Politecnico di Milano

School of Industrial and Information Engineering

Department of Aerospace Science and Technology

Master of Science in Space Engineering

Stereovision-Based Pose and Inertia Estimation for Unknown and Uncooperative Space Objects

Advisor:
Prof. Michèle Lavagna

Co-Advisor:
Prof. Riccardo Bevilacqua

Graduation Thesis of:
Vincenzo Pesce
801404

Academic Year 2013-2014

To my parents.

Bisogna cercare orizzonti lontani non solo per scopi
pragmatici, ma perché l'immaginazione e l'anima
vengano nutrite dalla meraviglia e dalla bellezza.

Wernher von Braun

Acknowledgments

The greatest thanks go to my parents, Francesco and Angela, for their constant support during these important years. They always bolstered my decisions and, with their advices, they tried to guide me through the tortuous path of the life. I would like to thank my brother, Aldo, my aunt, Teresa, and all my family. Particular thanks go to my advisor, Professor Michèle Lavagna for guiding me through this work and for being a source of inspiration and motivation. The valuable experience at the University of Florida would not have been possible without her mediation. I would like to thank Professor Riccardo Bevilacqua for the opportunity to work at the ADAMUS (ADvanced Autonomous MUltiple Spacecraft) Lab at the University of Florida and for his daily support and essential advices. A special thanks to all my friends Emiliano, Giacomo, Salvatore, Luca and especially The Trio, for sharing the best and worst of these years. Particular thanks to my 'historic', always present friends, Giancarlo, Juri, Manuel, Emanuele, Vincenzo e Fabio. I have really enjoyed working together with the ADAMUS team and I would like to thank David, Levin, Leo and Dave for the helpful discussions and their support. Finally, I would like to thank Reza and Max for their friendship and the time we spent together, enjoying the Floridian life.

Abstract

Autonomous close proximity operations are an arduous and attractive problem in space mission design. In particular, the estimation of pose, motion and inertia properties of an uncooperative object is a challenging task because of the lack of available a priori information. In addition, good computational performance is necessary for real applications. This thesis develops a method to estimate the relative position, velocity, angular velocity, attitude and inertia properties of an uncooperative space object using only stereo-vision measurements. The classical Extended Kalman Filter (EKF) and an Iterated Extended Kalman Filter (IEKF) are used and compared for the estimation procedure. The relative simplicity and low computational cost of the proposed algorithm allow for an online implementation for real applications. The developed algorithm is validated by numerical simulations in MATLAB using different initial conditions and uncertainty levels. The goal of the simulations is to verify the accuracy and robustness of the proposed estimation algorithm. The obtained results show satisfactory convergence of estimation errors for all the considered quantities. An analysis of the computational cost is addressed to confirm the possibility of an onboard application. The obtained results, in several simulations, outperform similar works present in literature. In addition, a video processing procedure is presented to reconstruct the geometrical properties of a body using cameras. This method has been experimentally validated at the ADAMUS (ADvanced Autonomous MUltiple Spacecraft) Lab at the University of Florida.

Keywords: Pose and Inertia Estimation; Stereo-Vision; Autonomous Close Proximity Operations; Video Processing.

Sommario

Lo sviluppo di missioni spaziali, implicanti operazioni di prossimità autonome, è un problema complesso e affascinante. In particolare, la stima di posizione, assetto e dinamica relativa e della matrice di inerzia di un oggetto completamente sconosciuto, è un arduo compito a causa della mancanza di informazioni note a priori. Inoltre, in un'applicazione reale, l'algoritmo utilizzato deve offrire buone prestazioni computazionali. Questa tesi introduce un metodo per stimare posizione, velocità, velocità angolare e assetto relativi di un oggetto spaziale non cooperativo oltre a ricostruirne le proprietà di inerzia. Tutto questo, solamente tramite misure collezionate grazie ad una coppia di camere in configurazione stereoscopica. La stima è ottenuta utilizzando un Filtro di Kalman Esteso (EKF) e un Filtro di Kalman Esteso e Iterato (IEKF). I risultati, con entrambi i metodi, sono successivamente confrontati. La semplicità dell'algoritmo e il suo basso costo computazionale lo rendono un ottimo candidato per un'applicazione reale. L'algoritmo è stato validato con simulazioni numeriche implementate in MATLAB volte a valutarne l'accuratezza. Diverse condizioni iniziali e livelli di incertezza sono stati utilizzati. I risultati ottenuti mostrano una buona convergenza dell'errore relativo di tutte le componenti dello stato. Infine, un'analisi delle prestazioni computazionali è presentata. Questa è necessaria per ottenere un'ulteriore conferma del possibile utilizzo in applicazioni reali. I risultati ottenuti, nella maggior parte delle simulazioni, sono migliori di quelli presenti in simili lavori disponibili in letteratura. E' infine proposto un metodo per la ricostruzione delle proprietà geometriche e di massa tramite l'elaborazione di video. Questo metodo è stato validato tramite una campagna sperimentale all'ADAMUS (ADvanced Autonomous MUltiple Spacecraft) Lab presso la University of Florida.

Parole Chiave: Stima di posizione, assetto e dinamica relativa; Visione Stereoscopica; Operazioni di prossimita' autonome; Elaborazione di Video.

Estratto in lingua italiana

Negli ultimi anni, l'utilizzo di spacecraft autonomi sta diventando un aspetto fondamentale nel progetto di missioni spaziali. Questo lavoro è focalizzato sulla possibilità di eseguire operazioni di prossimità in modo del tutto autonomo. In particolare, l'obiettivo è quello di implementare un algoritmo capace di stimare posizione, velocità, assetto e proprietà di inerzia di un corpo completamente sconosciuto. Per far ciò, le uniche misure disponibili sono fornite da due camere poste in configurazione stereoscopica. Grazie ad esse ed ad un accurato modello della dinamica, è possibile progettare un filtro robusto per la stima delle quantità desiderate. Il processo di stima descritto, risulta essere un'operazione complessa e difficile per via della mancanza di informazioni note a priori. Infatti, un oggetto spaziale sconosciuto e non cooperativo non può fornire nessuna informazione al satellite predisposto per operazioni di prossimità. Inoltre, è assunta la mancanza di punti caratteristici sull'oggetto esaminato come, per esempio, LED o retroriflettori. In questo modo, il campo di applicazione del metodo implementato è notevolmente maggiore. Ad esempio, detriti spaziali e asteroidi rientrano in questa categoria e molteplici applicazioni possono essere considerate. In particolare, l'algoritmo proposto potrebbe essere utilizzato su uno spacecraft autonomo per la rimozione di detriti spaziali o per la navigazione attorno ad un asteroide. Applicazioni in campi differenti da quello spaziale sono possibili. In più, il presente lavoro propone anche un metodo per la ricostruzione delle proprietà di inerzia dell'oggetto analizzato tramite solamente informazioni video.

Modello Dinamico

Una descrizione accurata del moto relativo è fondamentale in applicazioni spaziali con più di un oggetto. I metodi più comuni modellano gli oggetti coinvolti come puntiformi. Tuttavia, in operazioni di prossimità, questi metodi risultano non essere sufficienti. Infatti, per descrivere in maniera appropriata il moto relativo è necessario considerare i corpi come oggetti con sei gradi di libertà. In questo lavoro, le dinamiche traslazionali e rotazionali sono considerate essere disaccoppiate. Un modello non lineare è utilizzato per descrivere la dinamica traslazionale per superare le limitazioni derivanti dalle assunzioni necessarie per eseguire una linearizzazione (equazioni di Clohessy-Wiltshire). Per descrivere la dinamica rotazionale, sono utilizzate le classiche equazioni di Eulero. Grazie alla combinazione di questi due modelli, è possibile descrivere in maniera appropriata il moto di un punto diverso dal centro di massa.

Modello dell'Osservazione

Le informazioni disponibili sono ottenute grazie a due camere in configurazione stereoscopica. Per descrivere il loro funzionamento, è utilizzata la teoria del foro stenopeico. Con questo metodo, è possibile esprimere le componenti di un punto in 3D come proiezioni su un piano 2D. In più, è possibile collezionare informazioni sul flusso ottico. Da questo, possono essere estratte misure riguardanti velocità traslazionale e angolare. Tutte queste misure sono immagazzinate per ogni istante di tempo e usate come input nel processo di filtraggio. È importante sottolineare come, è stato scelto di utilizzare delle camere per evitare sensori attivi come, ad esempio, il LIDAR. In questo modo, il sensore passivo non interagisce in nessun modo con l'ambiente esterno. Sensori utilizzanti laser, oltre ad essere inutilizzabili in applicazioni con particolari tipi di materiali poco riflettenti, implicano l'utilizzo di molta potenza elettrica. Questo può risultare svantaggioso in applicazioni reali. Inoltre, una configurazione stereoscopica è stata scelta in quanto, utilizzando visione monoculare, non possono essere recuperate informazioni sulla profondità dell'oggetto.

Metodo di Stima

Per il metodo di stima, un Filtro di Kalman Esteso (EKF) e un Filtro di Kalman Esteso e Iterato (IEKF) sono confrontati. Entrambi sfruttano il modello della dinamica e le misure disponibili per stimare posizione, velocità, velocità angolare, assetto e proprietà di inerzia di un oggetto non cooperativo. L'IEKF risulta avere, in questa applicazione, prestazioni leggerente migliori ed essere più robusto. Differenti simulazioni sono analizzate per testare la convergenza e l'affidabilità del metodo proposto. In tutte le simulazioni, le componenti dello stato presentano una convergenza soddisfacente. La stima delle componenti di inerzia è tuttavia influenzata dalla velocità angolare dell'oggetto osservato. Infatti, risulta che con una bassa velocità angolare, le componenti della matrice di inerzia non influenzano in maniera significativa la stima dello stato e per questo, tendono a restare costanti nel processo di filtraggio. Per migliorare le prestazioni, in questo caso, è necessario aggiungere una pseudo misura per vincolare le componenti di inerzia. Tuttavia, questo implica la conoscenza dell'accelerazione angolare dell'oggetto analizzato. Questa misura può essere ottenuta solamente integrando numericamente le informazioni disponibili sulla velocità angolare relativa. Nel caso di elevata velocità angolare, l'inerzia influenza in maniera rilevante la dinamica rotazionale dell'oggetto. In questo caso, le componenti della matrice di inerzia convergono al valore esatto ma è presente un leggero decadimento generale delle prestazioni di stima. È necessario specificare che è assunto un moto senza alcun momento esterno. In questo caso, la matrice di inerzia ha solamente due gradi di libertà osservabili. Quindi, è possibile stimare solamente i rapporti delle componenti della matrice di inerzia. Per far ciò, la matrice di inerzia è parametrizzata in maniera opportuna.

Ricostruzione delle Proprietà di Inerzia

Come detto, in condizioni con un basso valore della velocità angolare, è necessario utilizzare metodi numerici per ottenere un valore approssimato dell'accelerazione angolare. Per evitare di aggiungere rumore ed aumentare l'errore utilizzando questi metodi, è sviluppato un processo di ricostruzione delle proprietà di inerzia, utilizzando dei video. Infatti, da un video è possibile estrarre immagini, quindi una nuvola di punti rappresentati l'oggetto analizzato e da questa ricostruirne una mesh 3D. Assumendo densità costante, è possibile ricostruire le proprietà di massa dell'oggetto analizzato, conoscendone una geometria 3D approssimata. La campagna sperimentale condotta mostra come, con questo metodo, è possibile stimare le proprietà di inerzia di un oggetto dalla forma complessa con errori contenuti.

Contents

1 **Introduction** 27

1.1 Motivation 28

1.1.1 Uncooperative Object 29

1.1.2 Relative Dynamics and Inertia Matrix Estimation 29

1.1.3 Stereo-Vision 29

1.2 Literature Review 30

1.2.1 Autonomous Spacecraft Proximity Operation Missions 30

1.2.2 Relative Spacecraft Motion Estimation 31

1.2.3 Relative Spacecraft Dynamical Model 32

1.2.4 Vision Based Navigation 33

1.2.5 Thesis Contributions and Algorithm Overview 34

1.3 Thesis Outline 36

2 **Dynamical Model** 37

2.1 Coordinate Frames 38

2.2 Rigid Body Dynamics 39

2.2.1 Relative Translational Dynamics 39

2.2.2 Relative Rotational Dynamics 41

2.3 Clohessy-Wiltshire equations 44

2.4 Coupled Relative Spacecraft Motion Model 45

2.5 Results 48

2.5.1 Simplified Coupled Model 48

2.5.2 CW comparison 51

2.5.3 Computational Performance 54

3 **Observation Model and Computer Vision** 57

3.1 Stereoscopic Camera Model 57

3.1.1 Pinhole Camera Model 57

3.1.2 Optical Flow and Image Velocity 60

3.1.3 Disparity 61
3.2 Computer Vision Algorithm 63
3.2.1 Feature Extraction and Matching 63
3.2.2 Feature Tracking 65

4 **Inertia Tensor Observability Analysis** 71

4.1 Non-linear Observability Analysis 72
4.1.1 Lie Derivatives 72
4.1.2 Inertia Observability Algebraic Test 73

5 **Estimation Methodology** 77

5.1 Extended Kalman Filter 77
5.2 Iterated Extended Kalman Filter 82

6 **Stereovision-Based State and Inertia Estimation** 85

6.1 Mathematical Formulation 85
6.1.1 Assumptions and Notation 85
6.1.2 Process Model 86
6.1.3 Observation Model 88
6.1.4 Estimation Procedure 89
6.2 Inertia Ratios Estimation 90
6.2.1 Inertia Parametrization 91
6.2.2 Equality Constraint 94
6.3 Numerical Simulations 96
6.3.1 Case A 99
6.3.2 Case B 110
6.3.3 Case C 120
6.3.4 Case D 120
6.3.5 Case E 124
6.3.6 Case F 129
6.3.7 Computational Analysis 134

7 **Inertia Matrix Reconstruction** 137

7.1 From Video to Images 138
7.2 From Images to Point Cloud 138
7.3 From Point Cloud to 3D Mesh 139
7.4 From 3D Mesh to Inertia Properties 140
7.4.1 Inertia Parameters From Triangular Surface Mesh 140

7.4.2 Constant Density Assumption 144
7.5 Experimental Validation 145
7.5.1 Experimental Setup 145
7.5.2 Case A - Simple Shape 146
7.5.3 Case B - Complex Shape 149

8 **Conclusions** 153

8.1 Review of Contributions 154
8.2 Future Work 154

A EKF Derivation 157

A.1 Prediction 157
A.2 Filtering 160

B Differentiation of noisy signal 165

B.1 Least Squares Polynomial Approximation 165
B.2 Tikhonov regularization 166
B.3 Tracking Differentiator 167

List of Figures

1.1 Orbital Express - Artistic Rendering 28
1.2 Georges Lemaitre ATV final approach 31
1.3 Algorithm Scheme 36

2.1 Leader - Target Reference Frames 38
2.2 Leader - Target graphical representation 46
2.3 Relative Position 49
2.4 Deviations 50
2.5 CW Relative Position 51
2.6 CW Zoom 52
2.7 CW Relative position for a docking case 53
2.8 Time variation with 1 feature point 55
2.9 Time variation with 3 feature points 55
2.10 Time variation with 6 feature points 56

3.1 RayDiagram 58
3.2 Pinhole Camera Model 58
3.3 Stereo Camera Configuration 59
3.4 Disparity Example 62
3.5 Surf Detected Points 64
3.6 MSER Detected Regions 64
3.7 FAST Detected Corners 65
3.8 Number of Detected Feature Points 66
3.9 Computational Time 67

5.1 EKF Scheme 81
5.2 IEKF Scheme 83

6.1 Relative Position Error 101
6.2 Relative Translational Velocity Error 102
6.3 Relative Angular Velocity Error 103

6.4 Relative Attitude Error 104
6.5 Inertia Ratio Error 105
6.6 Inertia Ratio Error 106
6.7 IEKF-EKF Differences 109
6.8 Relative Position Error 111
6.9 Relative Translational Velocity Error 112
6.10 Relative Angular Velocity Error 113
6.11 Relative Attitude Error 114
6.12 Inertia Ratio Error 115
6.13 Inertia Ratio Error 116
6.14 IEKF-EKF Differences 119
6.15 Relative Position Error 121
6.16 Relative Translational Velocity Error 122
6.17 Relative Angular Velocity Error 122
6.18 Relative Attitude Error 123
6.19 Inertia Ratio Error 123
6.20 Inertia Ratio Error 124
6.21 Relative Position Error 125
6.22 Relative Translational Velocity Error 125
6.23 Relative Angular Velocity Error 126
6.24 Relative Attitude Error 126
6.25 Inertia Ratio Error 127
6.26 Inertia Ratio Error 127
6.27 Inertia Ratio Error 128
6.28 Inertia Ratio Error 128
6.29 Relative Position Error 130
6.30 Relative Translational Velocity Error 130
6.31 Relative Angular Velocity Error 131
6.32 Relative Attitude Error 131
6.33 Inertia Ratio Error 132
6.34 Inertia Ratio Error 132
6.35 Inertia Ratio Error 133
6.36 Inertia Ratio Error 134
6.37 Computational Time 135

7.1 Sony HandiyCam HDR-CX110 145
7.2 3-DOF Simulator 146
7.3 High Pressure Tank 147

7.4 Tank Point Cloud 147
7.5 Tank 3D Mesh 148
7.6 3-DOF Simulator 149
7.7 3-DOF Simulator Point Cloud 149
7.8 3-DOF Simulator Mesh 150
7.9 3-DOF Simulator Mesh in MATLAB 151
7.10 3-DOF Simulator CAD Model 151

List of Tables

6.1 State Errors - Case A 107

6.2 State Errors - Case B 117

6.3 State Errors - Case C 120

6.4 State Errors - Case D 124

6.5 State Errors - Case E 129

6.6 State Errors - Case F 129

6.7 Computational Time 135

List of Algorithms

1 EKF 82
2 IEKF 84

Chapter 1

Introduction

Over the past few decades, spacecraft autonomy has become a very important aspect in space mission design. In this thesis, autonomous spacecraft proximity operations are discussed with particular attention to the estimation of position and orientation (pose), motion and inertia properties of an uncooperative object. The precise pose and motion estimation of an unknown object, such as a Resident Space Object (RSO) or an asteroid has many potential applications. In fact, it allows autonomous inspection, monitoring and docking. However, dealing with an uncooperative space body is a challenging problem because of the lack of available information about the motion and the structure of the target. The interest of the main space agencies, in these years, is focused on the gradual automation of the space missions because of its large number of practical applications. For example, in 2005, NASA sponsored the DART (Demonstration for Autonomous Rendezvous Technology) [1] project to develop and demonstrate automated navigation and rendezvous for a spacecraft. However, this mission failed with an unintentional collision with the target satellite. DARPA, in 2007, launched the Orbital Express mission [2] aimed at developing an approach for autonomous satellites servicing in orbit; ESA developed automated systems too. The ATV (Automated Transfer Vehicle) [3] to supply the ISS is one of the main examples and further studies are carried on about new rendezvous sensors and algorithms. Moreover, relative navigation between non-cooperative satellites can become a powerful tool in missions involving objects that cannot provide effective cooperative information, such as faulty or disabled satellite, space debris, hostile spacecraft, asteroids and so on.

In particular, the precise pose and motion estimation of an uncooperative object has possible applications in the space debris removal field. Space debris includes all man-made defunct objects, in Earth orbit or re-entering the atmosphere. The pose

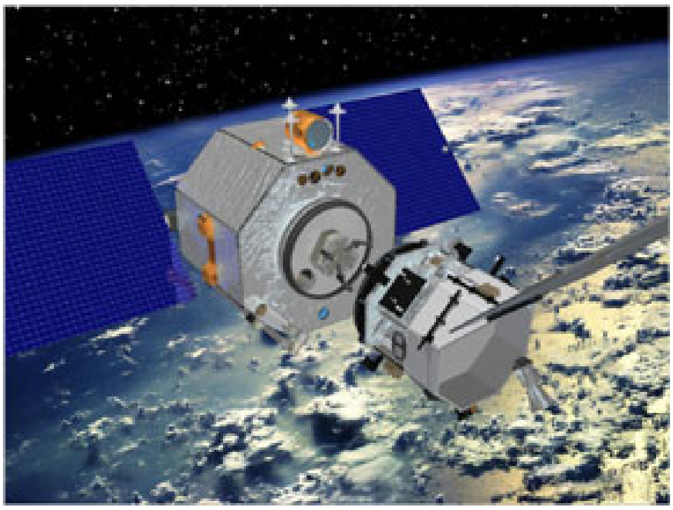


Figure 1.1. Orbital Express - Artistic Rendering

and the inertia matrix estimation is the first step to implement a system to recover and remove elements harmful to operational and active satellites. Additionally, the obtained algorithm can be installed on autonomous spacecraft for close-proximity operations to asteroids or for rendezvous manoeuvres. This thesis focuses on the problem of how to estimate the relative state and the inertia matrix of an unknown, uncooperative space object using only stereoscopic measurements. This information is provided by two cameras.

The methodology developed to solve this problem has many potential applications in other fields (iceberg-relative navigation [4], biomedical applications [5], vision-based unmanned aerial vehicle navigation [6], etc.). Current literature addresses the problem of relative state estimation with respect to an uncooperative object, assuming partial knowledge of the geometry or feature points of the target. In other cases, multiple spacecraft or sensors with high power consumption are utilized to compensate for the lack of information. None of these proposed algorithms has been physically implemented in a real application, one main reason being their high computational cost.

This chapter will review the current state of the art in spacecraft navigation and provide an overview of the contributions and outline of this thesis.

1.1 Motivation

Pose, motion and inertia matrix estimation is only the first step to achieve the goal of a complete autonomous system. In the future, smart satellites will operate and make decisions autonomously. However, in order to realize this futuristic vision, new methods for autonomous navigation and control are necessary. This work tries to take a step forward in this direction.

In particular, an algorithm for the estimation of the relative motion and shape of a completely unknown and uncooperative object is developed. Very simple sensors are assumed to be utilized. The only available information about the target is provided by stereo images collected by two cameras in a stereo configuration. In the following subsections, the motivation behind all these choices is addressed.

1.1.1 Uncooperative Object

The assumption of an uncooperative target has been carried out to have the possibility to apply the obtained results to a wide number of applications. Uncooperative objects have no target identifiers (such as optical feature points) and proximity sensors. This makes the estimation more difficult and less accurate, but more applicable to a vast range of in-space objects. With this assumption, it is possible to expand the applicability of this work to asteroids and completely unknown objects.

1.1.2 Relative Dynamics and Inertia Matrix Estimation

Autonomous close proximity operations are impossible without an accurate estimation of the relative dynamics. Knowledge of position, translational and rotational velocity and attitude is the necessary starting point for close approach and navigation around an unknown object.

In addition, the information about the inertia properties of a body is a crucial aspect to perform landing or docking with the inspected object. However, the estimation of the inertia matrix, in torque-free motion conditions, is a very difficult task and one of the open points that this thesis will try to solve.

1.1.3 Stereo-Vision

To estimate the relative dynamics of this unknown object, it is necessary to collect and interpret measurements. Specifically, this work assumes that only two cameras can provide stereoscopic images. Other works present in literature use different types of sensors. Several papers deal with the pose estimation using LIDAR (Light Detection and Ranging) [7], [8], [9]. This sensor is a laser based 3-D scanning system. LIDAR sends out an invisible laser light in all directions. These waves hit the surrounding objects and rebound back towards the sensor. Range and range rate can be estimated by measuring the time taken to transmit and receive the signal. Using LIDAR implies to manage a large amount of data and to have a large power consumption. For this reason, real-time missions can be hampered.

Moreover, these instruments decrease their efficiency in applications with poor reflectors, such as black, blue, violet objects or rounded surfaces and are sensitive to the sunlight. All these problematic issues discourage the use of this particular instrument for space proximity operations. Other authors used a single camera to estimate the relative dynamics [10],[11],[12], [13]. Nonetheless, the monocular vision cannot provide reliable depth information. This can be estimated but it tends to be less robust and accurate than the same information obtained by stereoscopic vision [14]. In addition, cameras are completely passive sensors. This means that they do not interact with the target and avoid the classical problems of active sensors. In fact, the signal, that active sensors send to the target, can be interfered by other sources and this affects the final measurement. Using cameras this problem is absent, since they simply read and interpret an input signal from the environment.

1.2 Literature Review

1.2.1 Autonomous Spacecraft Proximity Operation Missions

In last recent years, few on-orbit demonstrations of Autonomous Rendezvous and Proximity Operations (ARPO) have been carried out, with both successful and not fortunate results. Besides the mentioned DART, Orbital Express and ATV missions, there have been different ARPO attempts in the history of space exploration. One of the first mission was the Japanese Engineering Test Satellite 7 (ETS-7) [15] in 1998. This experiment was developed to demonstrate autonomous rendezvous and docking. The ETS-7 consisted of two satellites (chaser and target). An on-orbit capture was successfully proved using a vision-aided robotic manipulator. A few years later, The United States Air Force (USAF) XSS-10 [16] and XSS-11 [17] demonstrated on-orbit proximity operations without contact. XSS-11 in particular, accomplished a visual-inertia inspection of a non-cooperative target. However, it did not estimate the geometry or relative position, velocity, angular velocity and attitude. The autonomous relative navigation and docking is still a hot subject. In fact, NASA is developing a system for autonomous Asteroid Redirect Mission (ARM) [18]. This mission will use a sensor suite used for docking, including a visible camera, a long-wavelength infrared camera and a 3D lidar. This sensor system has been used for Sensor Test for Orion Relative Navigation Risk Mitigation (STORRM) experiment in 2011. A future test for this rendezvous sensor system will be performed by Raven. This mission will demonstrate the possibility of real-time relative navigation and autonomous rendezvous. Raven is slated to fly in 2016. Although many missions have obtained great results from the autonomy point of view, none has still estimated the relative state of a completely unknown and uncooperative space object.

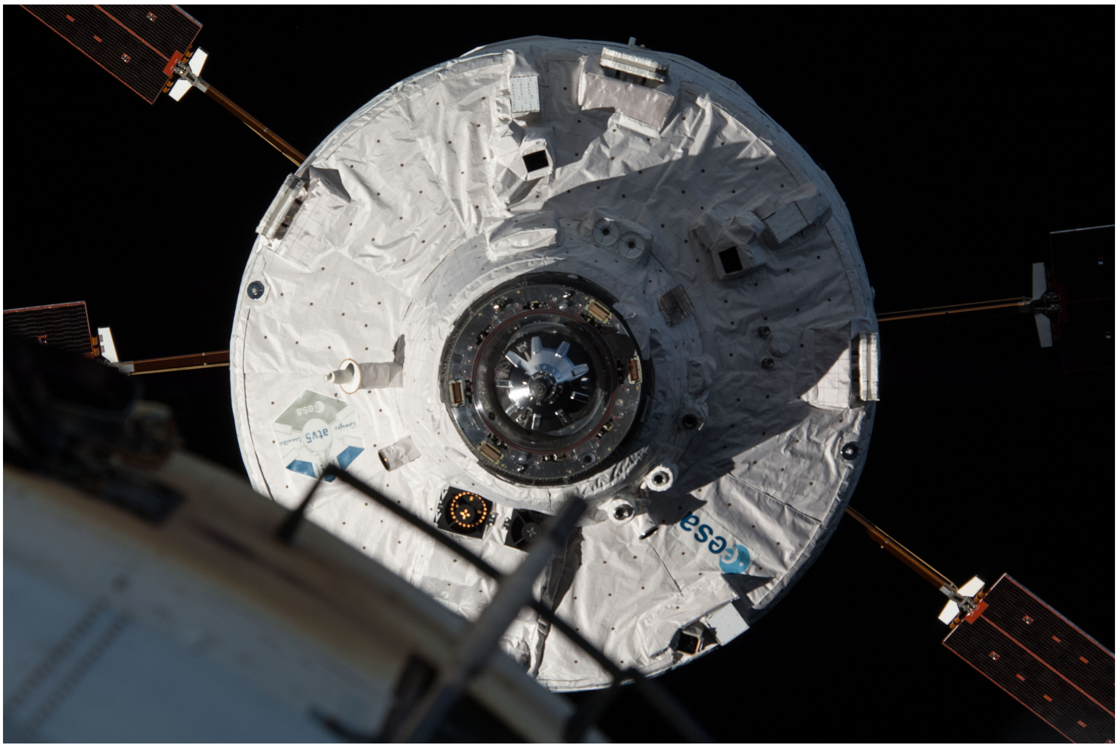


Figure 1.2. Georges Lemaitre ATV final approach

1.2.2 Relative Spacecraft Motion Estimation

An accurate relative spacecraft motion estimation is becoming very important in problems dealing with formation flying [19], [20] or docking [21]. In these two cases, the target is usually known and it shares information with the chaser. This makes the estimation easier than in the case of a completely unknown and uncooperative object. In addition, this estimation usually relies on relative sensors such as differential GPS receivers or light beacons. However, not much information is available in applications involving unknown objects. RSO, space debris, asteroids and so on, usually do not present any relative sensors. Dealing with these objects is typically more arduous because of the poor or absent a priori information about the target geometry or motion. No GPS measurements, no LEDs, no beacons and no motion information are present in case of uncooperative space missions. One of the main contribution in literature, addressing relative state estimation of an uncooperative target is from Lichter [22], [23]. He solves the problem of estimating the relative pose, motion and structure using a 3D vision sensor. This creates and processes point clouds to reconstruct the geometric shape of the object. From this information, he is able to extract a rough measurement of the centroid and rotation matrix. Then, two Kalman Filters (translation and rotation) are used to estimate the state and inertia properties. However, using 3D sensor involves more power consumption, computational cost and data to manage. In 2013, Segal and Gurfil presented a solution of the state estimation of a non-cooperative spacecraft using an Iterated Extended Kalman Filter (IEKF) [24], [25], [26]. Their approach is the baseline for this thesis. They develop and utilize a translational-rotational coupled model to describe the relative dynamics. Then, an IEKF is used to estimate the state. The basic assumption is to have only stereoscopic measurements. However, they do not estimate the inertia matrix, but they run N Kalman filters in parallel and, at the end, they choose the best value for the inertia tensor according to a Maximum A Posteriori (MAP) estimation. Thus, N filters must work simultaneously for an interval of time t to estimate the state. Then, all the estimated states are compared and the selected inertia matrix is the one that provides better results in terms of state error. This method clearly cannot be implemented on a real spacecraft because of the large computational cost needed in problems without previous knowledge of geometry information. A very interesting work is by Tweddle [27], [28]. Assuming a stationary leader, he developed a method to estimate the state and structure of an unknown object using a smoothing algorithm. Smoothing and Mapping (SAM) are commonly used for simultaneous localization and mapping (SLAM) problems. This method estimates the complete 'robot' trajectory in time and not only the current pose. The particular smoothing algorithm used by Tweddle is the Incremental Smoothing Algorithm (iSAM) introduced by Kaess et al. [29], [30]. This method performs fast incremental updates to compute a full map and trajectory of the object at any time. However, this method does not handle properly trajectory with loops and non-linear problems. This is an important aspect because the most general equations describing the motion of two satellites are non-linear. For this reason, he assumes a fixed leader and a very simple linearised dynamics. In our work, a Kalman Filter has been selected as estimator for its better computational performance. In fact, [31] clearly expresses how the smoothing algorithm is more accurate but also less computationally efficient than the classical filters. Moreover, filtering techniques work better under high uncertainties problems.

1.2.3 Relative Spacecraft Dynamical Model

The model of the relative dynamics between two or more spacecraft must be correct and precise when dealing with proximity operations. This is the first step to obtain accurate results. The first work trying to model the relative dynamics between two spacecraft was the one by Clohessy and Wiltshire (CW) [32]. This model describes the relative translational motion between two objects that are assumed to be on a circular orbit around a spherical Earth. Another assumption of this model is that the target must be very close to the chaser. This is a linearised model and it has been widely used for works dealing with formation flying of multiple spacecraft and docking. A different model for elliptic reference orbits was developed by Tschauner-Hempel [33]. In the last decade, orbital element differences are used to describe relative motion [34], [35], [36], [37] also considering perturbed motion [38], [39], [40]. Thus far, point-mass models are cited. However, in close proximity applications, it is necessary to use models that describe the relative motion of six-DOF spacecraft. In particular, the position of points different from the centre of mass has to be taken into account. Pan and Kapila [41] were among the first to consider the translational-rotational dynamical coupling between two spacecraft. Nonetheless, very few models describing the relative motion between 6-DOF bodies are present in literature. One of these is the one presented by Segal and Gurfil [42]. This model illustrates, in an accurate way, the completely coupled translational-rotational motion of two 6-DOF spacecraft.

1.2.4 Vision Based Navigation

Computer vision is a vast and complex field that is used in many classical application and recent innovative researches. Computer vision involves extraction, analysis and comprehension of information deriving from images. This process is somehow similar to what humans do. Vision, in human behaviour, is not only recording light, but also an interpretation of the received input. Computer vision, wants to imitate this human characteristic. Usually, the most common methods for navigation and mapping in an unknown environment are Visual Odometry [43] and Simultaneous Localization and Mapping [44]. Visual odometry takes its name from the wheel odometry. In fact, as it is possible to estimate the motion of a vehicle by integrating the number of turns of the wheels, in a similar way, the pose of an object can be estimated through the variations in the collected images. This method works tracking features correspondence between two subsequent images. Simultaneous Localization and Mapping is a different estimation technique. This method keeps track of a map of the environment, also when it is not needed, to recover the complete path of an object. Therefore, Visual Odometry recovers the path incrementally, analysing only the local trajectory; SLAM tries to 'close the loop', storing information about the past location of a point and matching it when it is seen again. Although SLAM could be more accurate, Visual Odometry results to be more robust, simpler and less computationally expensive. This is one of the reasons why the Visual Odometry has been widely used for real applications. In particular, it has been used for the Mars Exploration Rovers (MER) [45]. MER's visual odometry system reconstructs the pose of the rover by tracking terrain feature using two cameras. The estimation process is completed by a maximum likelihood estimation methodology. In addition, different papers in literature address the pose estimation using visual odometry technique [46]. Also SLAM approach has been used for pose estimation in space applications [47], [48]. In addition, other fields have used SLAM in similar problems. For example, there are papers involving autonomous iceberg navigation [4], unmanned aerial vehicle (UAV) applications [6] and also medical applications [5].

1.2.5 Thesis Contributions and Algorithm Overview

The goal of this work is to implement an algorithm suitable for real missions. It has to provide an estimation of the relative state of an unknown object and it has to be able to reconstruct its inertia properties. The only available measurements are images provided by two cameras in a stereoscopic configuration. In addition, in order to be suitable for real applications, the algorithm has to computationally efficient. The first important aspect to take into account is that the observed object, in our work, is unknown and uncooperative. Few works in the current literature [28], [25] address this kind of problem. This is because, without any information about the target object, the estimation of the relative quantities is more difficult and less accurate. Many approaches assume to have partial information about the relative state of the observed body. This allows a simpler solution of the estimation problem but also a smaller range of possible applications. Another aspect to consider is the modelling of the relative dynamics. In fact, it has to be accurate enough to properly describe the relative motion between two objects in space but also simple and computationally efficient. Our work uses a model that exploits the superposition of the rotational and translational dynamics instead of a completely coupled model. In this way, we are able to obtain good performance without adding complexity. Moreover, we have decided to do not use a simple linear model due to the too restricting involved assumptions. Another characteristic of our model is to rely only on stereo images. This allows to collect less data with respect to 3D sensors. Moreover, it guarantees computational efficiency and less power consumption with respect to sensors using lasers. For all these reasons, two cameras are the ideal candidates for a real implementation sensor. However, from the algorithm point of view, this means to have less accurate measurement and the necessity to rely on computer vision algorithms. In this work, since no experimental validation is performed, only a review of the available algorithms for feature extraction and matching is presented. In addition, an observation model, to be used in the filter, that exploits also the information about the optical flow is implemented. The most important aspect that this thesis tries to solve is the inertia tensor estimation. In the current literature, to the author's knowledge, only Tweddle [28] solved the problem to estimate the ratios of the inertia tensor, in torque free motion, relying only on stereo-vision measurements. In fact, most of the works present in literature try to solve the problem relying on 3D graphical reconstructions. These are usually computationally expensive and not suitable for an online implementation. Moreover, Tweddle's work only considers a spinning satellite and a stationary inspector spacecraft, using a very simple dynamical model. It also uses a smoothing algorithm that does not offer good computational performance. In fact, using the author's words: '*The results in this section do not illustrate a real time system, but rather a system that is better suited to offline applications*'. For this reason, this thesis solves the problem of estimating the inertia tensor exploiting a simple Kalman filter technique. Its extensive use in real time applications, allows for an efficient online implementation. In fact, instead of introducing new, computationally expensive, computer vision algorithm for only a partial relative state reconstruction, the idea of this algorithm is to rely on robust and reliable estimation procedures. At the end, this work also presents an offline 3D reconstruction of the geometrical properties of the object only using video processing. This is done to demonstrate that it is possible, using the data collected in the filtering process, to reconstruct, a posteriori, the geometry of the inspected body. Summarizing, the presented algorithm wants to estimate the *complete relative state* and the *inertia properties* of an *unknown and uncooperative* body, using only the stereo-vision. This is done without assuming any simplified dynamical model and trying to reduce the computational effort. To the author's knowledge, there are not actual works in literature dealing with this problem at this complexity level. In addition, a method for the estimation of the inertia tensor in torque free motion, therefore without assuming any capture of the object or imposed torque, is developed. Thus, this work proposes a solution for an open problem in the current literature. In addition, this thesis focuses its attention on the computational efficiency of the algorithm, differently than previous works. Concluding, in fig. 1.3 a general summary of the algorithm is presented. The most relevant aspects to solve the problem are highlighted. The filtering procedure will be analysed in the following chapters.

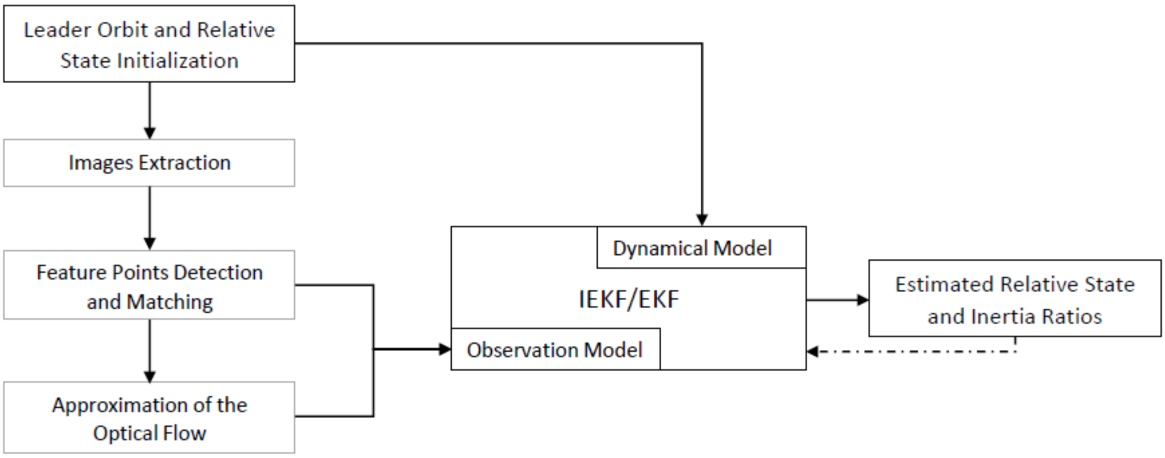

Leader Orbit and Relative State Initialization
Images Extraction
Feature Points Detection and Matching
Approximation of the Optical Flow
Dynamical Model
IEKF/EKF
Observation Model
Estimated Relative State and Inertia Ratios

Figure 1.3. Algorithm Scheme

1.3 Thesis Outline

The first chapter of this thesis is the chapter 2. It provides a brief review of the rigid body dynamics, describes the classical Clohessy-Wiltshire model and analyse recent coupled models. In addition, the dynamical model selected for this thesis is elucidated. Chapter 3 presents the observation and camera model of this work. It describes how to extract relative information of the target from stereo images. A review of computer vision algorithms is also presented. Chapter 4 provides a non-linear observability analysis of the inertia tensor to prove that it is not completely observable with no presence of external forces or torques. Chapter 5 introduces the concepts of Extended Kalman Filter (EKF) and Iterated Extended Kalman Filter, stressing the differences and the different formulations. Chapter 6 illustrates the developed estimation algorithm underling the new contributions introduced by this work. In chapter 7 a video processing procedure to recover inertia properties is illustrated. Experimental results are presented. Chapter 8 concludes this thesis, summarizing the contributions and the future work.

In appendix A the derivation of a discrete EKF is presented. Appendix B, instead, briefly describes some numerical methods to differentiate noisy signals.

Chapter 2

Dynamical Model

The accurate description of the relative motion is a key point in space systems involving more than one object. Correct modeling of relative translational and rotational motion is essential for autonomous missions. In literature, a large number of studies about point-mass models for relative spacecraft translational motion can be found. The most famous and used model is the one presented by Clohessy and Wiltshire [32]. Usually, these models are not sufficiently accurate when the faced problem deals with multiple cooperative spacecraft.

The interest in spacecraft formation flying and autonomous close proximity operation missions, motivated the research of a more accurate and complete relative spacecraft motion model. However, models for 6-DOF spacecraft have gained attention in literature only in recent years. In particular, Pan and Kapila [41], first addressed the coupled translational and rotational dynamics of two spacecraft. Nonetheless, in these models, the coupling is only induced by the gravity gradient.

In the last years, Segal and Gurfil [24] developed a model that takes into account a kinematic coupling. This coupling is obtained describing the relative dynamics of a point that is not the centre of mass (c.m.) of the spacecraft. They also show the importance of this coupling in vision-based relative attitude and position control, where arbitrary feature points on a target vehicle have to be tracked. However, this model implies complicated equations and a formulation that is not convenient to be used in a Kalman filter. For this reason, in this work, a simplified model is used to take into account the coupling without introducing complexity.

The purpose of this chapter is to provide a holistic review of the background material about rigid body dynamics, to briefly summarize the Clohessy-Wiltshire equation, describe the Segal-Gurfil coupled model and the simplified one.

2.1 Coordinate Frames

The location of a point in a three dimensional space must be specified with respect to a reference system. An appropriate description of the used coordinate systems is provided in this section. Two objects are considered: a leader L and a target T. In this thesis, the *leader* is the inspecting spacecraft and the *target* is the unknown, uncooperative object.

The standard Earth-centred, inertial, Cartesian right-hand reference frame is indicated with the letter $\mathcal{I}$. $\mathcal{L}$ is the local-vertical, local-horizontal Euler–Hill (LVLH) reference frame. It is fixed to the leader spacecraft's c.m., the $\hat{\mathbf{x}}$ unit vector directed from the spacecraft radially outward, $\hat{\mathbf{z}}$ normal to the leader orbital plane, and $\hat{\mathbf{y}}$ completing the frame.

Then with $\mathcal{J}$, a Cartesian right-hand body-fixed reference frame attached to the leader spacecraft's c.m. is denoted.

Finally $\mathcal{T}$, a Cartesian right-hand body-fixed reference frame centred on the target spacecraft's c.m.. It is also assumed that this frame is coincident with the principal axis of inertia.

In this work, the frames $\mathcal{J}$ and $\mathcal{L}$ are assumed to be aligned. From now on, only the letter $\mathcal{L}$ is used to refer to both of them.

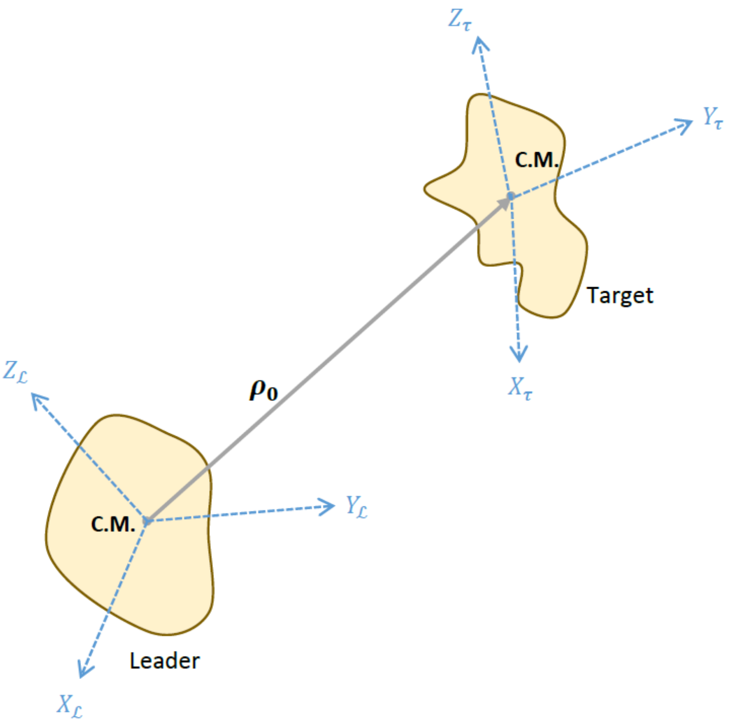


Figure 2.1. Leader - Target Reference Frames

2.2 Rigid Body Dynamics

In this section, both relative translational and rotational dynamics are analysed. These are described by Newton's Second Law and Euler's Equation of Rotational Motion. These laws are valid under the assumption of an inertial reference frame.

2.2.1 Relative Translational Dynamics

According to the Newton's Second Law, under the assumptions of no external or internal forces except gravity, spherical bodies, no tidal forces, attractor's mass much larger than the orbiting body's mass, the equations of motion of the Keplerian two-body problem can be written as:

$$\ddot{\mathbf{r}} + \mu \frac{\mathbf{r}}{r^3} = 0 \tag{2.1}$$

where $\mathbf{r} = [x, y, z]^T$ is the position vector in the $\mathcal{I}$ frame, μ is the Earth's gravitational constant and $r = ||\mathbf{r}||$. This equation of motion can be written for both leader and target, replacing $\mathbf{r}$ with $\mathbf{r}_L$ and $\mathbf{r}_T$, the positions of the leader and the target in the $\mathcal{I}$ frame.

$$\ddot{\mathbf{r}}_L + \mu \frac{\mathbf{r}_L}{{r_L}^3} = 0 \tag{2.2}$$

$$\ddot{\mathbf{r}}_T + \mu \frac{\mathbf{r}_T}{{r_T}^3} = 0 \tag{2.3}$$

Knowing the semi-major axes a_L and a_T, the eccentricities e_L and e_T and the true anomalies ϑ_L and ϑ_T of the leader and target orbit respectively, the expression of the magnitudes of the position vectors are obtained:

$$r_L = \frac{a_L(1 - {e_L}^2)}{1 + e_L \cos \vartheta_L}; \qquad r_T = \frac{a_T(1 - {e_T}^2)}{1 + e_T \cos \vartheta_T}; \tag{2.4}$$

Now, the relative position vector between the leader and the target can be defined as $\boldsymbol{\rho} = \mathbf{r}_T - \mathbf{r}_L$. Subtracting eq. (2.2) from eq. (2.3) yields:

$$\ddot{\boldsymbol{\rho}} = \frac{\mu(\mathbf{r}_L + \boldsymbol{\rho})}{||\ \mathbf{r}_L + \boldsymbol{\rho}\ ||^3} + \mu \frac{\mathbf{r}_L}{{r_L}^3} \tag{2.5}$$

The relative acceleration can also be expressed as:

$$\ddot{\boldsymbol{\rho}} = \frac{d^2\boldsymbol{\rho}}{dt^2} + 2\boldsymbol{\omega} \times \frac{d\boldsymbol{\rho}}{dt} + \frac{d\boldsymbol{\omega}}{dt} \times \boldsymbol{\rho} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{\rho}) \tag{2.6}$$

Considering the relative acceleration in the frame $\mathcal{L}$, $\boldsymbol{\omega}$ is the angular velocity of frame $\mathcal{L}$ with respect to $\mathcal{I}$. Being, by assumption, $\boldsymbol{\omega}$ normal to the orbital plane, leads to

$$\boldsymbol{\omega} = [0, 0, \dot{\vartheta}_L]^T \tag{2.7}$$

Knowing that

$$\boldsymbol{\rho}_{\mathcal{L}} = [x, y, z]^T \tag{2.8}$$

substituting eqs. (2.5), (2.7) and (2.8) into eq. (2.6) the following equations are obtained:

$$\ddot{x} - 2\dot{\vartheta}_L\dot{y} - \ddot{\vartheta}_L y - \dot{\vartheta}_L^2 x = \frac{\mu(r_L + x)}{[(r_L + x)^2 + y^2 + z^2]^{\frac{3}{2}}} + \frac{\mu}{{r_L}^2} \tag{2.9}$$

$$\ddot{y} + 2\dot{\vartheta}_L\dot{x} + \ddot{\vartheta}_L x - \dot{\vartheta}_L^2 y = -\frac{\mu y}{[(r_L + x)^2 + y^2 + z^2]^{\frac{3}{2}}} \tag{2.10}$$

$$\ddot{z} = -\frac{\mu z}{[(r_L + x)^2 + y^2 + z^2]^{\frac{3}{2}}} \tag{2.11}$$

$\dot{\vartheta}_L$ and $\ddot{\vartheta}_L$ are the orbital angular velocity and acceleration of the leader and are equal to

$$\dot{\vartheta}_L = \sqrt{\frac{\mu}{{a_L}^3(1 - {e_L}^2)^3}}(1 + e_L \cos\vartheta_L)^2 \tag{2.12}$$

$$\ddot{\vartheta}_L = \frac{-2\dot{r}_L\dot{\vartheta}_L}{r_L} \tag{2.13}$$

This set of non-linear ordinary differential equations describes the relative translational dynamics between two objects in space.

2.2.2 Relative Rotational Dynamics

As for the translational dynamics, a model that describes the rotational motion of the target relative to the leader is illustrated. First of all, the relative angular velocity of the target with respect to the leader is defined as $\boldsymbol{\omega} = \boldsymbol{\omega}_T - \boldsymbol{\omega}_L$ with $\boldsymbol{\omega}_T$ and $\boldsymbol{\omega}_L$ angular velocities of target and leader respectively. To parametrize the relative attitude, it has been decided to use a rotation matrix D which performs the transformation between the body fixed frame $\mathcal{T}$ relative to the target and the body fixed frame $\mathcal{L}$ relative to the leader. The components of this matrix are combinations of relative Euler parameters q_0, q_1, q_2, q_3. These are four parameters that describe a finite rotation about an arbitrary axis. They also form a quaternion $\mathbf{q} = [q_0, q_1, q_2, q_3]^T$ where $[q_1, q_2, q_3]$ is the vector part and q_0 is the scalar one. Therefore, the matrix D is defined as

$$D(\mathbf{q}) = \begin{bmatrix} {q_1}^2 - {q_2}^2 - {q_3}^2 + {q_0}^2 & 2(q_1 q_2 - q_3 q_0) & 2(q_1 q_3 + q_2 q_0) \\ 2(q_1 q_2 + q_3 q_0) & -{q_1}^2 + {q_2}^2 - {q_3}^2 + {q_0}^2 & 2(q_2 q_3 - q_1 q_0) \\ 2(q_1 q_3 - q_2 q_0) & 2(q_2 q_3 + q_1 q_0) & -{q_1}^2 - {q_2}^2 + {q_3}^2 + {q_0}^2 \end{bmatrix}$$

Note that all the components of the quaternion are relative components. The symbol δ has been avoided for sake of simplicity. In the next equations, a different notation is introduced. The symbol $a|^N$ indicates that the quantity 'a' is evaluated in the N frame, on the other hand, $(\frac{db}{dt})|_M$ indicates the derivative of the quantity 'b' in the M frame.

Knowing the rotation matrix D, the relative angular velocity in both target and leader frames can be calculated.

$$\boldsymbol{\omega}|^{\mathcal{L}} = D\boldsymbol{\omega}_T|^{\mathcal{T}} - \boldsymbol{\omega}_L|^{\mathcal{L}} \tag{2.14}$$

It is also possible to describe the relative attitude kinematics. Using the quaternion kinematic equations of motion:

$$\dot{\mathbf{q}} = \frac{1}{2} Q\boldsymbol{\omega}|^{\mathcal{T}} \tag{2.15}$$

with

$$Q(\mathbf{q}) = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}$$

The following derivation can be obtained. It is first necessary to derive the relative angular velocity in the inertial frame:

$$\left.\frac{d\boldsymbol{\omega}}{dt}\right|_{\mathcal{I}} = \left.\frac{d\boldsymbol{\omega}_T}{dt}\right|_{\mathcal{I}} - \left.\frac{d\boldsymbol{\omega}_L}{dt}\right|_{\mathcal{I}} \tag{2.16}$$

and using the rotation matrix D yields to:

$$\left.\frac{d\boldsymbol{\omega}}{dt}\right|_{\mathcal{I}}^{\mathcal{L}} = D\left.\frac{d\boldsymbol{\omega}_T}{dt}\right|_{\mathcal{I}}^{\mathcal{T}} - \left.\frac{d\boldsymbol{\omega}_L}{dt}\right|_{\mathcal{I}}^{\mathcal{L}} \tag{2.17}$$

However, it is known that:

$$\left.\frac{d\boldsymbol{\omega}}{dt}\right|_{\mathcal{I}} = \left.\frac{d\boldsymbol{\omega}}{dt}\right|^{\mathcal{L}} + \boldsymbol{\omega}_L \times \boldsymbol{\omega} \tag{2.18}$$

and expressed in the $\mathcal{L}$ frame is:

$$\left.\frac{d\boldsymbol{\omega}}{dt}\right|_{\mathcal{I}}^{\mathcal{L}} = \left.\frac{d\boldsymbol{\omega}}{dt}\right|^{\mathcal{L}} + \boldsymbol{\omega}_L|^{\mathcal{L}} \times \boldsymbol{\omega}|^{\mathcal{L}} \tag{2.19}$$

Equating the eq. (2.19) with eq. (2.17):

$$\left.\frac{d\boldsymbol{\omega}}{dt}\right|^{\mathcal{L}} = D\left.\frac{d\boldsymbol{\omega}_T}{dt}\right|_{\mathcal{I}}^{\mathcal{T}} - \left.\frac{d\boldsymbol{\omega}_L}{dt}\right|_{\mathcal{I}}^{\mathcal{L}} - \boldsymbol{\omega}_L|^{\mathcal{L}} \times \boldsymbol{\omega}^{\mathcal{L}} \tag{2.20}$$

Multiplying by the inertia tensor of the leader:

$$\mathbf{I}_L\left.\frac{d\boldsymbol{\omega}}{dt}\right|^{\mathcal{L}} = \mathbf{I}_L D\left.\frac{d\boldsymbol{\omega}_T}{dt}\right|_{\mathcal{I}}^{\mathcal{T}} - \mathbf{I}_L\left.\frac{d\boldsymbol{\omega}_L}{dt}\right|_{\mathcal{I}}^{\mathcal{L}} - \mathbf{I}_L\boldsymbol{\omega}_L|^{\mathcal{L}} \times \boldsymbol{\omega}^{\mathcal{L}} \tag{2.21}$$

This can be rewritten using the differentiation of the angular velocity in the body frames:

$$\mathbf{I}_L \frac{d\boldsymbol{\omega}}{dt}\bigg|^{\mathcal{L}} = \mathbf{I}_L D \frac{d\boldsymbol{\omega}_T}{dt}\bigg|_{\mathcal{T}}^{\mathcal{T}} - \mathbf{I}_L \frac{d\boldsymbol{\omega}_L}{dt}\bigg|_{\mathcal{L}}^{\mathcal{L}} - \mathbf{I}_L \boldsymbol{\omega}_L|^{\mathcal{L}} \times \boldsymbol{\omega}^{\mathcal{L}} \tag{2.22}$$

At this point, the Euler's equation for a rigid body is:

$$\frac{d\mathbf{L}_T}{dt}\bigg|_{\mathcal{I}} = \frac{d\mathbf{L}_L}{dt}\bigg|^{\mathcal{T}} + \boldsymbol{\omega}_T \times \mathbf{L}_T = \mathbf{N}_T \tag{2.23}$$

$$\frac{d\mathbf{L}_L}{dt}\bigg|_{\mathcal{I}} = \frac{d\mathbf{L}_L}{dt}\bigg|^{\mathcal{L}} + \boldsymbol{\omega}_L \times \mathbf{L}_L = \mathbf{N}_L \tag{2.24}$$

where $\mathbf{L}_T$ and $\mathbf{L}_L$ are the total angular momentum of target and leader respectively and $\mathbf{N}_T$ and $\mathbf{N}_L$ the correspondent external torques on the body. Since $\mathbf{L}_L = \mathbf{I}_L\boldsymbol{\omega}_L$ and $\mathbf{L}_T = \mathbf{I}_C\boldsymbol{\omega}_T$ the following equations are obtained:

$$\mathbf{I}_T \frac{d\boldsymbol{\omega}_T}{dt}\bigg|_{\mathcal{T}} + \boldsymbol{\omega}_T \times \mathbf{I}_T\boldsymbol{\omega}_T = \mathbf{N}_T \tag{2.25}$$

$$\mathbf{I}_L \frac{d\boldsymbol{\omega}_L}{dt}\bigg|_{\mathcal{L}} + \boldsymbol{\omega}_L \times \mathbf{I}_L\boldsymbol{\omega}_L = \mathbf{N}_L \tag{2.26}$$

Finally substituting eqs. (2.25) and (2.26) into eq. (2.22) yields:

$$\mathbf{I}_L \frac{d\boldsymbol{\omega}}{dt}\bigg|_{\mathcal{L}}^{\mathcal{L}} = \mathbf{I}_L D \mathbf{I}_T^{-1}[\mathbf{N}_T - \boldsymbol{\omega}_T|^{\mathcal{T}} \times \mathbf{I}_T\boldsymbol{\omega}_T|^{\mathcal{T}}] - \mathbf{I}_L\boldsymbol{\omega}_L|^{\mathcal{L}} \times \boldsymbol{\omega}_L|^{\mathcal{L}} - [\mathbf{N}_L - \boldsymbol{\omega}_L|^{\mathcal{L}} \times \mathbf{I}_L\boldsymbol{\omega}_L|^{\mathcal{L}}] \tag{2.27}$$

This final equation and eq. (2.15) describe the relative rotation dynamics and kinematics.

2.3 Clohessy-Wiltshire equations

The relative motion between two spacecraft is, in general, described by a set of non linear equations. This model can be linearised under certain assumptions. In fact, if the target's orbit in the inertial space is slightly elliptic and inclined with respect to the leader's orbit, the motion of the target will appear very close to the leader in a leader-fixed frame. In this case, the equations can be linearised and the resulting motion may be solved in closed-form. The obtained linearised equations of motion are called the *Clohessy-Wiltshire equations* or also the *Euler-Hill equations.* To obtain the more general derivation of the CW equation, it is necessary to start from eq. (2.2) and eq. (2.3), adding an external perturbation force F to the leader motion. The resulting equation is eq. (2.5) plus the new force. Under the aforementioned assumptions is possible to linearise this last equation through a binomial series, neglecting higher order terms. Avoiding all the derivation, the resulting equations of motion expressed in the leader reference frame are:

$$\ddot{\boldsymbol{\rho}} = -\frac{\mu}{{r_L}^3}\left[-\frac{3}{2}\mathbf{r}_L\frac{2\mathbf{r}_L\boldsymbol{\rho}}{{r_L}^2} + \boldsymbol{\rho}\right] + F \tag{2.28}$$

Equation (2.28) represents the inertial relative acceleration between target and leader spacecraft expressed in leader reference frame.

Moreover, expressing this equation in the inertial frame, yields:

$$\ddot{x} - 2n\dot{y} - 3n^2x = f_x \tag{2.29}$$

$$\ddot{y} + 2n\dot{x} = f_y \tag{2.30}$$

$$\ddot{z} + n^2z = f_z \tag{2.31}$$

where $n = \sqrt{\frac{\mu}{{a_L}^3}}$ is the angular velocity of the reference orbit, which it has been assumed to be nearly circular.

As already said, it is possible to obtain a closed form solution for the CW's equation. For the third expression, eq. (2.31), the solution is independent from the first two. Taking the Laplace transform:

$$z(s) = \frac{sz_0 + \dot{z}_0}{s^2 + n^2} + \frac{f_z(s)}{s^2 + n^2} \tag{2.32}$$

Hence, its expression in time domain, assuming f_z to be an impulse function of strength f_z, is simply:

$$z(t) = z_0 \cos(nt) + \frac{\dot{z}_0}{n} \sin(nt) + \frac{f_z}{n} \sin(nt) \tag{2.33}$$

On the other hand, x and y equations, being coupled, have to be solved simultaneously:

$$\begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \frac{1}{s^2(s^2+n^2)} \begin{bmatrix} s^2 & 2ns \\ -2ns & s^2-3n^2 \end{bmatrix} \begin{bmatrix} sx_0 + \dot{x}_0 - 2ny_0 + f_x \\ sy_0 + \dot{y}_0 - 2nx_0 + f_y \end{bmatrix} \tag{2.34}$$

Assuming again f_x and f_y to be impulsive forces, solving and applying the inverse Laplace transform, the solution is:

$$x(t) = [\dot{x}_0 + f_x]\frac{\sin(nt)}{n} - \left[3x_0 + \frac{2(\dot{y}_0 + f_y)}{n}\right]\cos(nt) + \frac{2}{n}(\dot{y}_0 + f_y) + 4x_0 \tag{2.35}$$

and

$$y(t) = [4(\dot{y}_0+f_y)+6nx_0]\frac{\sin(nt)}{n}+\frac{2}{n}(\dot{x}_0+f_x)\cos(nt)+y_0-\frac{2}{n}(\dot{x}_0+f_x)-3(\dot{y}_0+f_y+2nx_0)t \tag{2.36}$$

It is important, when using these equations, to remember that *x(t)* and *z(t)* have to be small. However, the imposed assumption of close proximity of the bodies and considering them as point masses, involves an intrinsic contradiction. In fact, if the spacecraft are close one to each other then, they can no longer be treated as point masses because the spacecraft shape and size affect the relative translation between off-c.m. points. This means that using the linear Clohessy–Wiltshire model can lead to considerable errors when applied to modelling of rendezvous and docking [42].

2.4 Coupled Relative Spacecraft Motion Model

In this section, the model presented by Segal and Gurfil is introduced. This kinematically coupled relative spacecraft dynamics model better describes the close proximity motion. In Gurfil-Segal model, the two spacecraft are not treated like two point masses and, as already said, they have their own reference frame. Since the space objects are not treated like point masses, feature points on the target object can be defined. The vision-based detection and tracking of the feature points is addressed in the next chapters. Being ${P_T}^i$ a point on the target object body, $\mathbf{P}_T{}^i = [{P_{xT}}^i, {P_{yT}}^i, {P_{zT}}^i]$ is defined as a vector directed from the origin of the coordinate system $\mathcal{T}$ to the point ${P_T}^i$. $\mathbf{P}_T{}^i$ is expressed in the $\mathcal{L}$ frame. The next step is to define the relative position vector between the c.m. of the leader and the generic point i on the target.

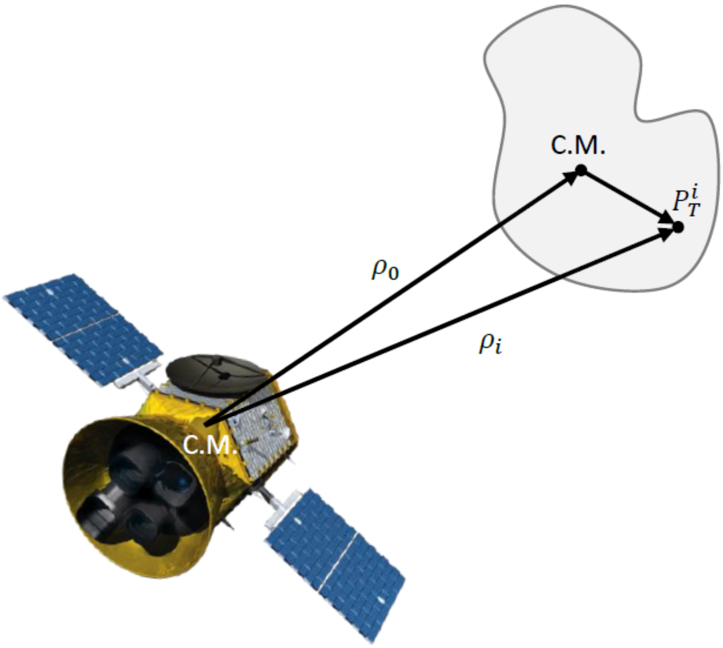


Figure 2.2. Leader - Target graphical representation

By observing fig. 2.2, the following relationship holds:

$$\boldsymbol{\rho}_i = \boldsymbol{\rho}_0 + \mathbf{P}_T{}^i \tag{2.37}$$

where $\boldsymbol{\rho}_0$ is the distance between the centres of mass of the two objects. The first and second derivatives consequently are:

$$\dot{\boldsymbol{\rho}}_i = \dot{\boldsymbol{\rho}}_0 + \boldsymbol{\omega} \times \mathbf{P}_T{}^i \tag{2.38}$$

$$\ddot{\boldsymbol{\rho}}_i = \ddot{\boldsymbol{\rho}}_0 + \dot{\boldsymbol{\omega}} \times \mathbf{P}_T{}^i + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{P}_T{}^i) \tag{2.39}$$

Considering $\boldsymbol{\rho}_0$ and $\boldsymbol{\omega}$ in the $\mathcal{L}$ frame and substituting eqs. (2.37) to (2.39) into eqs. (2.9) to (2.11), the general description of the translational motion between any arbitrary points on the target and the leader c.m. in absence of perturbing forces are obtained:

$$\begin{aligned}\ddot{x}_i &- [\omega_y(\omega_x {P_{yT}}^i - \omega_y {P_{xT}}^i) + \omega_z(\omega_z {P_{xT}}^i - \omega_x {P_{zT}}^i)] - \dot{\omega}_y {P_{zT}}^i + \dot{\omega}_z {P_{yT}}^i \\ &-2\dot{\vartheta}_L[\dot{y}_{ij} - (\omega_z {P_{xT}}^i + \omega_x {P_{zT}}^i)] - \ddot{\vartheta}_L(y_{ij} - {P_{yT}}^i) - \dot{\vartheta}_L^2(x_{ij} - {P_{xT}}^i) \\ &= \frac{-\mu(r_L + x_{ij} - {P_{xT}}^i)}{[(r_L + x_{ij} - {P_{xT}}^i)^2 + (y_{ij} - {P_{yT}}^i)^2 + (z_{ij} - {P_{zT}}^i)^2]^{\frac{3}{2}}} + \frac{\mu}{{r_L}^2}\end{aligned} \tag{2.40}$$

$$\begin{aligned}\ddot{y}_i &- [\omega_z(\omega_y {P_{zT}}^i - \omega_z {P_{yT}}^i) + \omega_x(\omega_x {P_{yT}}^i - \omega_y {P_{xT}}^i)] - \dot{\omega}_z {P_{xT}}^i + \dot{\omega}_x {P_{zT}}^i \\ &+2\dot{\vartheta}_L[\dot{x}_{ij} - (\omega_y {P_{zT}}^i + \omega_z {P_{yT}}^i)] + \ddot{\vartheta}_L(x_{ij} - {P_{xT}}^i) - \dot{\vartheta}_L^2(y_{ij} - {P_{yT}}^i) \\ &= \frac{-\mu(y_{ij} - {P_{yT}}^i)}{[(r_L + x_{ij} - {P_{xT}}^i)^2 + (y_{ij} - {P_{yT}}^i)^2 + (z_{ij} - {P_{zT}}^i)^2]^{\frac{3}{2}}}\end{aligned} \tag{2.41}$$

$$\begin{aligned}\ddot{z}_i &- [\omega_x(\omega_z {P_{xT}}^i - \omega_x {P_{zT}}^i) + \omega_y(\omega_y {P_{zT}}^i - \omega_z {P_{yT}}^i)] - \dot{\omega}_x {P_{yT}}^i + \dot{\omega}_y {P_{xT}}^i \\ &= \frac{-\mu(z_{ij} - {P_{zT}}^i)}{[(r_L + x_{ij} - {P_{xT}}^i)^2 + (y_{ij} - {P_{yT}}^i)^2 + (z_{ij} - {P_{zT}}^i)^2]^{\frac{3}{2}}}\end{aligned} \tag{2.42}$$

As it is possible to see, the relative angular velocity vector $\boldsymbol{\omega}$ makes eqs. (2.40) to (2.42) and the rotational motion eqs. (2.15) and (2.27) coupled. Considering also the equations of the orbital angular velocity and acceleration, the non-linear 6-DOF model of the rigid-body relative spacecraft motion is given by the following set of non linear coupled equations: eqs. (2.12), (2.13), (2.15), (2.27) and (2.40) to (2.42)

These equations describe in a more accurate way the motion between two spacecraft, but they also introduce complexity in the equations. In fact, considering N feature points, eqs. (2.40) to (2.42) must be integrated *2N* times to obtain the relative position of each point.

In order to avoid this complexity, the coupling effects can be simply described by combining the rotational and translational contributions.

The dynamics of a feature point expressed in the leader frame $\mathcal{L}$ is:

$$\dot{\mathbf{P}}_T^i|_{\mathcal{L}} = \dot{\mathbf{P}}_T^i|_{\mathcal{T}} + \boldsymbol{\omega} \times \mathbf{P}_T^i|_{\mathcal{L}} = \boldsymbol{\omega} \times \mathbf{P}_T^i|_{\mathcal{L}} \tag{2.43}$$

Knowing that the c.m. dynamics is described by eqs. (2.10) to (2.12), the position of a single feature point can be described as follows:

$$\boldsymbol{\rho}_i = \boldsymbol{\rho}_0 + \mathbf{P}_T^i|_{\mathcal{L}} \tag{2.44}$$

In this way, for each time step, it is only necessary to integrate the position of the c.m. and add the contribution due to the rotation of the feature points.

2.5 Results

In this section, similar results to [42] are proposed to show how it is possible to describe the same behaviour with simpler equations. In fact, in [42], the described coupled model is used. The presented results are, instead, obtained by using the simplified coupled model. Please note that the simplified model is not coupled but it describes a coupled behaviour. Hence, this model will be called *simplified coupled model*. Different simulations are proposed to evaluate close proximity operations and docking. In particular, the relative position is computed with both CW and coupled model in different conditions.

2.5.1 Simplified Coupled Model

First of all, the relative trajectories of different feature points are obtained, numerically integrating the set of non linear equations. A MATLAB code is implemented and ode45 is used as integration method. In this simulation, the leader is orbiting around the Earth in an orbit with the following parameters:

- semi-major axis: $a_L = 7136$ km
- eccentricity: $e_L = 0.05$;
- inclination: $i = 15°$;
- argument of the perigee: $\omega = 340°$;
- RAAN: $\Omega = 0°$;
- true anomaly: $\vartheta = 0°$.

Furthermore, the initial conditions of the relative motions are:

- relative position: $\boldsymbol{\rho} = [10, 150, 10]$ m ;
- relative velocity: $\dot{\boldsymbol{\rho}} = [0.01, -0.0226, 0.01]$ m/s
- relative quaternions: $\mathbf{q} = [0, 0, 0, 1]$;
- relative angular velocity: $\boldsymbol{\omega}_0 = [0.006, 0.006, 0.12]$ deg/s;

The feature points on target spacecraft can be chosen arbitrarily:

- $\mathbf{P}_0 = [0, 0, 0]$ m;
- $\mathbf{P}_1 = [2, 2, 0]$ m ;
- $\mathbf{P}_2 = [-2, -2, 0]$ m ;

It has been decided to select points far from the c.m. to better show the motion differences arising from the coupled model.

In fig. 2.3 the trajectory in time of the three components of two different feature points and the c.m. are evaluated. These are given by:

$$\rho_{10} = \rho_0 + \mathbf{P}_1|_{\mathcal{L}} \tag{2.45}$$

$$\rho_{20} = \rho_0 + \mathbf{P}_2|_{\mathcal{L}} \tag{2.46}$$

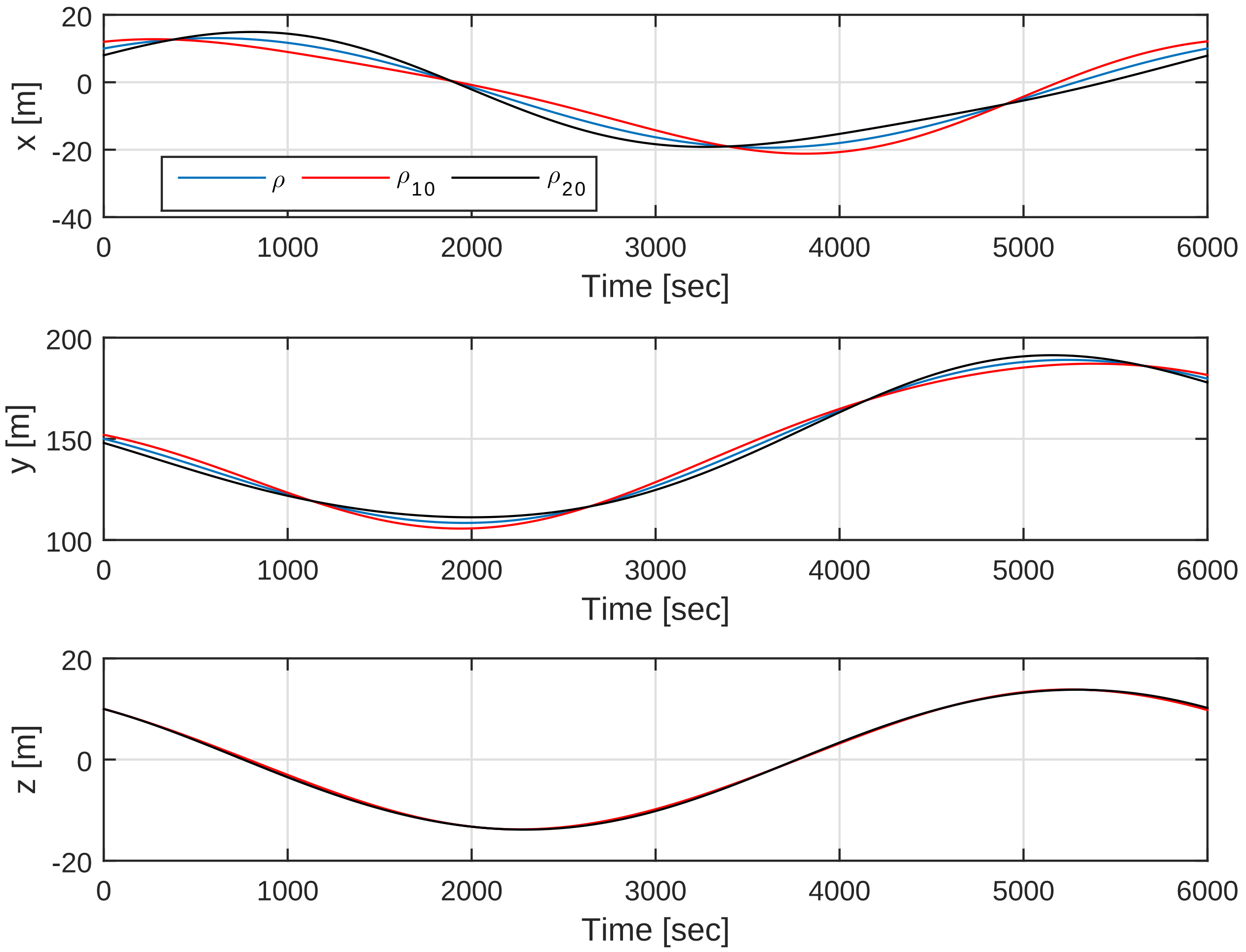


Figure 2.3. Relative Position

This result, clearly shows how there is a difference between the motion of the c.m. and the motion of a general feature point on the object. This difference varies according to the position of the point on the body of the target object. To better understand the motion of these points, in fig. 2.4, the deviation in the relative position of the feature points is shown. $\Delta\rho_1$ $\Delta\rho_2$ are the differences between ρ_{10}, ρ_{20} and the c.m.. From the graphs, the motion of the feature points appears to have a harmonic behaviour, with frequency of oscillation determined by the relative angular velocity.

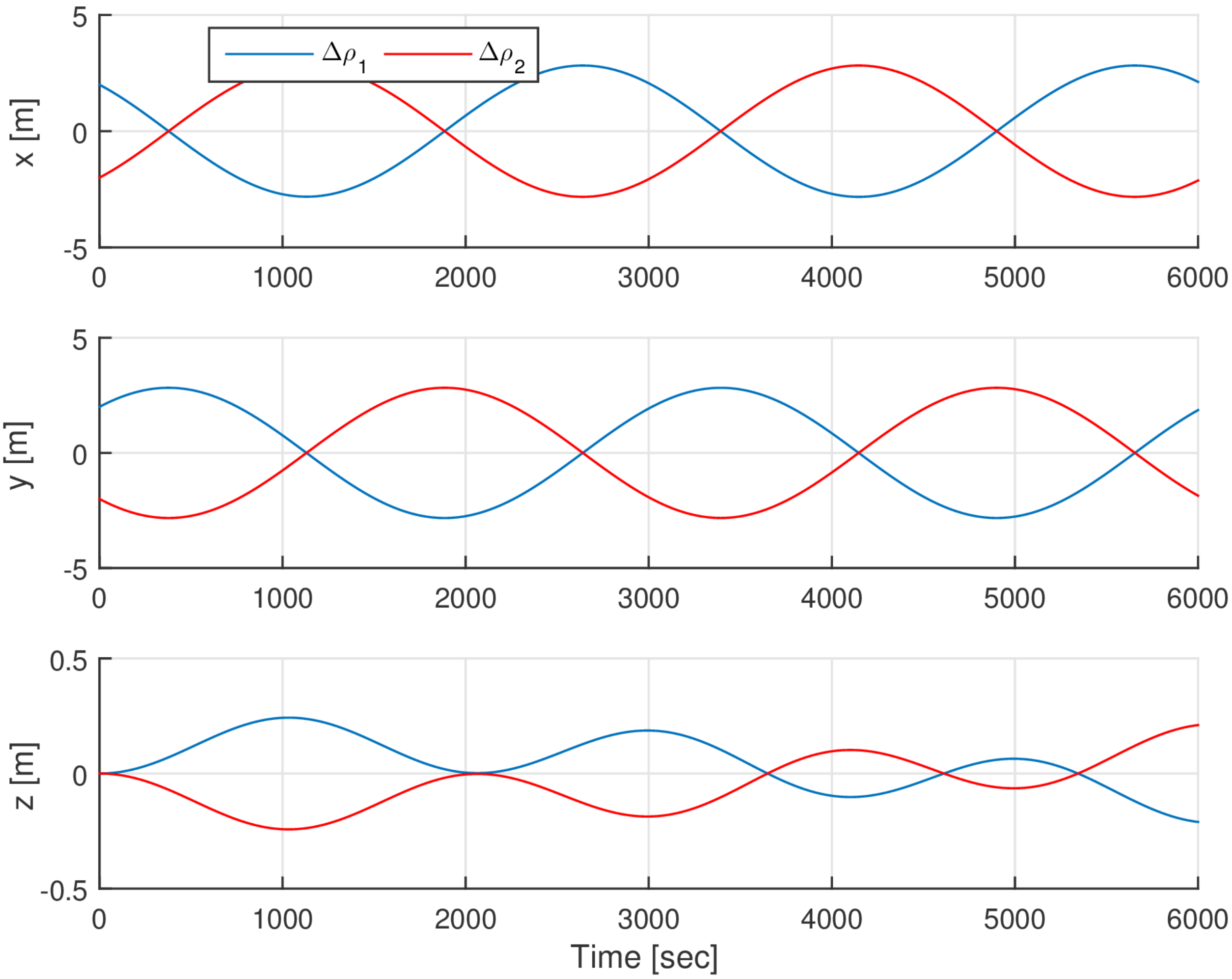


Figure 2.4. Deviations

With this model, a coupled behaviour, in terms of position, can be accurately detailed with simple equations. These results clearly demonstrate how this simplified coupled model describes the relative motion between two spacecraft in a different way with respect to the classical point-mass models.

2.5.2 CW comparison

It has been shown that the simplified coupled model can describe in a different, more accurate, way the relative dynamics between two objects. This is achieved taking into account also the motion of points different from the c.m.. In this section a comparison between CW equations and the simplified coupled model is presented. As first, the errors due to the assumptions of the CW are evaluated, then, it is shown how, not considering different feature points, leads to significant errors in case of docking applications.

As already said, one of the assumptions of the CW model is that the two objects are one close to the other. With this hypothesis, it is possible to linearise the equation of motion. This linearisation could lead to small errors in the description of the relative motion. In fig. 2.5 curves corresponding to CW and coupled model appear to be almost coincident.

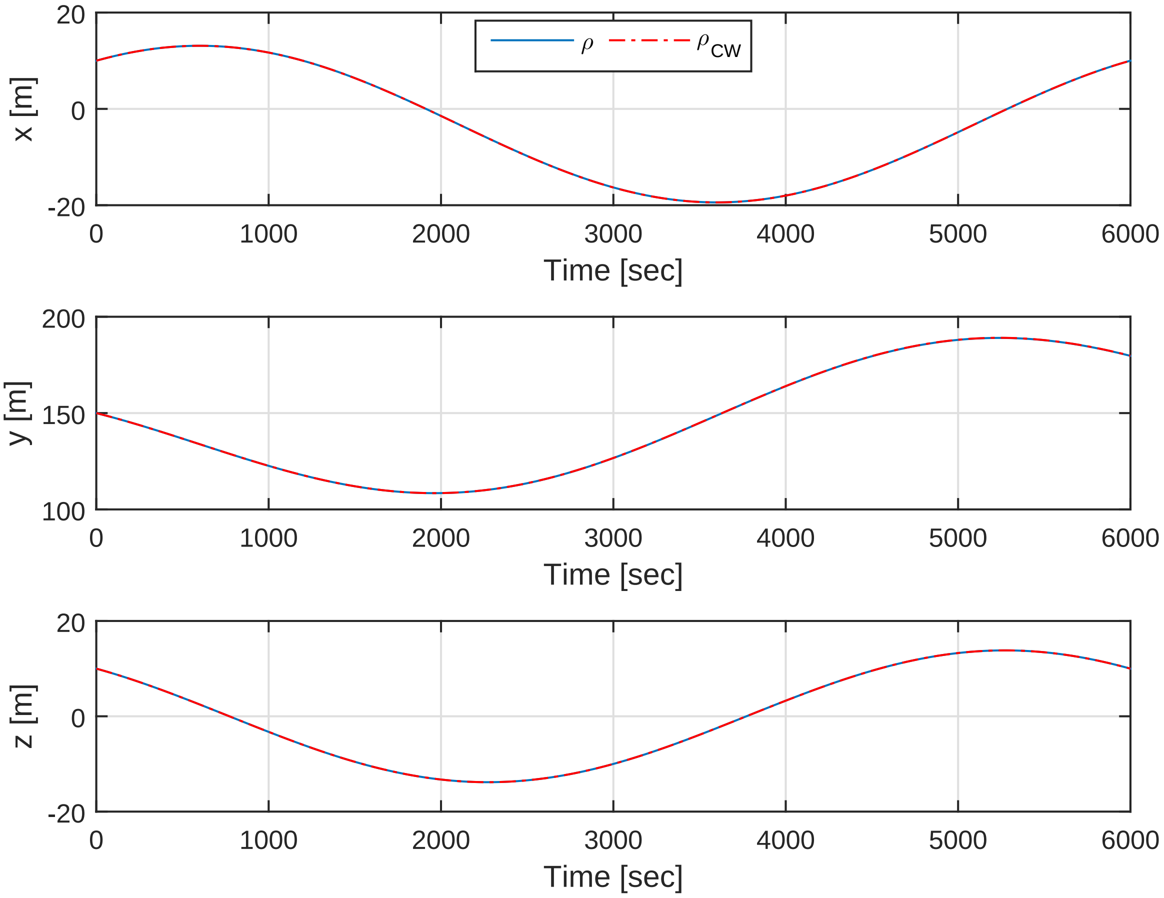


Figure 2.5. CW Relative Position

Zooming the two curves around 1000s, the error appears to be very small, in the order of centimetres, as shown in fig. 2.6. This means that the CW model can describe well the motion of the center of mass, assuming circular orbit and close objects.

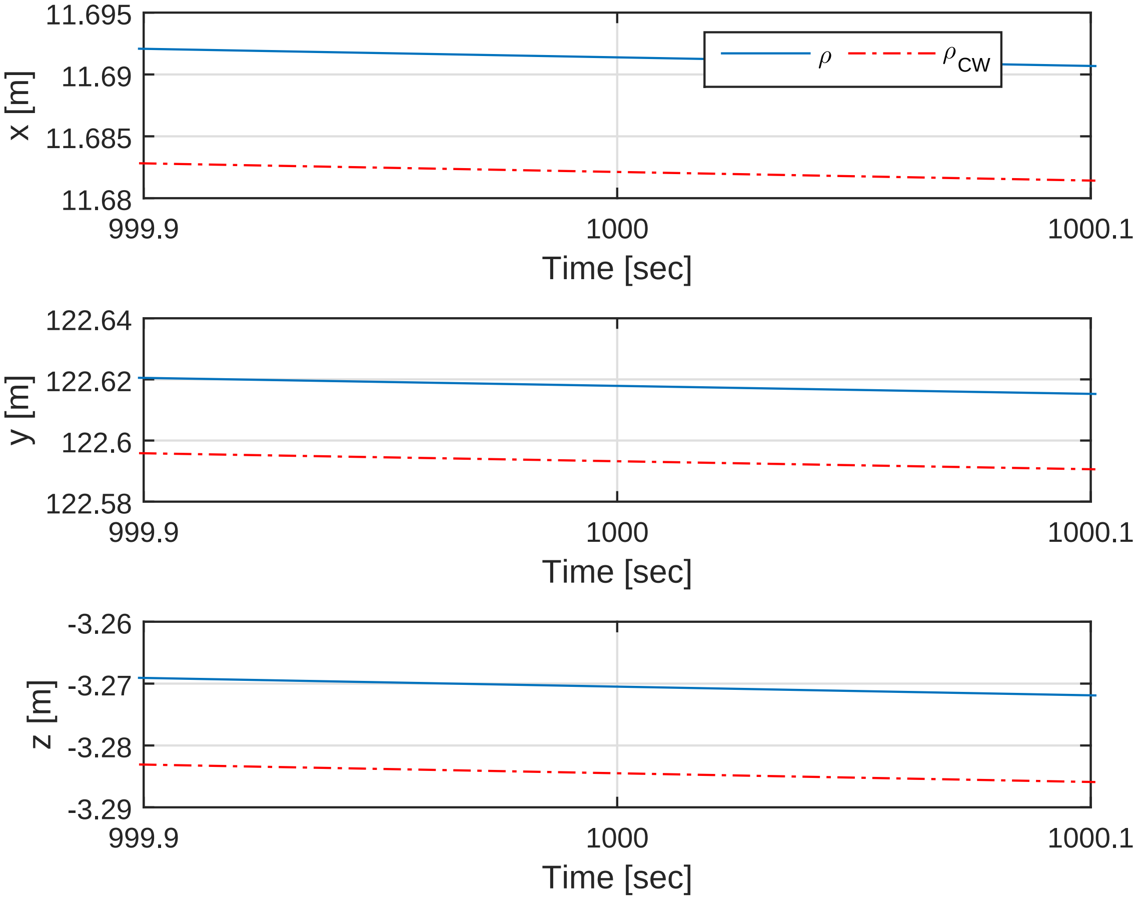


Figure 2.6. CW Zoom

However, it is not convenient to consider the two objects as point masses in docking applications. In order to evaluate the different behaviours, a docking simulation scenario is setted. Initial conditions about relative position and velocity are selected to have null components of the position after almost 80 seconds. In fig. 2.7 it is evident how the distance from the leader c.m. and a general point, different from the target c.m., is decreasing to zero but with a time offset. For this reason, the docking instant will depend on the position of the point.

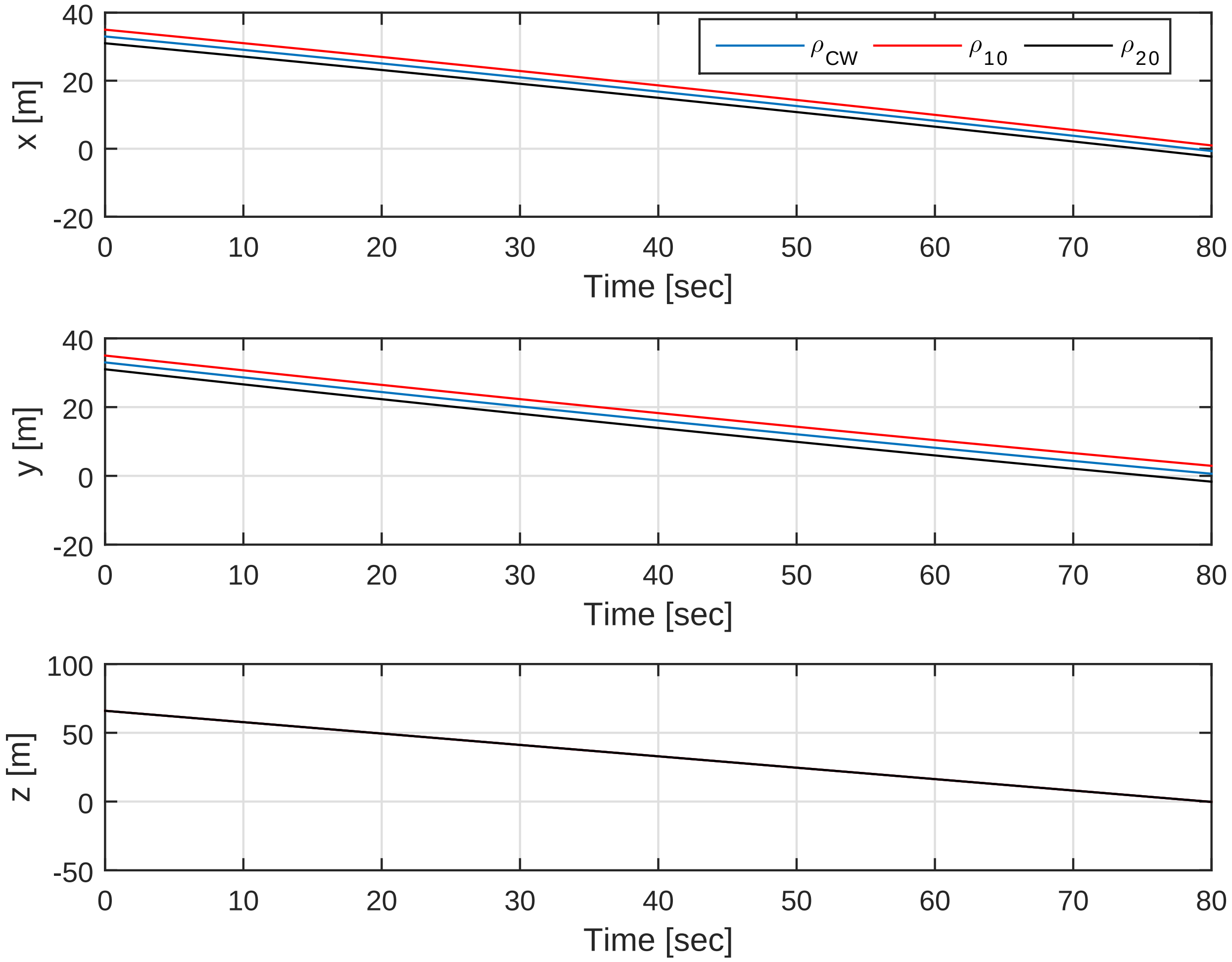


Figure 2.7. CW Relative position for a docking case

Concluding, the simplified coupled model describes the same behaviour of the complex model developed by Segal and Gurfil. Moreover, it better describes the relative motion of two spacecraft operating in close proximity with respect to the classical, only translational models. It can be possible to use the CW equations to compute the position of the c.m. and then add the rotational contribution. However, in presence of elliptical orbits, the CW model leads to significant errors. For this reason, the non-linear model has been chosen for this work.

2.5.3 Computational Performance

Computational performance is an important aspect to consider in online applications. As already said, the simplified coupled model has been chosen for its simplicity and computational efficiency. Proving the computational efficiency of an algorithm is not an easy task. In this section, the computational performance of both coupled and simplified model is analysed. The two methods are implemented in the same MATLAB script and the time for computing each algorithm is then evaluated. To have a reliable result, the time of each method is computed for 50 simulations. Then, the percentage variation is evaluated as:

$$v_{\%} = \frac{t_{coupl} - t_{simpl}}{t_{coupl}} \tag{2.47}$$

where t_{coupl} and t_{simpl} are the computational times for the coupled and the simplified model respectively. In this way, this quantity gives an idea on how fast the simplified model is with respect to the coupled model.

This variation is also evaluated for three different numbers of tracked feature points. In particular, the results are analysed for 1, 3 and 6 tracked feature points.

For one tracked point fig. 2.8, the difference is very small, as expected. The simplified model shows slightly better computational performance.

Increasing the number of points, the benefits deriving from using the simplified model should become significant. In fact, fig. 2.9 and fig. 2.10 show how, increasing the number of feature points, the relative difference increases and therefore, the simplified model results to be much faster than the completely coupled one.

The presented results justify the choice of the simplified coupled model in the filtering process. In fact, it shows comparable performance to the, more complex, coupled model and it outperforms the classical, only translational, dynamical models. In addition, it offers good computational efficiency also increasing the number of feature points. The simulations demonstrate how this method is faster than the completely coupled one. For an online implementation, these considerations lead to prefer the simplified coupled model. In fact, it offers a very accurate description of

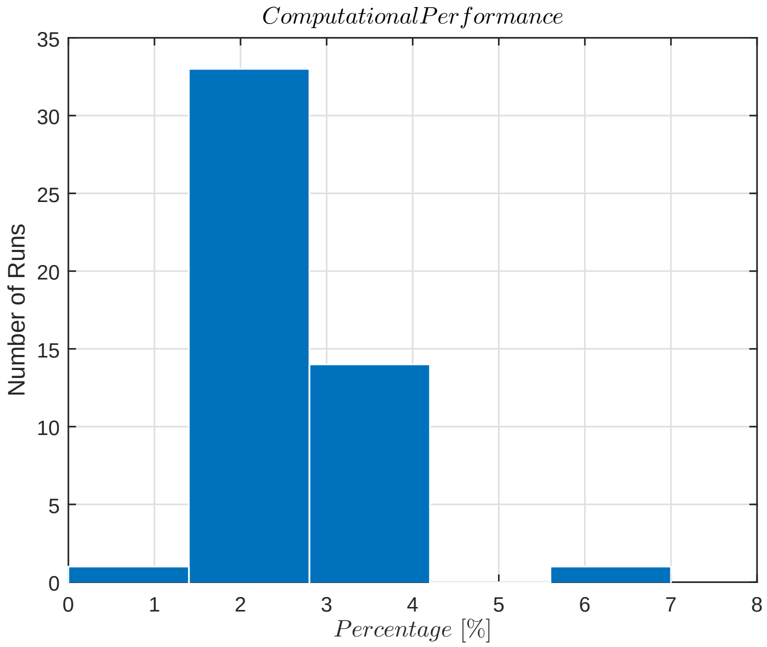


Figure 2.8. Time variation with 1 feature point

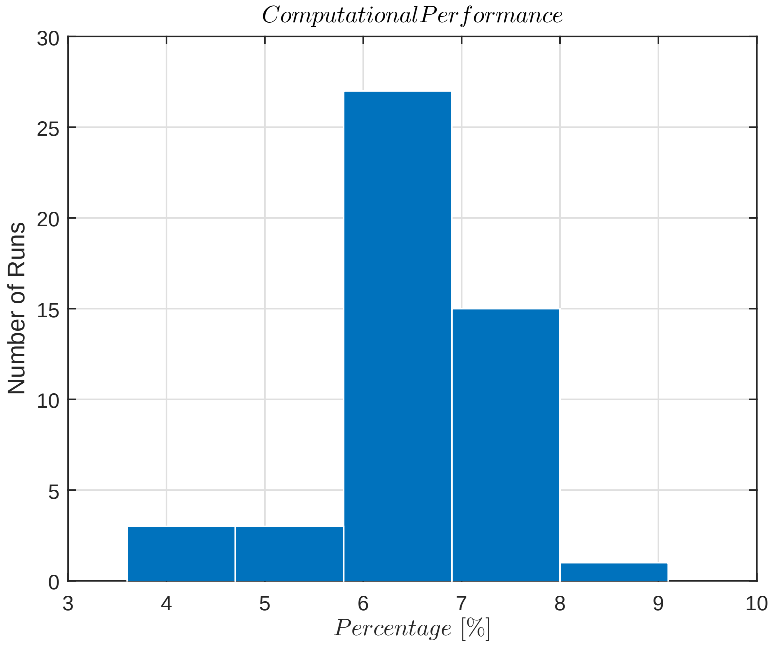


Figure 2.9. Time variation with 3 feature points

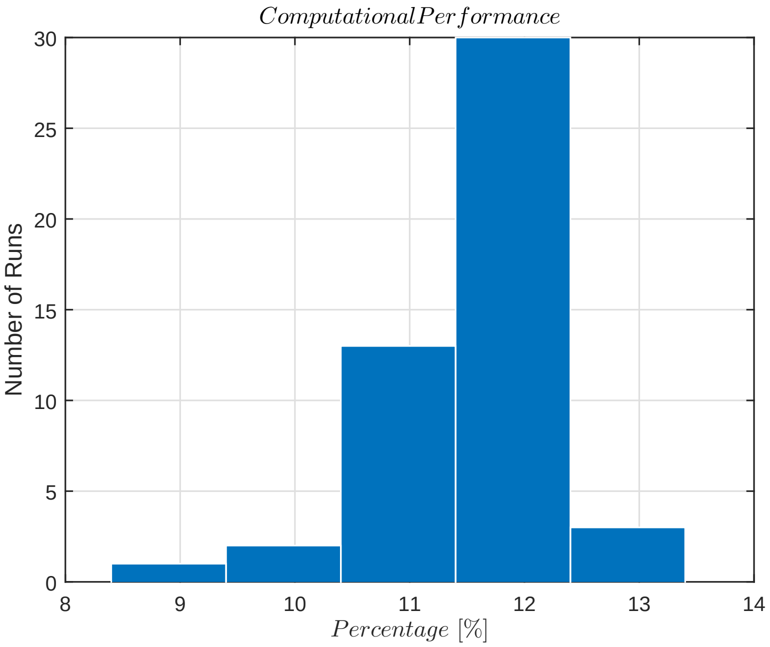


Figure 2.10. Time variation with 6 feature points

the relative motion without any strong assumption and any additional complexity. Moreover, its good computational performance, also with a large number of feature points, guarantees a possible online implementation.

Chapter 3

Observation Model and Computer Vision

The purpose of this chapter is to describe the observation model that allows to obtain information from the collected stereoscopic images. Moreover, this chapter presents the basic computer vision algorithms to recover information from pictures in real applications. Both computer vision algorithm and approximated model used for numerical simulations are analysed. In fact, without any experimental data, it is necessary to build a measurement model to approximate the actual information provided by a real sensor.

3.1 Stereoscopic Camera Model

3.1.1 Pinhole Camera Model

To describe how camera works, it is first necessary to explain how lenses work. They use refraction to produce images of objects. To better understand the image formation, the *Ray Diagrams* are introduced. A Ray Diagram, helps to interpret the path of the light through the lens. Figure 3.1 shows an example of Ray Diagram.

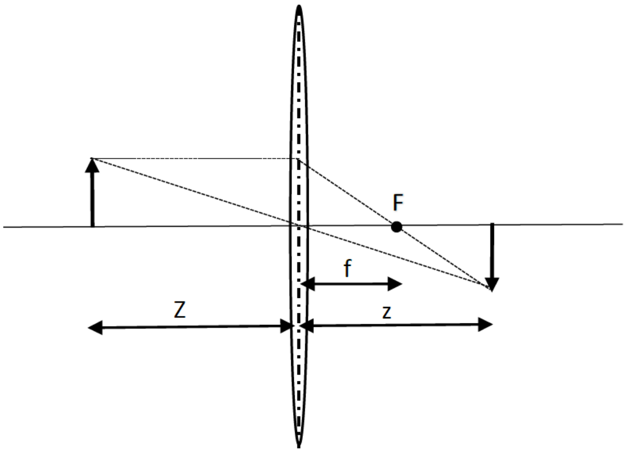


Figure 3.1. RayDiagram

Looking at fig. 3.1 it is possible to understand how an image formed by a convex lens is described by the lens equation:

$$\frac{1}{Z} + \frac{1}{z} = \frac{1}{f} \tag{3.1}$$

where Z is the distance of the object from the lens, z is the distance of the image from the lens and f is the focal length that is the distance of the focus from the lens. If the aperture of the lens decreases to zero, all rays are forced to go through the optical flow centre. This is one of the most important assumptions of the pinhole camera model. This is a model that mathematically describes the relationship between the 3D coordinates of a generic point and the 2D coordinates of its projection onto the image plane of the camera. As already said, it is assumed that all light travels through an infinitely small hole and is projected onto an image frame. Therefore, no lenses are used to focus light and distortion is not considered.

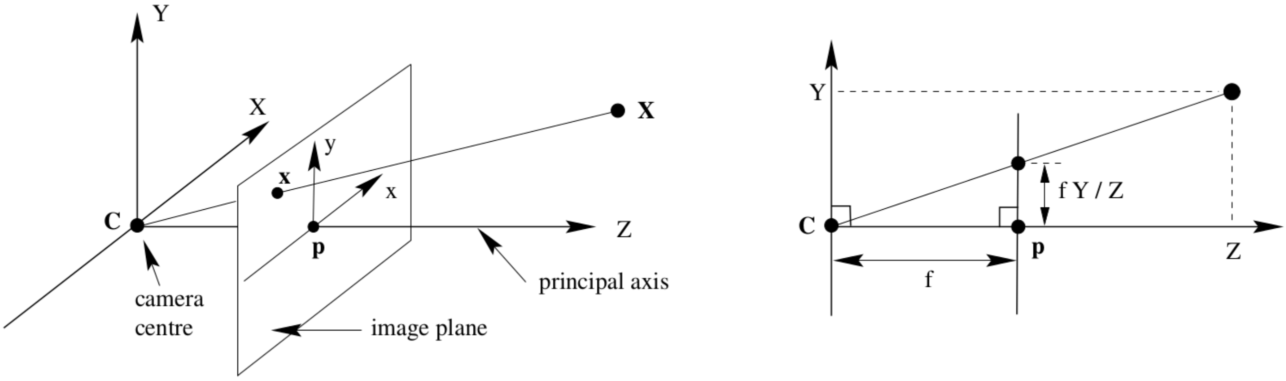


Figure 3.2. Pinhole Camera Model

Looking at fig. 3.2, a derivation of how the coordinates of the point $\mathbf{x}$ = [u,v], in the image plane, depend on the coordinates of $\mathbf{X}$ = [X,Y,Z] is provided. Exploiting a simple rule of similar triangles, the following expressions are obtained:

- $u = f\dfrac{X}{Z}$
- $v = f\dfrac{Y}{Z}$

In this thesis it is assumed to have two cameras forming a stereo rig.

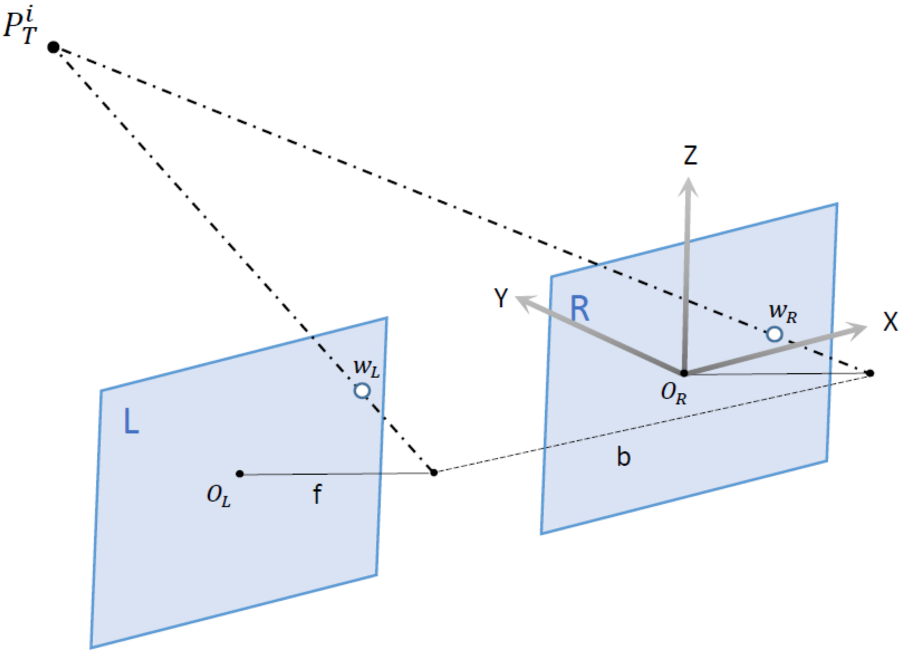


Figure 3.3. Stereo Camera Configuration

In fig. 3.3 the configuration and the related camera frames are shown. The right camera centre-of-projection O_R is assumed to coincide with the centre of mass of the leader spacecraft. The left camera centre-of-projection O_L is shifted by a baseline b along the $\mathbf{x}$ axis. The point O_R is also the origin of the Cartesian right-hand camera frame [X, Y, Z]. As already said, using the perspective projection model, a points in a 3D frame is described in the 2D image plane. With this method all the selected and tracked feature points are expressed in the 2D camera plane. For the line-of-sight ρ_i between a generic feature point and the leader centre of mass (or the camera frame origin O_R) holds $\rho_i = \rho_0 + D(q)P_i$. Assuming to have a focal length equal to 1, the following expressions are derived:

For the right camera

$$u_R(i) = \frac{x_i}{y_i} \tag{3.2}$$

$$v_R(i) = \frac{z_i}{y_i} \tag{3.3}$$

and for the left camera

$$u_L(i) = \frac{x_i - b}{y_i} \tag{3.4}$$

$$v_L(i) = \frac{z_i}{y_i} \tag{3.5}$$

where $\rho_i = [x_i, y_i, z_i]$ in the camera frame. We can also define $\mathbf{w}_R = [u_R\, v_R]$ and $\mathbf{w}_L = [u_L\, v_L]$.

3.1.2 Optical Flow and Image Velocity

Further information can be recovered from the acquired images, exploiting the optical flow. In this work, a general image sequence of 3D scenes is collected. It is assumed that both camera and observed object are moving. Each tracked feature point moves along a 3D path that can be projected onto the image plane as shown before. This 2D path $p(\vec{t}) = (u(t), v(t))$ has an instantaneous direction given by the velocity $\dot{w}_i = d\vec{p}/dt$. Collecting the velocities of all the visible points, the 2D motion field is reconstructed. The optical flow approximates the motion field from the information of the time-varying image intensity. Without entering into the details of the computation of the optical flow, knowing the approximated images velocity, it is possible to relate this information to the translational and rotational motion. In [49], [50], [51], [52] different formulations of the relation between the 3D motion and the optical flow are derived. In particular, in [49], a simple equation to describe this relation is formulated. Reviewing the physics of rigid-body motion under perspective projection, an equation relating the 3D motion to image velocity is derived. First, the velocity of any feature points can be written as

$$\dot{\boldsymbol{\rho}}_i = \dot{\boldsymbol{\rho}}_0 + \boldsymbol{\omega} \times \boldsymbol{\rho}_i \tag{3.6}$$

In the 2D plane, it is necessary to differentiate the eqs. (3.2) to (3.5) to obtain an expression of the image velocity in the 2D image plane for both right and left camera $\dot{w}_{Ri}$, $\dot{w}_{Li}$. Then, substituting these in eq. (3.6), we obtain an equation connecting the images velocity to the relative translational and rotational velocity between the camera and the observed object. This relationship is expressed by the following equations:

$$\dot{\mathbf{w}}_{R\,i} = \begin{bmatrix} \frac{1}{y_i}A(w_{R\,i}) & B(w_{R\,i}) \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\rho}}_{\mathbf{0}} \\ \boldsymbol{\omega} \end{bmatrix} \tag{3.7}$$

and

$$\dot{\mathbf{w}}_{L\,i} = \begin{bmatrix} \frac{1}{y_i}A(w_{L\,i}) & B(w_{L\,i}) \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\rho}}_{\mathbf{0}} \\ \boldsymbol{\omega} \end{bmatrix} \tag{3.8}$$

with

$$A = \begin{bmatrix} 1 & 0 & w_1 \\ 0 & 1 & w_2 \end{bmatrix} \tag{3.9}$$

and

$$B = \begin{bmatrix} -w_1 w_2 & 1 + {w_1}^2 & -w_2 \\ -1 - {w_2}^2 & w_1 w_2 & w_1 \end{bmatrix} \tag{3.10}$$

where w_1 and w_2 are the first and the second component of the vector $\mathbf{w}$. This is an approximation and, for real applications, the optical flow must be recovered.

3.1.3 Disparity

The first main challenge in a stereoscopic system is to find the correspondence between points in different images. In other words, it is necessary to determine the different location of the same point in the left and right image plane respectively. The resulting difference is called disparity and it is defined as:

$$d_i = u_L - u_R \tag{3.11}$$

With disparity it is possible to reconstruct information about the depth. The human brain does something similar, interpreting the difference in retinal position. In stereo vision applications, this can be performed exploiting the so called *triangulation*. Equation (3.11) shows how disparity and depth are inversely related. Considering the disparity for each pixel, a *disparity map* is obtained. This map provides information about the 3D coordinates of each point in the space from 2D images. In fig. 3.4 the disparity is computed using MATLAB. The left and right images, on top, are overlapped in the bottom figure. The yellow arrows correspond to the differences of the same detected feature points in the two images.

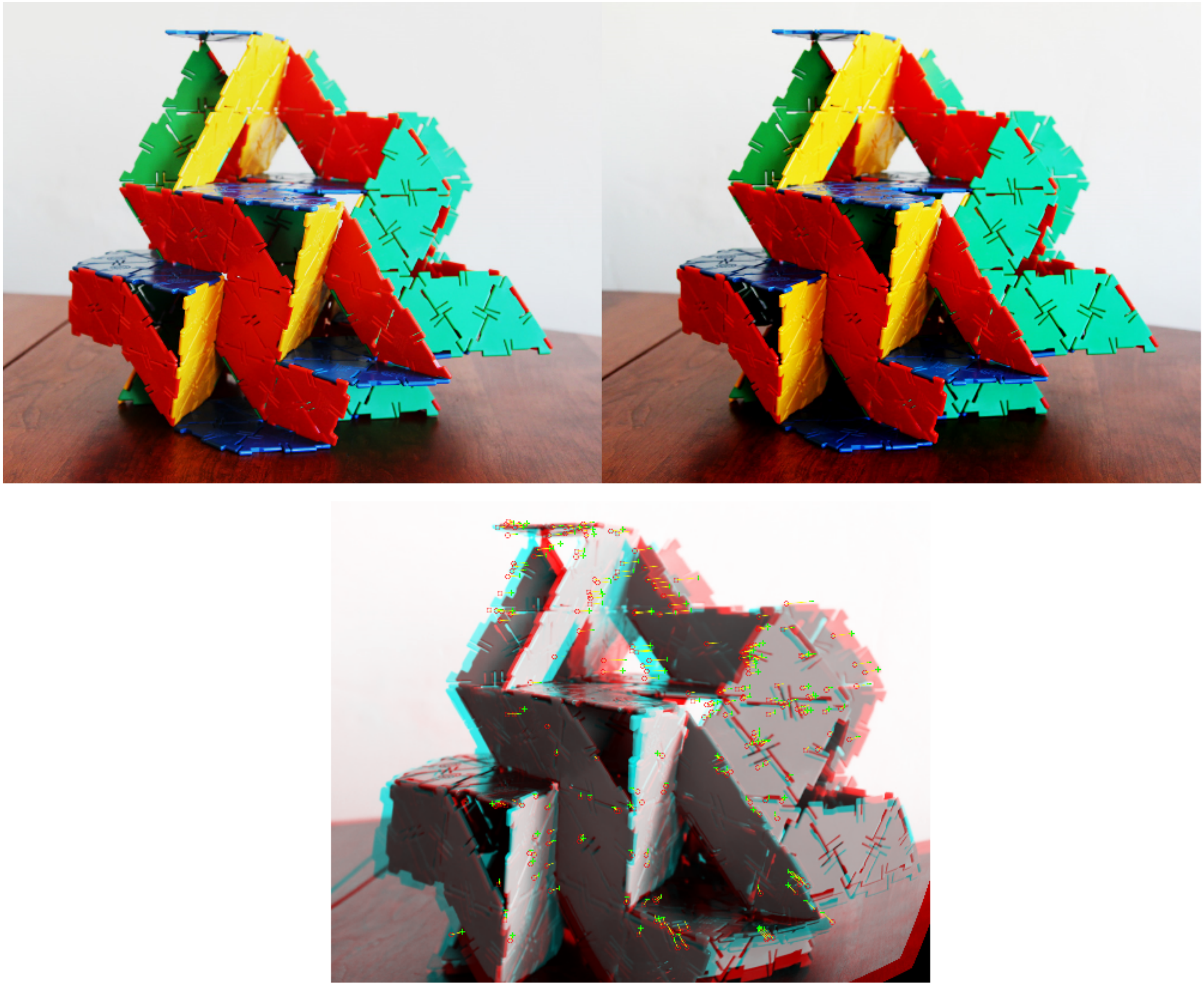


Figure 3.4. Disparity Example

3.2 Computer Vision Algorithm

In real applications, a way to extract information from images must be found. It is necessary to analyse images and produce numerical information from the real world. Computer vision completes this task. It started out in the 1960s and 1970s and nowadays is widely used in many fields. In this section, few simple algorithms are presented.

3.2.1 Feature Extraction and Matching

The first and very important aspect to take into account in computer vision is to detect feature points and match them in images taken from different locations. In our case, points have to be extracted from images taken by two cameras separated by a baseline. Once the points are detected, it is necessary to match them between the couple of images. First of all, there are many ways to detect feature points. In this work, the MATLAB Computer Vision System Toolbox is used. It has different embedded feature extraction functions. The most used is probably the Speeded Up Robust Features (SURF) method proposed in 2006 by Bay [53]. The main characteristic of this method is that is very fast with respect to previous detection methods. Using author's words: *'It approximates or even outperforms previously proposed schemes with respect to repeatability, distinctiveness, and robustness, yet can be computed and compared much faster. This is achieved by relying on integral images for image convolutions; by building on the strengths of the leading existing detectors and descriptors (in casu, using a Hessian matrix-based measure for the detector, and a distribution-based descriptor); and by simplifying these methods to the essential.'*

Another method, proposed by Rosten [54], is the Features from Accelerated Segment Test (FAST) algorithm. It uses a different approach, detecting corners. The main advantage is that this method has good computational efficiency and outperforms other similar corner detector methods such as the one presented by Harris-Stephens [55]. Finally, the Maximally Stable Extremal Regions (MSER) method is considered. It was firstly presented by Matas at al [56]. This is a technique for blob detection in images. It extracts a number of different regions that are *'stable connected component of some gray-level sets of the image'*. In other words, it collects regions that have same characteristics, fixing different thresholds. With the MATLAB Computer Vision Toolbox, all these three methods can be used to detect feature points. The sample image is a picture of the asteroid 243 IDA recorded by the *Galileo* flyby. In fig. 3.5, fig. 3.6 and fig. 3.7 results using the three different methods are presented.

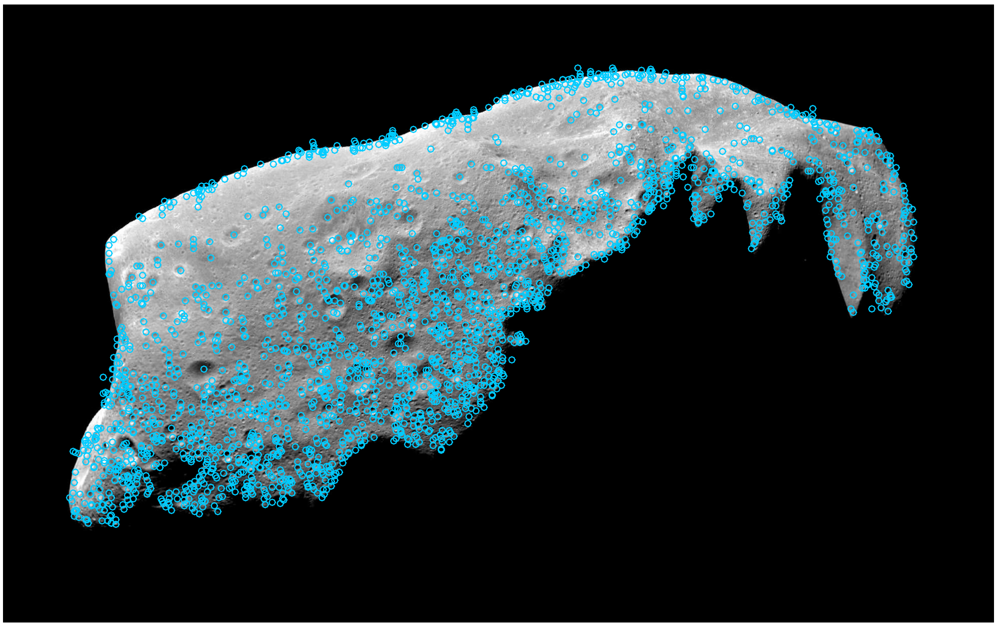


Figure 3.5. Surf Detected Points

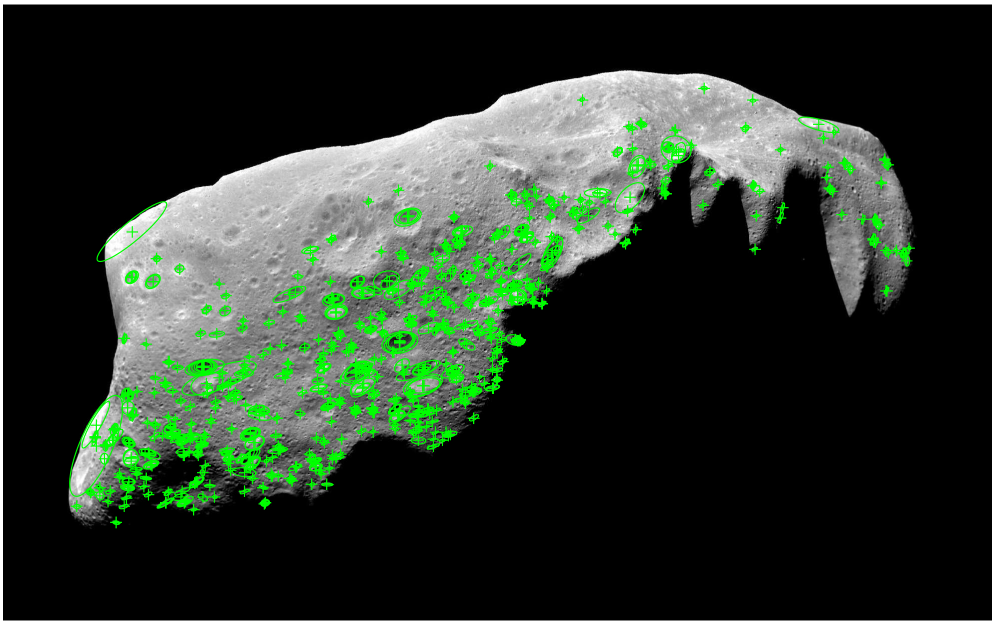


Figure 3.6. MSER Detected Regions

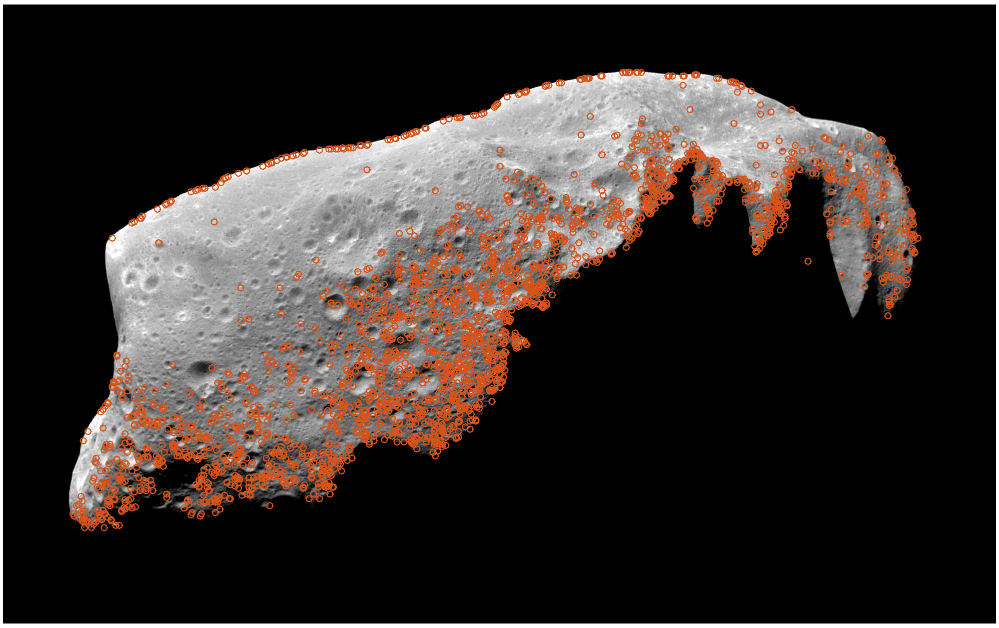


Figure 3.7. FAST Detected Corners

As it is possible to see, the SURF method has better results in terms of number of feature points and these also better cover the object surface. Two aspects are taken into account to evaluate the best *detector*. First, the number of extracted features is considered, then, the computational time is taken into account. In fig. 3.8 and fig. 3.9 the results are shown. In this case, the SURF method has better results in terms of number of detected feature points. FAST has comparable quantity of detected points. Instead, MSER shows poor performance. MSER results to be the worst detection method also for what concern computational time. FAST has the better computational performance. Concluding, SURF and FAST offer the best performance in terms of features extraction and computational cost.

3.2.2 Feature Tracking

Once detected, feature points must be tracked from one frame to the following one. This is not an easy process and implies many challenges. For example, it is not simple to autonomously understand which point can be tracked also because some of them could change their appearance due to rotation or different light conditions. For similar reason, points may appear and disappear in time. To efficiently track the features, the Kanade-Lucas-Tomasi (KLT) [57], [58] method can be used. This approach exploits the spatial intensity gradient of two different images to find a good match, relying on Newton-Raphson iterations. This method allows a faster computation reducing the number of possible matches. In this section, a brief

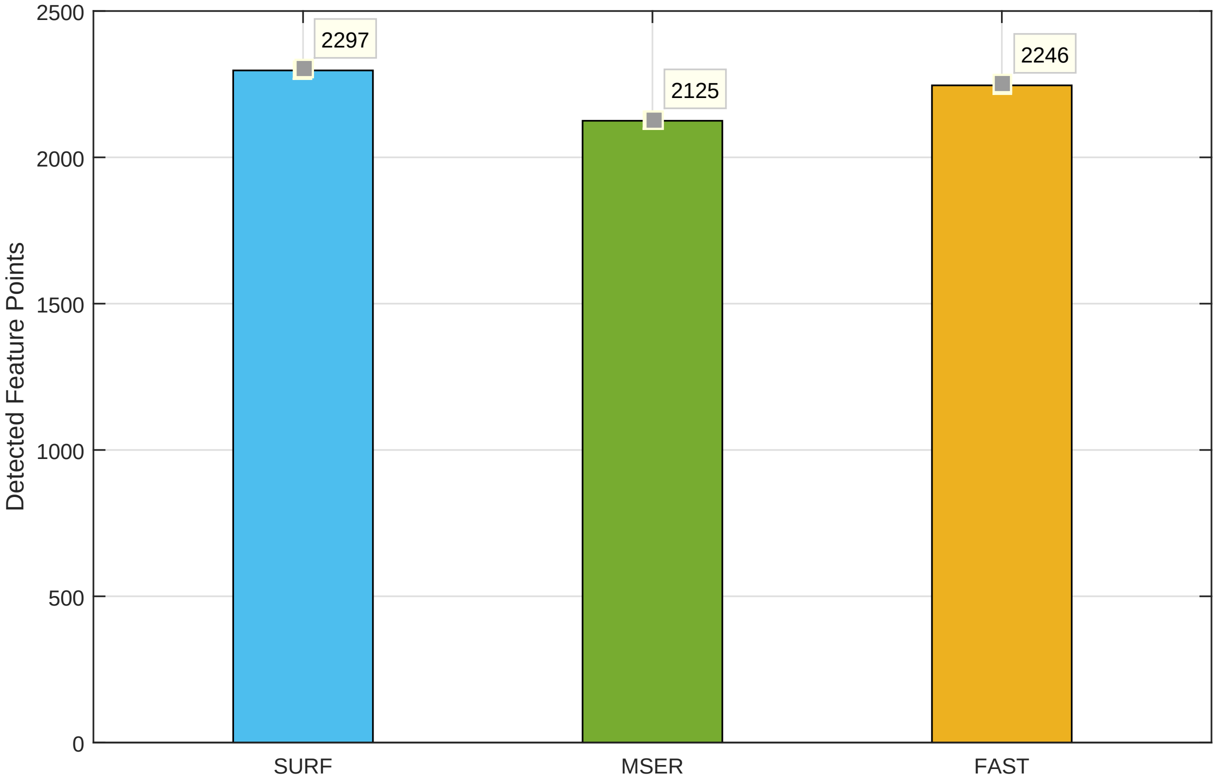


Figure 3.8. Number of Detected Feature Points

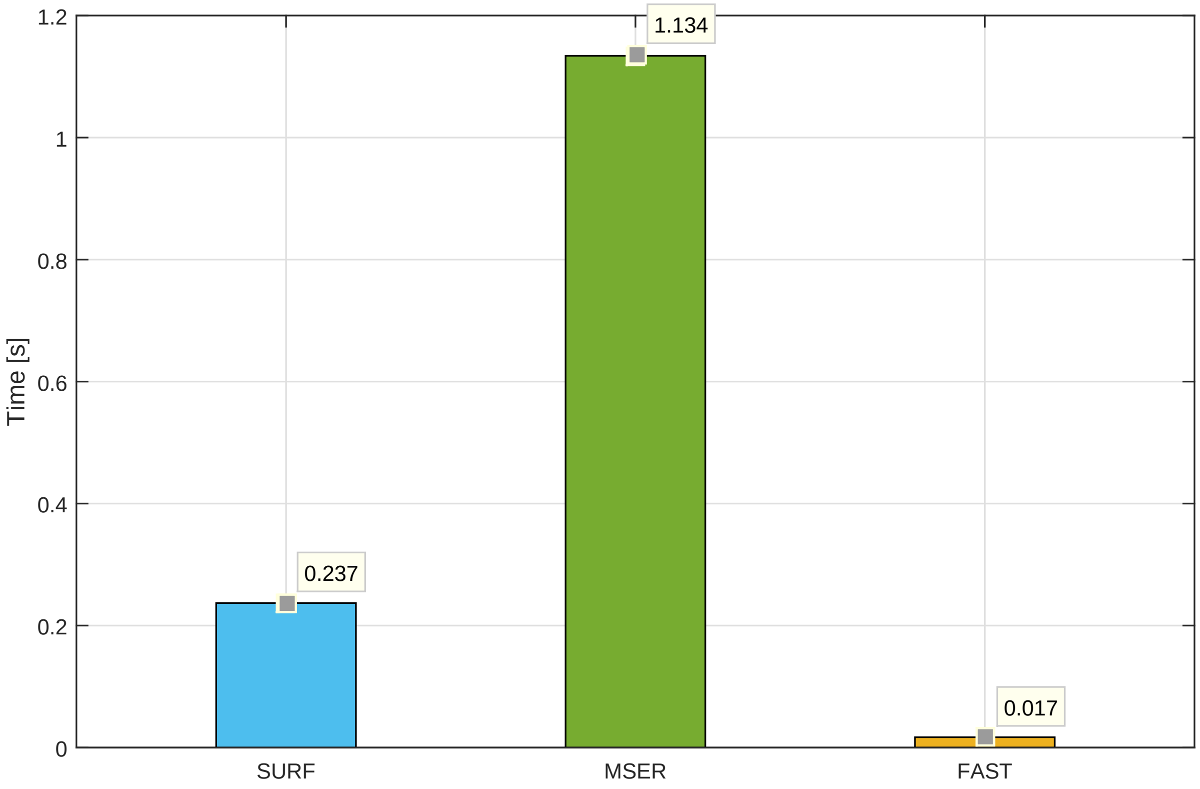


Figure 3.9. Computational Time

explanation on how this algorithm works is provided. One of the main assumptions of KLT is that the brightness is assumed constant. In this way, the points do not change their appearance in time. Moreover, the motion between two following frames is assumed to be small and spatial incoherence is not contemplated. This means that each feature moves in the same way of the others. Going into the details of the formulation, two functions *f(x)* and *g(x)* are defined. They give, as output, the pixel value corresponding to the point vector x in two different images. The goal of the method is to find, in a fast way, the vector h that minimizes the difference between *f(x + h)* and *g(x)*. In a general, multi dimensional case, the Euclidean norm of the error has to be minimized:

$$E = \sum_R [f(x+h) - g(x)]^2 \tag{3.12}$$

where R is the region of interest. At this point, f can be linearly approximated by:

$$f(x+h) \simeq h\frac{\partial f(x)}{\partial x} \tag{3.13}$$

The error minimization is then written as:

$$\frac{\partial E}{\partial h} = 0 \tag{3.14}$$

so,

$$\frac{\partial}{\partial h}\sum_x [f(x) + h\frac{\partial f}{\partial x} - g(x)]^2 = 0 \tag{3.15}$$

that can be also expressed like:

$$\sum_x 2\frac{\partial f}{\partial x}[f(x) + h\frac{\partial f}{\partial x} - g(x)] = 0 \tag{3.16}$$

from eq. (3.16) the expression for h can be recovered:

$$h \simeq \left[\sum_x \frac{\partial f}{\partial x}^T g(x) - f(x)\right]\left[\sum_x \left(\frac{\partial f}{\partial x}\right)^T \left(\frac{\partial f}{\partial x}\right)^{-1}\right] \tag{3.17}$$

The above formulation takes into account only a translational transformation between two different frames. To consider also complex transformations, such as rotation, scaling or shearing, a new g can be defined.

$$g(x) = f(xA + h) \tag{3.18}$$

The matrix A expresses the linear spatial transformation between the two functions *f(x)* and *g(x)*. In this case, the error to minimise is different:

$$E = \sum_R [f(xA + h) - g(x)]^2 \tag{3.19}$$

Again, a linearisation can be performed considering ΔA and Δh as small increments:

$$f(x(A + \Delta A) + (h + \Delta h) \simeq f(xA + h) + (x\Delta A + \Delta h)\frac{\partial f(x)}{\partial x} \tag{3.20}$$

Knowing the expression for the linearised *f*, the solution can be find, as before, imposing the derivative of the error E equal zero. This leads to a set of linear equations.

This method has been improved in the following years. In [58] a method to select only the 'traceable' points is developed. In [59] an algorithm to discriminate between good and bad features was implemented. In this work, the details of these formulations are not presented. This chapter wants only to present the available computer vision algorithms for the feature points extraction and matching, since no experimental validation is provided. Different methods are analysed. SURF and FAST show the best performance in terms of number of extracted features and computational time. They are both suitable and already used in real applications. The goal of this work is not to develop a more efficient computer vision algorithm. In fact, relying on the available methods is sufficient to have a fast and robust image processing. In addition, these methods have only to extract information about the position of the feature points in time. This is an input for the filtering process that has to provide the relative state estimation, exploiting the available measurements. As already said, in our work, this step is approximated by using the previously described observation model, since no real measurement are available.

Chapter 4

Inertia Tensor Observability Analysis

In this chapter, a non linear observability analysis is performed to test the observability of the inertia tensor in torque free motion. With a very easy example, it is possible to demonstrate how the inertia tensor is observable only up to a scale factor. In fact, redefining the inertia tensor $\mathbf{I}$ as $r\mathbf{I}$, the rotational dynamics of the body is not influenced. Recalling the Euler equation for a general rotating body

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} \tag{4.1}$$

replacing $\mathbf{I}$ with $r\mathbf{I}$, the following expression is derived

$$\dot{\boldsymbol{\omega}} = \frac{1}{r}\mathbf{I}^{-1}\boldsymbol{\omega} \times r\mathbf{I}\boldsymbol{\omega} \tag{4.2}$$

then,

$$\dot{\boldsymbol{\omega}} = \frac{1}{r}r\mathbf{I}^{-1}\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} \tag{4.3}$$

and finally

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} \tag{4.4}$$

eq. (4.4) shows how the inertia matrix is observable up to a generic scale factor. This is valid only in torque free motion assumptions.

With the observability analysis, the same result will be derived in a mathematical way.

4.1 Non-linear Observability Analysis

The observability is defined as the ability to infer the state of a system, given a set of measurements (external output). The first to introduce the concept of 'observability' was Rudolf Emil Kálmán in the early 1960s [60], [61], [62]. However, he defined the observability only for linear systems. Ten years later, different authors developed theories about the observability of nonlinear systems [63], [64], [65]. However, the most used and known method to study the observability of nonlinear systems was presented by Hermann and Krener in 1977 [66]. In this work, this method is used to analyse the observability of the inertia tensor, as already done by Tweddle [28]. Dealing with linear systems, the observability is determined from the rank of the observability matrix or observability Gramian matrix, depending if the system is time-invariant or time-varying. Thus, for linear systems, the observability is a global property. On the other hand, for a nonlinear system, the observability must be determined locally about a certain state. Hermann and Krener introduced the concept of 'local weak observability'. Local weak observability means that the states must be distinguishable only from their neighbours. In other words, a generic nonlinear system is *locally observable at x_0* if $I_{\mathcal{U}}(x_0) = x_0$ being $\mathcal{U}$ every open neighbourhood of x_0 (note that $I_{\mathcal{U}}(x)$ means that all points are distinguishable from x as long as both trajectories lie entirely in $\mathcal{U}$). The same system is *locally weakly observable at x_0* if exists an open neighbourhood $\mathcal{U}$ of x_0 such that $I_{\mathcal{V}}(x_0) = x_0$ for every open neighbourhood $\mathcal{V}$ of x_0 with $\mathcal{V} \subset \mathcal{U}$ Finally, a system is *locally weakly observable* if it is locally weakly observable at every x_0. Summarizing it in a scheme:

$$\begin{tikzcd}
\text{locally observable} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{observable} \arrow[d, Rightarrow] \\
\text{locally weakly observable} \arrow[r, Rightarrow] & \text{weakly observable}
\end{tikzcd}$$

The main advantage of this approach with respect to the others is that it lends to an algebraic test. In order to describe in a proper way this observability analysis method, it is necessary to introduce other mathematical concepts.

4.1.1 Lie Derivatives

Lie derivatives are often used in fluid dynamics. This kind of derivative, evaluates the change of a tensor field along the flow of another tensor field. Considering a generic dynamical system Σ:

$$\dot{\mathbf{x}} = \mathbf{f}_0(x) + \sum_{i=1}^{p} \mathbf{f}_i(x)\mathbf{u}_i \tag{4.5}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) \tag{4.6}$$

being $\mathbf{x}$ the state vector, $\mathbf{y}$ the vector of the observed outputs of the system and $\mathbf{u}$ the vector of control inputs. The Lie derivative $L_f h(x)$ of a function $h(x)$ with respect to $f(x)$ is:

$$L_f h(x) = \sum_{k=1}^{n} \mathbf{f}_k(x) \frac{\partial h(x)}{\partial x_k} \tag{4.7}$$

and recursively for the higher-order terms derivatives ${L_f}^j = L_f({L_f}^{j-1} h(x))$.

4.1.2 Inertia Observability Algebraic Test

As already said, Hermann and Krener introduced an algebraic test to evaluate the observability of a nonlinear system. Considering the previously defined system Σ, the vector subspace Ω can be defined. This is the smallest vector subspace containing h_j with j=1,...,n, closed with respect to Lie differentiation by $\mathbf{f}_0$ and $\mathbf{f}_i$. At this point, the observability matrix $\boldsymbol{\mathcal{O}}$ can be defined. $\boldsymbol{\mathcal{O}}$ is composed by the gradients of the Lie derivatives of Ω:

$$\boldsymbol{\mathcal{O}} = \nabla\Omega \tag{4.8}$$

Evaluating locally the rank of this matrix, an observability analysis is performed. If $\boldsymbol{\mathcal{O}}$ has full column rank at $\mathbf{x}_0$, then Σ satisfies the observability rank condition at $\mathbf{x}_0$. This means that the system is locally weakly observable at $\mathbf{x}_0$. If $\boldsymbol{\mathcal{O}}$ has full column rank for all $\mathbf{x}$, then Σ is weakly observable and satisfies the observability rank condition generally.

At this point, the Hermann and Krener test is applied to our problem. Let's consider the Euler's Equation of Rotational Motion for our target under the assumption of torque free motion (we omit the label T for simplicity):

$$I_x \dot{\omega}_x = (I_y - I_z)\omega_y \omega_z \tag{4.9}$$

$$I_y \dot{\omega}_y = (I_z - I_x)\omega_x \omega_z \tag{4.10}$$

$$I_z \dot{\omega}_z = (I_x - I_y)\omega_x \omega_y \tag{4.11}$$

Defining the state vector as:

$$\mathbf{x} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ I_x \\ I_y \\ I_z \end{bmatrix} \tag{4.12}$$

the dynamics of the system can be defined. The dynamics of the inertia components is null because, under the assumption of rigid body, they are constant in time.

$$\dot{\mathbf{x}} = \begin{bmatrix} \frac{I_y - I_z}{I_x}\omega_y\omega_z \\ \frac{I_z - I_x}{I_y}\omega_x\omega_z \\ \frac{I_x - I_y}{I_z}\omega_y\omega_x \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.13}$$

The measurement model is given by:

$$y = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{4.14}$$

This is because the observability of the inertia components has to be evaluate knowing the evolution in time of the angular velocity $\boldsymbol{\omega}$. At this point, the observability matrix $\boldsymbol{\mathcal{O}}$ can be computed. $\boldsymbol{\mathcal{O}}$ has maximum rank when it is defined as

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} \nabla {L_f}^0 h \\ \nabla {L_f}^1 h \end{bmatrix} \tag{4.15}$$

With a simple algorithm, it can be computed that the rank of the matrix $\boldsymbol{\mathcal{O}}$ is five. However the state vector has six components. This leads to the conclusion that one of the six degrees of freedom is unobservable. Another important quantity to evaluate is the null space of the matrix $\boldsymbol{\mathcal{O}}$. This subspace is called unobservable subspace. The states lying in this set will generate identically null input response. In fact, the null space of a generic matrix A is defined as $\mathcal{N}$ such as:

$$A\mathbf{y} = 0 \tag{4.16}$$

where $\mathbf{y} \in \mathcal{N}$.
In our case:

$$null(\boldsymbol{\mathcal{O}}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{I_x}{I_z} \\ -\frac{I_y}{I_z} \\ 1 \end{bmatrix} \tag{4.17}$$

Analysing this matrix, the last three terms are non null. This confirms that the inertia matrix is observable only up to a scale factor. This result implies that, in torque free motion condition, the relative dynamics can be propagated correctly also without knowing the single components of the inertia matrix.

Chapter 5

Estimation Methodology

In this chapter, the selected estimation methods are analysed and compared. In particular, the Extended Kalman Filter and the Iterated Extended Kalman Filter are described. The differences between these two methods are underlined and the motivation, justifying the final choice, is provided.

5.1 Extended Kalman Filter

The Kalman Filter was firstly developed by Rudolf E. Kalman in 1960's [67]. However, a similar algorithm was developed by Thorvald Nicolai Thiele [68] and Peter Swerling [69] many years before. Kalman proposed a recursive solution to the discrete-data linear filtering problem. This set of mathematical equations provides a solution to this kind of problem in a very computationally efficient way. This is one of the reasons why it was used several times in real applications, such as the Apollo navigation computer. It is erroneously thought that Kalman Filter only deals with Gaussian uncertainties. The first Kalman's paper, instead, referred to a generic case and only later was demonstrated that in the Gaussian case, the Kalman Filter provides an optimal solution. Another important characteristic of this filter is that it supports estimations of past, present and future states. Moreover, it can also be used in applications in which the dynamical model of the system is not well-known. In the linear case, it is modelled on a Markov chain built on linear operators perturbed by Gaussian noise. At each time step, the new state can be predicted using a linear operator with noise and eventually a control input. Then, another linear operator, generates the measurement output from the unknown state. Noise can be added also to this output. Therefore, the Kalman Filter could be, in a certain sense, analogous to an hidden Markov model. The only differences is that Kalman Filter is applicable for continuous state variables. Knowing the system and measurement dynamics, the statistical information about the noises, measurements errors and uncertainties in the dynamical model and the initial conditions of the state, the filter is able to estimate the current value of the variables of interest. Another aspect of the Kalman Filter is that it is a recursive process. This means that, especially in practical implementations, it has not to store all the previous data but it always processes new measurements. The Kalman Filter can, in this way, statistically minimize the error of the state combining the actual available measurements and the prior knowledge of the system. However, this approach involves some basic assumptions. As already said, the classical Kalman Filter is used only with linear system. Moreover, the noise is assumed to be white. This means that the noise value is not correlated in time and that its power is equal for each frequency. This would imply that the noise has infinite power and this is not applicable to any real case. The reason why this assumption is still used is that any physical system having a certain ’bandpass’ frequency can be driven by the wideband noise. This is a noise having constant power within the system bandpass. This means that the real wideband noise can be approximated with a white noise within a certain frequency interval. The third assumption is that the noise has to be a Gaussian distribution. This is justified by the fact that measurement noises are usually generated by many different small sources. This is exactly the case when multiple independent variables are added together. In these cases, the summed effect can be described as a Gaussian probability density function. Of course, from the practical point of view, both ’whiteness’ and ’Gaussianness’ assumptions imply using simpler mathematics and therefore faster algorithm. Nonetheless, many real-world applications use non-linear models for describing the system or the measurements. Linear models are only a purposeful way to approximate and describe more complex systems. However, developing an estimator on the basis of a wrong model is dangerous. In fact, depending on the tuning, the filter can be affected much more by the output of the internal model than by the input given by the real-world measurements. There are different techniques that allow to deal with a non-linear system. Some compensation techniques, such as pseudonoise or artificial bounds for the error covariance matrix elements, help to describe in a proper way a non-linear system using an approximate linear model. There are also other filtering techniques that use a better description of the true system describing it as a non-linear model. These approaches are the Linearised and the Extended Kalman Filter. These concepts are introduced by Maybeck [70] in the early 1980’s. In this work, the analysed and used method is the Extended Kalman Filter (EKF). The EKF is a way to overcome the problems arising from the approximation of the system dynamics with a linear model. The general idea is to use a non-linear description of the system model and linearise this model about the state estimation for each time step. In this way, as soon as a new estimate is predicted, a linearised, more accurate state trajectory is available in the estimation process. Remembering that the linearisation process involves the assumption of small deviations from the reference trajectory, this can be ensured by incorporating this updating of the reference trajectory in the estimation process. Let's assume that our process is governed by a non-linear stochastic differential equation. This equation can be expressed in non-linear state-space as:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{w} \tag{5.1}$$

where $\mathbf{x}$ is the state vector, $f(\mathbf{x})$ is a non-linear function describing the model, depending on the state and $\mathbf{w}$ is a random zero-mean white noise. The related power spectral density matrix is given by:

$$\mathbf{Q} = E[\mathbf{w}\mathbf{w}^T] \tag{5.2}$$

In the same way, a measurement equation can be defined as a non-linear function as:

$$\mathbf{z} = h(\mathbf{x}) + \mathbf{v} \tag{5.3}$$

and

$$\mathbf{R} = E[\mathbf{v}\mathbf{v}^T] \tag{5.4}$$

with $\mathbf{v}$ being the measurement noise. The non-linear function $h(\mathbf{x})$ relates the state $\mathbf{x}$ to the measurement $\mathbf{z}$. For systems in which the measurements are discrete, the measurement equation can be written as:

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \tag{5.5}$$

As already said, the system and the measurement equations are non-linear. This implies that a first-order approximation has to be used in the continuous Riccati equations for the systems dynamics matrix $\boldsymbol{F}$ and the measurement matrix $\boldsymbol{H}$ respectively. These two matrices are the Jacobian of partial derivatives of f and h with respect to $\mathbf{x}$:

$$\boldsymbol{F} = \left.\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right|_{x=\hat{x}} \tag{5.6}$$

$$\boldsymbol{H} = \left.\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\right|_{x=\hat{x}} \tag{5.7}$$

The fundamental matrix for the discrete Riccati equations is approximated by the Taylor-series expansion for $e^{\boldsymbol{F}\Delta t}$ and can be expressed as:

$$\boldsymbol{\Phi}_k = \boldsymbol{I} + \boldsymbol{F}\Delta t + \frac{\boldsymbol{F}^2\Delta t^2}{2!} + \frac{\boldsymbol{F}^3\Delta t^3}{3!} + ... \tag{5.8}$$

where Δt is the sampling time. In general, the Taylor-series expansion is approximated with only the first two terms:

$$\boldsymbol{\Phi}_k = \boldsymbol{I} + \boldsymbol{F}\Delta t \tag{5.9}$$

It has been demonstrated that adding more terms to the series does not significantly improve the performance of the filter[71]. In the following steps, the EKF estimation procedure is summarized.

- Assume to have the knowledge of the last state estimate $\hat{\mathbf{x}}(k|k)$
- Linearize the system dynamics around $\hat{\mathbf{x}}(k|k)$
- Apply the EKF prediction step to the linearized system dynamics to obtain $\hat{\mathbf{x}}(k+1|k)$ and $\boldsymbol{P}(k+1|k)$
- Linearize the observation dynamics around $\hat{\mathbf{x}}(k+1|k)$
- Update the state applying the filtering cycle to the measurement model to obtain $\hat{\mathbf{x}}(k+1|k+1)$ and $\boldsymbol{P}(k+1|k+1)$

The mathematical formulation is therefore derived. First of all, it is necessary to update the state $\mathbf{x}$ and the covariance matrix $\mathbf{P}$ from the previous step k to the current time step *k+1*. This step is called *prediction.*

$$\hat{\mathbf{x}}(k+1|k) = \hat{\mathbf{x}}(k|k) + \int_k^{k+1} f(\mathbf{x}(t))dt \tag{5.10}$$

$$\mathbf{P}(k+1|k) = \boldsymbol{\Phi}_k\mathbf{P}(k|k){\boldsymbol{\Phi}_k}^T + \mathbf{Q} \tag{5.11}$$

where $\boldsymbol{\Phi}_k$ is defined in eq. (5.9) and $\boldsymbol{Q}$ in eq. (5.2). Once obtained the predicted state, the updated state can be recovered. This step is the *updating* or *filtering* cycle.

The filter gain $\boldsymbol{K}$ can be obtained and then the state and covariance matrix updated:

$$\mathbf{K}(k+1) = \mathbf{P}(k+1|k)\mathbf{H}^T(k+1)[\mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}^T(k+1)+\mathbf{R}(k+1)]^{-1} \quad (5.12)$$

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1)[y_{k+1} - h_{k+1}(\hat{\mathbf{x}}(k+1|k))] \quad (5.13)$$

$$\mathbf{P}(k+1|k+1) = [\mathbf{I} - \mathbf{K}(k+1)\mathbf{H}(k+1)]\mathbf{P}(k+1|k) \quad (5.14)$$

With this step, the state and covariance matrix estimates are compared and corrected with the measurement y_{k+1}. Figure 5.1 offers a complete picture of the EKF mathematical formulation.

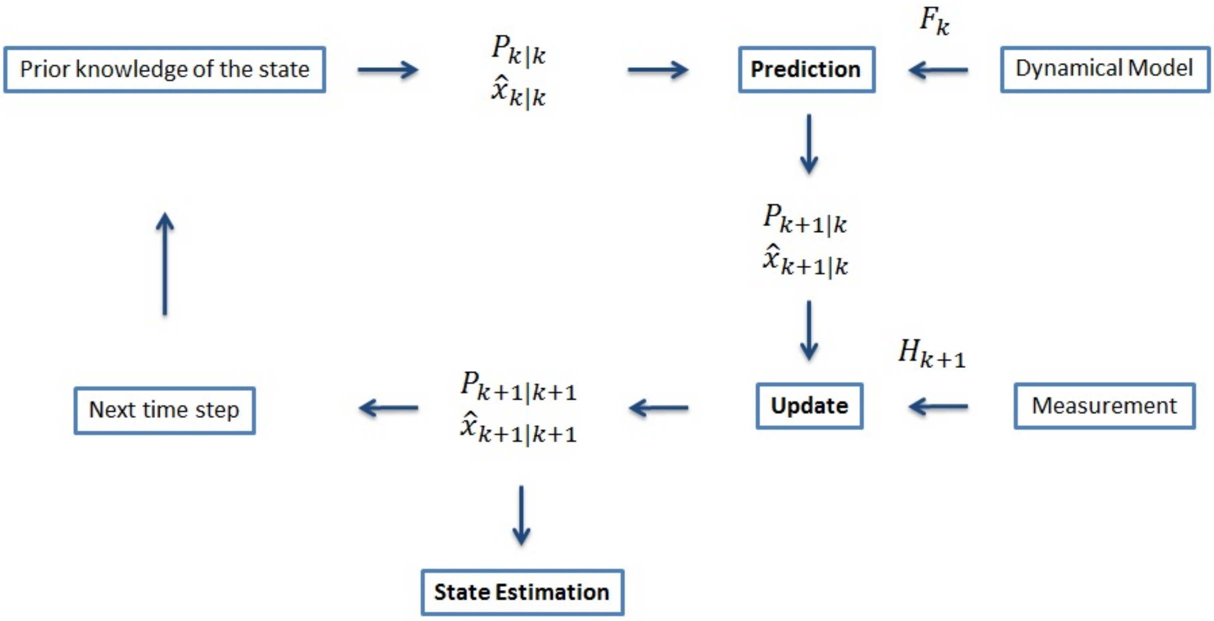


Figure 5.1. EKF Scheme

The EKF algorithm is presented:

It is important to underline that the estimation of the EKF is not optimal as the

Algorithm 1 EKF

```
Initialization: $\hat{\mathbf{x}}_0 = \mathbf{x}_0, \mathbf{P} = \mathbf{P}_0$
for k=0,...,n do
    State prediction: $\hat{\mathbf{x}}(k+1|k) = \hat{\mathbf{x}}(k|k) + \int_k^{k+1} f(\mathbf{x}(t))dt$
    Jacobian matrix of the process: $\boldsymbol{F} = \left.\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right|_{x=\hat{x}_{k+1}}$
    Transition matrix: $\boldsymbol{\Phi}_k = \boldsymbol{I} + \boldsymbol{F}\Delta t$
    Covariance prediction: $\mathbf{P}(k+1|k) = \boldsymbol{\Phi}_k \mathbf{P}(k|k) {\boldsymbol{\Phi}_k}^T + \mathbf{Q}$
    Measurement prediction: $h(\hat{\mathbf{x}}(k+1|k))$
    Innovation evaluation: $\mathbf{z}(k+1) = \mathbf{y}(k+1) - h(\hat{\mathbf{x}}(k+1|k))$
    Jacobian matrix of the observation model: $\boldsymbol{H} = \left.\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\right|_{x=\hat{x}_{k+1}}$
    Innovation covariance: $\mathbf{S}(k) = \mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}^T(k+1) + \mathbf{R}(k+1)$
    Kalman Filter gain: $\mathbf{K}(k+1) = \mathbf{P}(k+1|k)\mathbf{H}^T(k+1)[\mathbf{S}(k)]^{-1}$
    Updated state: $\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1)[\mathbf{z}(k+1)]$
    Updated covariance: $\mathbf{P}(k+1|k+1) = [\mathbf{I} - \mathbf{K}(k+1)\mathbf{H}(k+1)]\mathbf{P}(k+1|k)$
```

one provided by the linear Kalman Filter. Its implementation, in fact, is based on a set of approximations. Another difference is that the filter gain and the covariance matrix cannot be computed off-line. In fact, **F**, **H**, **K** and **P** are evaluated knowing the last estimate of the reference state. In this way, the equations for propagating and updating the estimation covariance matrix (5.11, 5.14) are coupled to the state estimate expression. As already said, this makes impossible to pre-compute the matrices **P** and **K** without the knowledge of the actual state estimates and measurement values. Looking at eq. (5.12), another observation can be done. In the expression of the Kalman filter gain, the Jacobian **H** is present. This term allows to propagate or magnify only some relevant components of the information contained in the collected measurement. In fact, in a simple case in which the measurements do not have a direct correspondence with each component of the state, the Jacobian **H** will magnify only the measurement components affecting the state. For what concern the evaluation of the performance, the analysis of the a priori covariance performance is not sufficient. In general, it is necessary to rely on a Monte Carlo analysis. This approach is conducted to investigate the behaviour and the performance of the filter.

5.2 Iterated Extended Kalman Filter

The presented Extended Kalman Filter, is a widely used, powerful estimation method. As already said, it is used in applications with a behaviour described by non-linear models. However, there are some real cases in which also the measurement equation can be highly non-linear. In the Maybech work [70], there is also a reference to an *Iterated Extended Kalman Filter* (IEKF). With this method, for each time step, local iterations and re-linearisation are performed on the updated reference state. In this way, at a fixed time k, the new state update is computed with the eq. (5.13) and this value is used as a better state estimate for evaluating h and computing the Jacobian $\mathbf{H}$. Knowing the following value of h and $\mathbf{H}$, the new state update can be computed. This iterative process can be stopped when the difference between two consecutive values of the state estimate is smaller than a fixed tolerance or after a fixed number of iterations. Also in this case, a simple scheme and the algorithm are presented.

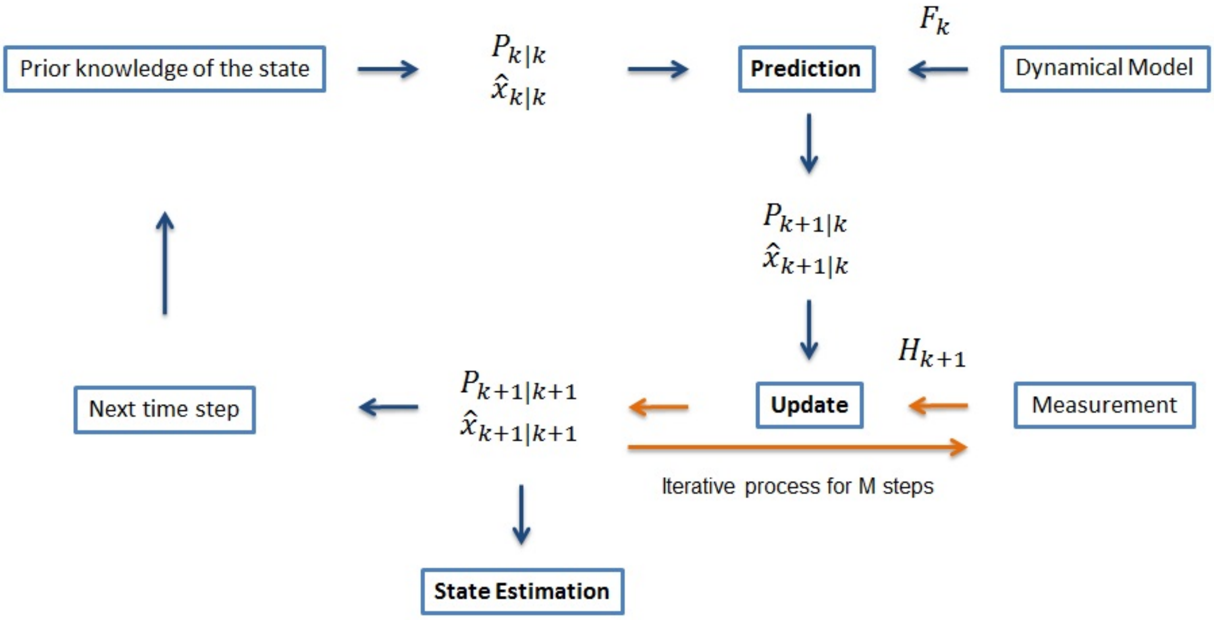


Figure 5.2. IEKF Scheme

The described algorithm outperforms the simple EKF re-computing, iteratively, h and $\boldsymbol{H}$ to obtain a better state estimate $\hat{\mathbf{x}}(k+1|k+1)$. This, of course, leads to have better estimation also in the future steps. Despite the iterative cycle, the filter maintains its simplicity and computational efficiency. In this thesis, both IEKF and EKF are used and compared. The classical formulations are used because they result to be robust and reliable. However, as described in the next chapter, new tools are introduced to estimate the inertia tensor using a proper parametrization.

Algorithm 2 IEKF

```
Initialization: $\hat{\mathbf{x}}_0 = \mathbf{x}_0,\ \mathbf{P} = \mathbf{P}_0$
for k=0,...,n do
    State prediction: $\hat{\mathbf{x}}(k+1|k) = \hat{\mathbf{x}}(k|k) + \int_k^{k+1} f(\mathbf{x}(t))dt$
    Jacobian matrix of the process: $\boldsymbol{F} = \left.\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right|_{x=\hat{x}_{k+1}}$
    Transition matrix: $\boldsymbol{\Phi}_k = \boldsymbol{I} + \boldsymbol{F}\Delta t$
    Covariance prediction: $\mathbf{P}(k+1|k) = \boldsymbol{\Phi}_k \mathbf{P}(k|k) {\boldsymbol{\Phi}_k}^T + \mathbf{Q}$
    for j = 1,...,m do
        Measurement prediction: $h^j(\hat{\mathbf{x}}(k+1|k))$
        Innovation evaluation: $\mathbf{z}^j(k+1) = \mathbf{y}(k+1) - h^j(\hat{\mathbf{x}}(k+1|k))$
        Jacobian matrix of the observation model: $\boldsymbol{H}^j = \left.\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\right|_{x=\hat{x}_{k+1}}$
        Innovation covariance: $\mathbf{S}(k)^j = \mathbf{H}^j(k+1)\mathbf{P}(k+1|k)\mathbf{H}^{j^T}(k+1)+\mathbf{R}(k+1)$
        Kalman Filter gain: $\mathbf{K}^j(k+1) = \mathbf{P}(k+1|k)\mathbf{H}^{j^T}(k+1)[\mathbf{S}^j(k)]^{-1}$
        Iterated state: $\hat{\mathbf{x}}^{j+1}(k+1|k+1) = \hat{\mathbf{x}}^j(k+1|k) + \mathbf{K}^j(k+1)[\mathbf{z}^j(k+1)]$
    Updated state: $\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1)[\mathbf{z}(k+1)]$
    Updated covariance: $\mathbf{P}(k+1|k+1) = [\mathbf{I} - \mathbf{K}(k+1)\mathbf{H}(k+1)]\mathbf{P}(k+1|k)$
```

Chapter 6

Stereovision-Based State and Inertia Estimation

This chapter deals with the estimation of relative position, velocity, angular velocity, attitude and inertia ratios of a space object, without any a priori information. As already specified, the inspector spacecraft is defined as *leader* and the unknown object is the *target*. The only available data are provided by two cameras placed on the leader. This problem is described by non-linear equations for the observation and process model. For this reason, EKF and IEKF are used. The purpose of this work is to design a robust filtering algorithm, capable of dealing with real application uncertainties. In addition, a new approach to estimate the inertia ratios is derived. In fact, a pseudo-measurement equation is added to the observation model, to constrain the inertia ratios. Hence, the idea is to use an approach that does not need any geometrical knowledge of the object. This implies less data to handle and better computational efficiency. In this chapter, a mathematical formulation of the problem is presented. Dynamical and observation model are explained and the approach to estimate the inertia ratios is introduced. Then, numerical studies and their probabilistic results are shown.

6.1 Mathematical Formulation

6.1.1 Assumptions and Notation

First, the assumptions of this formulation are summarized. The spacecraft and the space object are assumed to orbit around the Earth. A new leader/target notation is introduced to better describe the faced problem. The leader is assumed to collect and track N feature points on the target. In addition, from different frames in time, it can recover information about the optical flow and therefore, about relative angular and translational velocity. The reference frames are assumed to be the same as the ones described in chapter 2 but, for clearness, they are here recalled.

The standard Earth-centred, inertial, Cartesian right-hand reference frame is indicated with the letter $\mathcal{I}$. $\mathcal{L}$ is the local-vertical, local-horizontal Euler–Hill (LVLH) reference frame. It is fixed to the leader spacecraft's centre of mass, the $\hat{\mathbf{x}}$ unit vector directed from the spacecraft radially outward, $\hat{\mathbf{z}}$ normal to the leader orbital plane, and $\hat{\mathbf{y}}$ completing the frame.

It is assumed to be aligned with $\mathcal{J}$, a Cartesian right-hand body-fixed reference frame attached to the leader spacecraft's centre of mass.

Finally $\mathcal{T}$, a Cartesian right-hand body-fixed reference frame centred in the target spacecraft's centre of mass. It is also assumed that this frame is coincident with the principal axis of inertia.

The notation that is used in the formulation is now presented. The vector $\boldsymbol{\rho_0}$ is the vector connecting the leader centre of mass with the target centre of mass, expressed in the leader frame. Analogously, $\boldsymbol{\rho_i}$ can be defined as the position vector, in the leader frame, between the leader centre of mass and the feature point $\mathbf{P}_i$. Consequently, $\dot{\boldsymbol{\rho}}_\mathbf{0}$ and $\dot{\boldsymbol{\rho}}_{\boldsymbol{i}}$ are the translational velocities of the target centre of mass and of a generic feature point, expressed in $\mathcal{L}$. The relative angular velocity is expressed as $\boldsymbol{\omega}$. This vector is the difference of the angular velocities of the leader and target respectively, expressed in the leader frame:

$$\boldsymbol{\omega} = \boldsymbol{\omega}_T|_{\mathcal{L}} - \boldsymbol{\omega}_L|_{\mathcal{L}} \tag{6.1}$$

The relative attitude is described using the rotation quaternion $\mathbf{q} = [q_0, q_1, q_2, q_3]^T$ where the first component is the scalar part and the other three are the vector one.

6.1.2 Process Model

Once the fundamental parameters are defined, the problem is formulated, starting from the dynamical or process model. The classical formulation for the dynamical model in a Kalman Filter dealing with non-linear equations is:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{w}(t) \tag{6.2}$$

where $\mathbf{x}$ is the state vector, $f(\mathbf{x})$ is a non-linear function describing the process and $\mathbf{w}$ is a random zero-mean white noise.

In our case, the state vector is defined as:

$$\mathbf{x} = [\boldsymbol{\rho}_\mathbf{0}^T,\ \dot{\boldsymbol{\rho}}_\mathbf{0}^T,\ \boldsymbol{\omega}^T,\ \mathbf{q}^T,\ \mathbf{P}_i{}^T]^T \tag{6.3}$$

This is a 13 + 3N elements vector where N is the number of feature points. This is the reason why it has been decided to use a simplified dynamics instead of the completely coupled one. In that case, the resulting state vector is:

$$\mathbf{x}_{Coup} = [\boldsymbol{\rho}_\mathbf{0}^T,\ \dot{\boldsymbol{\rho}}_\mathbf{0}^T,\ \boldsymbol{\rho}_{\boldsymbol{i}}^T,\ \dot{\boldsymbol{\rho}}_{\boldsymbol{i}}^T,\ \boldsymbol{\omega}^T,\ \mathbf{q}^T,\ \mathbf{P}_i{}^T]^T \tag{6.4}$$

This is a 13 + 9N elements vector. In cases with a large number of features points, this can lead to poor computational performance and additional complexity.

At this point the dynamical model can be reviewed. As already said, the relative dynamics is described using a set of non-linear differential equations. Moreover, the translational and rotational behaviour are decoupled. This set of equations is given by eqs. (2.9) to (2.13), (2.15) and (2.27).

The motion of the feature points can be defined as :

$$\dot{\mathbf{P}}_T^i|_\mathcal{L} = \dot{\mathbf{P}}_T^i|_\mathcal{T} + \boldsymbol{\omega} \times \mathbf{P}_T^i|_\mathcal{L} = \boldsymbol{\omega} \times \mathbf{P}_T^i|_\mathcal{L} \tag{6.5}$$

This is expressed in leader frame. However, the dynamics of the feature points is simpler if expressed in the target frame. In fact, due to the rigid body assumption, a feature point cannot change its relative position with respect to the target centre of mass. This leads to:

$$\dot{\mathbf{P}}_T^i|_\mathcal{T} = 0 \tag{6.6}$$

Summarizing, the function $f(\mathbf{x})$ is implemented as:

$$f_1(\mathbf{x}) = x_4 \tag{6.7}$$

$$f_2(\mathbf{x}) = x_5 \tag{6.8}$$

$$f_3(\mathbf{x}) = x_6 \tag{6.9}$$

$$f_4(\mathbf{x}) = 2\dot{\vartheta}_L x_5 + \ddot{\vartheta}_L x_2 + \dot{\vartheta}_L^2 x_1 + \frac{\mu(r_L + x_1)}{[(r_L + x_1)^2 + x_2{}^2 + x_3{}^2]^{\frac{3}{2}}} + \frac{\mu}{r_L{}^2} \tag{6.10}$$

$$f_5(\mathbf{x}) = -2\dot{\vartheta}_L x_4 - \ddot{\vartheta}_L x_1 + \dot{\vartheta}_L^2 x_2 - \frac{\mu x_2}{[(r_L + x_1)^2 + x_2{}^2 + x_3{}^2]^{\frac{3}{2}}} \tag{6.11}$$

$$f_6(\mathbf{x}) = -\frac{\mu x_3}{[(r_L + x_1)^2 + {x_2}^2 + {x_3}^2]^{\frac{3}{2}}} \tag{6.12}$$

$$\begin{bmatrix} f_7(\mathbf{x}) \\ f_8(\mathbf{x}) \\ f_9(\mathbf{x}) \end{bmatrix} = D(\mathbf{x}_q)\mathbf{I}_T^{-1}[\mathbf{N}_T - D(\mathbf{x}_q)^T(\mathbf{x}_\omega + \boldsymbol{\omega}_L) \times \mathbf{I}_L D(\mathbf{x}_q)^T(\mathbf{x}_\omega + \boldsymbol{\omega}_L)] - \boldsymbol{\omega}_L \times \mathbf{x}_\omega - \mathbf{I}_L^{-1}[\mathbf{N}_L - \boldsymbol{\omega}_L| \times \mathbf{I}_L \mathbf{x}_\omega] \tag{6.13}$$

$$\begin{bmatrix} f_{10}(\mathbf{x}) \\ f_{11}(\mathbf{x}) \\ f_{12}(\mathbf{x}) \\ f_{13}(\mathbf{x}) \end{bmatrix} = \frac{1}{2} Q(\mathbf{x}_q) D(\mathbf{x}_q)^T \mathbf{x}_\omega \tag{6.14}$$

$$\begin{bmatrix} f_{14}(\mathbf{x}) \\ f_{15}(\mathbf{x}) \\ f_{16}(\mathbf{x}) \end{bmatrix} = \mathbf{0} \tag{6.15}$$

with $\mathbf{x}_\omega = [x_7,\, x_8,\, x_9]$, $\mathbf{x}_q = [x_{10},\, x_{11},\, x_{12},\, x_{13}]$.

6.1.3 Observation Model

The observation model has been already discussed in chapter 3. This paragraph summarizes the main equations used in the observation model. As previously explained, it is possible to extract information about the position of a generic feature point in a 3D space and project it in the 2D image plane. Considering the cameras configuration as in fig. 3.3, the perspective projection is computed as described in eqs. (3.2) to (3.5).

Another information that is possible to extract is related to the image velocity. In fact, exploiting the expression of the optical flow, a relation, depending of $\dot{\boldsymbol{\rho}}$ and $\boldsymbol{\omega}$, can be recovered. Equations (3.7) and (3.8) describe this relation.

The disparity can also be computed using eq. (3.11).

It is important to underline that, in reality, cameras collect images at a given sampling frequency. Using two subsequent frames, the optical flow can be estimated and the image velocity computed. However, in the presented observation model, it is assumed that the information about the image velocity is recovered at each time step. In addition, a set of feature points is chosen and it is assumed that they are always in the view of the cameras. Therefore, according to our assumptions, the initial set of points is always traceable. At each time step, the discrete measurement vector provided by the cameras is:

$$\mathbf{Z}_i = [\mathbf{w}_{Ri},\ \mathbf{w}_{Li},\ \dot{\mathbf{w}}_{Ri},\ \dot{\mathbf{w}}_{Li},\ d_i] \tag{6.16}$$

Therefore, the observation equation is:

$$\mathbf{Z}_i = h(\mathbf{x}) + \mathbf{v}(t) \tag{6.17}$$

with $\mathbf{v}$ being a random zero-mean white noise and $h(\mathbf{x})$ given by:

$$h(\mathbf{x}) = \begin{pmatrix} \frac{x_i}{y_i} \\ \frac{z_i}{y_i} \\ \frac{x_i - b}{y_i} \\ \frac{z_i}{y_i} \\ \frac{x_i - b}{y_i} - \frac{z_i}{y_i} \\ \begin{bmatrix} \frac{1}{y_i} A(w_{Ri}) & B(w_{Ri}) \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\rho}}_0 \\ \boldsymbol{\omega} \end{bmatrix} \\ \begin{bmatrix} \frac{1}{y_i} A(w_{Li}) & B(w_{Li}) \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\rho}}_0 \\ \boldsymbol{\omega} \end{bmatrix} \end{pmatrix} \tag{6.18}$$

6.1.4 Estimation Procedure

The widely discussed EKF and IEKF are used to estimate the state. Knowing the process (6.2) and measurement (6.17) equations and using the algorithm 1 and algorithm 2, the state, expressed by eq. (6.3), can be estimated. The EKF is a very powerful tool and it has been extensively used in real time applications for its computational efficiency and simplicity. However, the estimation accuracy depends on many factors. First of all, initial conditions play a determinant role on the convergence of a Kalman filter. These are usually random conditions, determined by a probability density function that depends on the initial covariance matrix. The first value of the covariance matrix, must be chosen in a realistic way to take into account the real possible uncertainties on the initial state. This can significantly affect the convergence and performance of a filter if it is not robust enough. In addition, the process and measurement covariance matrices must be tuned in a proper way. This is particularly important dealing with EKF. In fact, with the classical Kalman Filter, it is possible to have conditions in which both the process **Q** and measurement **R** noise covariance matrices are constant. In addition, it is not rare to find processes and measurements described by constant matrices **F** and **H**. In this case, the error covariance **P** and the Kalman Filter gain **K** will stabilize and remain constant. So, it is possible to pre-compute all the parameters running the filter off-line. Dealing with non-linear processes and with variable noise, this procedure is not applicable. For this reason, a good tuning of the matrices **R** and **Q** must be done. Unfortunately, there is not a rule of thumb for tuning the covariance matrices. This is a matter of experience, but simple considerations can be done. The noise measurement covariance matrix **R** can be defined in a simple way, knowing the accuracy of the sensor. In fact, by definition,

$$\mathbf{R} = E[\mathbf{v}\mathbf{v}^T] \tag{6.19}$$

where **v** is the noise associated to the measurements. This is true in linear applications, when a real sensor is present. In theory, **R** could also be assessed performing off-line sample tests on the measured signal. Therefore, it is usually chosen according to the value of the imposed measurement noise **v** and to the value of the filter innovation. Whereas, the tuning of **Q** is in general more difficult. With a good tuning, acceptable performance can be obtained also using a simple and poor model for the process. In this work, different simulations are performed to validate and prove the robustness of the presented algorithm.

6.2 Inertia Ratios Estimation

In the previous section, an estimation procedure is discussed. However, for the presented model, an a priori knowledge of the inertia matrix is necessary. This is not a realistic assumption since we are dealing with a completely unknown and uncooperative space object. To overcome this contradiction, an estimation of the basic inertial properties is necessary. A torque free motion is assumed. Moreover, no contact condition is imposed and therefore a known torque on the target object cannot be applied. In torque-free motion conditions, the non observability of the complete inertia matrix has been already discussed. Thus, two parameters are sufficient to represent the inertia matrix. With a parametrized inertia matrix, the motion can be propagated in the correct way. However, no geometrical or mass properties can be recovered. On the other hand, inertia components remain constant in time, under rigid body assumption, and no direct measurements are available. For this reason, a method to improve the filter accuracy and convergence has to be developed. This chapter provides a description of the chosen inertia parametrization and introduces an approach for the inertia components estimation.

6.2.1 Inertia Parametrization

It has been proved that, given an angular velocity trajectory $\boldsymbol{\omega}(t)$, the inertia matrix has only two observable degrees of freedom. This leads to the conclusion that the inertia should be re-expressed using only two parameters. In literature, only few works deal with the inertia matrix parametrization. Different approaches and formulations are used. Writing the Euler equations as:

$$\dot{\omega}_x = \frac{I_y - I_z}{I_x}\,\omega_y\omega_z \tag{6.20}$$

$$\dot{\omega}_y = \frac{I_z - I_x}{I_y}\,\omega_x\omega_z \tag{6.21}$$

$$\dot{\omega}_z = \frac{I_x - I_y}{I_x}\,\omega_y\omega_x \tag{6.22}$$

the inertia matrix could be easily parametrized considering the variables:

$$p_x = \frac{I_y - I_z}{I_x} \tag{6.23}$$

$$p_y = \frac{I_z - I_x}{I_y} \tag{6.24}$$

$$p_z = \frac{I_x - I_y}{I_x} \tag{6.25}$$

This is the formulation used by Aghili [72]. The first observation is that, with this parametrization, the number of parameters is larger than the number of degrees of freedom. For this reason, one of the three parameters can always be expressed by a combination of the other two. In addition, from a physical point of view, these parameters are subjected to the following constraint:

$$p_x = \frac{I_y - I_z}{I_x} > 1 \tag{6.26}$$

$$p_y = \frac{I_z - I_x}{I_y} > 1 \tag{6.27}$$

$$p_z = \frac{I_x - I_y}{I_x} > 1 \tag{6.28}$$

This is a strong constraint and it is not a reasonable formulation in applications dealing with a Kalman Filter. In fact, the initial value of these parameters has to be random, according to a Gaussian probability density function. Usually, this kind of distribution does not have any kind of constraint and it has an infinite physical domain. For these reasons, this formulation is not used in our work. The paper by Lichter [23], discussed in the literature review, proposes a different parametrization. He exploits the quaternion $\mathbf{q} = [q_0,\ q_1,\ q_2,\ q_3]^T$ to parametrize the inertia components. In particular, defining three parameters as:

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 2\left(q_1 q_3 + q_2 q_0\right) \\ 2\left(q_2 q_3 + q_1 q_0\right) \\ {q_0}^2 - {q_1}^2 - {q_2}^2 - {q_3}^2 \end{bmatrix} \tag{6.29}$$

the inertia matrix can be expressed as a combination of z_1, z_2, z_3:

$$I_x = |z_2| + |z_3| \tag{6.30}$$

$$I_y = |z_1| + |z_3| \tag{6.31}$$

$$I_z = |z_1| + |z_2| \tag{6.32}$$

Also for this formulation, the four parameters used to parametrize the inertia matrix are more than the two degrees of freedom. In addition, also if this formulation does not have any explicit constraints, for certain quaternions, the matrix components can assume values with no physical meaning. For example, the classical unit quaternion $\mathbf{q} = [0\ \ 0\ \ 0\ \ 1]^T$ leads to a value of 0 for the z component of the inertia matrix. Of course, a diagonal component of the inertia matrix equal to zero has no physical sense. Another approach was introduced by Tweddle [28]. This formulation introduces only two parameters:

$$k_1 = ln\left(\frac{I_x}{I_y}\right) \tag{6.33}$$

$$k_2 = ln\left(\frac{I_y}{I_z}\right) \tag{6.34}$$

This is the only formulation, so far, that relies on the minimum number of parameters. Additionally, these two parameters can be properly assumed to be Gaussian random variables. In fact, k_1 and k_2 do not have any additional constraints. The inertia ratios have to be greater than zero and they can be each value up to infinite. This is a consistent parametrization because the natural logarithm has the same validity domain. After this short review of different parametrizations present in literature, the obvious choice is the last one. In fact, this formulation seems to be the most robust and consistent for our implementation. The only critical condition is for k_1 and k_2 extremely high. In fact, when the value of the two parameters tends to infinite (the Gaussian distribution tends to zero), one of the inertia components tends to an extremely small value. This is physically not acceptable but, it is also clear that this condition is highly uncommon for a Gaussian distribution maximized by k_1 and k_2 equal to zero. Using this parametrization, the target inertia matrix can be expressed as:

$$\mathbf{I}_T = \begin{bmatrix} \frac{I_x}{I_y} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{I_z}{I_y} \end{bmatrix} = \begin{bmatrix} e^{k_1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-k_2} \end{bmatrix} \tag{6.35}$$

At this point, these two parameters must be estimated by the filter. Therefore, a new augmented state can be defined as:

$$\mathbf{x} = [\boldsymbol{\rho}_\mathbf{0}^T,\ \dot{\boldsymbol{\rho}}_\mathbf{0}^T,\ \boldsymbol{\omega}^T,\ \mathbf{q}^T,\ \mathbf{P}_i{}^T,\ k_1,\ k_2]^T \tag{6.36}$$

Also the dynamical model is different. In fact, the parametrized inertia matrix will substitute the previous value of the target inertia matrix in the rotational dynamics expression. Additionally, two equations for k_1 and k_2 are considered.

$$\frac{\partial k_1}{\partial t} = 0 \tag{6.37}$$

$$\frac{\partial k_2}{\partial t} = 0 \tag{6.38}$$

Equation (6.37) and (6.38) are valid under the assumption of rigid body motion and without considering any mass variation.

Considering conditions with low relative angular velocity, the filter will not converge properly to the exact value of the inertia ratios. In order to improve the convergence and the accuracy of the inertia ratios estimation, a novel concept is introduced. An equality state constraint is used to improve the inertia ratios estimation and convergence.

6.2.2 Equality Constraint

In this paragraph, a brief review of the current methods to introduce an equality constraint in the Kalman Filter formulation is addressed. It has already said that the EKF, in general, is not necessarily an optimal estimator. In same cases, a constraint given by the physics or geometry of the problem can be known. In that case, it can be reasonable to exploit this additional information to modify the Kalman Filter. To achieve this goal, several approaches exist. Some of them are the model reduction method, the pseudo measurement method, the estimate projection method and the system projection method. The most used and reliable are the first two. The model reduction method was introduced by Wen-Durrant [73]. This approach tries to reduce the order of the system model parametrization. In this way, a constraint filtering algorithm can be reduced to an unconstrained problem. Moreover, the dimension of the obtained model is lower and this could result in better computational results. However, this approach is not always applicable in cases with non-linear system dynamics. In addition, the physical meaning associated to the state can be lost. The pseudo measurement method, instead, exploits a perfect measurement to include the equality constraint. Many papers deal with this formulation in literature and [74], [75], [76] are only few examples in which this method is discussed. Adding a pseudo measurement, consists in including eq. (6.39) in the observation model.

$$\boldsymbol{l}(\mathbf{x}_k) = 0 \tag{6.39}$$

In this way, no real measurement is added. In fact, the output of this equation is zero. However, in this way, a constraint on the state is automatically imposed. It has to be noted that the measurement noise associated to the pseudo measurement is zero. This means that also the associate covariance is singular. In this thesis, the pseudo measurement method is used to estimate the inertia ratios. This approach has been widely used in different fields (health analysis [77], economic systems [78], biomedical systems [79], robotics [80]).

Going into the details of our formulation, the measurement model is augmented with an additional equation. This leads to:

$$\mathbf{Z}_i = [\mathbf{w}_{Ri},\ \mathbf{w}_{Li},\ \dot{\mathbf{w}}_{Ri},\ \dot{\mathbf{w}}_{Li},\ \mathbf{0}] \tag{6.40}$$

and $h(\mathbf{x})$ becomes:

$$h(\mathbf{x}) = \begin{pmatrix} \frac{x_i}{y_i} \\ \frac{z_i}{y_i} \\ \frac{x_i - b}{y_i} \\ \frac{z_i}{y_i} \\ \frac{x_i - b}{y_i} - \frac{z_i}{y_i} \\ \begin{bmatrix} \frac{1}{y_i} A(w_{Ri}) & B(w_{Ri}) \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\rho}}_{\mathbf{0}} \\ \boldsymbol{\omega} \end{bmatrix} \\ \begin{bmatrix} \frac{1}{y_i} A(w_{Li}) & B(w_{Li}) \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\rho}}_{\mathbf{0}} \\ \boldsymbol{\omega} \end{bmatrix} \\ \dot{\boldsymbol{\omega}}_T + \mathbf{I}_T{}^{-1}(\boldsymbol{\omega}_T \times \mathbf{I}_T \boldsymbol{\omega}_T) \end{pmatrix} \tag{6.41}$$

The last term of eq. (6.41) is the new pseudo measurement. It is the classical Euler equation for the rotational dynamics of the target. A fundamental aspect to take into account is that in the pseudo measurement equation, the target angular acceleration is present. Information about this quantity have to be recovered from the actual measurement. However, with knowledge of $\boldsymbol{\omega}_L$, $\dot{\boldsymbol{\omega}}_L$ and $\boldsymbol{\omega}$, there is not an analytical expression independent on $\mathbf{I}_t$ to compute $\dot{\boldsymbol{\omega}}_T$. This implies that the angular acceleration of the target has to be measured. With the knowledge of the optical flow, the value of $\boldsymbol{\omega}$ at each time step can be recovered. Then, a numerical differentiation can be performed to find the relative angular acceleration. The problem is that the measure of $\boldsymbol{\omega}$ is affected by noise, therefore, the numerical differentiation will not be accurate. Dealing with derivative of a quantity affected by errors is common in many scientific fields. For this reason, several methods to compute numerical derivative of noisy data exist. In our work, since no real measurements are available, the value of $\dot{\boldsymbol{\omega}}$ is obtained by adding noise to the quantity available from the real dynamical model. For completeness, some of the differentiation methods are described in appendix B. Once the relative angular acceleration is recovered, the target angular acceleration has to be computed.

Knowing that, in the leader frame:

$$\dot{\boldsymbol{\omega}} = \mathbf{D}\dot{\boldsymbol{\omega}}_T - \dot{\boldsymbol{\omega}}_L - \boldsymbol{\omega}_L \times \boldsymbol{\omega} \tag{6.42}$$

the target angular acceleration can be easily computed as:

$$\dot{\boldsymbol{\omega}}_T = \mathbf{D}^{-1}[\dot{\boldsymbol{\omega}} + \dot{\boldsymbol{\omega}}_L + \boldsymbol{\omega}_L \times \boldsymbol{\omega}] \tag{6.43}$$

The measurement on $\dot{\boldsymbol{\omega}}$ has been introduced with the pseudo-measurement and not as direct measurement, to directly correlate the angular acceleration to the target inertia ratios. Moreover, since the numerical integration can amplify the errors, in this way, an eventual poor measurement will only affect the inertia components estimation and not the other quantities.

6.3 Numerical Simulations

In this section, an evaluation of the performance and robustness of the filter is presented. Monte-Carlo simulations are performed for different values of the initial error covariance and initial relative position. Each simulation is performed considering a satellite and an object in low Earth orbits. In particular, the leader orbit is known. It is assumed that the orbit of the leader has eccentricity $e_L = 0.05$, semi-major axis $a_L = 7170\,km$, inclination $i_L = 15\,deg$, argument of the perigee $\omega = 340\,deg$ and right ascension of ascending node $\Omega = 0\,deg$. According to our parametrization, the leader inertia is

$$I_L = \begin{bmatrix} 0.83 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1.083 \end{bmatrix} kg\,m^2 \tag{6.44}$$

In addition, two parallel cameras, in a stereo configuration and pointing in the same direction are mounted on the leader spacecraft. The baseline between the cameras is assumed equal to $1\,m$. Moreover, only five feature points are supposed to be measured. This is an extreme case, in fact, more than five points are usually visible and detectable. However, this condition may occur when the object is not properly illuminated or if it is too bright. Additionally, considering only a small number of points, the robustness and convergence of the filter are tested also with poor available measurements. With few points to be tracked, the computational efficiency of the algorithm is improved. The detected features are assumed to be spread over the body of the target with a distance from the centre of mass in the order of $1.5\,m$. This can be varied according to the dimension of the target object.

After defining the initial condition for the leader orbit, the state has to be initialized. The initial state vector is:

$$\mathbf{x}_0 = [\boldsymbol{\rho}_0,\ \ \dot{\boldsymbol{\rho}}_0,\ \ \boldsymbol{\omega},\ \ \mathbf{q}_0,\ \ \mathbf{P}_T{}^i,\ \ k_1,\ \ k_2] \tag{6.45}$$

This vector will be defined for each simulation. At this point, the filter parameters have to be selected. In particular, the covariance matrices **Q**, **R**, **P** have to be chosen. As already said, **R** represents the noise of the measurement and it can be determined knowing the sensor accuracy. **Q** has to be selected to ensure the convergence of the filter. Finally, the initial value of **P**, the error covariance matrix, represents the uncertainties in the initial estimation of the state. In fact, the initial value of the state for the filter is computed with the MATLAB function *mvnrnd*. This function has as inputs the value of the mean and the corresponding covariance. As output, it produces a random vector from a multivariate normal distribution, having the selected mean and covariance. In our case, the mean is the real value of the initial state and the covariance is the initial error covariance **P**.

For the initial random value and for each time step, the quaternion is normalized. According to algorithm 2, after the initial condition initialization, the predicted value of the state has to be computed using the dynamical model. Therefore, equations from eq. (6.7) to eq. (6.15) are implemented in MATLAB. The function *ode45* is used to integrate the set of equations for each time step. Then, a centred difference method is used to compute the Jacobian of the process model. With this value, the transition matrix is computed and the new error covariance is predicted. The transition matrix is computed using the MATLAB function *expm* that computes the matrix exponential of a given matrix. At this point, a *while* cycle is used to implement the iterative procedure of the Iterated Extended Kalman Filter. A tolerance equal to 0.01 and a maximum number of iterations equal to 10 are used. For the observation model, the eq. (6.41) is solved and linearised with the same approximate method. The noise of the measurements is modelled as a zero-mean Gaussian with standard deviation of 10^{-5}. The *mvnrnd* function is used to generate it. Finally, the filter innovation, innovation covariance and gain are iteratively computed and state and covariance are updated. In our simulations, a time step of 1 second is used and the total time of the simulation is 100 seconds. The general scheme of the algorithm is presented in fig. 1.3. The computed errors are defined as:

$$e_\rho = \sqrt{(\rho_x - \underline{\rho}_x)^2 + (\rho_y - \underline{\rho}_y)^2 + (\rho_z - \underline{\rho}_z)^2} \tag{6.46}$$

with e_ρ being the error of the estimation of the centre of mass. In this notation,

$\underline{\rho}$ denotes the estimated value of ρ.

In the same way the relative translational velocity error can be defined:

$$e_{\dot{\rho}} = \sqrt{(\dot{\rho}_x - \underline{\dot{\rho}}_x)^2 + (\dot{\rho}_y - \underline{\dot{\rho}}_y)^2 + (\dot{\rho}_z - \underline{\dot{\rho}}_z)^2} \tag{6.47}$$

And the relative angular velocity error:

$$e_{\omega} = \sqrt{(\omega_x - \underline{\omega}_x)^2 + (\omega_y - \underline{\omega}_y)^2 + (\omega_z - \underline{\omega}_z)^2} \tag{6.48}$$

For k_1 and k_2 the error is simply:

$$e_{k_1} = \sqrt{(k_1 - \underline{k}_1)^2} \tag{6.49}$$

$$e_{k_1} = \sqrt{(k_2 - \underline{k}_2)^2} \tag{6.50}$$

The attitude error is defined in a different way. In fact, quaternion can be considered as a four-dimensional vector space. For this reason, the operations with quaternions have a different form. Recalling the definition of the inverse of a quaternion:

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{||\mathbf{q}||^2} \tag{6.51}$$

where $\mathbf{q}^*$ is the conjugate of $\mathbf{q}$, the error quaternion is equal to:

$$\mathbf{q}_e = \mathbf{q} \otimes \underline{\mathbf{q}}^{-1} \tag{6.52}$$

The symbol $\otimes$ is defined as the product of two quaternions. Given two quaternions

$$\mathbf{q} = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 \tag{6.53}$$

$$\mathbf{t} = t_0 + \mathbf{i}t_1 + \mathbf{j}t_2 + \mathbf{k}t_3 \tag{6.54}$$

their product is equal to

$$\mathbf{r} = r_0 + \mathbf{i}r_1 + \mathbf{j}r_2 + \mathbf{k}r_3 \tag{6.55}$$

with

$$r_0 = (t_0 q_0 - t_1 q_1 - t_2 q_2 - t_3 q_3) \tag{6.56}$$

$$r_1 = (t_0 q_1 + t_1 q_0 - t_2 q_3 + t_3 q_2) \tag{6.57}$$

$$r_2 = (t_0 q_2 + t_1 q_3 + t_2 q_0 - t_3 q_1) \tag{6.58}$$

$$r_3 = (t_0 q_3 - t_1 q_2 + t_2 q_1 + t_3 q_0) \tag{6.59}$$

Finally, the attitude estimation error can be defined as:

$$e_\theta = 2\cos^{-1}(q_{e_0}) \tag{6.60}$$

where in out notation, q_{e_0} is the scalar part of the error quaternion.

In the following examples, the performance of the filter is analysed. Several Monte Carlo simulations are discussed with different initial values of covariance matrix and initial conditions.

6.3.1 Case A

In the first case scenario, the filter is tested with the following initial conditions:

- $\boldsymbol{\rho}_0 = [10,\ \ 60,\ \ 10]\ m$
- $\dot{\boldsymbol{\rho}}_0 = [0.01,\ \ -0.0225,\ \ -0.01]\ m/s$
- $\boldsymbol{\omega}_0 = [-0.1,\ \ -0.1,\ \ 0.034]\ deg/s$
- $\mathbf{q}_0 = [0,\ \ 0,\ \ 0,\ \ 1]$

For this case, a small covariance matrix is used. Having small values for the covariance matrix means to have small uncertainties on the initial state. This is, in general, not true in real applications with unknown and uncooperative objects. However, this assumption should guarantee better filter performance. The covariance matrix $\mathbf{P}$ is given by:

$$\mathbf{P} = \begin{bmatrix} {\sigma_\rho}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & {\sigma_{\dot\rho}}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & {\sigma_\omega}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & {\sigma_q}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & {\sigma_P}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & {\sigma_I}^2 \end{bmatrix} \tag{6.61}$$

with:

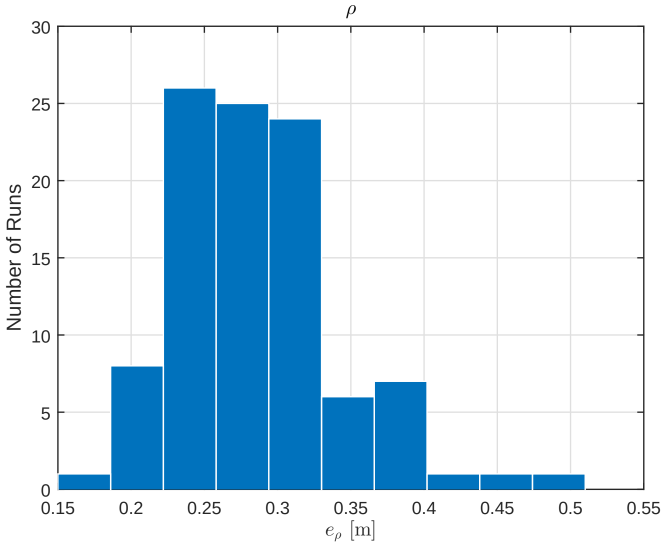
- ${\sigma_\rho}^2 = [1,\ 1,\ 1] \cdot 10^{-1}\ m^2$

- ${\sigma_{\dot{\rho}}}^2 = [1,\ 1,\ 1] \cdot 10^{-1}\ m^2/s^2$
- ${\sigma_{\omega}}^2 = [1,\ 1,\ 1] \cdot 10^{-1}\ deg^2/s^2$
- ${\sigma_{q}}^2 = [1,\ 1,\ 1,\ 1] \cdot 10^{-6}$
- ${\sigma_{P}}^2 = [1,\ 1,\ 1] \cdot 10^{-1}\ m^2$
- ${\sigma_{I}}^2 = [1,\ 1] \cdot 10^{-1}$

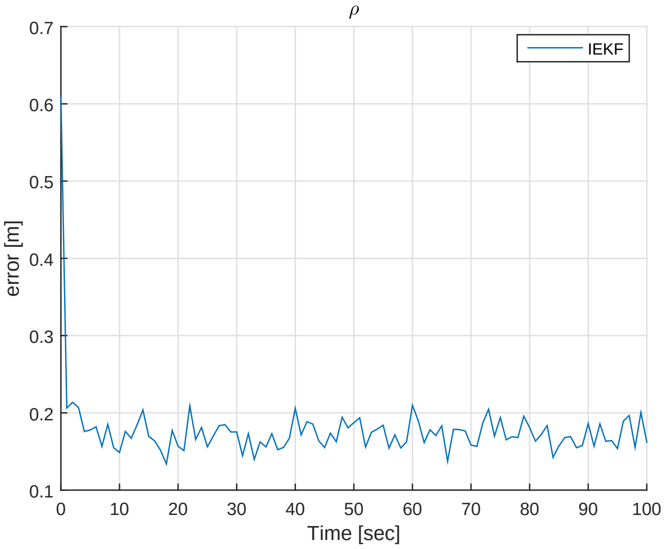
For this case, 100 simulations are considered. The mean relative errors after 10 seconds are evaluated according to eqs. (6.46) to (6.50) and (6.60). The results are analysed using histograms. In a table are then summarized the maximum error values for 100, 90, 70 and 50 percentiles. To evaluate the different performance of the two filters, for each simulation, the error difference between IEKF and EKF is computed as:

$$dif = e_{EKF} - e_{IEKF} \tag{6.62}$$

In this first case, the error trend for a single run, using the IEKF, is presented to show the general convergence trend of the filter.

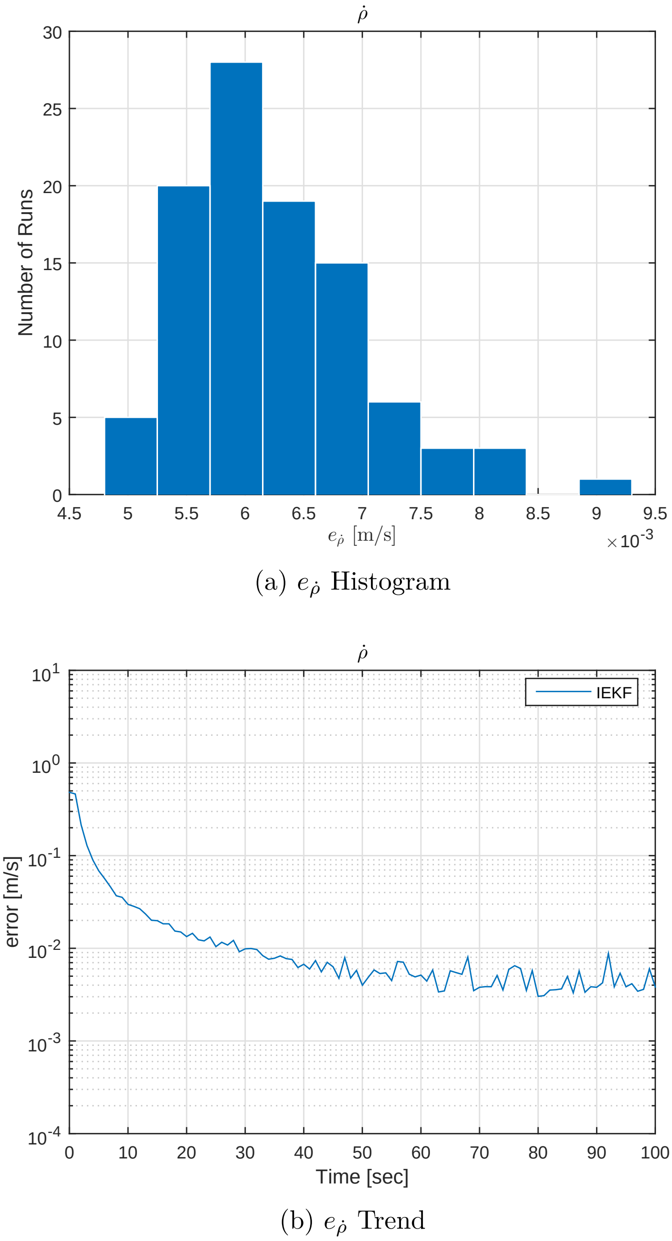


(a) e_ρ Histogram



(b) e_ρ Trend

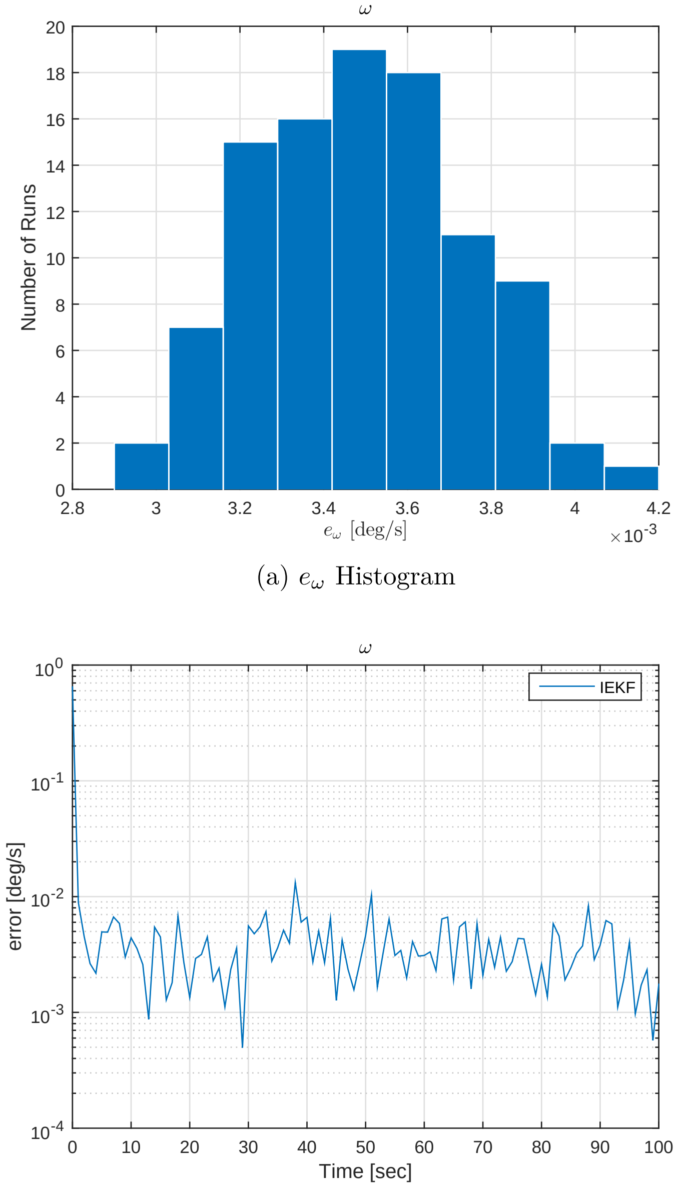
Figure 6.1. Relative Position Error



(a) $e_{\dot{\rho}}$ Histogram

(b) $e_{\dot{\rho}}$ Trend

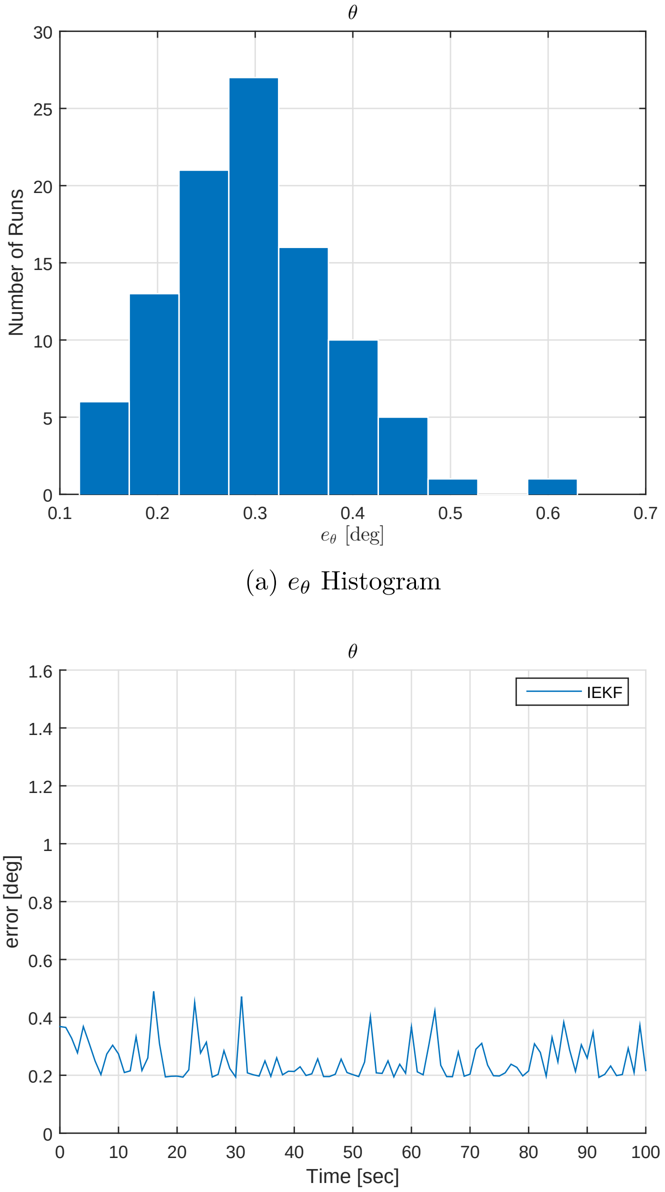
Figure 6.2. Relative Translational Velocity Error



(a) e_ω Histogram

(b) e_ω Trend

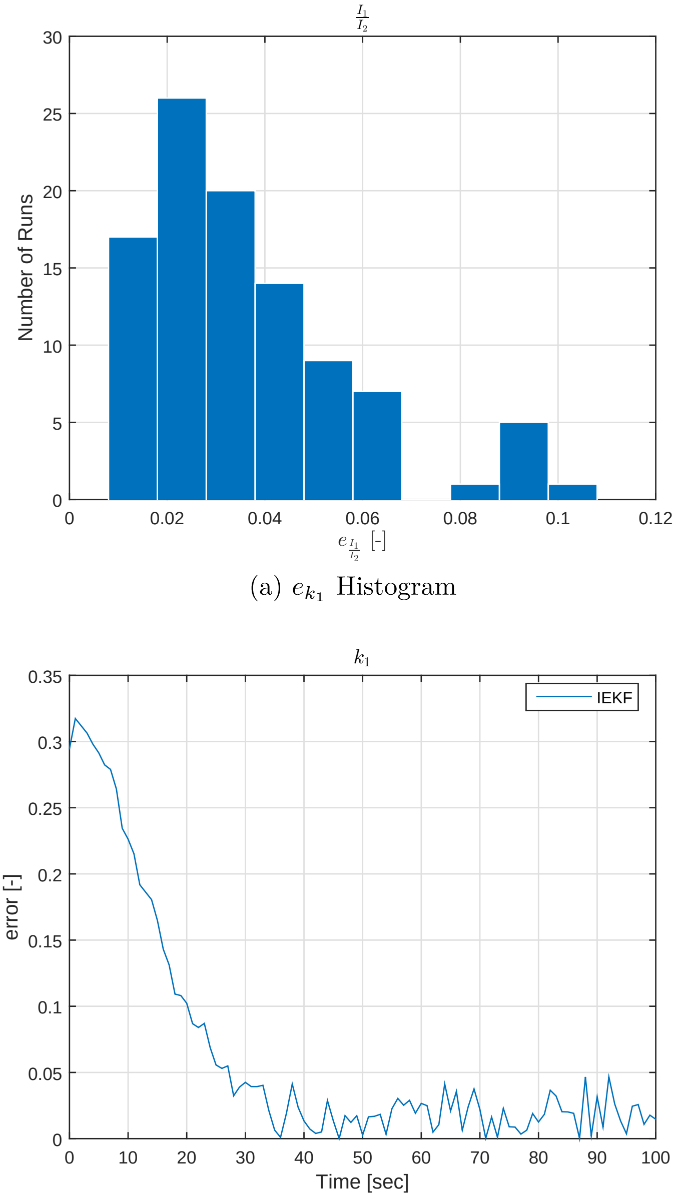
Figure 6.3. Relative Angular Velocity Error



(a) e_θ Histogram

(b) e_θ Trend

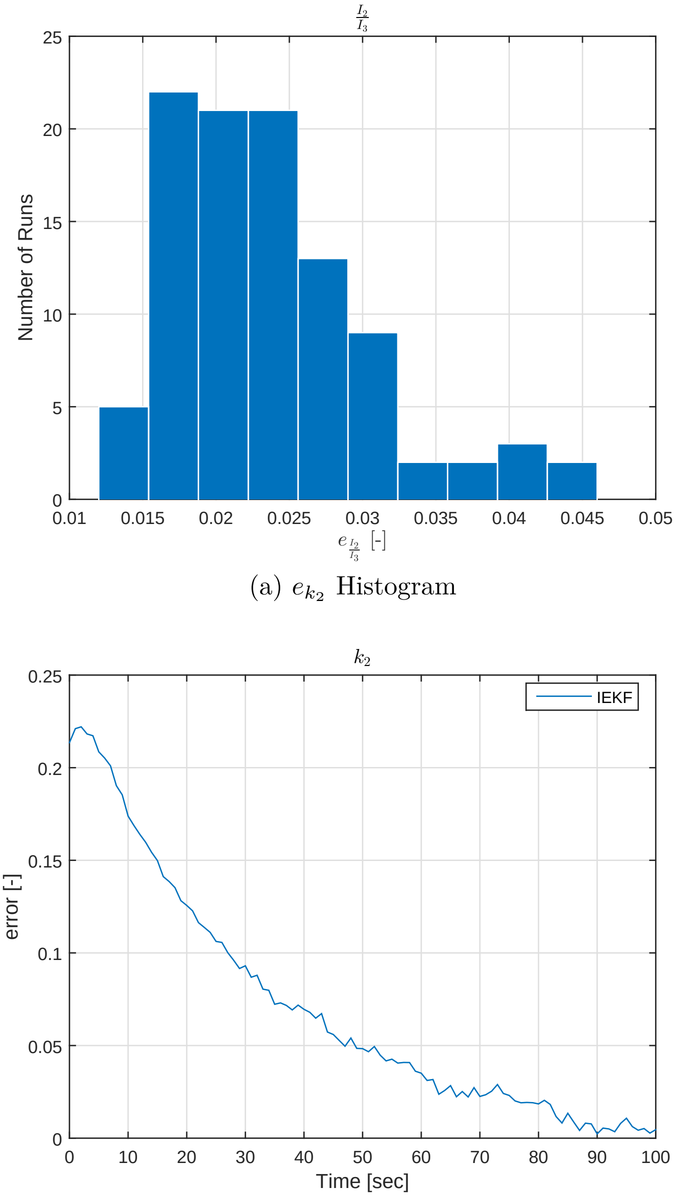
Figure 6.4. Relative Attitude Error



(a) e_{k_1} Histogram

(b) e_{k_1} Trend

Figure 6.5. Inertia Ratio Error



(a) e_{k_2} Histogram

(b) e_{k_2} Trend

Figure 6.6. Inertia Ratio Error

The presented results show robust convergence in all the analysed simulations. In particular, in fig. 6.1 the error relative to the position of the centre of mass is between 0.2 and 0.35 m in most of the cases.

Very good results are obtained for the relative angular and translational velocity in fig. 6.1 and fig. 6.3. This is probably connected to the fact that the optical flow equation is exploited.

The relative attitude is always difficult to estimate in a proper way and with good convergence. Figure 6.4 shows poor convergence of the relative angle error. The error tends to remain close to the initial value.

The two inertia ratios have good convergence thanks to the imposed equality constraint. Note that the results relative to the component of the state $\mathbf{P}_T{}^i$ are not reported. In fact, $\mathbf{P}_T{}^i$ does not represent any physical quantity of interest. In fact, in a real application, we could be interested in the evaluation of $\boldsymbol{\rho}_i$. This quantity depends not only on $\mathbf{P}_T{}^i$ but also on $\boldsymbol{\rho}_0$ and on the relative attitude.

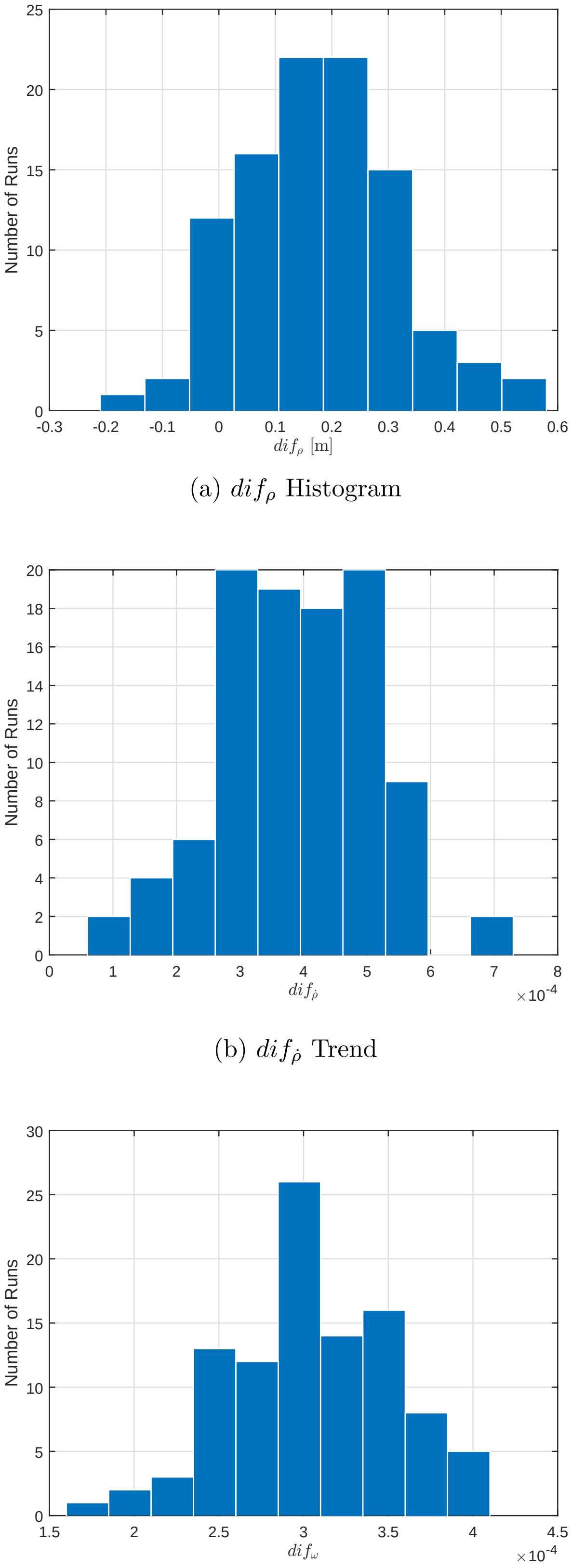
All the obtained results present oscillations. This is due to different factors. In fact, the IEKF tolerance and the maximum number of iterations can influence the errors trend. In addition, the time step for the linearisation of both measurement and process model plays a role. Of course, also the measurement and model noise can affect the error behaviour.

In table 6.1 , the results are summarized.

Table 6.1. State Errors - Case A

Percentiles	$\rho\,[m]$	$\dot{\rho}\,[m/s]$	$\omega\,[deg/s]$	$\theta\,[deg]$	$k_1\,[-]$	$k_2\,[-]$
50	0.27	0.0063	0.0035	0.29	0.033	0.022
70	0.31	0.0067	0.0036	0.33	0.043	0.026
90	0.36	0.0079	0.0038	0.41	0.06	0.031
100	0.51	0.0092	0.0041	0.62	0.1	0.045

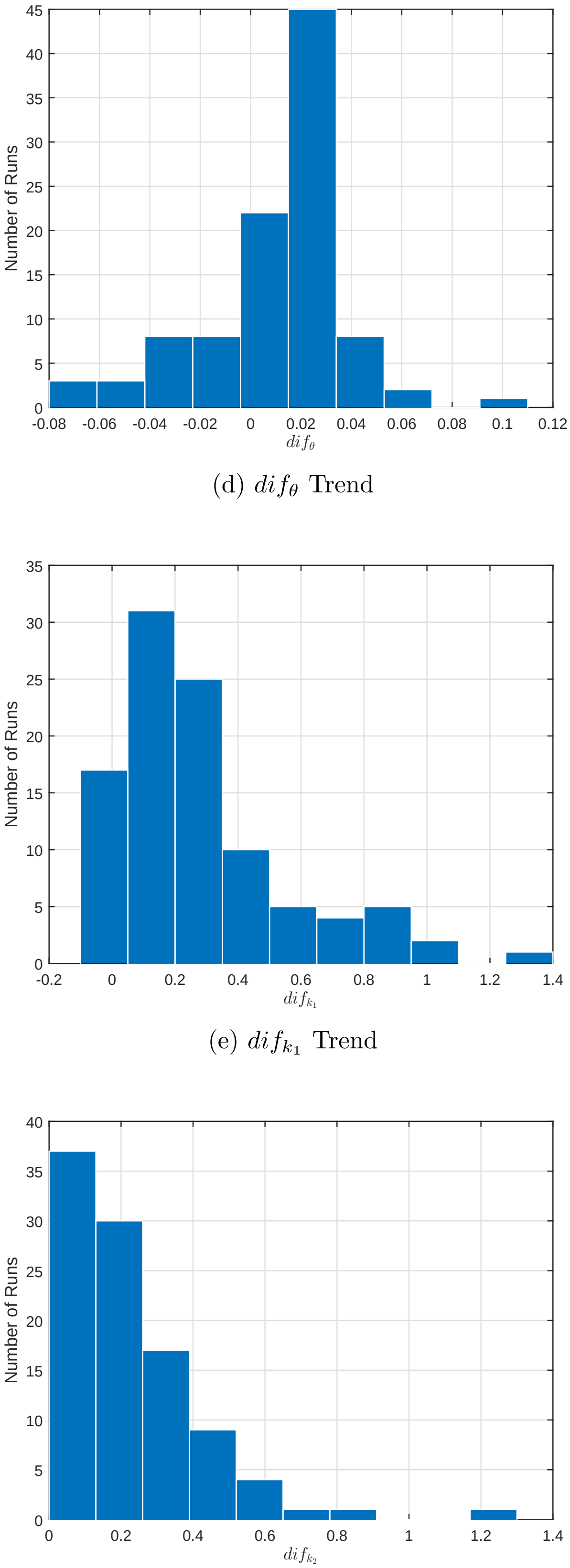
The results presented so far, are produced only with the IEKF. Histograms in fig. 6.7, describe the relative error, as defined in eq. (6.62).



(a) dif_ρ Histogram

(b) $dif_{\dot\rho}$ Trend

(c) dif_ω Trend



(d) dif_θ Trend

(e) dif_{k_1} Trend

(f) dif_{k_2} Trend

Figure 6.7. IEKF-EKF Differences

The histograms fig. 6.7 show that, in this case, the difference between EKF and IEKF is small. The EKF presents convergence in almost all the simulations with results comparable to IEKF. Due to the non-linear measurement equation of k_1 and k_2, larger differences are present for these two parameters. It is possible to conclude that, in this case and with this particular tuning, IEKF performs slightly better than EKF.

6.3.2 Case B

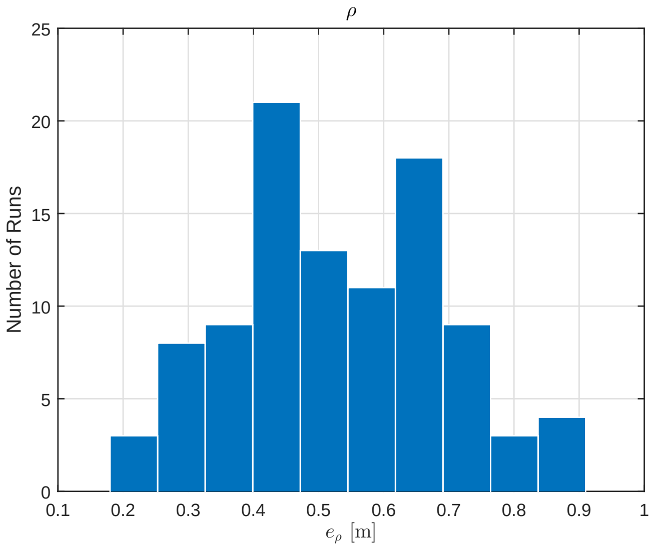
In the second case scenario, the filter is tested with the same initial conditions of the Case A:

- $\boldsymbol{\rho}_0 = [10,\ \ 60,\ \ 10]\ m$
- $\dot{\boldsymbol{\rho}}_0 = [0.01,\ \ -0.0225,\ \ -0.01]\ m/s$
- $\boldsymbol{\omega}_0 = [-0.1,\ \ -0.1,\ \ 0.034]\ deg/s$
- $\mathbf{q}_0 = [0,\ \ 0,\ \ 0,\ \ 1]$

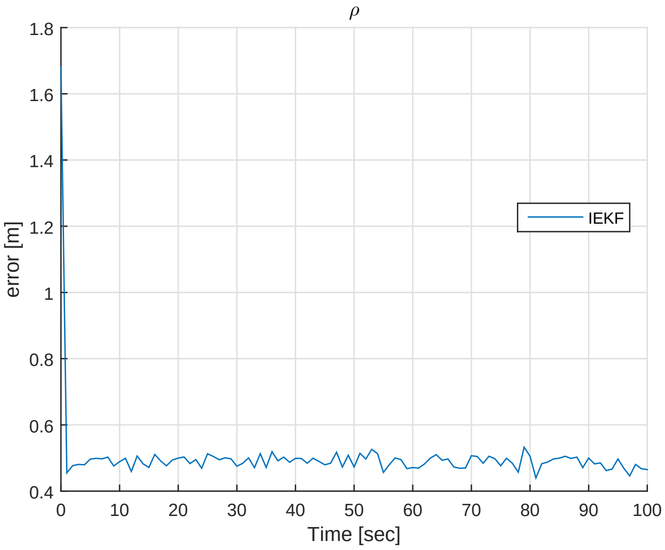
Different values for the covariance matrix are used. The components of the covariance matrix are assumed to be larger. This is done to represent a more realistic situation. However, with larger initial uncertainties, the filter can diverge more easily. Recalling eq. (6.61), the components of the covariance matrix $\mathbf{P}$ are:

- ${\sigma_\rho}^2 = [1,\ 1,\ 1]\ m^2$
- ${\sigma_{\dot{\rho}}}^2 = [1,\ 1,\ 1]\ m^2/s^2$
- ${\sigma_\omega}^2 = [1,\ 1,\ 1]\ deg^2/s^2$
- ${\sigma_q}^2 = [1,\ 1,\ 1,\ 1] \cdot 10^{-5}$
- ${\sigma_P}^2 = [1,\ 1,\ 1]\ m^2$
- ${\sigma_I}^2 = [1,\ 1]$

These values better describe the uncertainties in a real application. The probabilistic results for 100 runs are here presented.

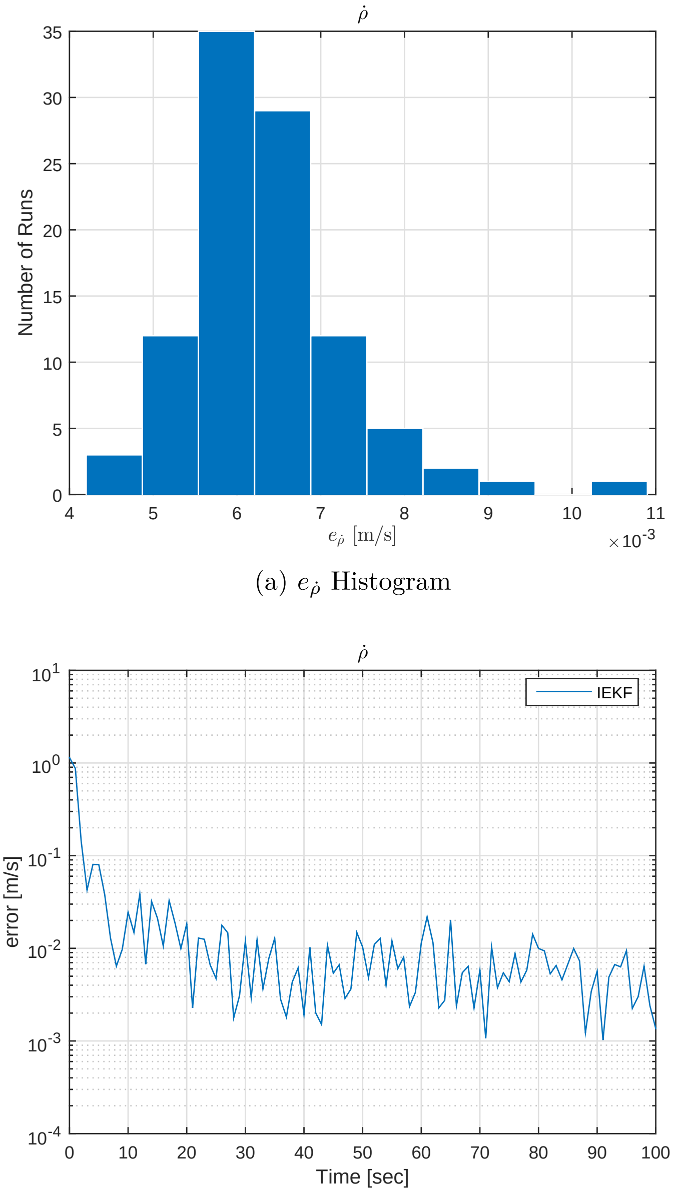


(a) e_ρ Histogram



(b) e_ρ Trend

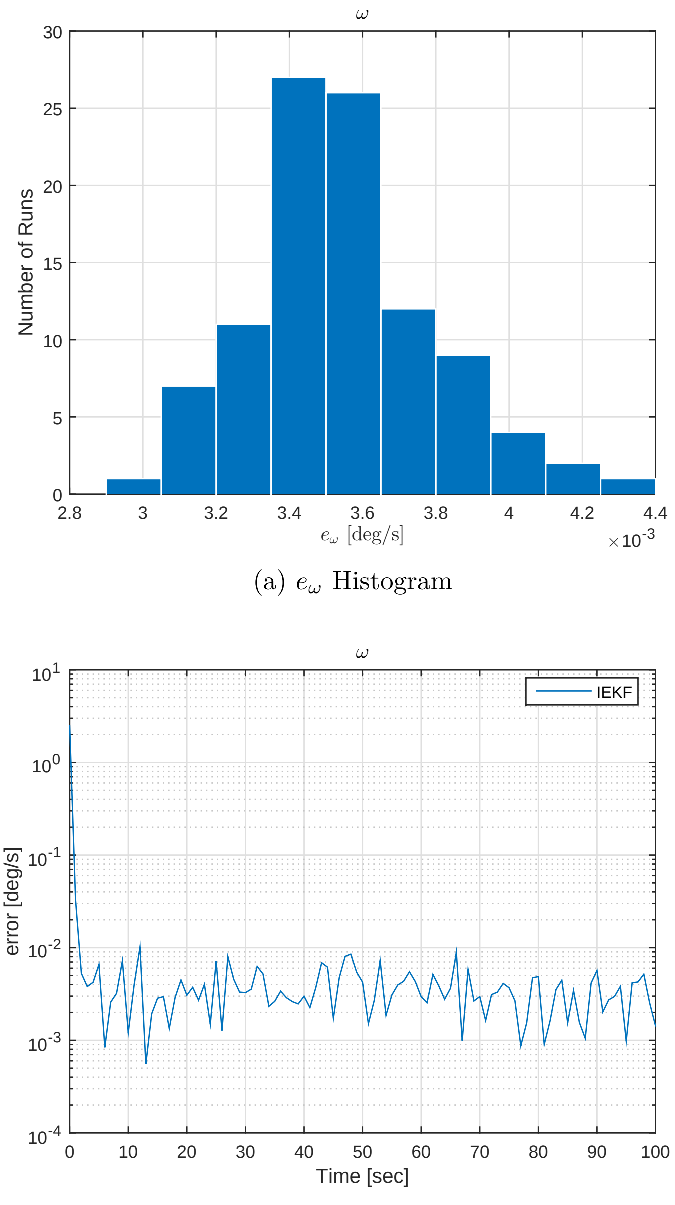
Figure 6.8. Relative Position Error



(a) $e_{\dot{\rho}}$ Histogram

(b) $e_{\dot{\rho}}$ Trend

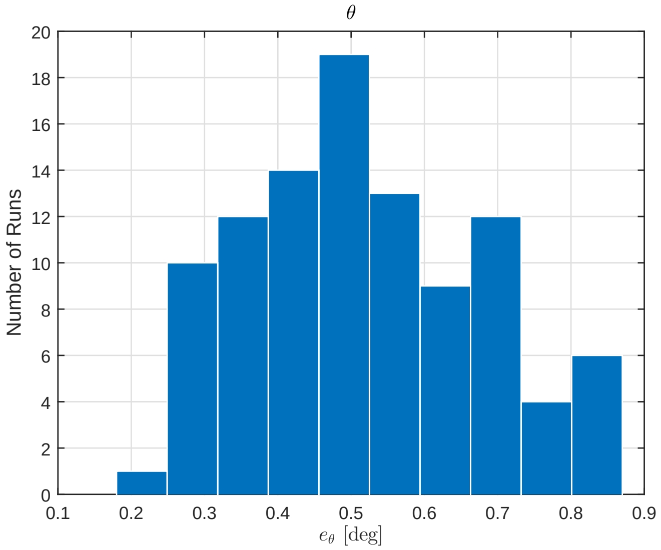
Figure 6.9. Relative Translational Velocity Error



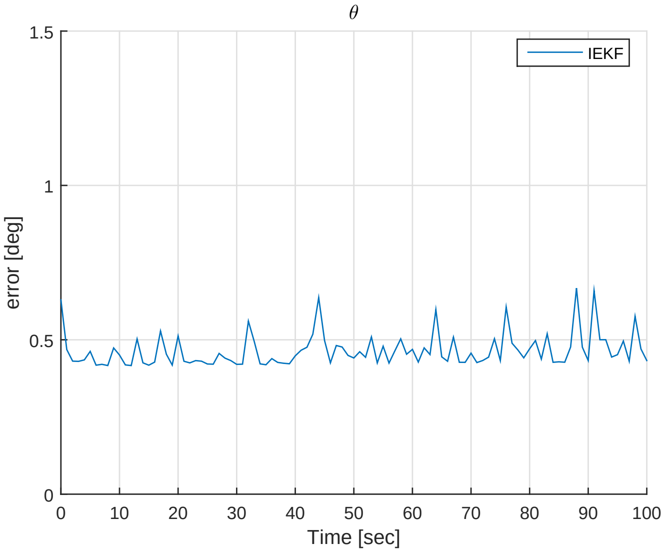
(a) e_ω Histogram

(b) e_ω Trend

Figure 6.10. Relative Angular Velocity Error

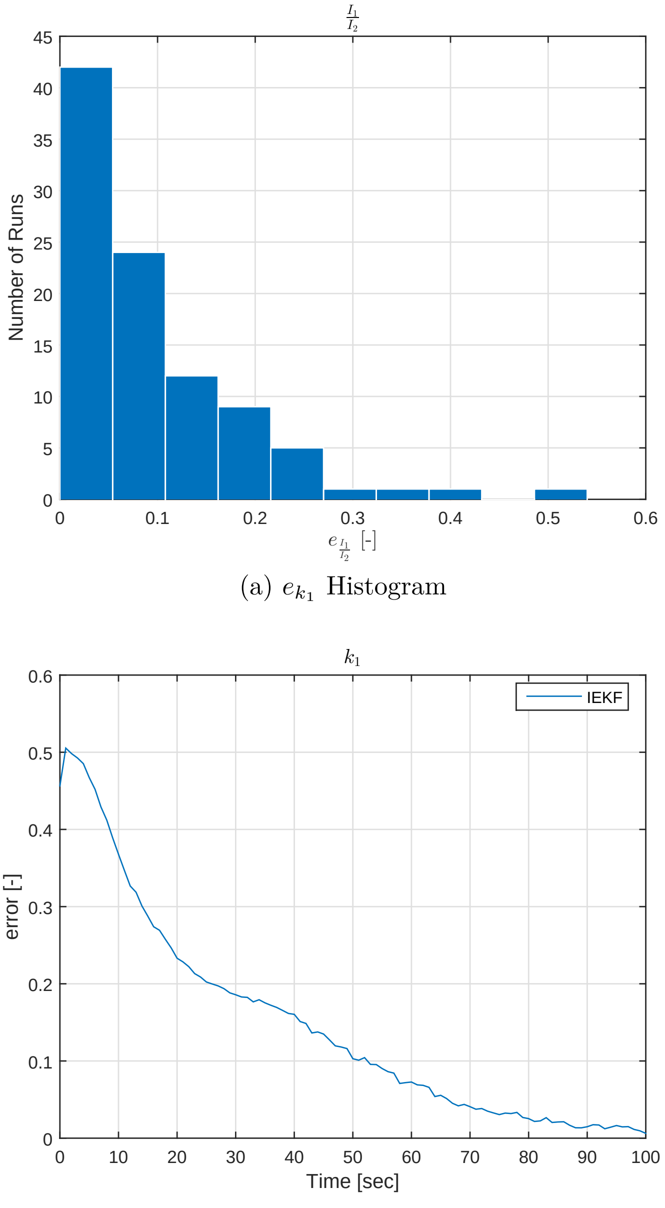


(a) e_θ Histogram



(b) e_θ Trend

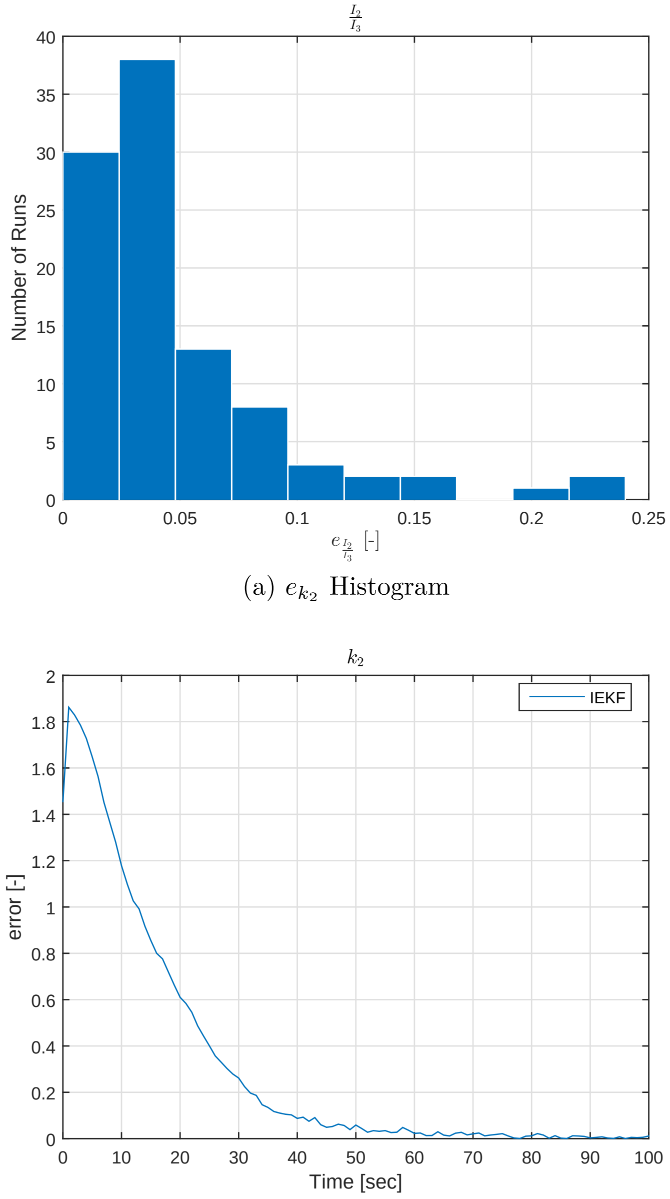
Figure 6.11. Relative Attitude Error



(a) e_{k_1} Histogram

(b) e_{k_1} Trend

Figure 6.12. Inertia Ratio Error



(a) e_{k_2} Histogram

(b) e_{k_2} Trend

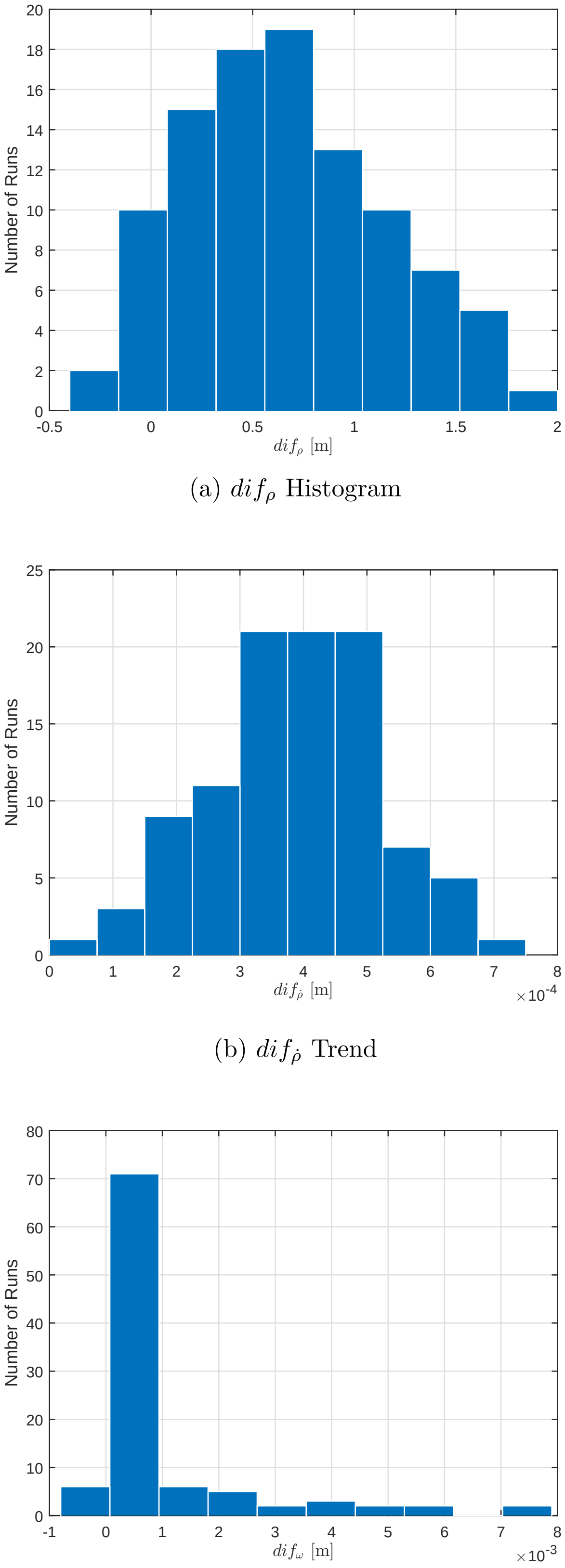
Figure 6.13. Inertia Ratio Error

The presented results still show robust convergence in all the analysed simulations. However, as expected, a small increase of the relative error of position, attitude and inertia parameters is present. The obtained error values are summarized in the following table for different percentiles.

Table 6.2. State Errors - Case B

Percentiles	$\rho\,[m]$	$\dot{\rho}\,[m/s]$	$\omega\,[deg/s]$	$\theta\,[deg]$	$k_1\,[-]$	$k_2\,[-]$
50	0.51	0.0062	0.0035	0.49	0.067	0.037
70	0.64	0.0067	0.0036	0.61	0.13	0.051
90	0.73	0.0073	0.0039	0.77	0.24	0.23
100	0.90	0.011	0.0043	0.87	0.53	0.23

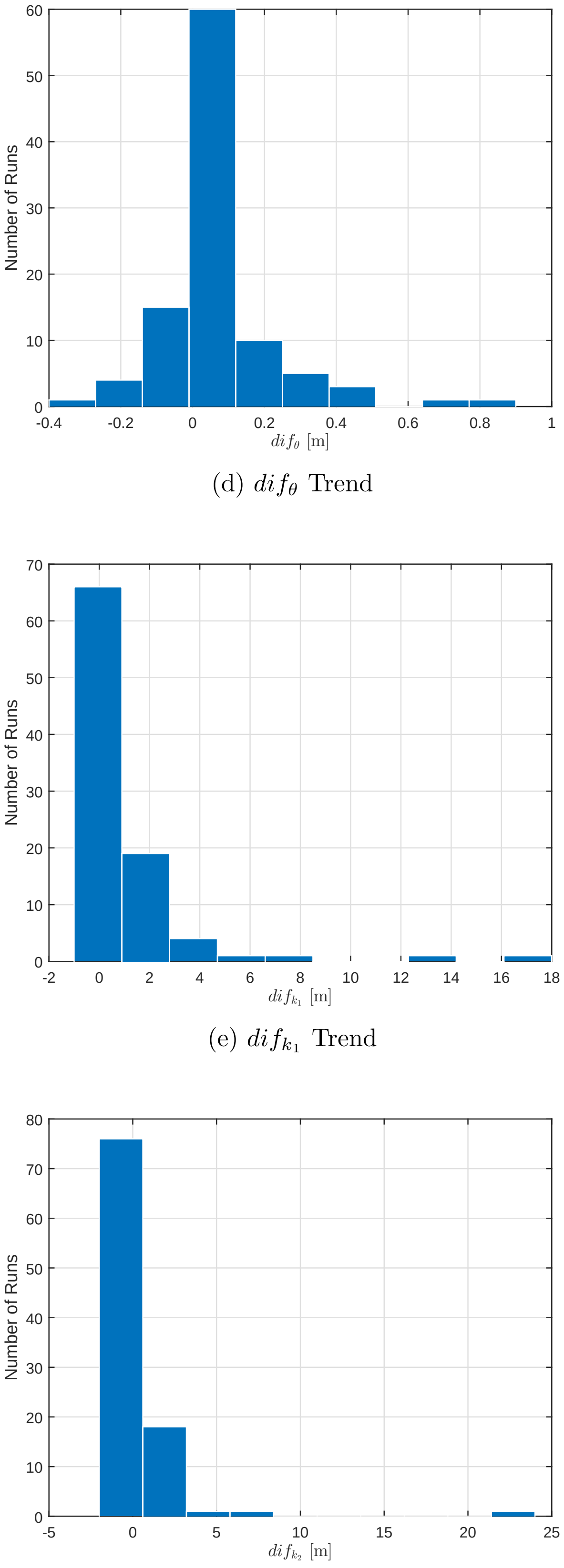
Also for this case, the difference between the error given by IEKF and EKF are compared. From fig. 6.14, the general trend shown in Case A, is confirmed. The IEKF performs slightly better than the EKF. In addition, in this case, in few simulations, a large difference is present. This means that the EKF is not stable as the IEKF and it tends to diverge with particular initial conditions and covariance values. The divergence is particularly large in the error of the inertia ratios. This is due to the fact that the pseudo measurement equation is non-linear. Therefore, consistently with the theory, the EKF poorly performs in this condition.



(a) dif_ρ Histogram

(b) $dif_{\dot\rho}$ Trend

(c) dif_ω Trend



(d) dif_θ Trend

(e) dif_{k_1} Trend

(f) dif_{k_2} Trend

Figure 6.14. IEKF-EKF Differences

6.3.3 Case C

In this third case, the initial conditions are the same of the previous cases:

- $\boldsymbol{\rho}_0 = [10,\ \ 60,\ \ 10]\ m$
- $\dot{\boldsymbol{\rho}}_0 = [0.01,\ \ -0.0225,\ \ -0.01]\ m/s$
- $\boldsymbol{\omega}_0 = [-0.1,\ \ -0.1,\ \ 0.034]\ deg/s$
- $\mathbf{q}_0 = [0,\ \ 0,\ \ 0,\ \ 1]$

High initial uncertainties are imposed on the position and translational velocity. Recalling eq. (6.61), the components of the covariance matrix **P** are:

- ${\sigma_\rho}^2 = [1,\ 1,\ 1] \cdot 10\ m^2$
- ${\sigma_{\dot\rho}}^2 = [1,\ 1,\ 1] \cdot 10\ m^2/s^2$
- ${\sigma_\omega}^2 = [1,\ 1,\ 1]\ deg^2/s^2$
- ${\sigma_q}^2 = [1,\ 1,\ 1,\ 1] \cdot 10^{-5}$
- ${\sigma_P}^2 = [1,\ 1,\ 1]\ m^2$
- ${\sigma_I}^2 = [1,\ 1]$

In this case, the trend of the errors and the differences between EKF and IEKF are similar to the previous cases. For this reason, only a summarizing error table is presented.

Table 6.3. State Errors - Case C

Percentiles	$\rho\,[m]$	$\dot\rho\,[m/s]$	$\omega\,[deg/s]$	$\theta\,[deg]$	$k_1\,[-]$	$k_2\,[-]$
50	0.55	0.0078	0.0035	0.6	0.064	0.034
70	0.6	0.0084	0.0037	0.75	0.1	0.045
90	0.74	0.0095	0.0038	0.89	0.22	0.078
100	0.91	0.019	0.0039	0.97	0.7	0.19

Table 6.3 shows how the filter can handle, in proper way, high values of uncertainties for position and translational velocity. Good convergence is shown in all the simulations.

6.3.4 Case D

In this fourth case, the initial conditions are different, in particular, a closer target is considered. So, the initial conditions are:

- $\boldsymbol{\rho}_0 = [2.7,\ \ 16.8,\ \ 2.5]\ m$

- $\dot{\boldsymbol{\rho}}_0 = [0.01, \ \ -0.0225, \ \ -0.01] \ m/s$
- $\boldsymbol{\omega}_0 = [-0.1, \ \ -0.1, \ \ 0.034] \ deg/s$
- $\mathbf{q}_0 = [0, \ \ 0, \ \ 0, \ \ 1]$

And the components of the covariance matrix **P** are:

- ${\sigma_\rho}^2 = [1, \ 1, \ 1] \ m^2$
- ${\sigma_{\dot{\rho}}}^2 = [1, \ 1, \ 1] \ m^2/s^2$
- ${\sigma_\omega}^2 = [1, \ 1, \ 1] \ deg^2/s^2$
- ${\sigma_q}^2 = [1, \ 1, \ 1, \ 1] \cdot 10^{-5}$
- ${\sigma_P}^2 = [1, \ 1, \ 1] \ m^2$
- ${\sigma_I}^2 = [1, \ 1]$

The results are very similar to what obtained before. Therefore, only histograms are presented.

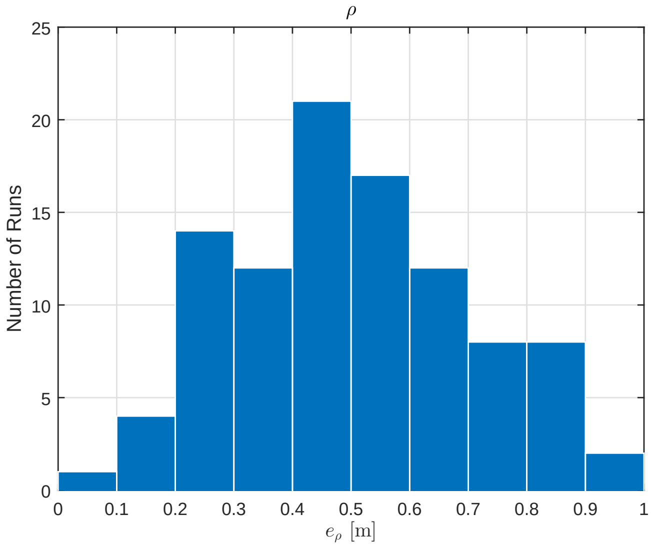


Figure 6.15. Relative Position Error

This result shows how, changing the initial conditions on the relative position, does not affect the filter convergence. For completeness, the table 6.4, with the simulations result is proposed.

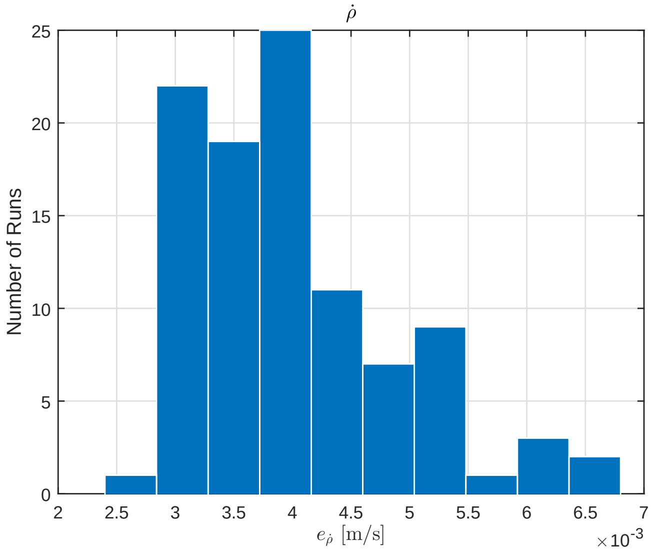


Figure 6.16. Relative Translational Velocity Error

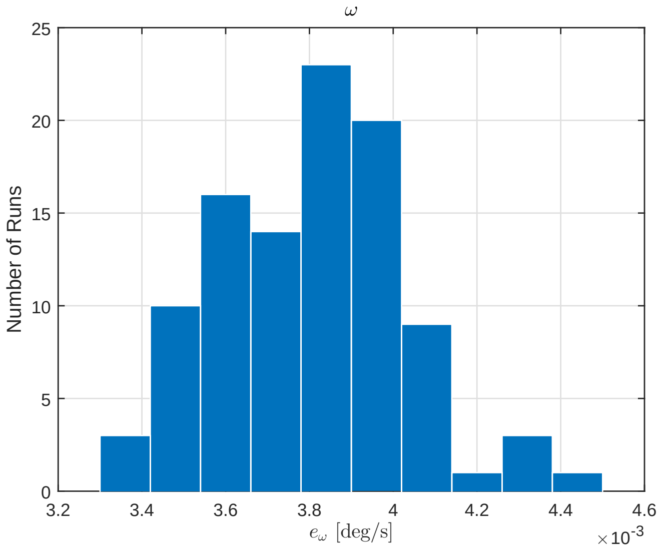


Figure 6.17. Relative Angular Velocity Error

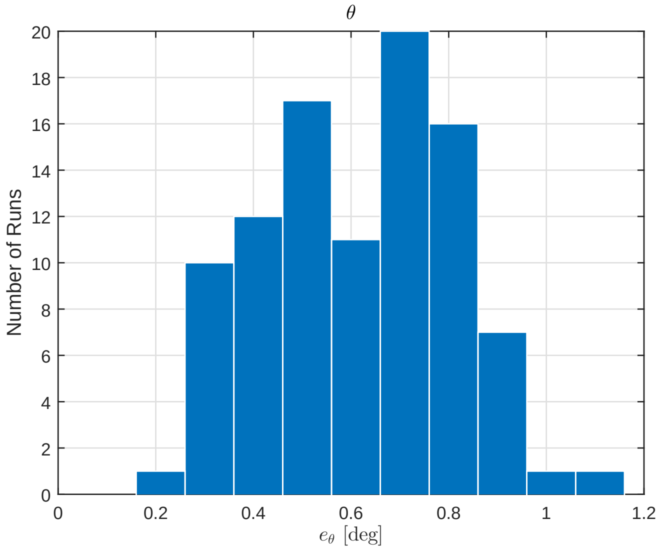


Figure 6.18. Relative Attitude Error

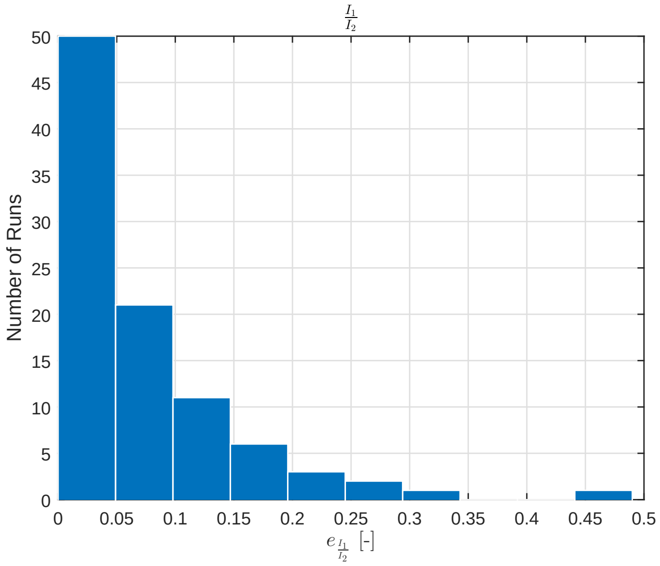


Figure 6.19. Inertia Ratio Error

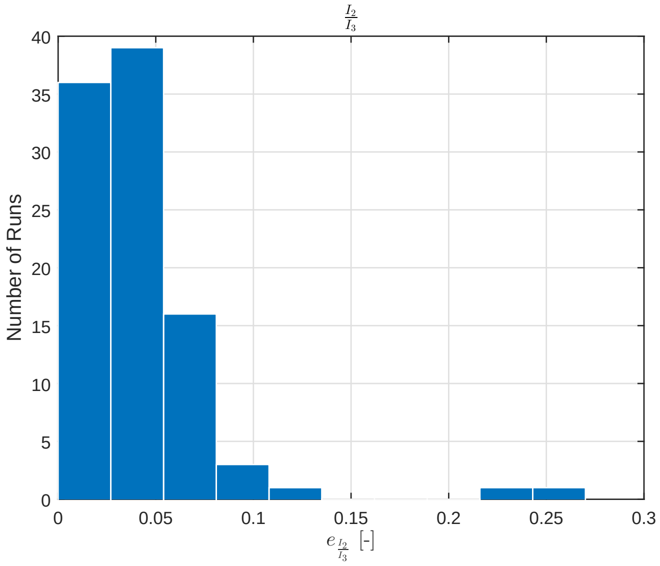


Figure 6.20. Inertia Ratio Error

Table 6.4. State Errors - Case D

Percentiles	$\rho\,[m]$	$\dot{\rho}\,[m/s]$	$\omega\,[deg/s]$	$\theta\,[deg]$	$k_1\,[-]$	$k_2\,[-]$
50	0.49	0.0039	0.0038	0.64	0.048	0.03
70	0.6	0.0042	0.0039	0.76	0.09	0.044
90	0.8	0.0052	0.004	0.88	0.22	0.077
100	0.96	0.0068	0.0044	1.1	0.48	0.27

6.3.5 Case E

In this case, it has been decided to remove the equality constraint. In this way, the filter performance is evaluated without a precise knowledge of the inertia properties. The initial conditions are the ones used for the Case B:

- $\boldsymbol{\rho}_0 = [10,\ \ 60,\ \ 10]\ m$
- $\dot{\boldsymbol{\rho}}_0 = [0.01,\ \ -0.0225,\ \ -0.01]\ m/s$
- $\boldsymbol{\omega}_0 = [-0.1,\ \ -0.1,\ \ 0.034]\ deg/s$
- $\mathbf{q}_0 = [0,\ \ 0,\ \ 0,\ \ 1]$

For the covariance matrix, a smaller value is assumed for the inertia ratios.

- ${\sigma_\rho}^2 = [1,\ \ 1,\ \ 1]\ m^2$
- ${\sigma_{\dot{\rho}}}^2 = [1,\ \ 1,\ \ 1]\ m^2/s^2$
- ${\sigma_\omega}^2 = [1,\ \ 1,\ \ 1]\ deg^2/s^2$
- ${\sigma_q}^2 = [1,\ \ 1,\ \ 1,\ \ 1] \cdot 10^{-5}$
- ${\sigma_P}^2 = [1,\ \ 1,\ \ 1]\ m^2$

- ${\sigma_I}^2 = [1,\ 1]/2$

With these assumptions, the following results are obtained.

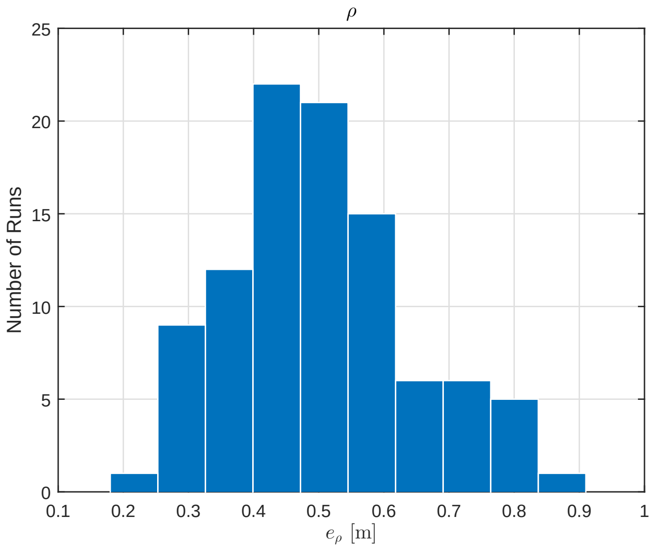


Figure 6.21. Relative Position Error

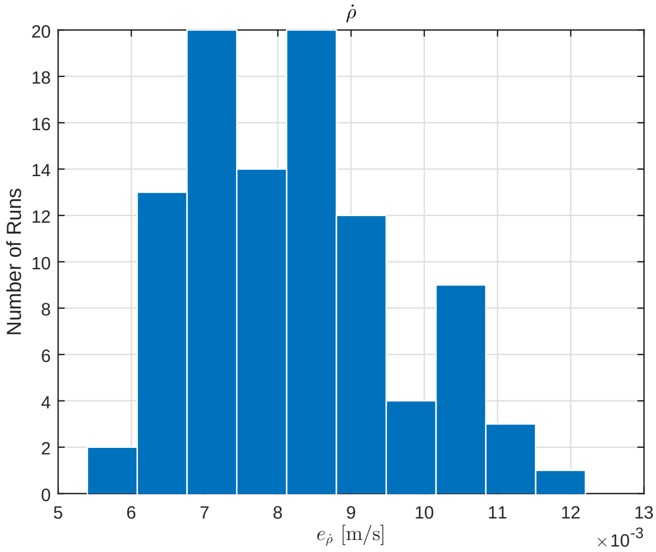


Figure 6.22. Relative Translational Velocity Error

The filter is robust and converges in almost all the runs. For the position and translational velocity error, the behaviour does not change. This is because the translational dynamics is not affected by the rotational one. The angular velocity and attitude has comparable results to the previous case. As already said, the filter can properly operate also with dynamics that do not describe in the proper way the process. Because of the measurements of the angular velocity, the behaviour, given

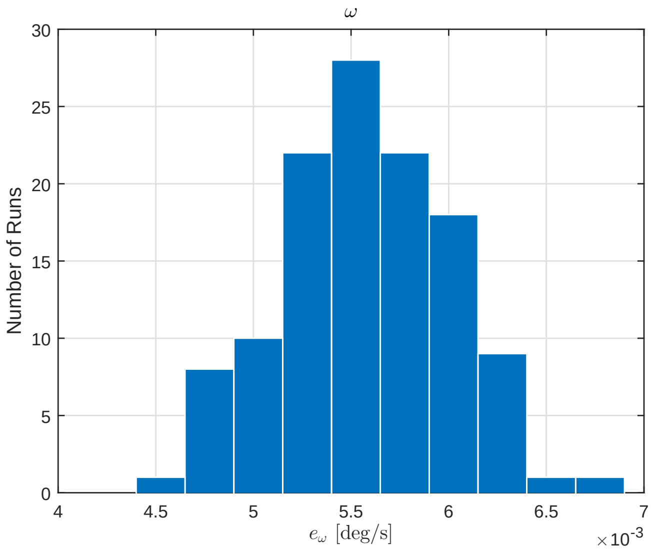


Figure 6.23. Relative Angular Velocity Error

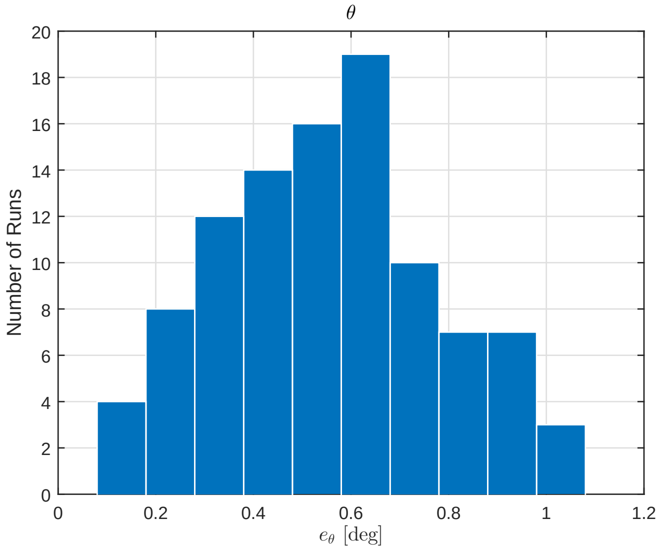


Figure 6.24. Relative Attitude Error

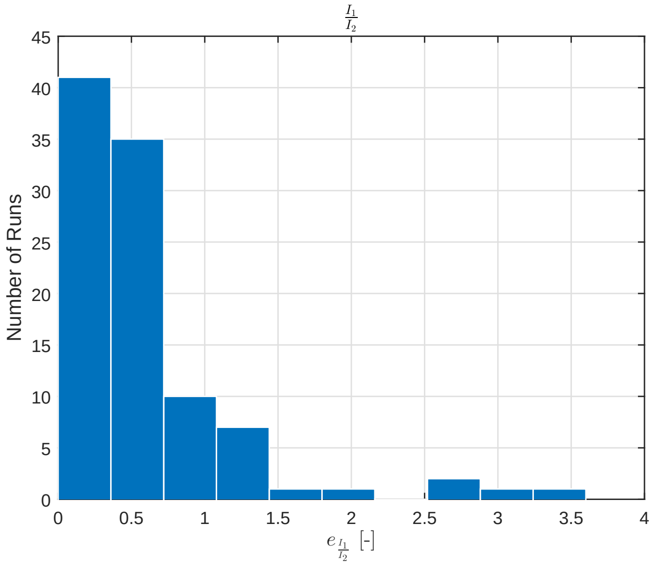


Figure 6.25. Inertia Ratio Error

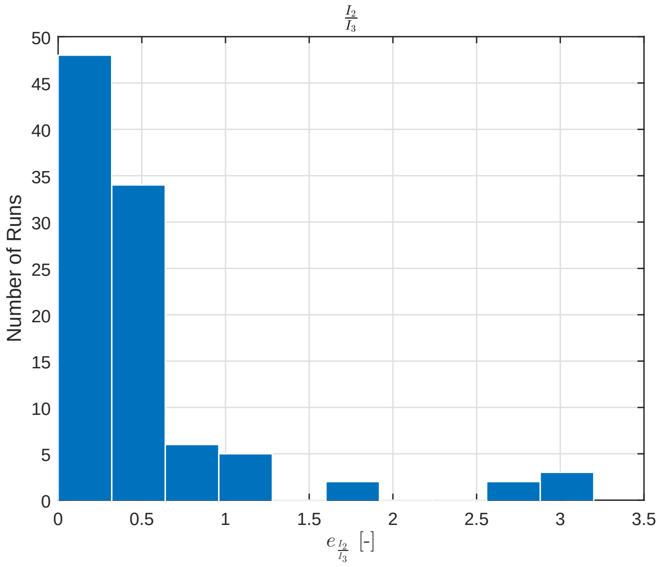


Figure 6.26. Inertia Ratio Error

by the wrong dynamical model, can be corrected. This is also possible because small values for the angular velocity are considered. In fact, in this case, the inertia matrix does not play a significant role in the rotational dynamics. From the presented results, the inertia ratios errors seem to converge to zero. However, this is only due to the fact that the initial covariance is small. In fact, looking at the trend of the error in fig. 6.27 and fig. 6.28, it is clear how the error tends to be constant. This means that the estimated inertia ratio remains constant and does not converge.

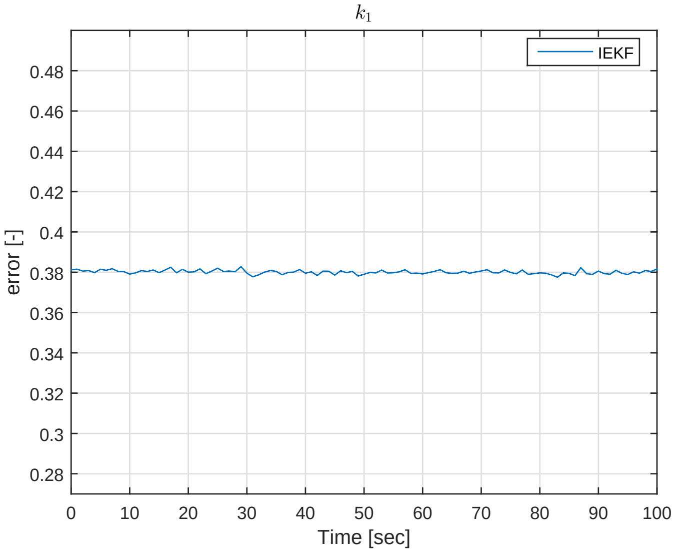


Figure 6.27. Inertia Ratio Error

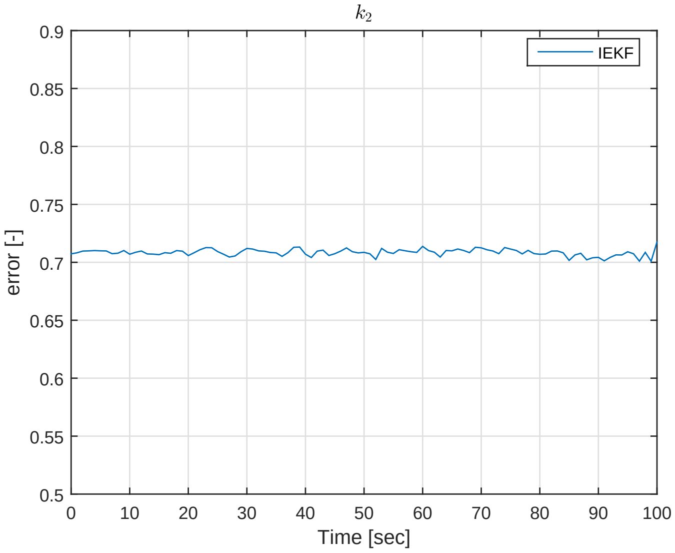


Figure 6.28. Inertia Ratio Error

In table 6.5, error results are summarized.

Table 6.5. State Errors - Case E

Percentiles	$\rho\,[m]$	$\dot{\rho}\,[m/s]$	$\omega\,[deg/s]$	$\theta\,[deg]$	$k_1\,[-]$	$k_2\,[-]$
50	0.51	0.0081	0.0058	0.59	0.33	0.35
70	0.6	0.009	0.0059	0.74	0.6	0.55
90	0.77	0.011	0.0063	0.88	1.4	1.6
100	0.96	0.013	0.0069	1.2	3.2	2.6

6.3.6 Case F

So far, only small values for the relative angular velocity have been considered. In this simulation, the performance of the filter without equality constraint is evaluated in a case with larger initial relative angular velocity. In particular:

- $\boldsymbol{\rho}_0 = [10,\ \ 60,\ \ 10]\ m$
- $\dot{\boldsymbol{\rho}}_0 = [0.01,\ \ -0.0225,\ \ -0.01]\ m/s$
- $\boldsymbol{\omega}_0 = [-1,\ \ -1,\ \ 0.934]\ deg/s$
- $\mathbf{q}_0 = [0,\ \ 0,\ \ 0,\ \ 1]$

The value of $\boldsymbol{\omega}_0$ is obtained increasing the value of $\boldsymbol{\omega}_T$. The covariance matrix is, as before:

- ${\sigma_\rho}^2 = [1,\ 1,\ 1]\ m^2$
- ${\sigma_{\dot{\rho}}}^2 = [1,\ 1,\ 1]\ m^2/s^2$
- ${\sigma_\omega}^2 = [1,\ 1,\ 1]\ deg^2/s^2$
- ${\sigma_q}^2 = [1,\ 1,\ 1,\ 1] \cdot 10^{-5}$
- ${\sigma_P}^2 = [1,\ 1,\ 1]\ m^2$
- ${\sigma_I}^2 = [1,\ 1]/2$

In this case, the following results are obtained:

Table 6.6. State Errors - Case F

Percentiles	$\rho\,[m]$	$\dot{\rho}\,[m/s]$	$\omega\,[deg/s]$	$\theta\,[deg]$	$k_1\,[-]$	$k_2\,[-]$
50	0.53	0.01	0.012	1.8	0.035	0.021
70	0.64	0.013	0.013	2	0.043	0.024
90	0.76	0.017	0.014	2.2	0.069	0.032
100	0.94	0.02	0.016	2.5	0.15	0.043

These are very interesting results. The estimation of the relative position and translational velocities is not affected by the change in the angular velocity since

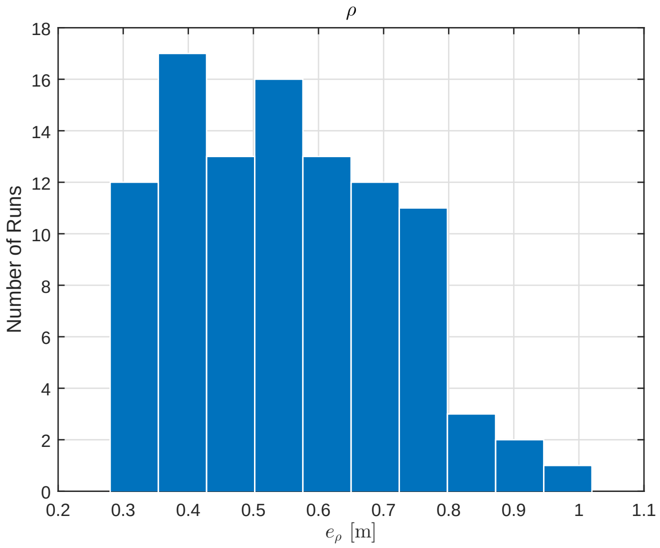


Figure 6.29. Relative Position Error

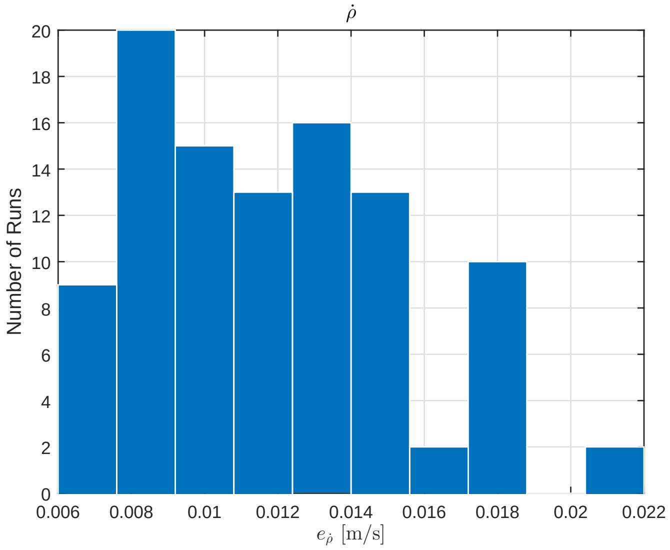


Figure 6.30. Relative Translational Velocity Error

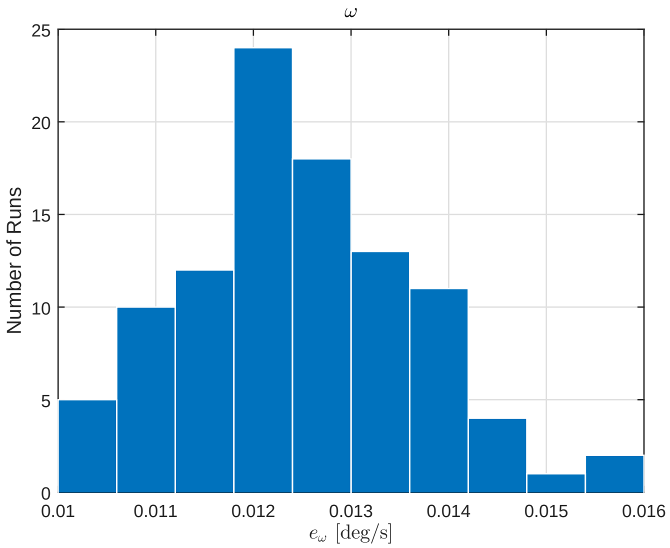


Figure 6.31. Relative Angular Velocity Error

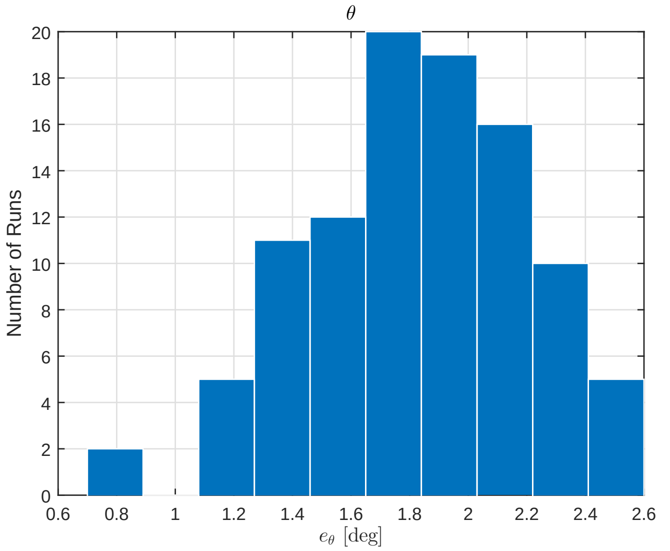


Figure 6.32. Relative Attitude Error

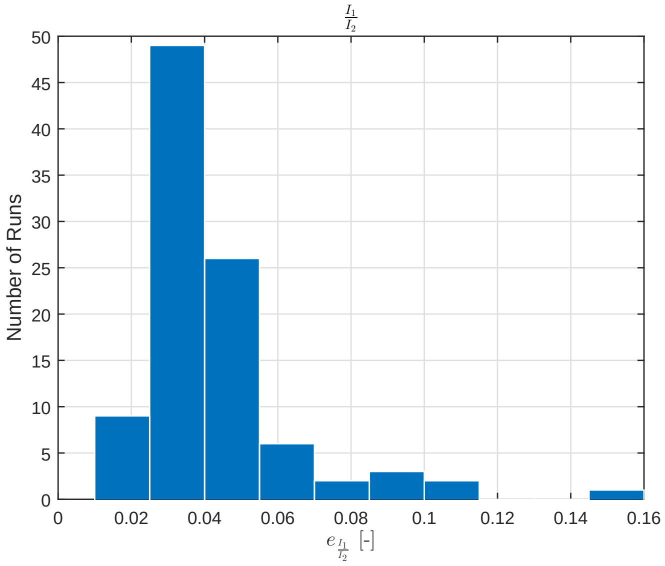


Figure 6.33. Inertia Ratio Error

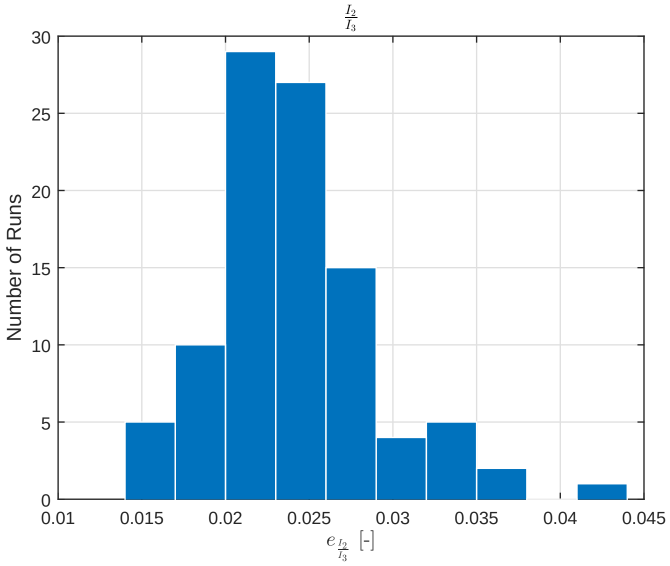


Figure 6.34. Inertia Ratio Error

the translation and the rotation dynamics are decoupled. The relative angular velocity and primarily the relative attitude are badly affected by this change. This is understandable because, in this case, the time step used for the simulation can be an issue for a good estimation of the state of a body, spinning with large angular velocity. Moreover, in this case, the error in the estimation of the inertia matrix affects more the dynamical model propagation. Therefore, the incorrect inertia ratios lead to a decay in the estimation performance for angular velocity and attitude. However, using a larger value for the target angular velocity implies better results in the inertia ratios estimation also without the equality constraint imposed with the pseudo measurement. Figure 6.35 and fig. 6.36 show the converging trend of the inertia ratios:

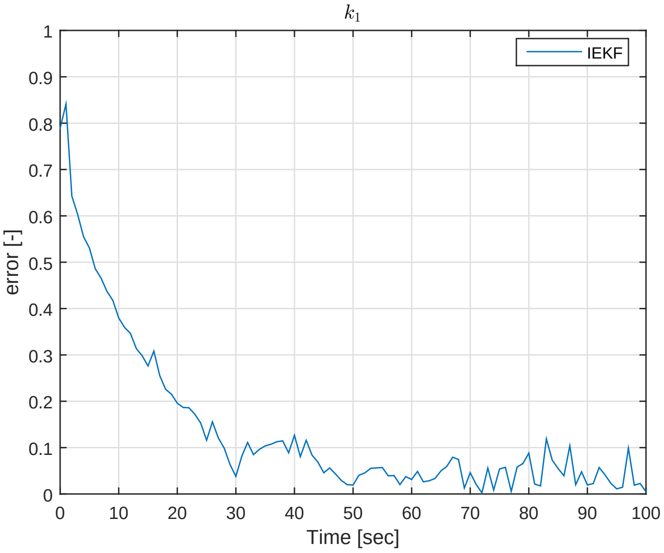


Figure 6.35. Inertia Ratio Error

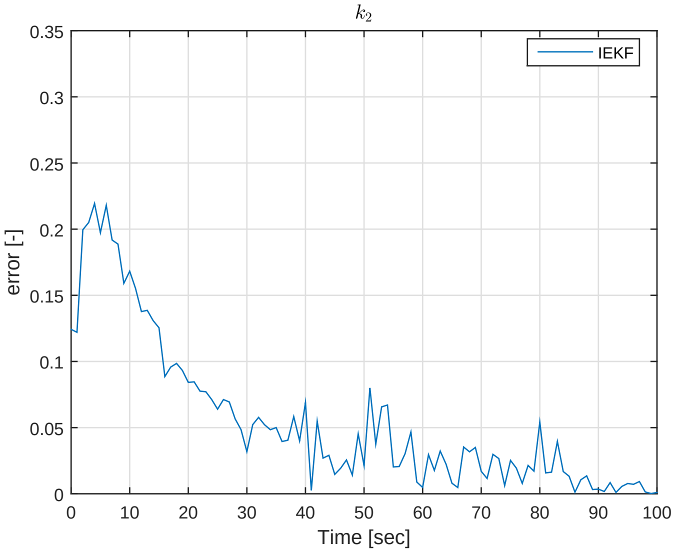


Figure 6.36. Inertia Ratio Error

This is justified by the fact that, with a larger angular velocity, the inertia matrix affects in a stronger way the rotational dynamics of the body. Hence, the dynamical model and the measurement equations of the angular velocity, force the inertia ratios to converge to a 'consistent' and exact value. This estimation can be obtained only in torque free motion conditions, parametrizing the inertia matrix in a proper way.

The presented results show how the algorithm, exploiting the equality constraint, can always estimate with satisfactory accuracy the complete relative state and the inertia components. Several simulations are presented to demonstrate the robustness of the algorithm with different covariance matrix values and initial conditions. It is also demonstrated how the inertia components, in the filtering process, can converge without the equality constraint but only with a sufficiently high value for the target angular velocity. In most of the cases, the presented algorithm shows better results with respect to similar works. Moreover, it is able to estimate the inertia components in a very efficient way. No previous work can do this with an estimation procedure.

6.3.7 Computational Analysis

In this section, an evaluation of the algorithm performance is presented. In this analysis, only the time of the filtering process is taken into account. In fact, the information extracted from the images is assumed to be known and only the observation model is used to process the available data. Hence, the time to extract and track the feature points is not taken into account. The algorithm is written in

MATLAB and is not optimized for an online application. It is run on a customized computer with an Intel Xeon E5-2609 2.5 Ghz 10Mb cache Ivy Bridge Processor and 16Gb DDR3 SDRAM. An extensive analysis was performed, running 200 simulations. Each simulation computes the estimated state for 100 seconds using the initial values and covariance matrix of the Case B. The computational time is therefore measured. The histogram in fig. 6.37 shows the time of each simulations.

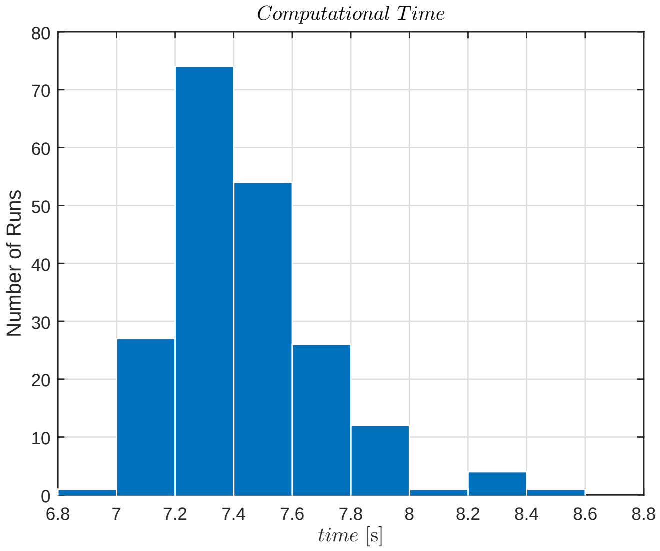


Figure 6.37. Computational Time

The presented results are good for a non optimized code, running on MATLAB. For a real application, a C++ implementation is preferable. In fact, in general, it offers better computational performance than MATLAB [81]. In table 6.7, maximum, average and minimum computational time are displayed.

Table 6.7. Computational Time

	Time [s]
Maximum	8.5
Average	7.4
Minimum	6.89

Figure 6.37 shows good computational performance for the implemented algorithm. As already said, this can only offer a rough idea of the computational time of the filtering process. A more detailed analysis, considering also the image processing, is necessary to have a realistic result.

Chapter 7

Inertia Matrix Reconstruction

In the previous chapter, the estimation process of the inertia ratios has been shown. However, the inertia parameters, with small relative angular velocity, are correctly recovered only if information about the angular acceleration of the target is provided. This implies to numerically derive the available measures of the angular velocity. They, as already said, are available from the optical flow. Numerical derivatives of a quantity that is usually noisy, can introduce instabilities and produce inaccurate results. In this chapter, on the other hand, a method to recover all the inertia components is described, without relying on any numerical method. A video and image process to recover mass properties is presented. This method is not computationally efficient and, in our application, relies on free and not optimized video/image processing software. The main idea is to collect a video or images of the observed body. These should be taken from different points of view, to reconstruct the whole body. From this set of images, a point cloud can be constructed according to video processing algorithms. Once a point cloud is available, a triangulate mesh can be built. The mesh gives us information about the geometry of the object. At this point, an assumption has to be done. In fact, knowing the geometry, the unknown density properties of the object do not allow a complete reconstruction of the mass properties of the body. However, generalizing the problem, the density can be assumed constant. This can be a very strong assumption but, many space objects can be approximated as bodies with constant density. In addition, density information is usually available. For example, often, for asteroids, the bulk density is reconstructed. This, in fact, can be estimated considering grain density of common meteorites or with different techniques. For space debris, the material of satellites and therefore their density is, in general, known, analysing pre-launch information. The next sections describe the reconstruction procedure, reviewing some basic algorithms. In addition, an experimental validation is performed to demonstrate the validity of this method also with complex geometries.

7.1 From Video to Images

The first step is to collect a video of the object. The most important aspect is to observe the entire surface of the body. If a navigation algorithm is implemented, the inspecting satellite can collect information from different points of view, in a given trajectory around the unknown object. Otherwise, the video must be collected exploiting the eventual rotation of the observed body. A video is nothing more than a series of still pictures mounted together. The number of picture per unit time of video is the frame rate. This must be setted in the correct way to improve the computational efficiency. In fact, as already said, this method is not computationally optimized and a large amount of data could be necessary to have good results. Setting, in a proper way, the frame rate could allow to store only the necessary amount of data. This is important in a real application. On the other hand, cameras are cheaper, most reliable and requires less power with respect to conventional 3D sensors.

7.2 From Images to Point Cloud

Once the set of images is available, this has to be processed and information has to be extracted. The method is not very different from what has been presented about stereovision algorithms. To obtain a point cloud from a series of images, it is necessary to detect feature in each frame, to match these with the same feature points seen in a different frame and to perform a bundle adjustment. The selected free software to obtain a point cloud from a set of images is called VisualSFM [82]. This is a software by Changchang Wu that performs a 3D reconstruction with a structure from motion algorithm. In particular, it uses a SIFT (Scale-invariant feature transform) [83] for the feature extraction and matching. This method works collecting interesting points for a set of reference images. These points allow to built a database that contains information about the possible detected feature points. Every time a new feature is detected, it is compared to the features present in the database to find a possible match based on Euclidian distance of their feature vectors. From the matched features, also a subset of keypoints can be determined to form a consistent cluster. This is done by implementing an Hough transform. The cluster is finally subjected to a linear least square solution of the parameters of the affine transformation, to verify the validity of the cluster. The affine transformation is a transformation for which collinearity and ratios of distance are preserved. It relates the motion of the model to the images. The visualSFM algorithm also exploits a bundle adjustment. This is a process to minimize the projection error selecting the proper 3D point positions and camera parameters. This is possible, solving a non-linear least squares problem to minimize the re-projection error. In particular, the bundle adjustment minimizes a cost function representing the total re-projection error with respect to all 3D points and camera parameters. The minimization is, in this software, performed with a Levenberg-Marquardt method. Without entering in the details of the formulation, this software implements a bundle adjustment that exploits the presence of multi-cores to perform a parallel computation. In this way, the computation is faster. The output of the software are a sparse point cloud and the position of the camera relative to each captured picture.

7.3 From Point Cloud to 3D Mesh

Having information of a sparse point cloud, a refined mesh can be computed. In this work, the free software *MeshLab* [84] is used. It is an advanced program to create and process a 3D mesh. It is developed by the italian ISTI-CNR research institute. MeshLab allows the inspection, cleaning and rendering of a variety of 3D formats. It is a new software but it is already widely used in the computer-vision research field. In our work, this software is used to create a refined mesh of the object from the known point cloud. This process is called surface reconstruction an it can be done in different ways. Ball pivoting, Marching Cubes and Poisson's Reconstruction are the available algorithms in MeshLab. Our surface reconstruction is computed with Poisson's method. This is a powerful approach introduced by Kazhdan [85]. The main idea is to compute a 3D indicator function χ that is 0 outside the model and 1 inside. Knowing this, an appropriate isosurface from the reconstructed surface can be extracted. To define the right indicator function, it is possible to exploit an integral relationship between the points and the wanted function. In particular, the normal of the points can be computed and, in this way, an information about the 'orientation' of the points is known. This oriented point samples can be viewed as samples of a gradient of the indicator function. This gradient will be always zero, except in the points close to the surface. Summarizing, the oriented points will identify a vector field $\vec{V}$. The indicator function χ is identified as the scalar function whose gradient best approximates the vector field $\vec{V}$. From a mathematical point of view, this is equivalent to solve

$$min_{\chi}||\nabla\chi - \vec{V}|| \tag{7.1}$$

Applying the divergence operator, this problem can be viewed as a standard Poisson problem:

$$\triangle\,\chi = \nabla \cdot \nabla\chi = \nabla\vec{V} \tag{7.2}$$

This means computing the scalar function χ whose Laplacian is equal to the divergence of the vector field $\vec{V}$.

Using this algorithm, MeshLab provides good results and the output is a refined 3D mesh. To compute geometrical properties, the 3D mesh surface must be closed and manifold. Importing the point cloud obtained with VisualSFM, a 3D mesh can be generated. This has to be refined to satisfy the surface constraints described before. Finally, a 3D mesh in .obj format is obtained.

7.4 From 3D Mesh to Inertia Properties

The final step is to recover mass properties from the geometric information of the body. In this work, a method to compute mass properties from a triangulate mesh is used. The 3D mesh is imported in MATLAB, then an algorithm that exploits the divergence theorem to compute the rigid body parameters, is used. In this section, the main steps of this algorithm are presented.

7.4.1 Inertia Parameters From Triangular Surface Mesh

Assume Ω being a region of space occupied by a rigid body with density ρ. $\rho(x, y, z)$ is in general a scalar function depending on the position (x, y, z) that describes the material density of the object in that precise point. A generalized quantity can be introduced. It is called 3D moment and it is defined as:

$$m_{p,q,r}(\Omega) = \int_{\Omega} \rho(x, y, z)\, x^p y^q z^r \; d\Omega \tag{7.3}$$

Assuming to have a body with constant density, the expression is reduced to:

$$m_{p,q,r}(\Omega) = \int_{\Omega} x^p y^q z^r \; d\Omega \tag{7.4}$$

Exploiting the divergence theorem, the integration domain can be changed. In fact, knowing that:

$$\int_{\Omega} (\vec{\nabla} \cdot \vec{F})\, d\Omega = \oint \vec{F} \cdot \vec{n} dS \tag{7.5}$$

the 3D moment can be integrated on the surface enclosing Ω. Comparing eq. (7.4) to eq. (7.5), the following relation holds

$$\vec{\nabla} \cdot \vec{F} = x^p y^q z^r \tag{7.6}$$

and from the definition of the Nabla operator

$$\frac{\partial F^1_{p,q,r}}{\partial x} + \frac{\partial F^2_{p,q,r}}{\partial y} + \frac{\partial F^3_{p,q,r}}{\partial z} = x^p y^q z^r \tag{7.7}$$

Moreover, the surface S can be seen as the sum of N different small triangles T_i (the triangles corresponding to the 3D mesh). Therefore,

$$m_{p,q,r}(S) = \sum_{i=1}^{N} \int_{T_i} \vec{F}_{p,q,r} \cdot \vec{n}_i dS = \sum_{i=1}^{N} m_{p,q,r}(T_i) \tag{7.8}$$

Since some information of the 3D mesh are the positions of vertices and faces, this expression can be derived in baricentric coordinates. For simplicity, we avoid this calculation. From the mesh, it is now possible to compute this general quantity called 3D moment. From the knowledge of this value, all the main parameters can be recovered. In fact, knowing that the volume occupied by the region Ω is given by:

$$V = \int_{\Omega} d\Omega \tag{7.9}$$

it can be computed utilizing the 3D moment formulation. In fact, from the definition of 3D moment, it is automatically equal to

$$V = m_{0,0,0} \tag{7.10}$$

Where $m_{0,0,0}$ is called zero moment. Knowing the volume, the position of the centroid can be easily recovered.

$$x_c = \frac{\int_{\Omega} x d\Omega}{\int_{\Omega} d\Omega} \tag{7.11}$$

$$y_c = \frac{\int_{\Omega} y d\Omega}{\int_{\Omega} d\Omega} \tag{7.12}$$

$$z_c = \frac{\int_\Omega z d\Omega}{\int_\Omega d\Omega} \tag{7.13}$$

And, in terms of the 3D moment:

$$x_c = \frac{m_{1,0,0}}{m_{0,0,0}} \tag{7.14}$$

$$y_c = \frac{m_{0,1,0}}{m_{0,0,0}} \tag{7.15}$$

$$z_c = \frac{m_{0,0,1}}{m_{0,0,0}} \tag{7.16}$$

At this point, the product and moments of inertia of a rigid body in an arbitrary frame are defined as:

$$I_{xx} = \int_\Omega y^2 + z^2 \, d\Omega = m_{0,2,0} + m_{0,0,2} \tag{7.17}$$

$$I_{yy} = \int_\Omega x^2 + z^2 \, d\Omega = m_{2,0,0} + m_{0,0,2} \tag{7.18}$$

$$I_{zz} = \int_\Omega x^2 + y^2 \, d\Omega = m_{2,0,0} + m_{0,2,0} \tag{7.19}$$

$$I_{xy} = I_{yx} = \int_\Omega yx \, d\Omega = m_{1,1,0} \tag{7.20}$$

$$I_{xz} = I_{zx} = \int_\Omega zx \, d\Omega = m_{1,0,1} \tag{7.21}$$

$$I_{yz} = I_{zy} = \int_\Omega zy \, d\Omega = m_{0,1,1} \tag{7.22}$$

However, by convention, the inertia tensor has to be defined with respect to a frame that has its origin located in the object's centroid. In this way, knowing the coordinates in this frame (x', y', z'), then:

$$x = x' + x_c \tag{7.23}$$

$$y = y' + y_c \tag{7.24}$$

$$z = z' + z_c \tag{7.25}$$

and the inertia moments and products with respect to this frame are expressed by:

$$I_{xx_c} = I_{xx} - ({y_c}^2 + {z_c}^2)\, m_{0,0,0} \tag{7.26}$$

$$I_{yy_c} = I_{yy} - ({x_c}^2 + {z_c}^2)\, m_{0,0,0} \tag{7.27}$$

$$I_{zz_c} = I_{zz} - ({x_c}^2 + {y_c}^2)\, m_{0,0,0} \tag{7.28}$$

$$I_{xy_c} = I_{xy} - x_c y_c\, m_{0,0,0} \tag{7.29}$$

$$I_{xz_c} = I_{xz} - x_c z_c\, m_{0,0,0} \tag{7.30}$$

$$I_{yz_c} = I_{yz} - y_c z_c\, m_{0,0,0} \tag{7.31}$$

Finally substituting eqs. (7.14) to (7.16) and eqs. (7.17) to (7.22) into eqs. (7.26) to (7.31), the final expression of the inertia tensor, function of the 3D moment is found.

$$I_{xx_c} = m_{0,2,0} + m_{0,0,2} - \frac{{m^2}_{0,1,0} + {m^2}_{0,0,1}}{m_{0,0,0}} \tag{7.32}$$

$$I_{yy_c} = m_{2,0,0} + m_{0,0,2} - \frac{{m^2}_{1,0,0} + {m^2}_{0,0,1}}{m_{0,0,0}} \tag{7.33}$$

$$I_{zz_c} = m_{2,0,0} + m_{0,2,0} - \frac{{m^2}_{1,0,0} + {m^2}_{0,1,0}}{m_{0,0,0}} \tag{7.34}$$

$$I_{xy_c} = m_{1,1,0} - \frac{m_{1,0,0} m_{0,1,0}}{m_{0,0,0}} \tag{7.35}$$

$$I_{xz_c} = m_{1,0,1} - \frac{m_{1,0,0} m_{0,0,1}}{m_{0,0,0}} \tag{7.36}$$

$$I_{yz_c} = m_{0,1,1} - \frac{m_{0,1,0} m_{0,0,1}}{m_{0,0,0}} \tag{7.37}$$

This demonstrates how it is possible to obtain inertia properties of a body, knowing its 3D mesh. As already said, this method is implemented in MATLAB and the inertia properties are recovered.

7.4.2 Constant Density Assumption

In the previous formulation, constant density is assumed. In this section, an analysis on the validity of this assumption is provided. In our work, we deal with unknown space objects. Reducing the generality of this assumption, only asteroids and space debris can be considered. For asteroids, their composition is not exactly known and it usually varies with the dimension of the asteroid. They are, in fact, considered as minor planets and the larger ones are usually planetoids. This means that, in general, larger asteroids present an heavy metal core, a mantle and a crust. It is understandable that in this case, the assumption of constant density is not realistic. However, many asteroids, commonly the smaller ones, appear to be aggregations of debris held together by gravity. In these cases, asteroids can be seen as homogeneous solids with uniform composition. Therefore, constant density assumption can be realistic. Moreover, different methods to estimate the density of an asteroid exist. A brief review of the most important ones is here discussed. A direct measurement of bulk density is possible knowing the values of the mass and volume of the asteroid. However, information about mass and volume of an asteroids is not easy to recover. In particular, to measure the mass of an asteroid, an analysis of its gravitational field influence on other objects has to be done. This is performed, studying the orbit defection during close encounters with other objects, or with a spacecraft flyby, sending radio signal towards the asteroids and measuring the Doppler shift to have information on its trajectory deflection. In addition, for asteroids with satellites, their mass can be estimated analysing optical and radar images to measure the satellite orbit. From this discussion, it is clear that, for small bodies, the mass estimation can be very hard and lead to very imprecise results. Volume estimation is usually done by knowing the diameter of the body. This can be computed through measurement of the absolute magnitude, or with radiometry. The volume estimation is, in general, available for a larger number of asteroids but it is also less accurate. Therefore, the density accuracy, is mainly limited by the volume measurement when a good mass estimate is present. In order to overcome the problems in knowing the mass of smaller bodies, other indirect methods for density estimation are available. In particular, from photometric observations of eclipses in asteroids binary systems, some constraints are derived. They are dimensionless as they represent ratios of physical quantities. From those parameters, estimation of the density can be recovered. This data has been collected and different catalogues with density information are available. For what concern space debris, more details about geometry, mass, volume and consequently density are available. In fact, pre-launch knowledge about material density of a satellite is accessible. For small size debris, breakup experiments on ground can be performed and the composition of residuals can be analysed. However, for large space debris, it is not always possible to assume to have a body with constant density. Depending on the case, this method can lead to considerable errors but it could be used as first approximation of the mass properties of the body.

7.5 Experimental Validation

In this section, experiments are conducted to evaluate the efficiency of the video processing method. Two different experiments are assessed, one with a body with a simple geometry and another one using an object with a complex geometry.

7.5.1 Experimental Setup

A very simple experimental setup is used. The video are collected using a Sony HandyCam HDR-CX110 as in fig. 7.1.

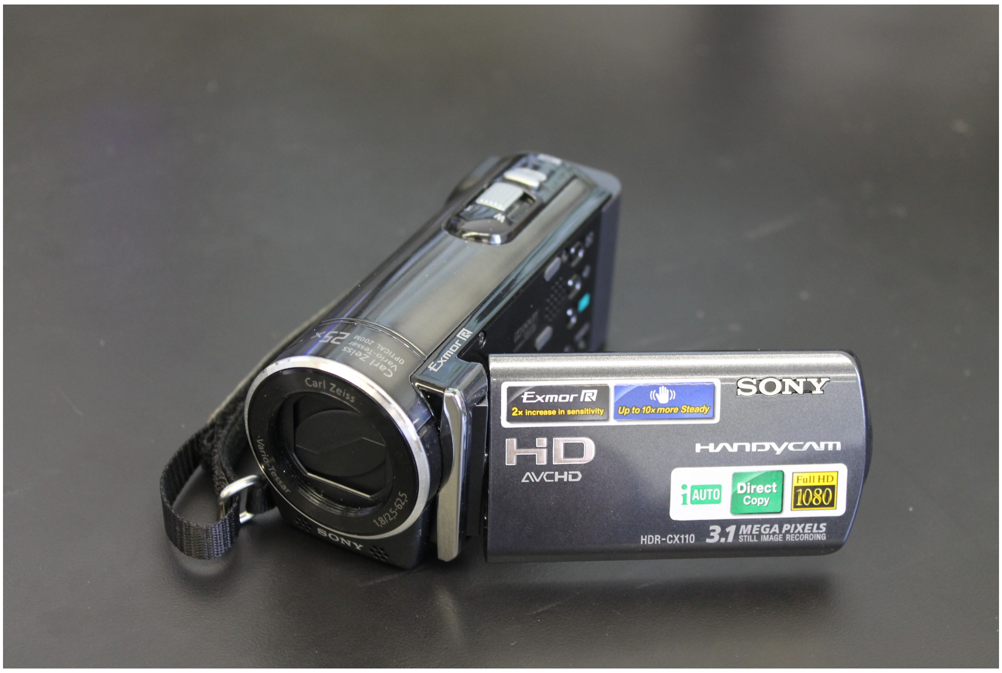


Figure 7.1. Sony HandiyCam HDR-CX110

It is an HD digital camcorder with a 3.1-mega pixels CMOS sensor. It has 35mm Carl Zeiss Lens with 25x optical zoom. The HD video are collected in AVCHD format with frame size per pixels equal to 1920x1080. The objects that are used as samples are a 3 degrees of freedom simulator with two air bearings fig. 7.2 and a high pressure tank fig. 7.3

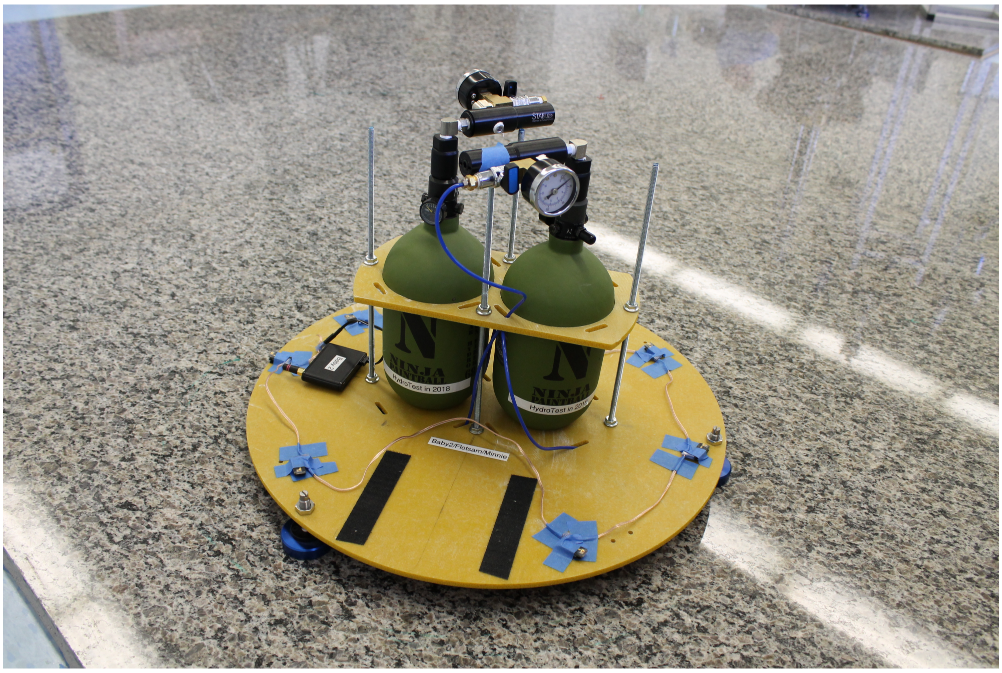


Figure 7.2. 3-DOF Simulator

The video is taken moving the camera in circle around the fixed object. A video capture lasts almost 15-20 seconds. Then, from 100 to 150 frames are extracted, depending on the video length. These images are imported in VisualSFM, the point cloud is generated according to the algorithms already discussed. Then, with MeshLab and MATLAB, the mass properties can be recovered. The computer used for the video processing is a customized one, running Windows 8. It mounts an Intel Xeon E5-2609 2.5 Ghz 10Mb cache Ivy Bridge Processor, 16Gb DDR3 SDRAM and a PNY Quadro K620 2Gb Video Card.

7.5.2 Case A - Simple Shape

The first experimental validation is performed taking a video of an object with a simple shape. In particular, an high pressure tank (fig. 7.3) is used. It has a simple cylindrical geometry with a small gauge and valves. These elements could influence the accuracy of the mesh. From the video, frames are extracted and processed with VisualSFM to create a point cloud. The result is shown in fig. 7.4.

From the obtained point cloud, a 3D mesh is created with MeshLab. The result obtained from a Poisson's reconstruction is then exported in MATLAB (fig. 7.5).

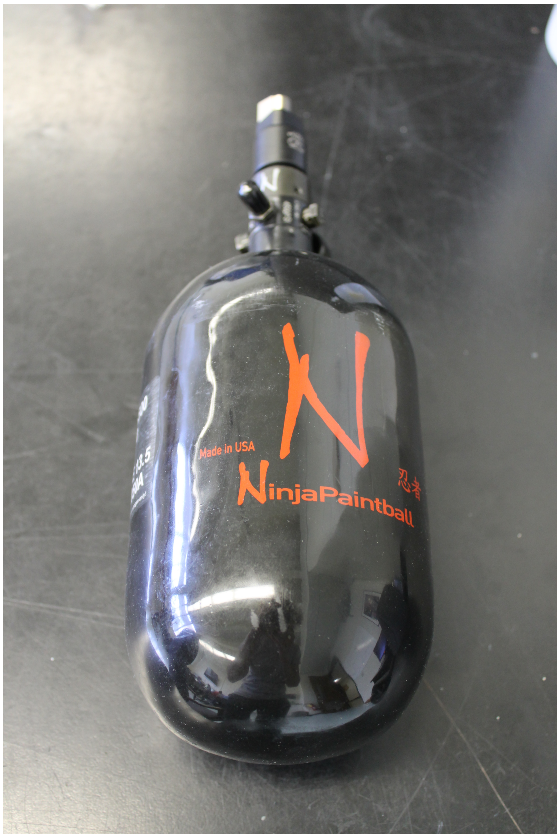


Figure 7.3. High Pressure Tank

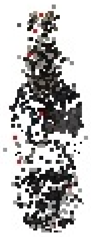


Figure 7.4. Tank Point Cloud

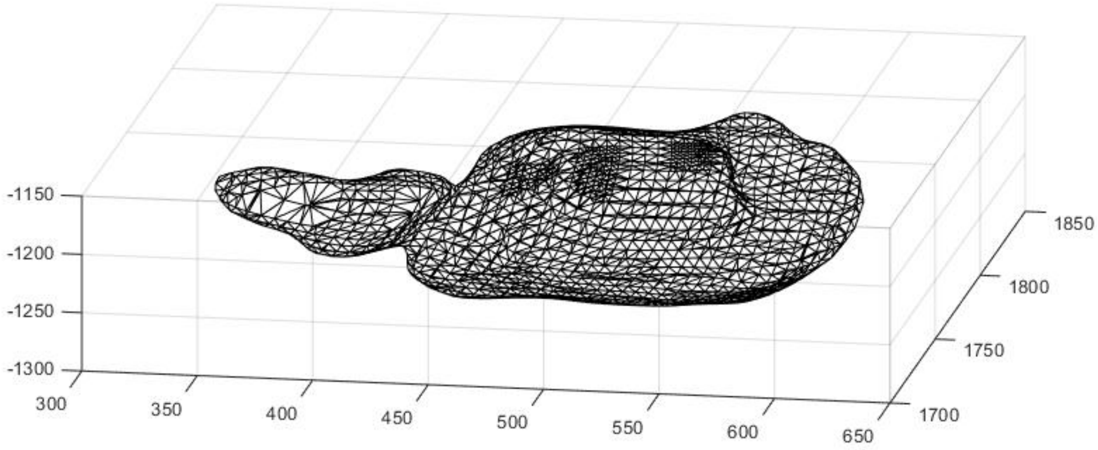


Figure 7.5. Tank 3D Mesh

The result is not perfect but it reproduces, in an acceptable way, the mass distribution of the body. In MATLAB, the mass properties are then computed. Assuming a constant density of $1\frac{g}{cm^3}$, the matrix of inertia expressed in principal axis is:

$$I = \begin{bmatrix} 67.31 & 0 & 0 \\ 0 & 79.51 & 0 \\ 0 & 0 & 27.57 \end{bmatrix} kg\, cm^2 \tag{7.38}$$

The 'real' value is computed using a CAD model of the tank. In SolidWorks, the inertia tensor is computed, assuming the same constant density of $1\frac{g}{cm^3}$. The result is:

$$I_{SW} = \begin{bmatrix} 85.3 & 0 & 0 \\ 0 & 85.34 & 0 \\ 0 & 0 & 26.67 \end{bmatrix} kg\, cm^2 \tag{7.39}$$

The errors for the single components are:

$$e_{I_{xx}} = 10.5\% \tag{7.40}$$

$$e_{I_{yy}} = 7.3\% \tag{7.41}$$

$$e_{I_{zz}} = 3.4\% \tag{7.42}$$

These results demonstrate that it is possible to estimate in a quite accurate way the inertia properties of a body using visual reconstruction.

7.5.3 Case B - Complex Shape

In this second experiment, a video of the 3-DOF simulator is processed. Figure 7.6 shows how complex is the shape of this object. In particular, the tanks are fixed with a perforated structure supported by four thin rods. It is very difficult to detect and model, in an accurate way, these elements and the wires present on the structure.

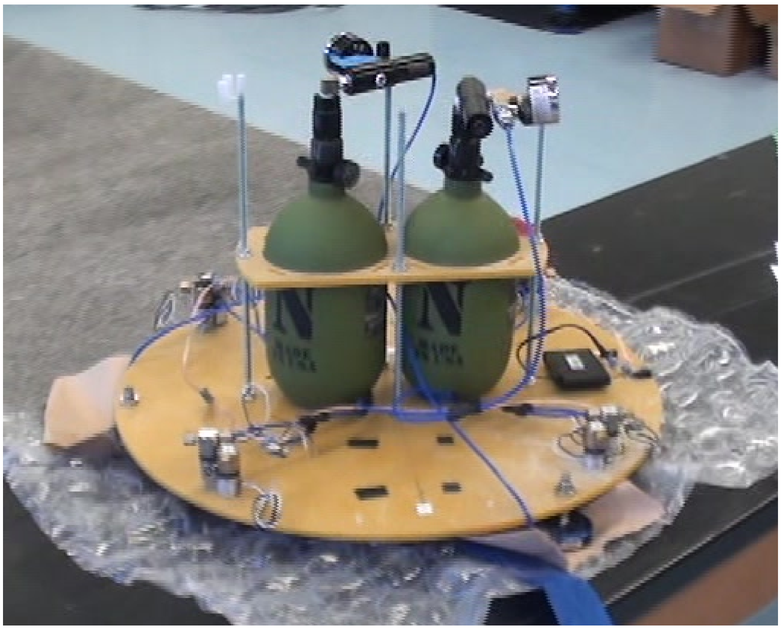


Figure 7.6. 3-DOF Simulator

As already said, the first step is to extract frames from the video. Figure 7.6 is one of them. Using VisualSFM, a point cloud like the one in fig. 7.7 is generated.

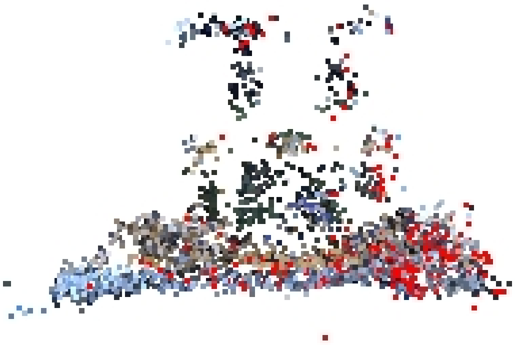


Figure 7.7. 3-DOF Simulator Point Cloud

From fig. 7.7, it is possible to see how the real shape of the object is not clearly represented. In the bottom part, all the elements are not precisely reconstructed. In the upper part, only the two tank regulators are clearly detectable. After processing this point cloud with MeshLab, generating a mesh and refining it, the result is much more accurate.

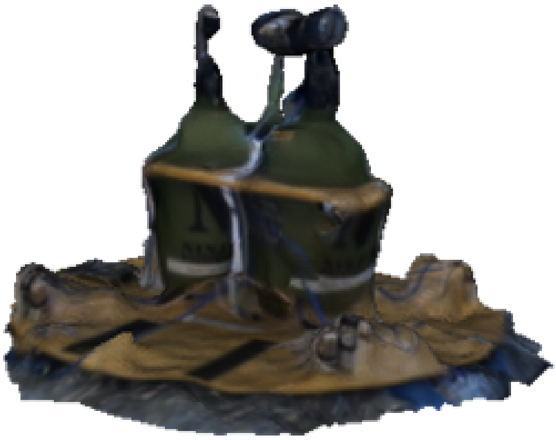


Figure 7.8. 3-DOF Simulator Mesh

Figure 7.8 shows a 3D mesh object with texture. This is a very good result, considering how poor was the initial point cloud. The base is still rough and all the details of the air bearings are lost. However, the average result is more than acceptable.

At this point, the matrix of inertia of this refined mesh has to be computed. The imported mesh in MATLAB appears like fig. 7.9

Using the previously described algorithm, the following results are obtained.

Assuming a constant density of $1\frac{g}{cm^3}$, the computed inertia matrix of the body is:

$$I = \begin{bmatrix} 795.49 & 0 & 0 \\ 0 & 855.07 & 0 \\ 0 & 0 & 978.07 \end{bmatrix} kg\,cm^2 \tag{7.43}$$

In order to compare this result with a 'real' value of the inertia matrix, the SolidWorks model of the 3-DOF Simulator is considered.

For simplicity, the wires are not reproduced in the model. SolidWorks computation of the inertia matrix, imposing the same constant density equal to of $1\frac{g}{cm^3}$, is:

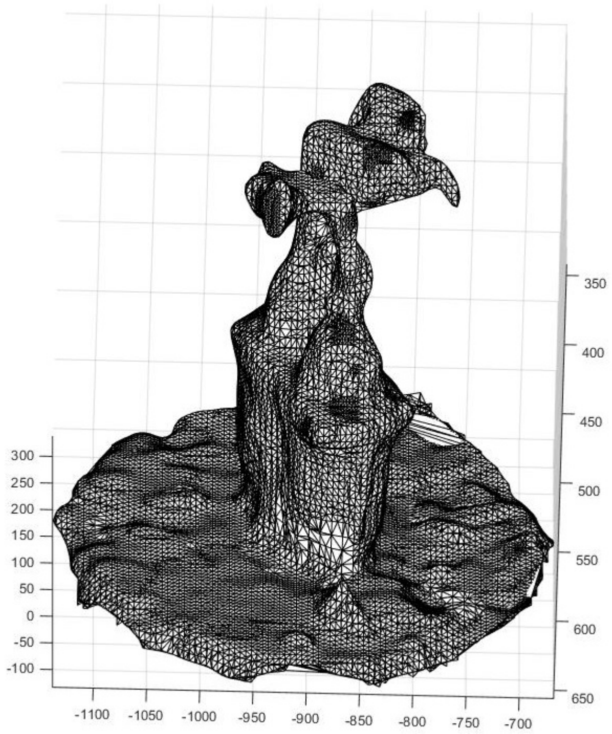


Figure 7.9. 3-DOF Simulator Mesh in MATLAB

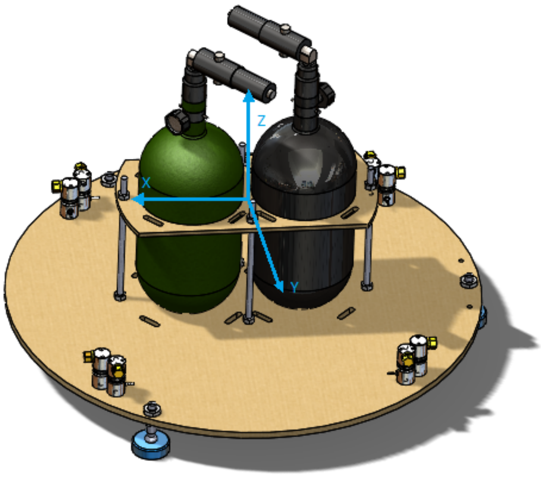


Figure 7.10. 3-DOF Simulator CAD Model

$$I_{SW} = \begin{bmatrix} 675.65 & 0 & 0 \\ 0 & 787 & 0 \\ 0 & 0 & 819.33 \end{bmatrix} kg\, cm^2 \tag{7.44}$$

Considering the error for the single components, the results are:

$$e_{I_{xx}} = 17.7\% \tag{7.45}$$

$$e_{I_{yy}} = 8.7\% \tag{7.46}$$

$$e_{I_{zz}} = 19.3\% \tag{7.47}$$

However, as already shown, in torque free motion, the ratios are sufficient to propagate the motion. For this reason, an evaluation of the error on the ratios is performed.

In this case, the relative errors are:

$$e_{I_{xx}} = 8.1\% \tag{7.48}$$

$$e_{I_{zz}} = 9.6\% \tag{7.49}$$

These results show how it is possible to obtain good inertia properties estimation only with camera measurements. However, this method is highly dependent on light conditions and observability of the body. This could be a key factor in space applications because, not always, favourable conditions are present.

Chapter 8

Conclusions

The main goal of this work was to develop an estimation procedure for the relative dynamics, capable to outperform the current methods present in the literature. Moreover, the algorithm should estimate the inertia parameters without relying on any 3D reconstruction procedure. In this way, the classical computational problems related to the image processing could be avoided. This thesis has proposed a new algorithm for estimating the pose, motion and inertia properties of an unknown, uncooperative space object. An IEKF and an EKF are used to estimate the relative state, exploiting the measurements collected by two cameras installed on the leader spacecraft. These sensors are selected since they are passive, highly reliable and have a low power consumption. On the other hand, the Extended and Iterated Extended Kalman filtering procedure have been chosen to guarantee simplicity and velocity but also accuracy and robustness. The IEKF appears to be more reliable with respect to its not iterated version. For this reason and also because it does not add any additional complexity, it is preferred to the EKF. An equality constraint is introduced to estimate the inertia properties of an object rotating with low angular velocity. It is included in the filtering procedure exploiting a pseudo-measurement equation, added to the observation model. However, this constraint can introduce numerical errors. For this reason, it has been demonstrated how, without this pseudo-measurement constraint, also the inertia ratios estimation converges with good results for certain values of the target angular velocity. In addition, it has been shown how the inertia properties can be reconstructed with a video processing procedure. In fact, the geometrical properties of a body can be reconstructed collecting multiple frames in time. In this way, imposing constant density, the mass properties of the observed object can be reconstructed. Several numerical simulations are assessed to test the filter in different conditions and with different covariance values. The obtained results show good convergence and robustness and outperform similar works available in the literature. Moreover, a preliminary computational analysis has been proposed. Simplicity and good computational performance are guaranteed. For the video processing procedure, experimental results are provided. The developed method allows good estimation of the inertia components of objects with both simple and complex shape. In this way, it has been demonstrated how this procedure allows for an offline approximation of the inertia matrix.

8.1 Review of Contributions

This work introduces some research contributions that are summarized in this section.

- The presented algorithm estimates relative translational and rotational dynamics, attitude and inertia properties assuming to use only stereo-vision. In the literature, no previous works estimate all these quantities without simplifying the relative dynamics or relying on different, more sophisticated sensors.

- Good results are obtained with different initial conditions. In particular, with a proper parametrization and adding a pseudo measurement constraint, the ratios of the inertia tensor components can be estimated also with a low target angular velocity.

- The pseudo measurement constraint needs the knowledge of the angular acceleration of the target object. This is obtained by a numerical differentiation. This procedure can introduce large errors. In order to avoid this, a video processing procedure is presented and experimentally validated. Good results are obtained also with complex shape objects.

8.2 Future Work

The next step of this work is the experimental implementation of the presented algorithm. This has to be optimized and eventually implemented with C/C++ to improve its computational performance. Moreover, an efficient computer vision algorithm has to be developed in order to extract useful information from the images in real applications. In our work, no perturbations are considered. A future step is to introduce orbital perturbations in the dynamical model. Another aspect that can be improved is the filtering process. EKF and IEKF perform linearisation that can affect the convergence and the performance of the filter. For this reason, an Unscented Kalman Filter (UKF) can be used. It uses an unscented transform to select sigma points around the desired mean. These points are then propagated without any linearisation and the covariance and the mean can be computed. Using this kind of filter could lead to better performance. The video processing procedure was implemented using free available software. For a real application, different tools has to be exploited or an ad hoc algorithm has to be developed. At the end, for an on-orbit implementation, many different aspects have to be taken into account. In particular, the lighting conditions can affect the computer vision algorithm and therefore the extracted measurements. Also the relative distance plays an important role in this sense. An evaluation on how the distance affects the estimation has to be done. In fact, this method could not be used for very far or very close objects. Concluding, this work shows how it is possible to estimate relative quantities in an autonomous way using only stereo-vision measurements. However, for a real application, in order to have a reliable system for every faced conditions, using additional sensors could result in a significant improvement of the estimation results.

Appendix A

EKF Derivation

This chapter presents the formal derivation of a discrete EKF. The procedure is divided in two steps, *prediction* and *filtering*.

A.1 Prediction

Supposing to have a Gaussian probability density function

$$p(x_k|{Y_1}^k) \sim \mathcal{N}(x_k - {\eta_F}^k, {V_F}^k) = \mathcal{N}(x_k - \hat{x}(k|k), P(k|k)) \tag{A.1}$$

with ${\eta_F}^k$ and ${V_F}^k$ being respectively the mean and the covariance matrix of the probability density function p.

Knowing the non-linear system discrete dynamics,

$$x_{k+1} = f_k(x_k) + w_k \tag{A.2}$$

and applying the Bayes law [86], the conditional probability function p of x_{k+1} given ${Y_1}^k$ can be expressed as:

$$p(x_{k+1}|{Y_1}^k) = \int_{-\infty}^{\infty} p(x_{k+1}|x_k)p(x_k|{Y_1}^k)dx_k \tag{A.3}$$

and can also be written as:

$$p(x_{k+1}|{Y_1}^k) = \int_{-\infty}^{\infty} p_{w_k}(x_{k+1} - f_k(x_k))p(x_k|{Y_1}^k)dx_k \tag{A.4}$$

with

$$p_{w_k}(x_{k+1} - f_n(x_k)) = \frac{1}{(2\pi)^{\frac{n}{2}}[\det Q_k]^{\frac{1}{2}}} e^{\frac{1}{2}(x_{k+1}-f_k(x_k))^T {Q_k}^{-1}(x_{k+1}-f_k(x_k))} \tag{A.5}$$

Because of x_k has a non-linear form, the probability density function is not Gaussian. At this point, the function $f_k(x_k)$ can be linearised around the mean ${\eta_F}^k$, neglecting higher order terms. Introducing the Jacobian of the function as:

$$F_k = \left.\frac{\partial f(x_k)}{\partial x_k}\right|_{\eta_F} \tag{A.6}$$

the linearised form of f is:

$$f_k(x_k) \cong f_k({\eta_F}^k) + F_k|_{\eta_F}[x_k - {\eta_F}^k] \tag{A.7}$$

defining a variable s_k as:

$$s_k = f_k({\eta_F}^k) - F_k|_{\eta_F}{\eta_F}^k] \tag{A.8}$$

the new linearised expression for the system dynamics is:

$$x_{k+1} = F_k|_{\eta_F}x_k + w_k + s_k \tag{A.9}$$

Equation eq. (A.9) represents a linear system, with s_k being a term depending on previous value of the state estimate with null conditional expected value.

Substituting eq. (A.7) in eq. (A.4), the probability density function can be re-written as:

$$p(x_{k+1}|{Y_1}^k) = \int_{-\infty}^{\infty} p_{w_k}(x_{k+1} - F_k|_{\eta_F}x_k - s_k)p(x_k|{Y_1}^k)dx_k \tag{A.10}$$

and knowing eq. (A.1),

$$p(x_{k+1}|{Y_1}^k) = \int_{-\infty}^{\infty} \mathcal{N}(x_{k+1} - F_k|_{\eta_F}x_k - s_k,\, Q_k)\mathcal{N}(x_k|{Y_1}^k,\, {V_F}^k)dx_k \tag{A.11}$$

At this point, a change of variable can be introduced.

$$q_k = F_k\, x_k \tag{A.12}$$

It is also possible to evaluate the mean and the covariance of the vector r_k:

$$E[q_k] = F_k E[x_k] = F_k {\eta_F}^k \tag{A.13}$$

$$E[q_k {q_k}^T] = F_k {V_F}^k {F_k}^T \tag{A.14}$$

Considering the probability density function expressed as eq. (A.5) and the introduced quantities, the following expression can be derived:

$$\mathcal{N}(x_k - {\eta_F}^k,\, {V_F}^k) = \frac{1}{(2\pi)^{\frac{n}{2}}[\det V_F]^{\frac{1}{2}}} e^{-\frac{1}{2}(x_k-\eta_F)^T {V_F}^{-1}(x_k-\eta_F)} \tag{A.15}$$

multiplying by F_k and its inverse and manipulating the expression, yields:

$$\mathcal{N}(x_k - {\eta_F}^k,\, {V_F}^k) = \frac{1}{(2\pi)^{\frac{n}{2}}[\det V_F]^{\frac{1}{2}}} e^{-\frac{1}{2}(F_k x_k-F_k\eta_F)^T F_k\, V_F\, {F_k}^{-1}(F_k x_k-F_k\eta_F)} \tag{A.16}$$

and again:

$$\mathcal{N}(x_k - {\eta_F}^k,\, {V_F}^k) = \det F_k \frac{1}{(2\pi)^{\frac{n}{2}}[\det F_k\, V_F\, {F_k}^T]^{\frac{n}{2}}} e^{-\frac{1}{2}(F_k x_k-F_k\eta_F)^T F_k\, V_F\, {F_k}^{-1}(F_k x_k-F_k\eta_F)} \tag{A.17}$$

So it is possible to recover the following expression:

$$\mathcal{N}(x_k - {\eta_F}^k,\, {V_F}^k) = \det F_k \mathcal{N}(F_k x_k - F_k{\eta_F}^k\, F_k V_F\, {F_k}^T) \tag{A.18}$$

Once the eq. (A.18) is known, this expression can be substituted in eq. (A.11) to obtain:

$$p(x_{k+1}|{Y_1}^k) = \int_{-\infty}^{\infty} \mathcal{N}(x_{k+1} - F_k x_k - s_k,\, Q_k)\mathcal{N}(F_k x_k - F_k\, V_F\, {F_k}^T) d(F_k x_k) \tag{A.19}$$

and from the previous equation,

$$p(x_{k+1}|{Y_1}^k) = \int_{-\infty}^{\infty} \mathcal{N}(x_{k+1} - s_k,\, Q_k) * \mathcal{N}(x_k - F_k{\eta_F}^k\, F_k V_F\, {F_k}^T) \tag{A.20}$$

The symbol $*$ represents the convolution of two functions.

The final value of the probability density function therefore is:

$$p(x_{k+1}|{Y_1}^k) = \mathcal{N}(x_{k+1} - F_k{\eta_F}^k - s_k,\, Q_k + F_k\, V_F\, {F_k}^T) \tag{A.21}$$

This probability function has mean ${\eta_F}^n$ and covariance ${V_F}^n$. Then, with the linearisation of the dynamics around the mean ${\eta_F}^n$ the $p(x_{k+1}|{Y_1}^k)$ at the new step is obtained. This probability density function has mean ${\eta_P}^{k+1}$ and covariance ${V_P}^{k+1}$. These two values are respectively:

$${\eta_P}^{k+1} = f_k({\eta_F}^k) \tag{A.22}$$

and

$$ {V_P}^{k+1} = Q_k + F_k\, {V_F}^k {F_k}^T \tag{A.23} $$

These values are the predicted state estimated and the associated covariance obtained by the EKF prediction step:

$$ \hat{x}(k+1|k) = {\eta_P}^{k+1} \tag{A.24} $$

$$ P(k+1|k) = {V_P}^{k+1} \tag{A.25} $$

or also

$$ \hat{x}(k+1|k) = f_k(\hat{x}(k|k)) \tag{A.26} $$

$$ P(k+1|k) = Q_k + F_k\, P(k|k) {F_k}^T \tag{A.27} $$

A.2 Filtering

For the filtering step, the measurements at time instant *k+1* have to be used to update the probability density function $p(x_{k+1}|{Y_1}^k)$ and therefore obtain $p(x_{k+1}|{Y_1}^{k+1})$ The Bayes law can be again recalled:

$$ p(x_{k+1}|{Y_1}^{k+1}) = \frac{p({Y_1}^k)}{p({Y_1}^{k+1})}[p(y_{k+1}|x_{k+1})p(x_{k+1}|{Y_1}^k)] \tag{A.28} $$

from the knowledge of the measurement equation

$$ y_{k+1} = h_{k+1}(x_{x+1}) + v_{k+1} \tag{A.29} $$

the probability density function of y_{k+1} conditioned on the state x_{k+1} is:

$$ p(y_{k+1}|(x_{k+1})) = \frac{1}{(2\pi)^{\frac{n}{2}}[\det R_{k+1}]^{\frac{1}{2}}} e^{\frac{1}{2}(y_{k+1}-h_{k+1}(x_{k+1}))^T {R_{k+1}}^{-1}(y_{k+1}-h_{k+1}(x_{k+1}))} \tag{A.30} $$

as already done for the dynamical model, it is possible to define the Jacobian of the measurement equation and perform a linearisation.

$$ H_k = \left.\frac{\partial h(x_{k+1})}{\partial x_{k+1}}\right|_{\eta_P} \tag{A.31} $$

Linearising $h_{k+1}(x_{k+1})$ around ${\eta_P}^{k+1}$ and ignoring high order terms yields:

$$h_{k+1}(x_{k+1}) \cong h_{k+1}({\eta_P}^{k+1}) + H_k|_{\eta_P}[x_{k+1} - {\eta_P}^{k+1}] \tag{A.32}$$

therefore, the linearised observation equation is:

$$y_{k+1} \cong H_{k+1}x_{k+1} + v_{k+1} + r_{k+1} \tag{A.33}$$

with r_{k+1} being

$$r_{k+1} = h_{k+1}({\eta_P}^{k+1}) - H_{k+1}{\eta_P}^{k+1} \tag{A.34}$$

The equation eq. (A.33) may be considered a linear equation and therefore, it is possible to compute the related probability density function $p(y_{k+1}|(x_{k+1}))$ as:

$$p(y_{k+1}|(x_{k+1})) = \mathcal{N} y_{k+1} - r_{k+1} - H_{k+1}x_{k+1},\, R_{k+1} \tag{A.35}$$

Using a similar transformation to what done for the prediction cycle, the probability density function can be expressed as:

$$p(y_{k+1}|{Y_1}^k) = \det H_k \mathcal{N}(H_{k+1}x_{k+1} - H_{k+1}{\eta_P}^{k+1}\, H_{k+1}V_P\, {F_{k+1}}^T) \tag{A.36}$$

Multiplying eq. (A.35) and eq. (A.36), the following expression is obtained:

$$p(y_{k+1}|(x_{k+1}))\, p(y_{k+1}|{Y_1}^k) \sim \mathcal{N}(H_{k+1}x_{k+1} - \mu, V) \tag{A.37}$$

with μ and V being the mean and the covariance matrix. Their expression is:

$$\begin{aligned}\mu = {} & H_{k+1}{V_P}^{k+1}{H_{k+1}}^T(H_{k+1}{V_P}^{k+1}{H_{k+1}}^T + R_{k+1})^{-1}[-r_{k+1} + y_{k+1}] \\ & + R_{k+1}(H_{k+1}{V_P}^{k+1}{H_{k+1}}^T + R_{k+1})^{-1}H_{k+1}{\eta_P}^{k+1}\end{aligned} \tag{A.38}$$

$$V = H_{k+1}{V_P}^{k+1}{H_{k+1}}^T(H_{k+1}{V_P}^{k+1}{H_{k+1}}^T + R_{k+1})^{-1}R_{k+1} \tag{A.39}$$

Recalling the expression of r_{k+1} in eq. (A.34) and substituting it in eq. (A.39), the following expression for μ is derived:

$$\begin{aligned}\mu = {} & H_{k+1}{V_P}^{k+1}{H_{k+1}}^T(H_{k+1}{V_P}^{k+1}{H_{k+1}}^T + R_{k+1})^{-1}[-h_{k+1}({\eta_P}^{k+1}) + H_{k+1}{\eta_P}^{k+1} + z_{k+1}] \\ & + R_{k+1}(H_{k+1}{V_P}^{k+1}{H_{k+1}}^T + R_{k+1})^{-1}H_{k+1}{\eta_P}^{k+1}\end{aligned} \tag{A.40}$$

that can also be written as:

$$\mu = H_{k+1}{V_P}^{k+1}+(H_{k+1}{V_P}^{k+1}{H_{k+1}}^T(H_{k+1}{V_P}^{k+1}{H_{k+1}}^T+R_{k+1})^{-1}[y_{k+1}-h_{k+1}({\eta_P}^{k+1})] \tag{A.41}$$

It is also known that:

$$\mathcal{N}(x_{k+1}-\mu_1, V_1) = \det H_{k+1}\mathcal{N}(H_{k+1}x_{k+1}-\mu, V) \tag{A.42}$$

comparing eq. (A.42) with eq. (A.37), the following expression for μ_1 is obtained:

$$\mu_1 = {\eta_P}^{k+1}+{V_P}^{k+1}{H_{k+1}}^T(H_{k+1}{V_P}^{k+1}{H_{k+1}}^T+R_{k+1})^{-1}[y_{k+1}-h_{k+1}({\eta_P}^{k+1})] \tag{A.43}$$

Therefore it is possible to conclude that:

$${\eta_F}^{k+1} = {\eta_P}^{k+1} + {V_P}^{k+1}{H_{k+1}}^T(H_{k+1}{V_P}^{k+1}{H_{k+1}}^T + R_{k+1})^{-1}[y_{k+1} - h_{k+1}({\eta_P}^{k+1})] \tag{A.44}$$

and for the covariance matrix:

$$V = H_{k+1}{V_F}^{k+1}{H_{k+1}}^T \tag{A.45}$$

this has to be solved relative to ${V_F}^{k+1}$ and therefore:

$$H_{k+1}{V_P}^{k+1}{H_{k+1}}^T = H_{k+1}{V_F}^{k+1}{H_{k+1}}^T{R_{k+1}}^{-1}H_{k+1}{V_P}^{k+1}{H_{k+1}}^T + H_{k+1}{V_F}^{k+1}{H_{k+1}}^T \tag{A.46}$$

or also,

$${V_F}^{k+1} = {V_P}^{k+1}[I + {H_{k+1}}^T R_{k+1}H_{k+1}{V_P}^{k+1}]^{-1} \tag{A.47}$$

Recalling the lemma of the inversion of the matrices, the following expression can be derived:

$${V_F}^{k+1} = {V_P}^{k+1} - {V_P}^{k+1}{H_{k+1}}^T[R_{k+1}H_{k+1}{V_P}^{k+1}{H_{k+1}}^T]^{-1}H_{k+1}{V_P}^{k+1} \tag{A.48}$$

So, in conclusion, having information of the measurements y_{k+1} and linearising the observation model around the mean ${\eta_P}^{k+1} = \hat{x}(k+1|k)$, the result is a Gaussian probability density function $p(x_{k+1}|{Y_1}^{k+1})$ with mean ${\eta_F}^{k+1}$ and covariance matrix ${V_F}^{k+1}$. At the end, exploiting the derived equations, the EKF is summarized.

First of all, the non-linear equations for the system and the observation model can be written in a linear form:

$$x(k+1) = F(k)x(k) + w_k + s_k \tag{A.49}$$

$$y(k+1) = H(k+1)x(k+1) + v_{k+1} + r_{k+1} \tag{A.50}$$

The prediction is given by:

$$\hat{x}(k+1|k) = f_k(\hat{x}(k|k)) \tag{A.51}$$

$$P(k+1|k) = F(k)P(k|k)F(k)^T + Q(k) \tag{A.52}$$

And the update consequently is:

$$K(k+1) = P(k+1|k)H^T(k+1)[H(k+1)P(k+1|k)H^T(k+1)+R(k+1)]^{-1} \tag{A.53}$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1)[y_{k+1} - h_{k+1}(\hat{x}(k+1|k))] \tag{A.54}$$

$$\begin{aligned} P(k+1|k+1) =& P(k+1|k) - P(k+1|k)H^T(k+1)[H(k+1)P(k+1|k)H^T(k+1)+ \\ & R(k+1)]^{-1}H(k+1)P(k+1) \end{aligned} \tag{A.55}$$

Equation (A.55) can also be written as:

$$P(k+1|k+1) = [I - K(k+1)H(k+1)]P(k+1|k) \tag{A.56}$$

Appendix B

Differentiation of noisy signal

In this appendix, a brief survey of differentiation methods is presented. In particular, approaches that handle noisy data and signals are described. These methods are used with a set of data. In an online implementation, the sampled data set could be determined by a *shifting windows* that takes only the last N values of the signal that have to be differentiated. The general problem is to find an approximation of the derivative of a generic smooth function g in the interval *(a,b)*, knowing data points belonging to this function in the same interval. In particular, an approximation of the function g has to be determined. This approximated function f can be differentiated to obtain a value d that approximates the derivative of the original function g.

From the mathematical point of view, the data set can be seen as:

$$y_i = g(x_i) + v_i \tag{B.1}$$

where v_i is the noise contaminating the function g. The fitting function f can be seen as a random variable and therefore, the mean squared error can be defined at each point as:

$$e_x = E[(g(x_i) - f(x_i))^2] \tag{B.2}$$

Many different methods are developed to solve this problem. In this appendix, only few of them are presented.

B.1 Least Squares Polynomial Approximation

This method consists in fitting the set of data with a sequence of polynomials f and solve a least square problem. For these polynomials, a value of the order

o has to be selected. It is the smallest value for which the residual norm is lower than a fixed tolerance. The approximated value d is therefore the derivative of the function f.

B.2 Tikhonov regularization

This method introduces a regularization term in the formulation of the least square problem. This term improves the conditioning of the ill-posed problem. In particular, the value $g(a)$ (where a is the left bound of the domain of interest) is known and the function g is described in the Sobolev space $H^{k+1}(a, b)$. The approximated problem is:

$$A\,d = \int_a^x d(t)\,dt \tag{B.3}$$

From the knowledge of d, the function f can be easily computed as the integral of the function d.

The system can be also represented in a discrete way:

$$A\,d = \hat{y} \tag{B.4}$$

where $\hat{y}$ is the difference between y and $g(a)$.

The problem in eq. (B.4) is ill-conditioned and therefore, a regularization factor is added.

$$E(d) = ||A\,d - \hat{y}||^2 + \alpha||D\,d||^2 \tag{B.5}$$

The regularization factor is composed by a nonnegative parameter α and from the differential operator D. Depending on the order, various expressions of the differential operator are possible. The general solution to this problem is:

$$(A^t A + \alpha D^t D)d = A^t\hat{y} \tag{B.6}$$

Many others numerical approaches exist. They exploit splines, convolutions and other methods to approximate the derivative. In this work, only the most famous and reliable numerical methods are presented.

B.3 Tracking Differentiator

Another different approach was proposed by Han [87]. He introduced the concept of tracking-differentiator. This is a tool that allows to filter a noisy signal and to compute its derivative. It outperforms numerical differentiation methods in many online applications.

The basic idea is to describe a system that, given a reference noisy signal $y(t)$, provides two signals y_1 and y_2 such that $y_1(t) = y(t)$ and $y_2(t) = \dot{y}(t)$

This is equivalent to solve the system:

$$\begin{cases} \dot{y}_1(t) = y_2(t) \\ \dot{y}_2(t) = f(y_1(t), y_2(t)) \end{cases} \tag{B.7}$$

If this system satisfies $lim_{t\to\infty}(y_1(t)\, y_2(t)) = 0$, then for any bounded function $v(t)$ and any constant $T > 0$, the solution of the system in eq. (B.8),

$$\begin{cases} \dot{y}_1(t) = y_2(t) \\ \dot{y}_2(t) = R^2\, f(y_1(t) - v(t),\, \frac{y_2(t)}{R}) \end{cases} \tag{B.8}$$

satisfies

$$\lim_{R\to\infty} \int_0^T |y_1(t) - v(t)|dt = 0 \tag{B.9}$$

This system represents a nonlinear tracking differentiator. One of the feasible systems is represented by the following expression:

$$\begin{cases} \dot{y}_1 = y_2 \\ \dot{y}_2 = -R\, sat(y_1 - v + \frac{y_2|y_2|}{2R},\, \delta) \end{cases} \tag{B.10}$$

With R and δ being two tuning factors, respectively velocity and filtering factor. Additionally, sat is a nonlinear saturation function expressed by:

$$sat(A, \delta) = \begin{cases} sgn(A), \quad |A| > \delta \\ \frac{A}{\delta}, \quad |A| \leq \delta \end{cases} \tag{B.11}$$

As already said, R and δ can be properly tuned in order to achieve certain performance. In particular, with a large value of δ the noise tends to decrease. Large value of R, instead, implies faster transition and tracking.

Bibliography

[1] S Croomes, “Overview of the dart mishap investigation results”, *NASA Report*, pp. 1–10, 2006 (cit. on p. 27).

[2] R. B. Friend, “Orbital express program summary and mission overview”, in *SPIE Defense and Security Symposium*, International Society for Optics and Photonics, 2008, pp. 695 803–695 803 (cit. on p. 27).

[3] F Chabert, “Automatic transfer vehicle (atv) gnc overview”, in *IAF abstracts, 34th COSPAR Scientific Assembly*, vol. 1, 2002, p. 55 (cit. on p. 27).

[4] P. Kimball, “Iceberg-relative navigation for autonomous underwater vehicles. phd”, *Stanford University, Stanford, CA*, vol. 8, p. 2011, 2011 (cit. on pp. 28, 34).

[5] O. G. Grasa, J. Civera, and J. Montiel, “Ekf monocular slam with relocalization for laparoscopic sequences”, in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 4816–4821 (cit. on pp. 28, 34).

[6] M. Bryson and S. Sukkarieh, “Building a robust implementation of bearing-only inertial slam for a uav”, *Journal of Field Robotics*, vol. 24, no. 1-2, pp. 113–143, 2007 (cit. on pp. 28, 34).

[7] K. Shahid and G. Okouneva, “Intelligent lidar scanning region selection for satellite pose estimation”, *Computer Vision and Image Understanding*, vol. 107, no. 3, pp. 203–209, 2007 (cit. on p. 29).

[8] P. Jasiobedzki, S. Se, T. Pan, M. Umasuthan, and M. Greenspan, “Autonomous satellite rendezvous and docking using lidar and model based vision”, in *Defense and Security*, International Society for Optics and Photonics, 2005, pp. 54–65 (cit. on p. 29).

[9] R. C. Fenton, *A LADAR-Based Pose Estimation Algorithm for Determining Relative Motion of a Spacecraft for Autonomous Rendezvous and Dock.* ProQuest, 2008 (cit. on p. 29).

[10] Z. Shijie, L. Fenghua, C. Xibin, and H. Liang, “Monocular vision-based two-stage iterative algorithm for relative position and attitude estimation of docking spacecraft”, *Chinese Journal of Aeronautics*, vol. 23, no. 2, pp. 204–210, 2010 (cit. on p. 30).

[11] S. Zhang, X. Cao, F. Zhang, and L. He, “Monocular vision-based iterative pose estimation algorithm from corresponding feature points”, *Science China Information Sciences*, vol. 53, no. 8, pp. 1682–1696, 2010 (cit. on p. 30).

[12] H. Pan, J. Y. Huang, and S. Y. Qin, “Relative pose estimation under monocular vision in rendezvous and docking”, *Applied Mechanics and Materials*, vol. 433, pp. 799–805, 2014 (cit. on p. 30).

[13] S. D’Amico, M. Benn, and J. L. Jørgensen, “Pose estimation of an uncooperative spacecraft from actual space imagery”, *International Journal of Space Science and Engineering*, vol. 2, no. 2, pp. 171–189, 2014 (cit. on p. 30).

[14] P. Azad, T. Asfour, and R. Dillmann, “Stereo-based vs. monocular 6-dof pose estimation using point features: a quantitative comparison”, in *Autonome Mobile Systeme 2009*, Springer, 2009, pp. 41–48 (cit. on p. 30).

[15] N. Inaba and M. Oda, “Autonomous satellite capture by a space robot: world first on-orbit experiment on a japanese robot satellite ets-vii”, in *Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on*, IEEE, vol. 2, 2000, pp. 1169–1174 (cit. on p. 30).

[16] T. M. Davis and D. Melanson, “Xss-10 microsatellite flight demonstration program results”, in *Defense and Security*, International Society for Optics and Photonics, 2004, pp. 16–25 (cit. on p. 30).

[17] R. W. Madison, “Micro-satellite based, on-orbit servicing work at the air force research laboratory”, in *Aerospace conference proceedings, 2000 IEEE*, IEEE, vol. 4, 2000, pp. 215–226 (cit. on p. 30).

[18] C. Moore, “Technology development for nasa’s asteroid redirect mission”, (cit. on p. 30).

[19] S.-G. Kim, J. L. Crassidis, Y. Cheng, A. M. Fosbury, and J. L. Junkins, “Kalman filtering for relative spacecraft attitude and position estimation”, *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 133–143, 2007 (cit. on p. 31).

[20] Y. Xing, X. Cao, S. Zhang, H. Guo, and F. Wang, “Relative position and attitude estimation for satellite formation with coupled translational and rotational dynamics”, *Acta Astronautica*, vol. 67, no. 3, pp. 455–467, 2010 (cit. on p. 31).

[21] N. Philip and M. Ananthasayanam, “Relative position and attitude estimation and control schemes for the final phase of an autonomous docking mission of spacecraft”, *Acta Astronautica*, vol. 52, no. 7, pp. 511–522, 2003 (cit. on p. 31).

[22] M. D. Lichter and S. Dubowsky, “State, shape, and parameter estimation of space objects from range images”, in *Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*, IEEE, vol. 3, 2004, pp. 2974–2979 (cit. on p. 31).

[23] ——, “Shape, motion, and parameter estimation of large flexible space structures using range images”, in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, IEEE, 2005, pp. 4476–4481 (cit. on pp. 31, 92).

[24] S. Segal, A. Carmi, and P. Gurfil, “Vision-based relative state estimation of non-cooperative spacecraft under modeling uncertainty”, in *Aerospace Conference, 2011 IEEE*, IEEE, 2011, pp. 1–8 (cit. on pp. 32, 37).

[25] ——, “Stereovision-based estimation of relative dynamics between noncooperative satellites: theory and experiments”, *Control Systems Technology, IEEE Transactions on*, vol. 22, no. 2, pp. 568–584, 2014 (cit. on pp. 32, 34).

[26] S. Segal, P. Gurfil, and A. Carmi, “Experimental validation of stereoscopic satellite relative state estimation”, in *AIAA Guidance, Navigation, and Control Conference*, AIAA Reston, VA, 2011, pp. 1–16 (cit. on p. 32).

[27] B. E. Tweddle and A. Saenz-Otero, “Relative computer vision-based navigation for small inspection spacecraft”, *Journal of Guidance, Control, and Dynamics*, pp. 1–9, 2014 (cit. on p. 32).

[28] B. E. Tweddle, “Computer vision-based localization and mapping of an unknown, uncooperative and spinning target for spacecraft proximity operations”, *PhD, Massachusetts Institute of Technology, Cambridge, MA*, 2013 (cit. on pp. 32, 34, 35, 72, 92).

[29] M. Kaess, A. Ranganathan, and F. Dellaert, “Isam: incremental smoothing and mapping”, *Robotics, IEEE Transactions on*, vol. 24, no. 6, pp. 1365–1378, 2008 (cit. on p. 32).

[30] ——, “Isam: fast incremental smoothing and mapping with efficient data association”, in *Robotics and Automation, 2007 IEEE International Conference on*, IEEE, 2007, pp. 1670–1677 (cit. on p. 32).

[31] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular slam: why filter?", in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, IEEE, 2010, pp. 2657–2664 (cit. on p. 32).

[32] W. Clohessy W.H., "Terminal guidance system for satellite rendezvous", 1960 (cit. on pp. 33, 37).

[33] T. J and P Hempel, "Rendezvous zu einem in elliptischer bahn umlaufenden ziel", *Astronautica Acta*, vol. 11, no. 2, p. 104, 1965 (cit. on p. 33).

[34] C. D. Karlgaard and F. H. Lutze, "Second-order relative motion equations", *Journal of guidance, control, and dynamics*, vol. 26, no. 1, pp. 41–49, 2003 (cit. on p. 33).

[35] S. S. Vaddi, S. R. Vadali, and K. T. Alfriend, "Formation flying: accommodating nonlinearity and eccentricity perturbations", *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 2, pp. 214–223, 2003 (cit. on p. 33).

[36] P. Gurfil and N. J. Kasdin, "Nonlinear modelling of spacecraft relative motion in the configuration space", *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 154–157, 2004 (cit. on p. 33).

[37] P. Gurfil and K. V. Kholshevnikov, "Manifolds and metrics in the relative spacecraft motion problem", *Journal of guidance, control, and dynamics*, vol. 29, no. 4, pp. 1004–1010, 2006 (cit. on p. 33).

[38] S. Vaddi, K. T. Alfriend, S. Vadali, and P Sengupta, "Formation establishment and reconfiguration using impulsive control", *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 2, pp. 262–268, 2005 (cit. on p. 33).

[39] J.-F. Hamel and J. D. Lafontaine, "Linearized dynamics of formation flying spacecraft on a j2-perturbed elliptical orbit", *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1649–1658, 2007 (cit. on p. 33).

[40] D.-W. Gim and K. T. Alfriend, "Satellite relative motion using differential equinoctial elements", *Celestial Mechanics and Dynamical Astronomy*, vol. 92, no. 4, pp. 295–336, 2005 (cit. on p. 33).

[41] H. Pan and V. Kapila, "Adaptive nonlinear control for spacecraft formation flying with coupled translational and attitude dynamics", in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, IEEE, vol. 3, 2001, pp. 2057–2062 (cit. on pp. 33, 37).

[42] S. Segal and P. Gurfil, "Effect of kinematic rotation-translation coupling on relative spacecraft translational dynamics", *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 3, pp. 1045–1050, 2009 (cit. on pp. 33, 45, 48).

[43] D. Nistér, O. Naroditsky, and J. Bergen, “Visual odometry”, in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, IEEE, vol. 1, 2004, pp. I–652 (cit. on p. 33).

[44] S. Thrun, W. Burgard, and D. Fox, “A probabilistic approach to concurrent mapping and localization for mobile robots”, *Autonomous Robots*, vol. 5, no. 3-4, pp. 253–271, 1998 (cit. on p. 33).

[45] Y. Cheng, M. W. Maimone, and L. Matthies, “Visual odometry on the mars exploration rovers-a tool to ensure accurate driving and science imaging”, *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 54–62, 2006 (cit. on p. 34).

[46] B. Van Pham, S. Lacroix, M. Devy, T. Voirin, C. Bourdarias, and M. Drieux, “Fusion of visual odometry and landmark constellation matching for spacecraft absolute navigation: analysis and experiments”, in *Proceedings of the 11th Symposium on Advanced Space Technologies in Robotics and Automation*, 2011 (cit. on p. 34).

[47] S. Augenstein and S. M. Rock, “Improved frame-to-frame pose tracking during vision-only slam/sfm with a tumbling target”, in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 3131–3138 (cit. on p. 34).

[48] C Cocaud and T Kubota, “Surf-based slam scheme using octree occupancy grid for autonomous landing on asteroids”, in *Int. Symp. on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), Sapporo, Japan*, vol. 30, 2010, pp. 275–282 (cit. on p. 34).

[49] D. J. Heeger and A. D. Jepson, “Subspace methods for recovering rigid motion i: algorithm and implementation”, *International Journal of Computer Vision*, vol. 7, no. 2, pp. 95–117, 1992 (cit. on p. 60).

[50] H. C. Longuet-Higgins and K. Prazdny, “The interpretation of a moving retinal image”, *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 208, no. 1173, pp. 385–397, 1980 (cit. on p. 60).

[51] A. R. Bruss and B. K. Horn, “Passive navigation”, *Computer Vision, Graphics, and Image Processing*, vol. 21, no. 1, pp. 3–20, 1983 (cit. on p. 60).

[52] A. M. Waxman and S. Ullman, “Surface structure and three-dimensional motion from image flow kinematics”, *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 72–94, 1985 (cit. on p. 60).

[53] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: speeded up robust features”, in *Computer vision–ECCV 2006*, Springer, 2006, pp. 404–417 (cit. on p. 63).

[54] E. Rosten and T. Drummond, “Fusing points and lines for high performance tracking”, in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, IEEE, vol. 2, 2005, pp. 1508–1515 (cit. on p. 63).

[55] C. Harris and M. Stephens, “A combined corner and edge detector.”, in *Alvey vision conference*, Manchester, UK, vol. 15, 1988, p. 50 (cit. on p. 63).

[56] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions”, *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004 (cit. on p. 63).

[57] B. D. Lucas, T. Kanade, *et al.*, “An iterative image registration technique with an application to stereo vision.”, in *IJCAI*, vol. 81, 1981, pp. 674–679 (cit. on p. 65).

[58] C. Tomasi and T. Kanade, “Shape and motion from image streams under orthography: a factorization method”, *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992 (cit. on pp. 65, 69).

[59] J. Shi and C. Tomasi, “Good features to track”, in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, IEEE, 1994, pp. 593–600 (cit. on p. 69).

[60] R. E. Kalman *et al.*, “Contributions to the theory of optimal control”, *Bol. Soc. Mat. Mexicana*, vol. 5, no. 2, pp. 102–119, 1960 (cit. on p. 72).

[61] R. Kalman, “On the general theory of control systems”, *IRE Transactions on Automatic Control*, vol. 4, no. 3, pp. 110–110, 1959 (cit. on p. 72).

[62] R. E. Kalman, “Mathematical description of linear dynamical systems”, *Journal of the Society for Industrial & Applied Mathematics, Series A: Control*, vol. 1, no. 2, pp. 152–192, 1963 (cit. on p. 72).

[63] Y. Y. Roinodot'tenberg, “Observability of nonlinear systems”, *SIAM Journal on Control*, vol. 8, no. 3, pp. 338–345, 1970 (cit. on p. 72).

[64] M Hwang and J. Seinfeld, “Observability of nonlinear systems”, *Journal of Optimization Theory and Applications*, vol. 10, no. 2, pp. 67–77, 1972 (cit. on p. 72).

[65] S. R. Kou, D. L. Elliott, and T. J. Tarn, “Observability of nonlinear systems”, *Information and Control*, vol. 22, no. 1, pp. 89–99, 1973 (cit. on p. 72).

[66] R. Hermann and A. J. Krener, “Nonlinear controllability and observability”, *IEEE Transactions on automatic control*, vol. 22, no. 5, pp. 728–740, 1977 (cit. on p. 72).

[67] R. E. Kalman, “A new approach to linear filtering and prediction problems”, *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960 (cit. on p. 77).

[68] T. N. Thiele, “Om anvendelse af mindste kvadraters methode i nogle tilfælde, hvor en komplikation af visse slags uensartede tilfældige fejlkilder giver fejlene en ‘systematisk’karakter”, *Det Kongelige Danske Videnskabernes Selskabs Skrifter-Naturvidenskabelig og Mathematisk Afdeling*, pp. 381–408, 1880 (cit. on p. 77).

[69] P. Swerling, “First-order error propagation in a stagewise smoothing procedure for satellite observations”, 1959 (cit. on p. 77).

[70] P. S. Maybeck, *Stochastic models, estimation, and control.* Academic press, 1982, vol. 3 (cit. on pp. 78, 83).

[71] Z. Paul and M. Howard, “Fundamentals of kalman filtering-a practical approach”, *Virginia, Published by the American Institute of Aeronautics and Astronautics*, 2005 (cit. on p. 80).

[72] F. Aghili, “A prediction and motion-planning scheme for visually guided robotic capturing of free-floating tumbling objects with uncertain dynamics”, *Robotics, IEEE Transactions on*, vol. 28, no. 3, pp. 634–649, 2012 (cit. on p. 91).

[73] W Wen and H. F. Durrant-Whyte, “Model-based multi-sensor data fusion”, in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, IEEE, 1992, pp. 1720–1726 (cit. on p. 94).

[74] J. Porrill, “Optimal combination and constraints for geometrical sensor data”, *The International Journal of Robotics Research*, vol. 7, no. 6, pp. 66–77, 1988 (cit. on p. 94).

[75] A. Alouani and W. Blair, “Use of a kinematic constraint in tracking constant speed, maneuvering targets”, *Automatic Control, IEEE Transactions on*, vol. 38, no. 7, pp. 1107–1111, 1993 (cit. on p. 94).

[76] L.-S. Wang, Y.-T. Chiang, and F.-R. Chang, “Filtering method for nonlinear systems with constraints”, *IEE Proceedings-Control Theory and Applications*, vol. 149, no. 6, pp. 525–531, 2002 (cit. on p. 94).

[77] D. Simon and D. L. Simon, "Kalman filtering with inequality constraints for turbofan engine health estimation", *IEE Proceedings-Control Theory and Applications*, vol. 153, no. 3, pp. 371–378, 2006 (cit. on p. 94).

[78] H. E. Doran, "Constraining kalman filter and smoothing estimates to satisfy time-varying restrictions", *The Review of Economics and Statistics*, pp. 568–572, 1992 (cit. on p. 94).

[79] T. L. Chia, P.-C. Chow, and H. J. Chizeck, "Recursive parameter identification of constrained systems: an application to electrically stimulated muscle", *Biomedical Engineering, IEEE Transactions on*, vol. 38, no. 5, pp. 429–442, 1991 (cit. on p. 94).

[80] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control.* Wiley New York, 2006, vol. 3 (cit. on p. 94).

[81] T. Andrews, "Computation time comparison between matlab and c++ using launch windows", 2012 (cit. on p. 135).

[82] C. Wu, "Visualsfm: a visual structure from motion system", *URL: http://homes. cs. washington. edu/˜ ccwu/vsfm*, vol. 9, 2011 (cit. on p. 138).

[83] D. G. Lowe, "Object recognition from local scale-invariant features", in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, Ieee, vol. 2, 1999, pp. 1150–1157 (cit. on p. 138).

[84] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "Meshlab: an open-source mesh processing tool", in *Eurographics Italian Chapter Conference*, The Eurographics Association, 2008, pp. 129–136 (cit. on p. 139).

[85] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction", in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, 2006 (cit. on p. 139).

[86] T. Bayes, D. W. Bunn, H. Raiffa, R. Schlaifer, and D. Von Winterfeldt, "An essay toward solving a problem in the doctrine of chances", *Philosophical Transactions of the Royal Society of London*, vol. 53, 1984 (cit. on p. 157).

[87] J. Han and W. Wang, "Nonlinear tracking-differentiator", *J. Syst. Sci. Math. Sci*, vol. 14, no. 2, pp. 177–183, 1994 (cit. on p. 167).

[88] L. Holguin, S. P. Viswanathan, and A. Sanyal, "Guidance and control for spacecraft autonomous rendezvous and proximity maneuvers using a geometric mechanics framework", in *AIAA Guidance, Navigation, and Control Conference*, 2012.

[89] D. Scaramuzza and F. Fraundorfer, “Visual odometry [tutorial]”, *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 80–92, 2011.

[90] E. D. Sontag, “A concept of local observability”, *Systems Control Lett.*, vol. 5, no. 1, pp. 41–47, 1984.

[91] S. Kokarev, “Lectures “elements of the theory of smooth manifolds”,(i): lie derivatives and their applications”, *Coll. papers of RSEC” Logos”, Yaroslavl*, vol. 4, pp. 77–166, 2009.

[92] J Kelly, “On the observability and self-calibration of visual-inertial navigation systems”, *University of Southern California, Los Angeles, USA, Tech. Rep. CRES-08-005*, 2008.

[93] A. Abdelgawad and M. Bayoumi, *Resource-Aware data fusion algorithms for wireless sensor networks.* Springer Science & Business Media, 2012.

[94] G. Welch and G. Bishop, “An introduction to the kalman filter. 2006”, *University of North Carolina: Chapel Hill, North Carolina, US*,

[95] M. I. Ribeiro, “Kalman and extended kalman filters: concept, derivation and properties”, *Institute for Systems and Robotics*, p. 43, 2004.

[96] T. Tuytelaars and K. Mikolajczyk, “Local invariant feature detectors: a survey”, *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.

[97] S. J. Julier and J. J. LaViola, “On kalman filtering with nonlinear equality constraints”, *Signal Processing, IEEE Transactions on*, vol. 55, no. 6, pp. 2774–2784, 2007.

[98] D. Simon, “Kalman filtering with state constraints: a survey of linear and nonlinear algorithms”, *IET Control Theory & Applications*, vol. 4, no. 8, pp. 1303–1318, 2010.

[99] B. Carry, “Density of asteroids”, *Planetary and Space Science*, vol. 73, no. 1, pp. 98–118, 2012.

[100] I. Knowles and R. J. Renka, “Methods for numerical differentiation of noisy data”, in *Electronic Journal of Differential Equations, Conference*, vol. 21, 2014, pp. 235–246.