the accuracy of the pose estimator will be analysed.

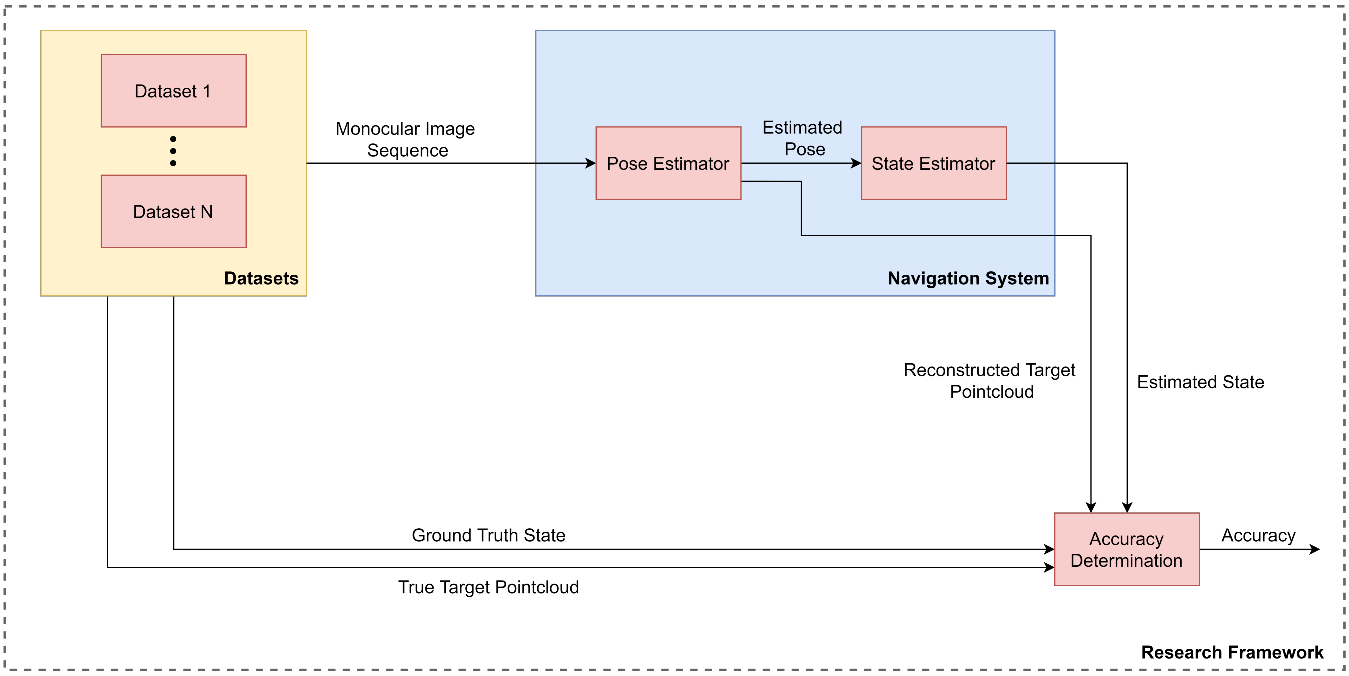


Fig. 1: Technical Research Framework

A. *Datasets*

For monocular-based SLAM, the input data consists of a sequence of monocular images that capture a target in relative motion to the camera frame. For the purpose of researching the accuracy, robustness, and suitability of the pose estimation system, the ground-truth values of each frame and the target 3D model are required. The suitability of the designed navigation system is determined based on its performance under various realistic orbital scenarios, and by using similar hardware that is currently exploited in spacecraft. Since there is a general unavailability of real space imagery capturing relevant close-proximity operations including ground-truth data, it has been determined that synthetically generated data is required in order to perform a valid assessment of the suitability of SLAM for spacecraft relative navigation. However, as synthetically generated data is never as representative as real data, verification of the realistic quality of the data should be performed.

B. *Navigation System*

The generated datasets are used as input in a navigation system. The navigation system consists of the pose estimator and the state estimator. As the objective of the thesis is to assess the pose estimator, it has been decided that the state estimation only computes the derivatives of the pose under a constant velocity model. As a result, the translational and angular velocity are calculated as the difference between two subsequent pose measurements divided by the time step. More sophisticated approaches of state estimation are developed by Capuano et al. [5] for example. In this work, the computation of the pose derivatives will support the analyses of the performance of the pose estimator in section V.

C. *Accuracy Determination*

The focus of this work is on monocular pose estimation. The scale of the target is therefore assumed to be known by the pose estimator. Consequently, The scale parameter will have to be determined in order to assess the accuracy of the pose estimation system. Additionally, a transformation between the true target frame and SLAM-generated target frame is required. The determination method of the transformation and scale is highlighted next.

The navigation system does not obtain any prior knowledge of the target. So, SLAM puts the origin of the map (target) frame at the location of the camera once the map is initialised. This means that if at *t=0s* the map gets initialised, the target frame and the camera frame exactly coincide. From each time step onward, the pose estimator describes the camera pose relative to its pose at *t=0s*. Since the camera frame, indicated by C, never coincides with the actual geometric target frame, from now on indicated by T, a transformation is required between the actual target frame T, and the SLAM-generated target frame N. The pointcloud/map of the target that is constructed by the SLAM algorithm also originates from frame N. The ground truth data is measured in their respective frame of reference, T. To find the transformation $A^N_{T/N}$, the Coherent Point Drift (CPD) algorithm [13] will be exploited. This algorithm compares two pointclouds and performs rigid registration to find the transformation matrix between the two pointclouds. Also, it gives an estimate of the scale difference between the point clouds which is necessary to obtain a scale-unambiguous pose estimate. In order to do this, the pointcloud in frame N is extracted from SLAM, and the pointcloud in frame T is generated from the target CAD model. To illustrate this more clearly, the problem of relative navigation is outlined.

The objective of vision-based relative navigation is to find the rigid-body transformation $A^S_{T/S}$, where S is the body-fixed frame of the chaser/servicer satellite. In monocular-only mode, the pose estimator will produce a scale ambiguous version of the transformation $A^N_{C/N}$. If C' is the camera frame estimated by the pose estimator, then $t^{C'}_{N/C'}$ is the scale ambiguous position vector of frame N with regards to camera frame C' as expressed in frame C' coordinates. If the unknown scale is defined as λ, then it means that $t^C_{N/C} = \lambda t^{C'}_{N/C'}$. From this it follows that:

$$A^C_{N/C} = \begin{bmatrix} R_{N/C'} & \lambda t^{C'}_{N/C'} \\ 0 & 1 \end{bmatrix} \quad (1)$$

The above transformation matrix only holds when all detected features on the target frame are fixed with respect to the target body. Since the designed pose estimator discards background features, this assumption is valid. The desired transformation $A^S_{T/S}$ is then obtained by cascading transformations:

$$A^S_{T/S} = A^S_{C/S} A^C_{N/C} A^N_{T/N} \quad (2)$$

Where, to summarise, $A^S_{C/S}$ is known by the designers of the chaser spacecraft, $A^C_{N/C}$ is obtained from the pose estimator, and $A^N_{T/N}$ is obtained from the CPD algorithm by transforming the point cloud of the reconstructed target to that of the target 3D model.

III. Pose Estimation System

The pose estimation system consists of an existing SLAM algorithm that is used as baseline, after which it is optimised for spacecraft relative navigation.

A. Baseline Algorithm

Based on the superior computational efficiency of Oriented FAST and Rotated BRIEF (ORB) features, and their invariance to illumination, scale and view-angle, it has been determined that an ORB-based SLAM will be utilised as baseline algorithm. Recently, ORB-SLAM3 [4] has been developed as an open-source SLAM algorithm built on the work of previous versions of ORB-SLAM [11] [12]. The novelties of this work lay in the improvement of the accuracy, the implementation of a multi-map system which allows for map merging in the case tracking gets lost, and the improvement of loop closing. In the field of robotics and computer vision, this algorithm is deemed the new reference, being as robust as the current state-of-the-art, while also being more accurate [4]. Consequently, this algorithm is used as the baseline of the pose estimator.

B. Optimisation for Spacecraft Relative Navigation

The baseline algorithm is tested on benchmark datasets that contain images on Earth. This means that the algorithm is not optimised for spacecraft relative navigation. With respect to the standard algorithm, some modifications have been made to improve the robustness for spacecraft relative navigation. The algorithm optimisation is based on three main challenges identified for SLAM applications in space: robustness to background objects (i), robustness to pose initialisation with low parallax (ii), and improvement of feature matching on space imagery (iii).

Robustness to Background Objects

SLAM is generally applied to static environments, meaning all features are static and stay fixed with respect to each other. The application of SLAM to spacecraft relative navigation requires robustness in dynamic environments as well. When an object is in the background of the frame, it does not have the same motion as the target satellite. When there are multiple objects in the frame in relative motion to each other, the environment changes to dynamic. A plausible scenario occurs when the Earth comes in the frame. The Earth does not have the same motion as the target satellite in the camera frame, resulting in an inaccurate and likely unstable SLAM algorithm. To mitigate any influence of these background features on the performance of the algorithm, it would be ideal if those features are not taken into account in the pose estimation. Therefore, the standard algorithm has been adjusted by implementing a *far points threshold*. The algorithm measures the distance of the features and only stores the features that are below the distance threshold for pose estimation. The other features are discarded. Therefore, features detected on Earth could be filtered out by setting the distance threshold at a distance much further than the expected distance of the target.

The pseudo-code of the algorithm is described in algorithm 1, where $dist_{f,c_{KFi}}$ and $dist_{thres}$ are the distance of feature f to the camera c in keyframe (KF) i, and the threshold distance respectively. This algorithm simply says that if the distance of a new feature to the camera in two separate keyframes is smaller than the set threshold, it will be used for pose estimation. Otherwise it will be discarded.

Algorithm 1: Background filtering in pseudo-code

```
for $f \in F$ do
    if $(dist_{f,c_{KF1}} < dist_{thres}) \wedge (dist_{f,c_{KF2}} < dist_{thres})$ then
        store f
    else
        discard f
    end
end
```

Robustness to Pose Initialisation with Low Parallax

In order to match features between different frames, sufficient parallax is needed between the frames. Parallax is the apparent displacement of an object due to the change of view of the observer. Sufficient parallax is needed in order to obtain a valid initial estimate of the relative pose. Spacecraft relative navigation is typically characterised by high relative distance, and low relative motion between the chaser and target. Hence, pose initialisation is difficult to obtain as demonstrated by Dor and Tsiotras [6], who tried pose initialisation of SLAM on the Hubble telescope during relative navigation operations of the space shuttle. In SLAM, parallax is created by observing a feature from different neighbouring keyframes. The amount of neighbouring frames to take into account for feature matching is a trade-off between computational time and initialisation robustness. For low parallax, more neighbouring keyframes should be searched for feature matching, this will increase the computational time. Hence, the number of keyframes to search is optimised when it is as high as possible, while still allowing to run in real-time.

The algorithm is visualised in algorithm 2. From the algorithm it can be seen that by increasing the number of neighbouring keyframes, a higher chance of sufficient parallax is obtained, as keyframes that are further away are included as well. The loop might get computationally more expensive with many neighbouring keyframes, and thus real-time performance might suffer. Therefore, the parameters should be rightly set.

Algorithm 2: SLAM Initialisation procedure in pseudo-code

```
STATE = Initialise ;
for $KF2 \in NeighKFs$ do
    baselineratio = $dist_{KF1-KF2} / MapMedianDepth_{KF2}$ ;
    if baselineratio > 0.01 then
        Search Matches ;
        Triangulate Matches ;
        if Reprojection Error < Threshold then
            add matched map points to map;
            estimate pose ;
            STATE = Track ;
            exit loop ;
        end
    end
end
```

Space Imagery Optimisation

Due to the high noise in spacecraft images, features are not as clear as during the benchmark datasets that the algorithm has been tested on. Hence, this is not optimised for spacecraft relative navigation. The relative descriptor distance between features in different frames is a measure of similarity between features. A distance threshold decides if features are similar enough to be classified as a match. If the threshold goes down, fewer features are matched, but the certainty that the features are correctly matched goes up. This is a trade-off between robustness and accuracy of the pose. Therefore, in higher quality images the threshold can go down, but with lower resolution and high noise images the threshold might be too strict, making feature tracking more difficult. Hence, the threshold can be tuned to sacrifice accuracy for better robustness, and the other way around. The pseudo-code is described in algorithm 3, where f is a feature. It should be noted that there are other conditions for a feature match as well, but not discussed here.

Algorithm 3: Feature matching procedure in pseudo-code

```
for all $f$ in 2 KFs do
    find dist between each combination of $f$;
    if $dist(f_1, f_2) < dist_{thres}$ then
        $f_1, f_2$ is a match (if other conditions are true as well)
    end
end
```

IV. Datasets

A. Dataset Generation

The synthetic datasets are generated using Blender [3]. The following parameters are defined in the creation of the dataset: *target*, *trajectory*, *background*, *lighting*, and *image post-processing*. The design of each of these factors are discussed next.

Target

The target in the datasets is the Delfi-n3Xt satellite, depicted in Figure 2. For this study, the CAD model as well as a real mock-up of the satellite are available.

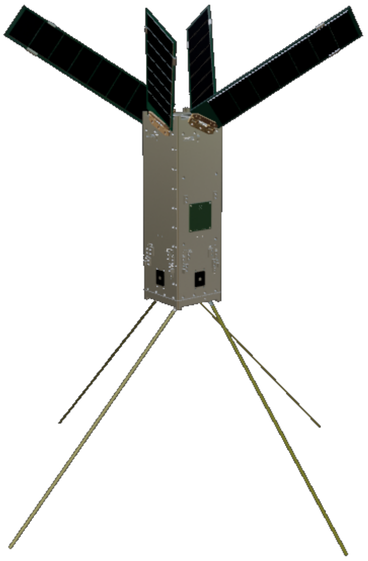
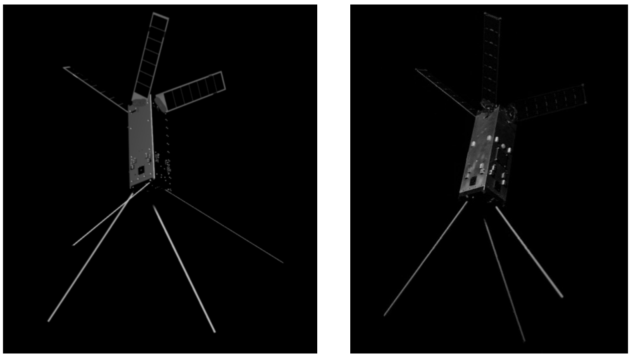


Fig. 2: Delfi-n3Xt CAD Model

It is important that the appearance of the simulated target is as realistic as possible. Therefore, the Delfi-n3Xt materials have been simulated in Blender using the properties of the real materials. It is essential that the reflection of sunlight on the material is similar to real space imagery, since the visibility of features have significant influence on the performance of the pose estimator. The realistic quality of the target is verified by means of comparison to images generated in the lab, and to images in the SPEED dataset [1]. The SPEED dataset is deemed a good reference since this is seen as a benchmark dataset for monocular-based spacecraft pose estimation. The lab-generated imagery is created using a realistic solar simulator. The comparison to the lab-generated image is displayed in Figure 3a and Figure 3b. Both images are captured under the same angle, and the approximate same direction of sunlight.



(a) Delfi-n3Xt synthetic

(b) Delfi-n3Xt under real space lighting

Fig. 3: Comparison synthetic and lab-generated target

Apart from the lack of an antenna in the lab-generated satellite and the visible distortions of the real camera, it is noticed that the visibility and reflectivity of the components are relatively close. The main difference is the texture on the aluminium surface of the lab-generated satellite, which is lacking in the simulated one. Due to the lack of texture, it can be assumed that the performance on the synthetic images will be conservative, since the real imagery will thus contain more texture for features detection. Also, it is noticeable that the solar panels of the CAD model are slightly different than the real mock-up. The solar cells do not cover as much surface in the CAD model, making the underlying panel itself more visible. The visible edge of the panel in the lab image is comparable in brightness. Hence, the reflective properties of the solar panels are considered rightly chosen.

The second comparison is with an image of the SPEED dataset, visualised in Figure 4. It should be noted that the target satellite is different. So a real comparison cannot be made. However, the visibility of the features can be compared and assessed. It is observed that the shading on the different materials in the SPEED image is also uniform, just like the self-created one. Furthermore, the spectrum of brightness, shading, and visibility of features is comparable in both images. Thereby verifying the quality of the self-generated datasets.

Trajectory

The trajectory has been defined as a circle around the target, illustrated in Figure 5. This is chosen due to several reasons. First, a pointcloud of the target needs to be constructed by the pose estimator in order to obtain pose accuracy information using rigid registration methods. The more complete the pointcloud is, the greater the validity of the obtained results in this research. By travelling in full circles around the target, the target can be reconstructed from all sides, creating a more complete pointcloud. Secondly, the influence of the lighting conditions on the pose estimator can be evaluated as the lighting direction remains fixed, but the camera will be rotating with respect to the light direction. Hence, it can be verified if tracking will become more difficult on the darker sides of the target. Thirdly, Dor and Tsiotras[6] showed that map initialisation is difficult for spacecraft relative navigation due to the low parallax between frames. By travelling in a circle around the target, parallax requirements for map initialisation can be obtained. The relative distance to the target is then not of influence, as the decisive factor for sufficient parallax is the angular velocity. Consequently, this will simplify the testing scenarios for relative motion as the map initialisation will be directly related to the relative angular velocity.

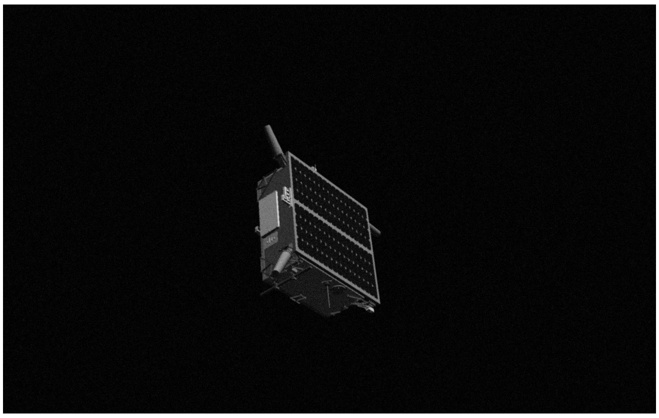


Fig. 4: SPEED dataset imagery [1]

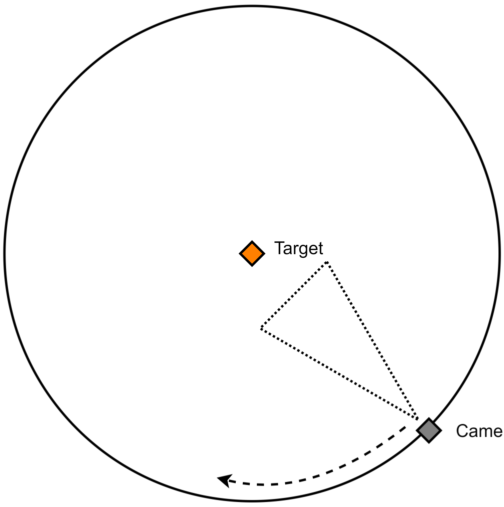


Fig. 5: Trajectory definition

Background

The background is modelled as plain black, since this is most representative for real space imagery and is also used in the

SPEED dataset [1]. In the datasets for background robustness tests, the Earth will be modelled. The Earth is modelled by overlaying three images on top of each other, after which the image is wrapped around a blue sphere. The simulated Earth is shown in Figure 6. Depending on the direction of the simulated sunlight in Blender, the Earth's night and day is also realistically modelled.

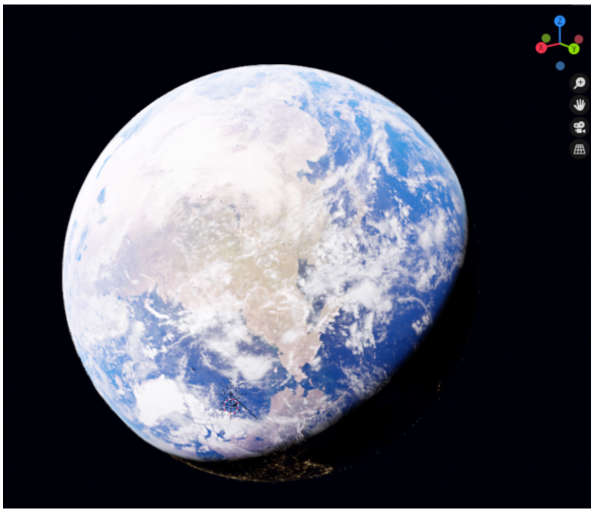


Fig. 6: Earth model in Blender

Lighting

The visible light in orbit comprises sunlight and albedo from celestial objects. In orbit, the majority of albedo comes from Earth's surface. The sunlight in Blender has been set to pure white, and the power has been adjusted such that the brightness of the target is similar to the SPEED dataset, meaning that the features are of similar visibility. The Earth albedo is adjusted such that it is 30% of the sunlight's power [15].

Image Post-processing

The camera that is simulated in Blender is a pinhole camera without any modelled distortions. Furthermore, rendering of the frames yields no visible noise or motion blur. As these phenomena are inherent to real-life cameras, especially in the field of spacecraft VBN, the rendered frames are post-processed. In the post processing step, Gaussian white noise with $\sigma^2 = 0.0022$ and Gaussian motion blur with $\sigma^2 = 0.0022$ are added to each frame. Identical post-processing is performed in the SPEED dataset.

B. Test Cases

An overview of the test scenarios is outlined in Table I. It is noted that these are the baseline datasets to assess the pose estimator. In case inconclusive results are obtained, additional datasets are generated.

Relative Motion [RM]

The RM datasets test the robustness of the pose estimator to a range of relative angular velocities. As mentioned earlier, the angular velocity is the critical factor for the functionality of the pose estimator in terms of map initialisation. This is because the most challenging part of pose estimation using SLAM is the initialisation phase, especially for spacecraft applications as demonstrated [6]. Since the motion is circular around the target, sufficient parallax between frames is determined by the relative angular velocity (and frame rate). A realistic range for the relative angular velocity for spacecraft relative navigation is determined to be between 1 to 10 deg/s. This is also the range that various other pose estimation algorithms for space applications have been tested on [2] [5]. As a result, the test cases for RMVELhigh, RMVELmid, and RMVELlow are respectively 10, 5, and 1 deg/s, while keeping the other dataset parameters fixed. As a baseline for other datasets, 5 deg/s for the relative angular velocity has been chosen.

Image Resolution [HWRES]

For feature detection, the resolution of the images plays a significant role. When the objects contains fewer pixels, feature detection will be more challenging. The maximum resolution is based on the resolution of the SPEED images, which have a resolution of 1920x1200 pixels [1]. The medium resolution is based on the images of the Hubble telescope by the space shuttle, which have a resolution of 1000x1000 pixels [6]. Since CubeSats generally have low processing power available, a lower resolution is also tested. This results in the following resolutions to be tested: 1920x1440px, 1080x810px, 860x635px. All images have an aspect ratio of 4:3. As a baseline resolution, the maximum resolution of 1920x1440px will be used. This is motivated by the fact that this allows for finding the ultimate limitations of the pose estimator when it is tested on other parameters.

Frame Rate [HWFR]

The frame rate is of great importance for multiple reasons. It is important for feature matching, but also for map initialisation in correlation with relative angular velocity. The higher the frame rate, the less the view-angle is changed between subsequent frames. The upper frame rate is determined by running the algorithm in real-time on an Intel Core i7 9th gen processor. At an image resolution of 1920x1440px, the pose estimator can process frames at a rate of 0.066 seconds per frame. It should be noted that the performance of the processor has not been maximised during the execution of the algorithm. on average the algorithm consumed 25% of the CPU. As a result, the frame rate range has been set at 10, 5, and 1 frames per second.

Brightness [EEBR]

As brightness can influence the performance of the pose estimator, it is essential that this is correctly modelled. To take into account any uncertainty in the illumination model, datasets are generated with 50% and 150% of the original lighting. The performance of the pose estimator is then compared to the original, after which conclusions can be drawn with respect to the validity of the brightness conditions. Finally, the lighting settings in Blender have been set to their minimum to simulate eclipse. Since no accurate

TABLE I: Overview of synthetically generated datasets

Scenario	Ang Vel [deg/s]	Resolution [px]	Frame rate [Hz]	Brightness [%]	Background [-]	Revolutions [-]
RMVELhigh	10	1920x1440	10	100	no	3
RMVELmed	5	1920x1440	10	100	no	3
RMVELlow	1	1920x1440	10	100	no	3
HWRESmed	5	1080x810	10	100	no	3
HWRESlow	5	860x645	10	100	no	3
HWFRmed	5	1920x1440	5	100	no	3
HWFRlow	5	1920x1440	1	100	no	3
EEBRhigh	5	1920x1440	10	150	no	3
EEBRlow	5	1920x1440	10	50	no	3
EEbackground	5	1920x1440	10	100	yes	3
EEeclipse	5	1920x1440	10	1	no	3

verification method is obtained of lighting conditions in eclipse, real eclipse performance cannot be assessed. However, it is interesting to observe what brightness conditions are required for feature detection and matching.

Background [EEBG]

A dataset is created to test the robustness of the pose estimator to background objects. The background dataset consists of an Earth, as stated earlier. The Earth comes in the scene at certain positions and orientations of the camera.

V. Results

A. Accuracy Uncertainty

It should be noted that the accuracy might not be exact, due to the method of accuracy determination,. This is because there is an uncertainty in the scale and transformation matrix obtained from CPD. A slight adjustment in the scale significantly affects the accuracy. The CPD algorithm lines up the true and estimated pointcloud to its best capability, but a certain uncertainty will always be present. Therefore, slight changes in true position and orientation are expected among the various tests. As the velocity and angular velocity are directly obtained from the derivation of the pose and its preceding pose estimate, they are a good indication of the accuracy as well, as the propagation accuracy between the pose at subsequent frames is to a certain extent independent of the scaling. This is also seen in the results as position and orientation accuracy can be inconsistent, but the derivatives are relatively consistent throughout all scenarios. A good illustration of this is shown in Figure 7, where the position is overlaid. There is a translational offset in the z-axis, but the propagation of the pose looks very accurate. This can indicate that the CPD transformation is not perfect. Hence in the analysis, the derivatives should also be looked at due to the nature of their estimation method and to obtain valid conclusions.

B. Environmental Influence

Regarding the influence of environmental factors like background objects and illumination conditions, one needs to consider scenarios RMVELmed, EEBRhigh, EEBRlow, EEbackground and EEeclipse. The results are summarised in Table II.

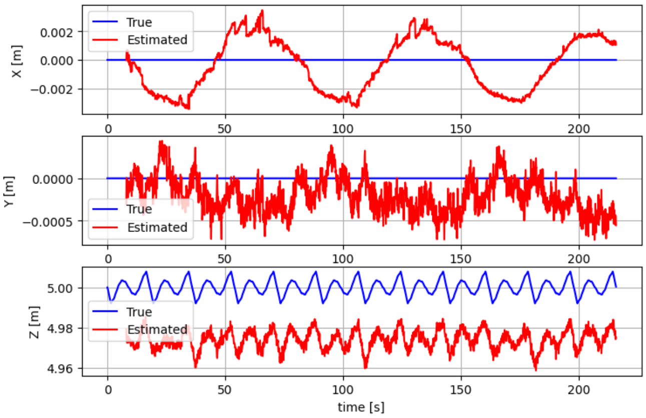


Fig. 7: Pose propagation accuracy and absolute pose accuracy

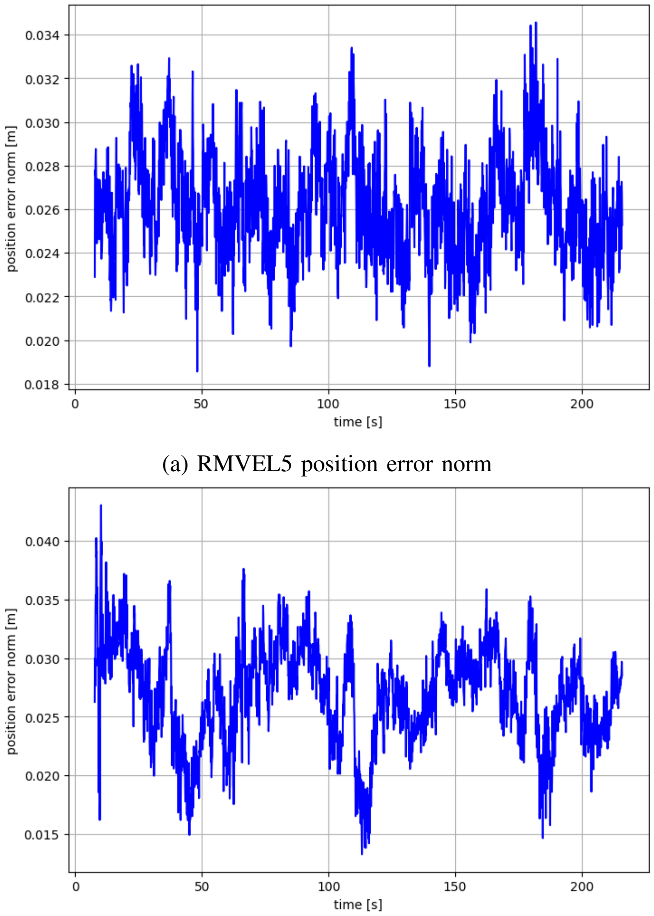
TABLE II: Results Environmental Influence

Scenario	Accuracy			
	Position [m]	Orientation [deg]	Velocity [m/s]	Ang Vel [deg/s]
RMVELmed	0.026	0.322	0.002	0.072
EEBRhigh	0.077	0.186	0.002	0.074
EEBRlow	0.033	0.196	0.002	0.077
EEbackground	0.027	0.230	0.002	0.071
EEeclipse	0.301	0.483	0.002	0.087

Background

To be robust to background objects, EEbackground is a dataset containing a rotating Earth in the background which is dynamic with respect to the chaser and target frame. The purpose of this dataset is to test the background robustness of the algorithm. In this simulation, the threshold for far points has been set at 50 meters. This is a safe distance as all feature points of the target are well within that distance, while the Earth exceeds that distance. The position and orientation error of the simulations with and without background object is shown in Figure 8 and Figure 9.

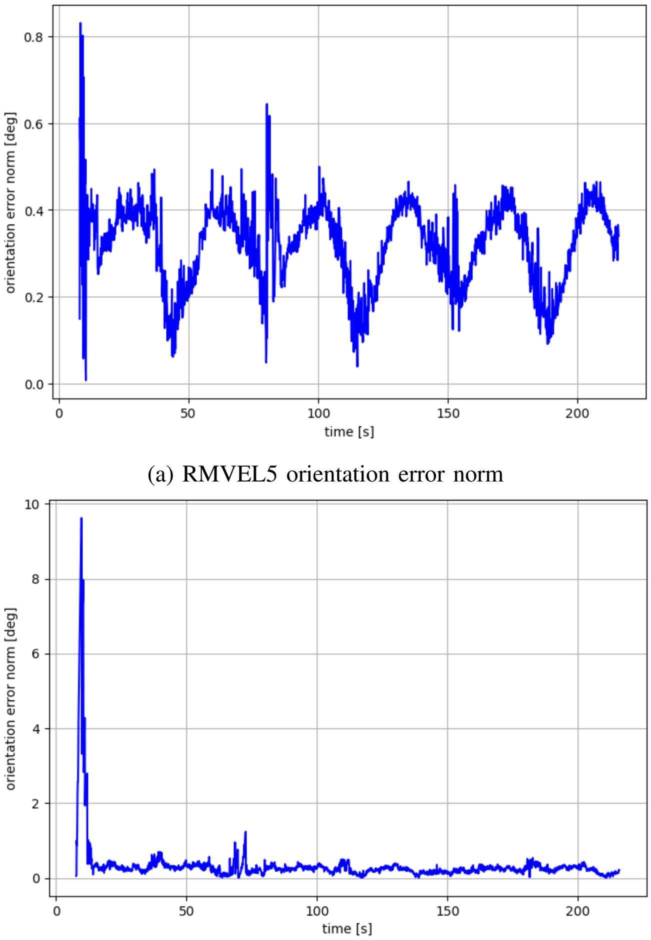
The Earth is in frame in the following periods. From 57 to 70 s, from 129 to 142 s, and from 201 to 214 seconds. It can be noticed that the initial pose estimate of the background scenario is not very accurate. However, this is unrelated to the fact that that dataset contains the Earth in the background, as the Earth does not come in the frame until 57 seconds in the simulation. In that simulation, the initial pose estimate was more inaccurate, but it converges over time to approximately



(a) RMVEL5 position error norm

(b) EEbg position error norm

Fig. 8: Position Error Norm comparison between RMVEL5 and EEbg



(a) RMVEL5 orientation error norm

(b) EEbackground orientation error norm

Fig. 9: Orientation error norm comparison between RMVEL5 and EEbackground

the same root mean square (RMS) error. Apart from a little disruption in the background orientation graph at around 70 seconds, no abnormal behaviour is noted, meaning the background robustness implementation is shown to work. Another visualisation of the working of the background elimination algorithm is shown in subsection V-E.

Brightness

When comparing the brightness scenarios, it is observed that the propagation of the pose between EEBRhigh, EEBRlow and RMVELmed (as baseline scenario) are almost identical. This shows on one hand the robustness of the algorithm to illumination variations, but on the other hand also that highly accurate illumination conditions in the input data do not significantly benefit the validity of the performance assessment. Of course, this is under the condition that the illumination conditions are approximately representative of the real illumination in space. The verification of the lighting conditions has been discussed in section IV. Observing that the algorithm is robust to certain brightness variations, also contributes to the validity of the simulated results.

In eclipse, when the illumination is very low, significant differences can be observed in terms of pose accuracy. However, the realistic quality of the eclipse simulation has not been verified, meaning that the simulation can not be actually stated as an eclipse simulation. Therefore, The main conclusion that can be drawn from the eclipse scenario is that under very low lighting power, when the target is still visible, initialisation and tracking can be achieved, even though the accuracy drops a little bit.

C. Hardware Influence

The hardware analysis results are analysed in terms of frame rate and resolution. The results are shown in Table III and Table IV.

Resolution

From the simulations it is noticed that the resolution significantly affects the performance. The results yield a positive correlation between resolution and accuracy. Furthermore, With the current noise model on the images and relative motion scenario, resolutions lower than 800x600 pixels yield no results due to instabilities of the pose estimate. Various lower resolutions have been tried, but no stable results could be obtained. This shows that resolution is an important factor for pose estimation on space imagery. At resolutions higher than 800x600 pixels, the accuracy is still within a degree on orientation and cm-level on position. This has to do with the fact that SLAM needs a certain accuracy in their map, as map points are matched across frames by reprojecting them in the current frame. When this reprojection is highly inaccurate, matching becomes inaccurate as well, and thus unstable behaviour of the SLAM algorithm is observed. This indicates that the accuracy does not linearly drop with resolution, but a stable pose estimate simply cannot be obtained when the resolution is too low. It is likely that for detailed pose estimation design for certain missions and applications, stable pose estimates at lower resolutions could be achieved. But since this work is mainly a high-level research on the suitability of SLAM for space applications, the perfect optimisation of SLAM algorithms is deemed outside the scope of this work.

TABLE III: Results Resolution Influence

Scenario	**Accuracy**			
	Position [m]	**Orientation [deg]**	**Velocity [m/s]**	**Ang Vel [deg/s]**
RMVELmed	0.026	0.322	0.002	0.072
HWRESmed	0.062	0.265	0.004	0.203
HWRESlow	0.084	0.469	0.005	0.299

Frame Rate

Frame rate and relative angular velocity have similar effect on the pose accuracy. The higher the frame rate or the lower the angular velocity, the less the target displaces between frames. Hence, the frame rate has been analysed with a constant relative motion to visualise the effect more clearly. However, a combined conclusion should be drawn with the analysis of relative motion in the next paragraph. It is observed that for the scenario of 1 frame per second at a relative angular velocity of 5 deg/s (HWFRlow), no initialisation could be achieved. This has to do with the fact that the relative motion is too high (parallax is too big) for ORB features to be matched. ORB has a certain invariance to view point, but only up to a certain degree. It is clear that in this case, that was out of its limits. Therefore, it is interesting to investigate in this application of spacecraft relative navigation what the maximum degree of relative rotation of the target with respect to the camera per frame is. For RMVELhigh and HWFRmed, this is one degree per frame. To investigate if this is still valid for a frame rate of 1 frame per second, an extra dataset has been generated, VEL1FR1, where the relative velocity is 1 deg/s, and the frame rate is 1 fps. During this simulation. stable pose estimates are achieved. The other test scenarios all implement rotation per frame of either 0.1 deg/frame in RMVELlow, or 0.5 deg/frame in the remaining simulations. Additional tries have been made to achieve a larger rotation than 1 deg/frame, but without success. Therefore, one can conclude that in order to apply monocular SLAM for spacecraft relative navigation, orientation propagation of the target between frames exceeding 1 deg/frame might result in unstable pose outputs. This is an important finding and indication of the limitations of monocular SLAM for spacecraft relative navigation, as the state of the target is typically unknown for uncooperative pose estimation.

TABLE IV: Results Frame Rate Influence

Scenario	**Accuracy**			
	Position [m]	**Orientation [deg]**	**Velocity [m/s]**	**Ang Vel [deg/s]**
RMVELmed	0.026	0.322	0.002	0.072
HWFRmed	0.023	0.442	0.002	0.086
HWFRlow	-	-	-	-
VEL1FR1	0.053	0.146	0.002	0.062

D. Relative Motion Influence

The general range in which the pose estimation should work has been tested in RMVELhigh, RMVELmed, and RMVELlow. It was able to achieve pose estimation in all simulations. Initially, initialisation could not be achieved for RMVELlow as the parallax between subsequent frames was too low. But after adaption of the algorithm to use more neighbouring keyframes for feature matching, this was achieved. Even though all scenarios produce stable pose estimates (demonstrating the robustness of the algorithm), it is observed that the accuracy drops with the relative angular velocity. This is strongly related to the achievement of parallax between frames. If there is not much movement between frames, the uncertainty in the pose estimate increases due to the relatively small viewpoint change. This shows that the algorithm performs better in the range of 0.5 to 1 deg/frame. Further conclusions of this section tie in to the frame rate conclusions, as a higher frame rate is required for higher relative angular velocities in order to achieve algorithm initialisation.

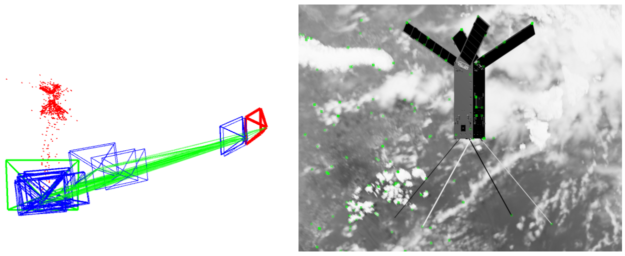
TABLE V: Results relative motion influence

Scenario	**Accuracy**			
	Position [m]	**Orientation [deg]**	**Velocity [m/s]**	**Ang Vel [deg/s]**
RMVELhigh	0.014	0.298	0.002	0.082
RMVELmed	0.026	0.322	0.002	0.072
RMVELlow	0.046	1.221	0.002	0.494

E. Effects of Algorithm Optimisation

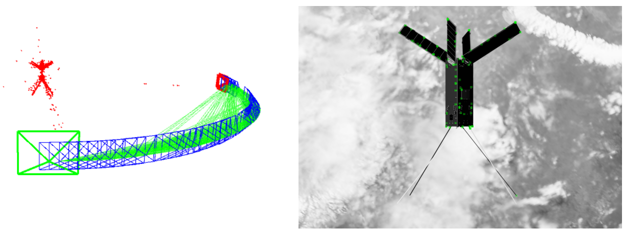
Robustness to Background

The effect of the background elimination system algorithm is visualised in Figure 10 and Figure 11. It was noticed that when the Earth came in frame, unstable behaviour of the pose estimator was observed. This resulted in an inaccurate map of the target, for which no accuracy comparison could be made due to the inaccuracy of the map points when used in CPD. Hence, the effect is best observed in the live map viewer during running of the algorithm, visualised in Figure 10a and Figure 11a. In the live map viewer, the blue squares are the keyframes, red is the initial frame of the map, and the red dots are the map points. As the camera runs in a circle relative to the target, the key frames should be lined up along that circle, as is shown in Figure 11a. Unstable behaviour is observed when these keyframes are not lined up in order any more, but cluttered up on top of each other. This is the result of dynamic environments in the camera frame.



(a) Unstable behaviour in the live map viewer (b) Background feature detection

Fig. 10: SLAM unstable behaviour



(a) Stable behaviour in the live map viewer (b) No background feature detection

Fig. 11: SLAM stable behaviour

Robustness to Pose Initialisation with Low Parallax

The dataset VEL1FR10 has a relatively high frame rate and low relative velocity, resulting in low separation of the keyframes and thus low parallax. So, initially no initialisation was achieved in this simulation. Consequently, The parallax for feature matching has been extended by increasing the number of neighbouring keyframes for feature matching in the algorithm. The resulting succesfull position estimation is displayed in Figure 12.

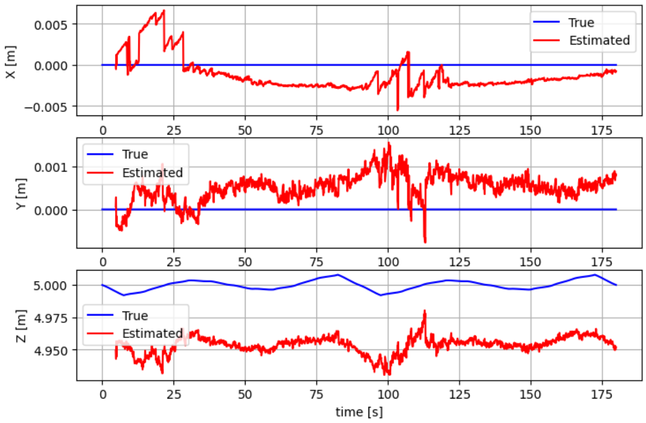
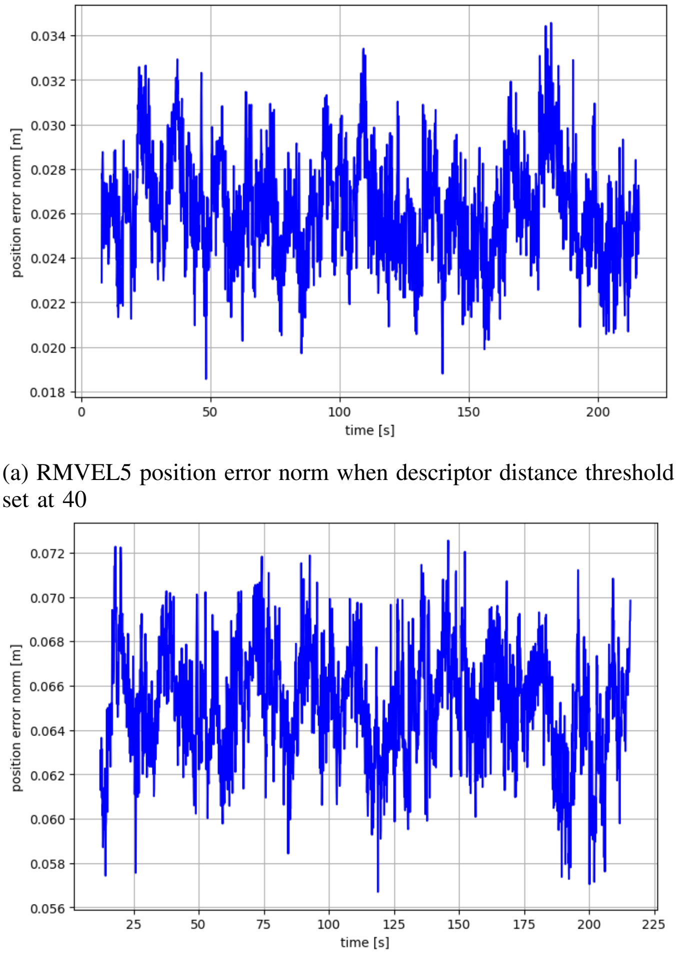


Fig. 12: VEl1FR10 with initialisation successful just after 5 seconds

Space Imagery Optimisation

The difference in the effect of the feature descriptor matching threshold is visualised when running two different versions of the algorithm on the same dataset. This is shown in Figure 13a and Figure 13b. The plot in Figure 13a shows the result when the threshold is set at 40, the other when the threshold is set at 60. The difference is significant. The lower threshold version has an accuracy that oscillates around 0.026 m, the higher threshold version around 0.065 m. It was noted that a distance below 30 resulted in some occasions in a low amount of feature detection and high initialisation phase. Thus, 40 has been determined as optimum.



(a) RMVEL5 position error norm when descriptor distance threshold set at 40

(b) EEbg position error norm when descriptor distance threshold set at 60

Fig. 13: Effect of descriptor distance threshold

VI. Discussion

From the results, it is concluded that resolution has a significant impact on robustness. Stable and accurate pose estimation has been achieved for resolutions higher than 800x600 pixels. However, lowering the resolution further resulted in unsuccessful pose estimation. Additionally, a limitation is observed related to relative motion and frame rate. When the orientation propagation of the target between frames exceeds 1 deg/frame, pose initialisation cannot be obtained. An ideal range in terms of accuracy is found for a propagation between 0.5 and 1 deg/frame. This is the range that allows for sufficient parallax between frames that significantly benefits the accuracy. Lower propagation levels also succeed, but it is noticed that the accuracy drops slightly. These results impose limitations on the space system encompassing the processor and monocular sensor. It is important that the on-board processor can process these resolutions at a sufficient frame rate.

Additionally, it is found that the brightness of the target does not significantly affect the performance of the algorithm. This demonstrates the robustness of the pose estimator to variations in illumination conditions, but also increases the validity of the results, since an extremely accurate illumination representation is not required for the dataset generation.

Furthermore, all algorithm optimisations for spacecraft relative navigation have been shown to work. Robustness to background objects has been achieved by implementing an algorithm that discards features further than a threshold distance. Also, it has been demonstrated that initialisation of very low parallax motion can be achieved.

On average this system achieved an accuracy of sub-degree level on orientation, and cm-level on relative position. In related literature, Capuano et al.[5] achieved an accuracy of sub 1 degree and 0.3 meters. Jin et al. [9] obtained accuracies of sub 1 degree as well and 0.1 meters. This means that the pose estimation system used in this work has similar, if not better, accuracy than the state-of-the-art.

The suitability of the pose estimator for spacecraft relative navigation is assessed on three factors. Firstly, achievement of robustness to orbital relative navigation conditions is required. This includes relative motion, background, space image quality, and illumination robustness. The robustness to these factors is demonstrated in the simulations. The second factor is based on the capability of this system to work in real-time on space hardware. In this work, it is found that a lower limit of 800x600 pixels on image resolution guarantees stable performance of the pose estimator. Additionally, The frame rate should be high enough such that the target's orientation propagation between frames does not exceed 1 deg/frame. When observing hardware used on VBN systems, it is noted that the RNS camera in [14] could capture resolutions of 1024x1024px at a frame rate of 3 Hz. Based on the fact that the design of that system was developed more than ten years ago, and the fact that technology advances significantly over the years, it is deemed feasible that the algorithm could be implemented on space hardware. The third and last point is related to the achievable accuracy. On average, this system achieved an accuracy of sub-degree level in orientation, and centimetre-level in relative position. For particular mission phases of RPO, strict accuracy requirements need to be met. A reference example of requirements set for the e.Deorbit mission is found in [16]. They state accuracies for the closing phase (50-100m) of sub-5 deg and sub-10m, and sub-2 deg and sub-0.05 m for the sync and capture phase (0-50m). Additionally, during Servicing Mission 4 of the Hubble telescope [14], accuracies were required of sub 1 deg and sub 0.1m for the most critical phase (below 5 meters). Since the required accuracies of both the e.Deorbit mission and Servicing Mission 4 have been achieved in this work, the designed pose estimation system is deemed suitable for spacecraft relative navigation.

VII. Conclusions

This paper investigated the suitability of SLAM algorithms for spacecraft relative navigation. The work is motivated by the continously growing number of objects in space, resulting in an increasing risk of involunary collisions between space debris and operational spacecraft.

In this work, The algorithm of Campos Martínez et al. [4] (ORB-SLAM3) is adapted for spacecraft pose estimation. It has been optimised for spacecraft relative navigation. Background robustness is achieved by implementing a distance threshold for feature storing. Additionally, pose initialisation is optimised by enhancing the robustness under low-parallax conditions, while maintaining real-time performance. Finally, the feature descriptor distance threshold is tuned to improve the accuracy of the algorithm for spacecraft relative navigation purposes. The designed pose estimator is assessed on synthetically generated datasets, capturing the Delfi-n3Xt satellite. The realistic quality of the datasets is verified by comparison with lab-generated imagery and the SPEED dataset [1]. The designed datasets allow for analysis of relative motion effects, environmental effects, like brightness changes and background objects, and for the analysis of hardware requirements.

The suitability of the pose estimator for spacecraft relative navigation is assessed on three factors: its robustness (i), its capability to work in real-time on space hardware (ii), and its accuracy (iii). In this work, some limitations were observed with respect to resolution, frame rate and relative angular velocity. It is found that a lower-limit of 800x600 pixels on image resolution guarantees stable performance of the pose estimator. Additionally, The frame rate should be high enough such that the target's orientation propagation between frames does not exceed 1 deg/frame. Furthermore, the system achieved an accuracy of sub-degree level in orientation, and centimetre-level in relative position. By comparing these limitations and accuracies with the specifications of a space-proven navigation system, and relative navigation requirements of the e.Deorbit and SM4 missions, it is concluded that the implemented pose estimation system –and therefore SLAM in general– is deemed suitable for spacecraft relative navigation.

The novelties of this work related to robustness analysis of the implemented pose estimator to spacecraft relative motion, the capability of handling background objects in the frame, and the analysis of hardware requirements, provide a relevant contribution to the work already performed performed in this field.

For future work, it is recommended to validate the results on real RPO footage and to test the optimised algorithm in relative navigation facilities. Furthermore, the effect of sensor fusion for scale recovery on the pose estimation performance is suggested to be analysed.

Acknowledgement

This paper is based on the work performed during the MSc thesis at Delft University of Technology. This work has been supervised by Dr. J. Guo. The quality of this work would not have been as it is without his support and guidance. Therefore, the author would like to credit and express his gratitude to Dr. Guo.

References

[1] Ali Ismail Awad and Mahmoud Hassaballah. *Image Feature Detectors and Descriptors*. Vol. 630. Springer International Publishing, 2016, pp. 10–45. ISBN: 9783319288543. DOI: 10.1007/978-3-319-28854-3.

[2] Kuldeep Rambhai Barad. "Robust Navigation Framework for Proximity Operations around Uncooperative Spacecraft (MSc Thesis)". MA thesis. Delft University of Technology, 2020. URL: http://repository.tudelft.nl/..

[3] Blender. *Blender Website*. https://www.blender.org/. Oct. 2021.

[4] Carlos Campos Martínez et al. *ORB-SLAM3: An accurate Open-source library for visual, Visual-inertial and Multi-map SLAM*. 2020. arXiv: 2007.11898.

[5] Vincenzo Capuano et al. "Monocular-based pose determination of uncooperative space objects". In: *Acta Astronautica* 166.September 2019 (2020), pp. 493–506. ISSN: 00945765. DOI: 10.1016/j.actaastro.2019.09.027. URL: https://doi.org/10.1016/j.actaastro.2019.09.027.

[6] Mehregan Dor and Panagiotis Tsiotras. "ORB-SLAM applied to spacecraft non-cooperative rendezvous". In: *Space Flight Mechanics Meeting, 2018* (2018). DOI: 10.2514/6.2018-1963.

[7] ESA Space Debris Office. *ESA ' s Annual Space Environment Report*. Tech. rep. September. ESA, 2020.

[8] Danil Ivanov, Mikhail Ovchinnikov, and Marianna Sakovich. "Relative pose and inertia determination of unknown satellite using monocular vision". In: *International Journal of Aerospace Engineering* 2018 (2018). ISSN: 16875974. DOI: 10.1155/2018/9731512.

[9] Zeming Jin et al. "Monocular-Based Pose Estimation of Non-Cooperative Space Targets Using EKF and EKPF". In: *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)* (2020), pp. 46–51. DOI: 10.1109/cisp-bmei51763.2020.9263615.

[10] Ting Lei et al. "Pose estimation of a noncooperative target based on monocular visual SLAM". In: *International Journal of Aerospace Engineering* 2019 (2019). ISSN: 16875974. DOI: 10.1155/2019/9086891.

[11] Raul Mur-Artal, J. M.M. Montiel, and Juan D. Tardos. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5 (2015), pp. 1147–1163. ISSN: 15523098. DOI: 10.1109/TRO.2015.2463671. arXiv: 1502.00956.

[12] Raul Mur-Artal and Juan D. Tardos. "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras". In: *IEEE Transactions on Robotics* 33.5 (2017), pp. 1255–1262. ISSN: 15523098. DOI: 10.1109/TRO.2017.2705103. arXiv: 1610.06475.

[13] Andriy Myronenko and Xubo Song. "Point set registration: Coherent point drifts". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.12 (2010), pp. 2262–2275. ISSN: 01628828. DOI: 10.1109/TPAMI.2010.46. arXiv: 0905.2635.

[14] Bo J. Naasz et al. "Flight results from the HST SM4 Relative Navigation Sensor system". In: *Advances in the Astronautical Sciences* 137 (2010), pp. 723–744. ISSN: 00653438.

[15] NASA. *Measuring Earth's Albedo*. https://earthobservatory.nasa.gov/images/84499/measuring-earths-albedo. Oct. 2021.

[16] Wolfgang Rackl et al. "GNC architecture for the e . Deorbit mission". In: *7th European Conference for Aeronautics and Space Sciences (EUCASS)* (2017). DOI: 10.13009/EUCASS2017-317.