Politecnico di Milano
Department of Aerospace Science and Technology
Doctoral Programme in Aerospace Engineering

Autonomous Navigation for Close Proximity Operations around Uncooperative Space Objects

Doctoral Dissertation of:
Vincenzo Pesce

Supervisor:
Prof. Michèle Lavagna

Tutor:
Prof. Aldo Frezzotti

Coordinator:
Prof. Pierangelo Masarati

Year 2018 – Cycle XXXI

"Fall in love with some activity, and do it! Nobody ever figures out what life is all about, and it doesn't matter. Explore the world. Nearly everything is really interesting if you go into it deeply enough. Work as hard and as much as you want to on the things you like to do the best. Don't think about what you want to be, but what you want to do."

— Richard Feynman

Abstract

SPACECRAFT autonomous relative navigation is an arduous and attractive problem for future space missions. In particular, autonomy is becoming indispensable, allowing to cope with the inability to rely on commands from ground control stations (due to communications latencies and black-outs), but providing increased mission frequency, robustness, and reliability. In this thesis, innovative techniques for relative state estimation in case of uncooperative known and unknown objects, using cameras, are proposed. Several mission scenarios are examined, considering the consequent effects on the architecture of the estimation technique, its robustness and implementability. An innovative approach for vision-based relative state estimation using a mono-camera is presented along with numerical and experimental results. The novelty of this approach lies in the pose acquisition algorithm, based on a customized implementation of the RANSAC algorithm which exploits the Principal Component Analysis (PCA) and the knowledge of a simplified target model, and in the navigation filter, exploiting a linear H-∞ Filter for the translational motion and an innovative 2^{nd} Order Non-linear Filter on the Special Orthogonal group ($SO(3)$) for the rotational part. Moreover, this work offers an extensive comparative analysis between different filtering techniques for relative attitude estimation. In the same framework, relative navigation in a cislunar environment is analyzed, considering the case of a passively cooperative target and a chaser equipped with stereo-camera. Then, a novel estimation technique combining Radial Basis Function Neural Network (RBFNN) and an adaptive form of a Kalman Filter is presented and applied to relative navigation scenarios. Finally, the problem of navigating, mapping and planning around a small body (uncooperative unknown target) is tackled exploiting Partially Observable Markov Decision Process (POMDP). All the different approaches and algorithms presented in this dissertation provide satisfactory and promising results, representing possible answers to the main challenges of vision-based proximity relative navigation with uncooperative objects.

Ringraziamenti

Non amo i ringraziamenti perchè spesso si riducono ad una mera formalità, algide parole messaggere di un ben più profondo concetto. Tuttavia, sono l'unico strumento per dare il giusto riconoscimento a tutti quelli che, in maniera diversa, hanno contribuito al raggiungimento di questo traguardo. In primis, un grazie alla prof. Michèle Lavagna. Grazie per avermi dato l'opportunità di intraprendere questo percorso, di avermi sempre sostenuto e incoraggiato, di avermi trasmesso parte della sua passione e di essere stata fautrice della realizzazione di alcuni dei miei sogni. Un grazie particolare ai miei genitori, Angela e Franco. Grazie per avermi reso quello che sono, per avermi concesso di coltivare le mie passioni e per non aver mai influenzato le mie scelte. Grazie per avermi sostenuto nei momenti più difficili e gioito insieme a me nei momenti migliori. Grazie di esserci sempre, nel migliore dei modi.

Grazie al resto della famiglia, al mio fratellone Aldo e Manuela. Un grazie speciale anche a mia zia Teresa che non mi ha mai fatto mancare attenzioni degne di una madre.

Grazie agli amici che mi hanno accompagnato in questo percorso. Grazie in particolare ad Emiliano, persona unica ed amico speciale. La tua generosità e il tuo carattere solare hanno illuminato anche i giorni più bui. Grazie ad Andrea con cui ho iniziato, condiviso e terminato questo percorso. Grazie ai miei amici storici che, nonostante gli anni, sono ancora lì: Juri, Emanuele, Giancarlo, Manuel. Grazie a Claudio, Luca e Giacomo che continuano a ricordarmi di quanto terribile sia il mondo del lavoro.

Grazie ad Andrea (Bella), Paolo, Simone e Riccardo per aver contribuito a rendere le due settimane di campagna a Bordeaux indimenticabili (anche merito del 48). Un ringraziamento anche ai miei colleghi che hanno reso un piacere andare in ufficio ogni giorno: Stefano con cui ho condiviso tanto seppur in così breve tempo, Luca e le giornate al LAST, il surf, la PDA e tanto altro, Dani compagno di palestra e conterraneo (anche se il pane di Matera è più buono), Cave il cugino buono milanista, Bucci ideatore del contratto dello Spritz, di Mr Enzo, mi mancheranno le nostre discussioni KGB, Capa, i suoi orientamenti religiosi e il nostro LiciaCube, Fabio collega oltreoceano, Marco ormai principe delle startup, e ancora Louis, Chiara, Diogene. Grazie ad Abbis, il secco, grande amico e compagno interista. Grazie alla F.C. Lega BombarDAER per avermi distratto il giusto necessario durante la scrittura della tesi. Un grazie ai miei compagni di avventure al JPL, Unmesh, Bradd, Emilie, Sammy ed in particolare Luis, con cui ho condiviso notti stellate nella Death Valley, corse folli giù per il Grand Canyon e la quotidianeità della NASA. Una persona eccezionale che si è rivelata un grande amico. Un ringraziamento anche al mio mentore durante quest'esperienza oltreoceano, Ali per avermi fatto davvero capire cosa voglia dire inseguire un sogno e fare dei sacrifici per ottenerlo.

Grazie a tutti voi.

Table of Contents

List of Figures

List of Tables

List of Acronyms

A	
ADR	Active Debris Removal.
AEKF	Adaptive Extended Kalman Filter.
ANN	Artificial Neural Network.
B	
BA	Bundle Adjustment.
C	
CR3BP	Circular Restricted Three Body Problem.
D	
DCM	Direction Cosine Matrix.
E	
EKF	Extended Kalman Filter.
EO	Electro-Optical.
F	
FF	Formation Flying.
FIOS	fixed inclination orbit sequence.
FOV	Field of View.
G	
GEO	Geostationary Earth Orbit.

GNC	Guidance Navigation & Control.
H	
HEO	Highly Elliptical Orbit.
HIL	Hardware In the Loop.
J	
JPL	Jet Propulsion Laboratory.
K	
KF	Kalman Filter.
L	
LEO	Low Earth Orbit.
LIDAR	LIght Detection And Ranging.
LOS	Line-of-Sight.
M	
MEF	Minimum Energy Filter.
MEKF	Multiplicative Extended Kalman Filter.
MEKF	Multiplicative Extended Kalman Filter.
MEO	Medium Earth Orbit.
MLI	Multi-Layer Insulation.
MLP	Multi-Layer Perceptron.
MSE	Mean Square Error.
N	
NRHO	Near Rectilinear Halo Orbit.
O	
O-OS	On-Orbit Servicing.
OOS	optimal orbit sequence.
P	
PB-CAGE	Primitive Body Coverage and Geometry Evaluator.
PCA	Principal Component Analysis.
pdf	probability distribution function.
PF	Particle Filter.
POMDP	Partially Observable Markov Decision Process.

R

RANSAC	RANdom SAmple Consensus.
RBFNN	Radial Basis Function Neural Network.
RBFNN-AEKF	Radial Basis Function Neural Network Adaptive Extended Kalman Filter.
RMSE	Root Mean Square Error.
RSO	random sequence of orbits.

S

SBDT	Small Body Dynamics Toolkit.
SfM	Structure From Motion.
SLAM	Simultaneous Localitazion And Mapping.
SPC	Stereophotoclinometry.
SPG	Stereophotogrammetry.
SRP	solar radiation pressure.

U

UKF	Unscented Kalman Filter.

CHAPTER 1

Introduction

> Watching a coast as it slips by the ship is like thinking about an enigma. There it is before you, smiling, frowning, inviting, grand, mean, insipid, or savage, and always mute with an air of whispering, "Come and find out"
>
> *Heart of Darkness (1899)*
>
> Joseph Conrad

Spacecraft navigation is the action of processing measurements to determine the actual and probable future position and attitude of a vehicle. Taking a cue from the analogy in [1], spacecraft navigation can be compared to primordial ships navigation. In fact, once the ship left the port, its location was very uncertain. The sailors could then exploit some rudimentary sensors, such as the sextant or its ancestor the Jacob's staff, having a vague idea of the location of the home port and of the destination to *estimate* their position. In modern spacecraft, the navigation task is essential and it is usually carried out by exploiting sensors, dynamical models, powerful computers and ground support. With the word *navigation* we refer to absolute spacecraft navigation, i.e. the problem of localizing itself with respect to a known, inertial reference frame. In this thesis, we are interested in relative navigation, i.e. the problem of finding the relative location and attitude of two different space objects' reference frames. Moreover, the necessity to significantly enhance the level of autonomy of spacecraft to enable next-generation space missions has been recently acknowledged by major institutions like NASA [2]. In fact, autonomy allows to cope with the inability to rely on commands from ground control stations (due to communications latencies and black-outs), but it can also provide increased mission frequency, robustness, and reliability [3], [4] . In this framework, this thesis investigates the problem of autonomous relative navigation between an active satellite (chaser) and another space object (target) orbiting in close-proximity. This activity is of interest to a variety of applications, namely Formation Flying (FF) [5], On-Orbit Servicing (O-OS) of functional satellites [6] or space station [7] and Active Debris Removal (ADR) [8], [9]. Indeed, in such mission scenarios, the on-board processing unit of the chaser must be able to autonomously estimate its relative state ensuring both high accuracy and update-rate, thus being able to satisfy control requirements and minimize collision risks. During the last decades, few missions have tried to perform autonomous operations in space. However, the interest of the main space agencies towards a gradual automation of space missions is increasing because of its numerous advantages. The first mission involving autonomous proximity operations in space was the Engineering Test Satellite ETS-7 in 1997 [10]. Autonomous rendezvous and docking was performed between two unmanned spacecraft using laser, GPS and proximity sensors. In 2005, NASA sponsored the DART (Demonstration for Autonomous Rendezvous Technology) [11] mission to demonstrate automated spacecraft relative navigation and rendezvous. However, this mission failed with an unintentional collision between the two spacecraft. DARPA, in 2007, launched the Orbital Express mission [12] aimed at developing an approach for autonomous satellites servicing. The chaser spacecraft successfully performed autonomous in-orbit refueling. Another important mission demonstrating autonomous spacecraft proximity operations is PRISMA, launched in 2010 [13], [14]. During the mission, the chaser performed different maneuvers and relative approaches around the target spacecraft, demonstrating the possibility to use relative navigation techniques (i.e. GPS-based, RF or vision-based) to enable future FF missions. Depending on the category of target, different application scenario can be identified [15]. Table 1.1 reports different possible mission scenarios with the associated chaser and target hardware for each case.

Table 1.1: Relative Navigation Scenarios

Target category	**Chaser hardware**	**Target hardware**	**Mission scenario**
Actively cooperative	RF/GPS antennas	RF/GPS antennas	FF, O-OS
Passively cooperative	Relative sensors (e.g. cameras, LIDAR)	Artificial markers	FF, O-OS
Uncooperative known	Relative sensors	-	ADR, O-OS
Uncooperative unknown	Relative sensors	-	ADR, Small-body approach

In the actively cooperative target case, both chaser and target have the knowledge of their own state with a certain degree of uncertainty and they exchange information by means of a communication link. In this case, the navigation performance is usually very high and this solution can be applied to FF and O-OS scenarios in which high accuracy is required [16]. In some cases, the target may also be cooperating in a passive way, through artificial markers on the spacecraft body that can be detected and tracked by the chaser spacecraft [7]. Also in this case, very accurate navigation can be performed. When dealing with uncooperative targets, the navigation performance inevitably degrades because of the lack of information provided by the target spacecraft. For this scenario, passive or active sensors have to be used along with advanced software techniques to derive a relative state estimate. However, even a limited knowledge of the target spacecraft geometry (e.g. CAD model) can significantly improve the relative state estimation. Specific attention is also addressed to the case of uncooperative targets which are particularly difficult to approach. In fact, the lack of any a priori knowledge of the target body and the high uncertainty on its motions make the relative navigation problem particularly difficult to tackle. Consequently, advanced, ad-hoc, technological and algorithmic solutions shall be envisaged. With regards to the technological aspects, Electro-Optical (EO) sensors have been identified as the best option for relative navigation purposes when close-proximity maneuvers (e.g., rendezvous and docking) towards uncooperative targets are required [15]. In particular, either active LIght Detection And Ranging

(LIDAR) systems or passive monocular and stereo cameras can be used. The selection of the navigation sensor must consider the resources available on board in terms of mass, electrical and processing power, on one side, the mission scenario and the costs to be sustained for design and development of the satellite system, on the other side [17]. This thesis considers the use of monocular and stereo cameras. Indeed, they provide advantages with respect to LIDARs in terms of lower hardware complexity, cost, weight and power consumption and because of the possibility to be simultaneously used for supervised applications. It is worth underlying that monocular cameras have much larger operational range, not limited by the size of the platform with respect to the stereo approach but they cannot provide depth information. However, the adoption of cameras shall involve the identification of strategies to cope with their operational drawbacks, like the sensitivity to adverse illumination conditions [18] (e.g., saturation under direct Sun illumination, or absence of light during eclipse). The problem of estimating the state of an uncooperative target using passive cameras can be tackled by exploiting either a loosely-coupled or a tightly-coupled relative navigation architecture. The tightly-coupled approach consists in directly processing natural features (e.g., corners [19], edges [20], or more complex descriptors [21], [22]), extracted from the acquired images, within a filtering scheme (e.g., a Kalman Filter (KF)). Specifically, the Line-of-Sight (LOS) or the 3D position vector of each feature shall be included in the state vector [23], [24]. Consequently, the computational effort increases with the number of detected features. Moreover, tightly-coupled architectures are not robust in case of fast relative dynamics or if the target has complex geometry, since the capability to adequately track features can be compromised. This can happen also when approaching spacecraft with moving parts (e.g. Geostationary Earth Orbit (GEO) communication satellite). On the other hand, it is important to underline that these approaches are the best option when dealing with uncooperative unknown targets, for which also information on the inertia parameters can be recovered [25], [26]. Instead, in the case when at least basic information about the target geometry are available, loosely-coupled architectures are typically preferred. This term is used to indicate the fact that the relative navigation architecture is composed of two separate, consecutive blocks. First, the pose determination block analyzes the acquired images to provide an independent estimate of the target/chaser relative position and attitude. Then, this pose estimate is used within the measurement equation of the KF.

1.1 Desired Attributes of a Navigation Filter

This section contains an overview of the desirable attributes that a navigation filter should have. The qualities outlined in this list have to be taken into account while designing a navigation filter and they are strongly dependent on the application scenario.

1.1.1 Dynamical Modeling

The choice of the model to describe the state dynamics is one of the most important aspects when designing navigation filters. Analyzing a relative navigation scenario, neglecting external disturbances and flexibility of the spacecraft, the translational and the rotational dynamics are decoupled. The translational dynamics can be described by nonlinear or simplified, linearized models, in terms or relative position and velocity. Different levels of model fidelity are available considering specific formulations. In Chapter 3 the main relative translational models, adopting a classical cartesian formulation are presented. In this work, only cartesian formulations are considered but it is worth mentioning that formulations based on orbit elements can be adopted and that they are usually more efficient. For the rotational part, attitude is naturally expressed as a rotation matrix. However, classical filtering techniques exploit vectorial parametrization of rotation for the attitude representation, such as quaternions. This parametrization allows to use familiar filters like Extended Kalman Filter (EKF) but implies the non-uniqueness of the attitude representation. This is a strong weakness of this particular representation because it lead to the unwinding phenomenon [27] during the control phase. Using rotation matrix is therefore beneficial but it implies a more complicated algebra and filters. Another important aspect to consider while modeling the filter dynamics is the addition of orbital perturbations. This clearly implies an increase of the computational cost of the algorithm but it is necessary for filters with low frequency update or long operational time, when the perturbations effects become significant, like in the case of FF.

1.1.2 Robustness

A navigation filter must be robust to measurements and initialization errors. This implies that an appropriate filter should converge to the desired solution in a reasonable time and with an opportune accuracy, also in the case of measurement noise levels different from the expected ones. Moreover, it should be robust to uncertainty in the dynamical model, especially in the case of unknown or partially known targets. Finally, it should be very robust to tuning settings for both measurement and process covariance matrix.

1.1.3 Computational Cost

A suitable navigation filter for space application should be computationally efficient and inexpensive. This is due to the fact that high frequencies can be required but the computational power on-board is limited. A desired attribute is certainly the recursiveness of the adopted filtering technique. This allows to use only the last measurement instead of the past acquired data (recursive filter vs. batch filter). Recursiveness should be preferably preserved also when adaptation is required. Furthermore, it is desirable to have a practical tuning process that should not be dependent on off-line training or long processes.

1.2 Thesis Overview and Contributions

This work mostly focuses on the relative navigation techniques and possible application scenarios. In particular, innovative techniques for relative state estimation in case of uncooperative known and unknown objects, using cameras, are proposed. Several mission scenarios are examined, considering the consequent effects on the architecture of the estimation technique, its robustness and implementability. Chapter 2 introduces the basics concepts of estimation theory. In particular, some of the most common, linear and nonlinear filtering strategies are presented. In Chapter 3 the dynamical models used all along the thesis to describe the relative motion between two spacecraft are detailed. In Chapter 4 the main topic of the dissertation is discussed. An innovative approach for vision-based relative state estimation using a mono-camera is described along with numerical and experimental results. Chapter 5 offers an extensive comparison between different filtering techniques for relative attitude estimation, underlying benefits and drawbacks of each examined algorithm. In Chapter 6, relative navigation in a cislunar environment is analyzed. In particular, the case of a passively cooperative target is investigated, considering a chaser equipped with stereo-camera. Chapter 7 presents a novel estimation technique combining Radial Basis Function Neural Network (RBFNN) and an adaptive form of KF. The resulting filter is then applied to a relative navigation scenario. In Chapter 8, the problem of navigating, mapping and planning around a small body (uncooperative unknown target) is tackled. A potential solution for autonomous planning for mapping and navigation is proposed. In Chapter 9 some relevant remarks are emphasized and possible future extension of this work are introduced. The indented contribution of this this and the main innovative aspects are summarized below:

- an original loosely-coupled architecture for relative navigation toward a non-cooperative, known target, based on a single passive camera, is proposed. Innovative aspects are relevant to both pose determination algorithms and filtering scheme.
- an extensive numerical performance comparison of classical and recent nonlinear filtering techniques for relative attitude estimation is presented. The influence of the poor knowledge of the inertia matrix in filters exploiting dynamical models is investigated.
- a navigation filter formulation for relative navigation in a cislunar environment is introduced. Preliminary performance analyses are carried out, underlying possible criticality.
- a novel estimation technique combining RBFNN and adaptive KF techniques is proposed and tested in a realistic relative navigation scenario.
- the problem of planning for mapping and navigation around small bodies is tackled. An autonomous orbit selection method to maximize the small body coverage is proposed, by developing formal definitions of quantitative measures characterizing the accuracy of the small body map.

1.3 Bibliographic Disclaimer

The research presented in this thesis is the result of my original work and collaborations with others during my PhD. In particular, the work introduced in Chapter 4 was partially carried out in collaboration with the University of Naples in the framework of the project VINAG, funded by the Italian Space Agency, aimed at developing a highly integrated unit (including vision, inertial and GNSS systems) for autonomous absolute and relative navigation of spacecraft. In Chapter 8, instead, the work carried out during a 6-month period at the Jet Propulsion Laboratory is described. Furthermore, most of the work and results presented in this dissertation have already been published in different journal papers or presented at conferences. A list of major publications is here provided.

- V. Pesce, S. Silvestrini, and M. Lavagna, “Radial Basis Function Neural Network aided Adaptive Extended Kalman Filter”, *Journal of Guidance, Control and Dynamics [under review]*, 2018
- V. Pesce, M. F. Haydar, M. Lavagna, *et al.*, “Comparison of Filtering Techniques For Relative Attitude Estimation of Uncooperative Space Objects”, *Aerospace Science and Technology*, 2018. DOI: `10.1016/j.ast.2018.10.031`
- V. Pesce, R. Opromolla, S. Sarno, *et al.*, “Autonomous Relative Navigation Around Uncooperative Spacecraft Based on a Single Camera”, *Aerospace Science and Technology [under review]*, 2018
- V. Pesce, M. Lavagna, and R. Bevilacqua, “Stereovision-based pose and inertia estimation of unknown and uncooperative space objects”, *Advances in Space Research*, vol. 59, no. 1, pp. 236–251, 2017. DOI: `10.1016/j.asr.2016.10.002`
- V. Capuano, V. Pesce, R. Opromolla, *et al.*, “A Highly Integrated Navigation Unit for On-Orbit Servicing Missions”, in *69th International Astronautical Congress (IAC 2018)*, 2018, pp. 1–13
- P. Lunghi, V. Pesce, L. Losi, *et al.*, “Ground testing of vision-based GNC systems by means of a new experimental facility”, in *Proc. 69th International Astronautical Congress*, 2018
- V. Capuano, V. Pesce, R. Opromolla, *et al.*, “VINAG: A Highly integrated system for autonomous on-board absolute and relative spacecraft navigation”, in *Small Satellites Systems and Services - The 4S Symposium*, 2018
- V. Pesce, A.-a. Agha-mohammadi, and M. Lavagna, “Autonomous navigation & mapping of small bodies”, in *2018 IEEE Aerospace Conference*, IEEE, 2018, pp. 1–10. DOI: `10.1109/aero.2018.8396797`
- V. Pesce, R. Opromolla, S. Sarno, *et al.*, “Vision-Based Pose Estimation and Relative Navigation Around Uncooperative Space Objects”, in *10th International ESA Conference on Guidance, Navigation & Control Systems (GNC 2017)*, 2017, pp. 1–16
- V. Pesce, L. Losi, and M. Lavagna, “Vision-Based State Estimation of an Uncooperative Space Object”, in *68th International Astronautical Congress (IAC 2017)*, International Astronautical Federation, IAF, 2017, pp. 7018–7026

CHAPTER 2

Filtering Techniques

> [...] the first step to sanity is filtering. Filter
> the information: extract for knowledge.
>
> *David's Sling*
>
> Marc Stiegler

This chapter discusses the main alternatives when dealing with state estimation problem. In general terms, the aim of filtering is to estimate a state, at a certain time, given some measurements, exploiting dynamical and observation models. Filters are a crucial and indispensable element for spacecraft navigation. For this reason, they have to guarantee accuracy and robustness while limiting the computational burden. A detailed description of some of the most used filtering techniques is presented, underlying pros and cons of each approach.

2.1 Linear Filters

2.1.1 Kalman Filter

The most widely used and known filtering technique is certainly the KF [37], [38]. It takes its name from Rudolf E. Kálmán, a pioneer of modern estimation theory. In this section, the standard form of discrete-time KF is presented. Let's consider a linear discrete-time system:

$$\begin{aligned} \mathbf{x}_k =& \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{y}_k =& \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{2.1}$$

with $\mathbf{x}_k$ being the state vector, $\mathbf{u}_k$ the control input, $\mathbf{w}_k$ and the $\mathbf{v}_k$ the process and measurement noise with associated covariance matrix $\mathbf{Q}_k$ and $\mathbf{R}_k$; $\mathbf{y}_k$ the measurement output. $\mathbf{F}_k$, $\mathbf{G}_k$ and $\mathbf{H}_k$ are the state-transition, model-input and observation model matrices respectively. Given the estimate of the state $\hat{\mathbf{x}}_{k-1}$, the best estimate is obtained by propagating the state using the dynamical model:

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{G}_{k-1}\mathbf{u}_{k-1}. \tag{2.2}$$

This is the state prediction step of the KF. Please note that the $\hat{\mathbf{x}}^+$ represents the a posteriori estimate: the expected value of $\mathbf{x}_k$ conditioned on all the measurements up to time k. Instead, $\hat{\mathbf{x}}^-$ is the a priori estimate: the expected value of $\mathbf{x}_k$ conditioned on all the measurements up to time $k-1$. In a similar way, according to probability theory, the covariance of a linear discrete-time system can be propagated as:

$$\mathbf{P}_k^- = \mathbf{F}_{k-1}\mathbf{P}_{k-1}^+\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}. \tag{2.3}$$

This is the covariance prediction step of the KF. The next step is to derive the correction step, according to the available measurements $\mathbf{y}_k$. Without entering into the details of the derivation, the KF correction step is:

$$\begin{aligned} \hat{\mathbf{x}}_k^+ =& \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k-1}^-) \\ \mathbf{P}_k^+ =& (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k-1}^- \end{aligned} \tag{2.4}$$

with $\mathbf{K}_k$ being the KF gain matrix:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}. \tag{2.5}$$

It is worth underlying that there is not any assumption on the Gaussian nature of process and measurements noise $\mathbf{w}_k$ and $\mathbf{v}_k$. It results that the KF is the optimal filter when the noise is Gaussian, zero-mean, uncorrelated and white, but, it is still the best *linear* estimator if the Gaussian assumption does not hold.

2.1.2 H-∞ Filter

As previously discussed, for linear systems with process and measurement noise represented by a zero-mean Gaussian distribution, the KF represents the optimal estimator. In fact, if the standard deviations of the process and measurement noise are known, the KF is the minimum variance estimator. However, to satisfy these assumptions and to guarantee a good tuning of the filter, the mean of $\mathbf{w}_k$ and $\mathbf{v}_k$ and their covariance $\mathbf{Q}_k$ and $\mathbf{R}_k$ have to be known. If these assumptions are not satisfied or if the tuning is off-nominal, a filter that minimizes the worst-case estimation error rather than the variance of the estimation error could outperform the KF. This kind of filter is called H-∞ filter or also minimax filter. It minimizes the ∞-norm of the estimation error and it does not make any assumption about the statistics of the process and measurement noise [39]. The formulation of the H-∞ filter is very similar to the one of the KF. In fact, the prediction step is performed in the same way (see Equation (2.2) and 2.3) but, instead, a slightly difference is present in the correction equation:

$$\begin{aligned} \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1}^-) \\ \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_{k-1}^-, \end{aligned} \tag{2.6}$$

where the filter gain $\mathbf{K}_k$ has not the same expression of the KF. In particular $\mathbf{K}_k$ has to be chosen such that $\|\mathbf{T}_{ew}\|_\infty < \frac{1}{\theta}$, where $\mathbf{T}_{ew}$ represents the difference between the predicted and real state and θ is a tuning parameter. For the H-∞ filter, the expression of the gain is:

$$\mathbf{K}_k = \mathbf{P}_k^-[\mathbf{I} - \theta\mathbf{P}_k^- + \mathbf{H}_k{}^T\mathbf{R}_k{}^{-1}\mathbf{H}_k\mathbf{P}_k^-]^{-1}\mathbf{H}_k{}^T\mathbf{R}_k{}^{-1}. \tag{2.7}$$

Please note that the performance of the H-∞ filter is sensitive to the selection of the tuning parameter θ. However, this kind of filter is the best option when dealing with very uncertain systems and time-varying dynamics, offering a robust alternative to KF at the cost of slightly reduced accuracy.

2.2 Nonlinear Filters

KF and H-∞ Filter are linear filters that can be used when the dynamics underlying the investigated phenomenon is linear. However, linear systems are only approximations of more complex nonlinear dynamics. Many real applications involve systems that can be approximated with linear dynamics but this is not always true. When the behavior of a system cannot be described with a linear function, linear filters are no longer adequate. In all these cases, nonlinear estimators have to be employed.

2.2.1 Extended Kalman Filter

A common approach to nonlinear state estimation is to use a modified version of the standard KF to cope with the nonlinearities in the equations of the dynamics. This is the so-called EKF [40]. The idea behind the EKF is straightforward. In practice, the nonlinear system is linearized around the state estimate and the state estimate is obtained from the linearized system. In this section the discrete-time EKF is presented. Let's consider the nonlinear model:

$$\begin{aligned} \mathbf{x}_k &= f_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \\ \mathbf{y}_k &= h_k(\mathbf{x}_k, \mathbf{v}_k) \end{aligned} \tag{2.8}$$

with, as before, $\mathbf{x}_k$ the state vector, $\mathbf{u}_k$ the control input, $\mathbf{w}_k$ and the $\mathbf{v}_k$ the process and measurement noise with associated covariance matrix $\mathbf{Q}_k$ and $\mathbf{R}_k$, and $\mathbf{y}_k$ the measurement output. f_k and h_k are the functions describing the state dynamics and observation model. Performing local linearization around the state estimate, the following algorithm is obtained:

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= f_{k-1}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, 0) \\ \mathbf{P}_k^- &= \mathbf{F}_{k-1}\mathbf{P}_{k-1}^+\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \\ \mathbf{K}_k &= \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - h_k(\hat{\mathbf{x}}_k^-, 0)) \\ \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- \end{aligned} \tag{2.9}$$

where

$$\begin{aligned} \mathbf{F}_{k-1} &= \left.\frac{\partial f_{k-1}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1}^+} \\ \mathbf{H}_{k-1} &= \left.\frac{\partial h_{k-1}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_k^-} . \end{aligned} \tag{2.10}$$

The main drawback of this kind of filter is that, relying on linearization for state covariance and mean propagation, it is usually difficult to tune. Moreover, it can also produce unreliable results for highly nonlinear systems [39]. Higher order EKF can be formulated, retaining more terms in the Taylor series expansion equation (2.10).

2.2.2 Unscented Kalman Filter

The main alternative to the EKF is the Unscented Kalman Filter (UKF) [41]. This filter avoids the local linearization used by the EKF and it uses an unscented transformation to propagate the mean and covariance of the state. Let's consider a nonlinear system:

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_k \\ \mathbf{y}_k &= h(\mathbf{x}_k) + \mathbf{v}_k. \end{aligned} \tag{2.11}$$

The update step is performed by considering $2n$ sigma points, where n is the dimension of the state vector. The sigma points are selected according to:

$$\begin{aligned} \hat{\mathbf{x}}_{k-1}^{(i)} &= \hat{\mathbf{x}}_{k-1}^{+} + \tilde{\mathbf{x}}^{(i)} \quad i = 1, ..., 2n \\ \tilde{\mathbf{x}}^{(i)} &= \left(\sqrt{n\mathbf{P}_{k-1}^{+}}\right)_i^T \quad i = 1, ..., n \\ \tilde{\mathbf{x}}^{(n+i)} &= -\left(\sqrt{n\mathbf{P}_{k-1}^{+}}\right)_i^T \quad i = 1, ..., n \end{aligned} \tag{2.12}$$

and then propagated using the dynamical model:

$$\hat{\mathbf{x}}_k^{(i)} = f(\hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_k). \tag{2.13}$$

Then, we can combine the $2n$ vectors to obtain the a priori state estimate and its covariance:

$$\begin{aligned} \hat{\mathbf{x}}_k^{-} &= \frac{1}{2n}\sum_{i=1}^{2n} \hat{\mathbf{x}}_k^{(i)} \\ \mathbf{P}_k^{-} &= \frac{1}{2n}\sum_{i=1}^{2n} \left(\hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k^{-}\right)\left(\hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k^{-}\right)^T + \mathbf{Q}_{k-1}. \end{aligned} \tag{2.14}$$

In a similar way, the correction step is performed. First other $2n$ sigma points are selected based on the a priori estimate:

$$\begin{aligned} \hat{\mathbf{x}}_{k-1}^{(i)} &= \hat{\mathbf{x}}_{k}^{-} + \tilde{\mathbf{x}}^{(i)} \quad i = 1, ..., 2n \\ \tilde{\mathbf{x}}^{(i)} &= \left(\sqrt{n\mathbf{P}_{k}^{-}}\right)_i^T \quad i = 1, ..., n \\ \tilde{\mathbf{x}}^{(n+i)} &= -\left(\sqrt{n\mathbf{P}_{k}^{-}}\right)_i^T \quad i = 1, ..., n. \end{aligned} \tag{2.15}$$

For a faster algorithm, the sigma points obtained before can be used for this step but this will result in lower performance. At this point, the observation model is used to predict the measurements:

$$\hat{\mathbf{y}}_k^{(i)} = h(\hat{\mathbf{x}}_k^{(i)}). \tag{2.16}$$

Finally, the predicted measurement covariance and the cross covariance can be computed:

$$\begin{aligned} \mathbf{P}_y =& \frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{\mathbf{y}}_k^{(i)}-\hat{\mathbf{y}}_k^-\right)\left(\hat{\mathbf{y}}_k^{(i)}-\hat{\mathbf{y}}_k^-\right)^T+\mathbf{R}_{k-1} \\ \mathbf{P}_{xy} =& \frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{\mathbf{x}}_k^{(i)}-\hat{\mathbf{x}}_k^-\right)\left(\hat{\mathbf{y}}_k^{(i)}-\hat{\mathbf{y}}_k^-\right)^T . \end{aligned} \tag{2.17}$$

This lead to the a posteriori state estimate:

$$\begin{aligned} \hat{\mathbf{x}}_k^+ =& \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k-\hat{\mathbf{y}}_k) \\ \mathbf{P}_k^+ =& \mathbf{P}_k^- - \mathbf{K}_k\mathbf{P}_y\mathbf{K}_k^T \\ \mathbf{K}_k =& \mathbf{P}_{xy}\mathbf{P}_y^{-1}. \end{aligned} \tag{2.18}$$

The presented formulation is valid under the assumption of additive noise. If this assumption is not valid, a different formulation has to be derived [39]. The UKF presents some advantages with respect to the EKF and overcome its intrinsic limitations. In fact, the computation of Jacobians is not required in this case. However, computing the propagation of $2n$ sigma points can be computationally demanding in case of complex system dynamics. An efficient formulation of the UKF has recently been proposed in [42]

2.2.3 Particle Filter

Particle Filters (PFs) represent an alternative to EKF and UKF. They were first introduced in 1993 [43] with the name of bootstrap filter. The key idea underlying the particle filter is to approximate the filtering density function as a weighted set of samples, also called particles. Its representation is fundamentally different from the one used in the KF, where a specific functional form of the density function is assumed and the estimate is then represented by the parameters, (the mean and the covariance) parameterizing this density. In the PF the filtering density is represented as a set of random samples approximately distributed according to this density. As before, a generic nonlinear system can be described as:

$$\begin{aligned} \mathbf{x}_k =& f_{k-1}(\mathbf{x}_{k-1},\mathbf{u}_{k-1},\mathbf{w}_{k-1}) \\ \mathbf{y}_k =& h_k(\mathbf{x}_k,\mathbf{v}_k) \end{aligned} \tag{2.19}$$

For PF the *Bayesian* approach to nonlinear state estimation is introduced. The aim of this strategy is to compute or approximate the posterior distribution of the state, given the observations. In particular, the *Bayesian* recursive solution to compute the posterior distribution ($p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$) of the state, given past observation, is given by 2.20.

$$\begin{aligned} Initialization: p(\mathbf{x}_0 \mid \mathbf{Y}_{k-1}) &= p(\mathbf{x}_0) \\ Prediction: p(\mathbf{x}_k \mid \mathbf{Y}_{k-1}) &= \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1})\, p(\mathbf{x}_{k-1} \mid \mathbf{Y}_{k-1}) d\mathbf{x}_{k-1} \\ Correction: p(\mathbf{x}_k \mid \mathbf{Y}_k) &= \frac{p(\mathbf{y}_k \mid \mathbf{x}_k)\ p(\mathbf{x}_k \mid \mathbf{Y}_{k-1})}{\int p(\mathbf{y}_k \mid \mathbf{x}_k)\ p(\mathbf{x}_k \mid \mathbf{Y}_{k-1})} \end{aligned} \tag{2.20}$$

with $\mathbf{Y}_k = \mathbf{y}_{1:k}$.

For a general case, there is not an explicit solution for this integral. However, for a linear system with Gaussian noises, the classical KF recursive form provides the solution for the presented *Bayesian* problem. For a generic non linear system, with non-Gaussian noises, it is necessary to rely on numerical approximations. PFs offer a tool to approximately solve the recursion for a generic system. In particular, M random points (particles) are generated at the beginning of the estimation, based on the initial probability density function (pdf) of the state. In fact, it is reasonable to approximate the pdf as sum of δ and in particular:

$$p(\mathbf{x}_k \mid \mathbf{Y}_k) \simeq \hat{p}(\mathbf{x}_k \mid \mathbf{Y}_k) = \frac{1}{M}\sum_{i=1}^{M} \delta(\mathbf{x}_k - \bar{\mathbf{x}}_{k,i}) \tag{2.21}$$

where $\bar{\mathbf{x}}_{k,i}$ are the particles extracted from the true conditional density. Therefore, the recursive algorithm can be derived as follows:

$$\begin{aligned} Initialization: p(\mathbf{x}_0) &\simeq \frac{1}{M}\sum_{i=1}^{M} \delta(\mathbf{x}_0 - \bar{\mathbf{x}}_{0,i}) \\ Estimation: \hat{p}(\mathbf{x}_{k-1} \mid \mathbf{Y}_{k-1}) &\simeq \frac{1}{M}\sum_{i=1}^{M} \delta(\mathbf{x}_{k-1} - \bar{\mathbf{x}}_{k-1,i}) \\ Prediction: p(\mathbf{x}_k \mid \mathbf{Y}_{k-1}) &\simeq \frac{1}{M}\sum_{i=1}^{M} \delta(\mathbf{x}_k - \bar{\bar{\mathbf{x}}}_{k,i}) \\ Correction: p(\mathbf{x}_k \mid \mathbf{Y}_k) &= \sum_{i=1}^{M} \mathbf{q}_i\, \delta(\mathbf{x}_k - \bar{\bar{\mathbf{x}}}_{k,i}) \end{aligned} \tag{2.22}$$

where $\bar{\bar{\mathbf{x}}}_{k,i} = f(\bar{\mathbf{x}}_{k-1,i}, \bar{\mathbf{w}}_{k-1,i})$, $\bar{\mathbf{w}}_{k-1,i} \sim p(\mathbf{w}_{k-1})$ and $\mathbf{q}_i = \dfrac{p(\mathbf{y}_k \mid \bar{\bar{\mathbf{x}}}_{k,i})}{\sum_{i=1}^{M} p(\mathbf{y}_k \mid \bar{\bar{\mathbf{x}}}_{k,i})}$

Note that the second set of points, $\bar{\bar{\mathbf{x}}}_{k,i}$ is extracted from the already defined grid and the particles are propagated according to the non-linear dynamic of the system. The term $\mathbf{q}_i$ indicates the relative probability of each particle and it can be seen as a weight. The sum of all the weights is equal to 1. However, if we do not perform resampling, the particle filter would end to a set independent simulations, each with its own weight or probability. Most likely, the so called sample depletion would occur. This means that, not having any feedback from the observation, the results would be to have all the weights tending to zero, except for one that will be almost equal to 1. The high value of the weight does not mean that the estimated state is close to the real one but just that one sequence in the set of particles is more likely that the others. Resampling introduces the feedback from the observation and guarantees that the good state estimation does not disappear. Like the UKF, the PF method does not rely on any local linearization technique and do not have any constraint on the noise distribution. This flexibility can be very useful in several applications in which the EKF does not perform well (highly non linearities involved). However, all these advantages have a cost. In fact, the bottleneck of the PF is its computational load [44]. If we think at the EKF, only one function evaluation of $f(\mathbf{x}_k, \mathbf{w}_k)$ and $h(\mathbf{x}_k, \mathbf{v}_k)$ is required at each time step (note that if the Jacobian is not analytically available, more than one evaluation of these functions is needed). Per contra, M evaluations are needed with particle filter. This can become very demanding in systems with highly non-linear and complicated dynamics and sampling with an high number of particles. This is one of the reasons why PF has never been practically implemented on any spacecraft. Summarizing, the PF overcomes the intrinsic limitations of the EKF but the computational cost is much higher.

2.2.4 Minimum Energy Filter

Minimum Energy Filter (MEF) is a particular estimator in which dynamics and measurement noises are modeled deterministically. This is different from stochastic filtering techniques, i.e. KF, for which the noise in the dynamical propagation and observation model are assumed to be stochastic white-noise. In general, MEF can be seen as a generalization of the least square methods. Mortensen [45] firstly proposed to apply this method to filtering. His MEF formulation coincides with a KF for linear systems but it differs when dealing with nonlinear ones. In these cases, a MEF approach can outperform standard nonlinear KF techniques without significantly increasing the computational effort. The author has chosen not to present any particular formulation of MEF in this introductory section because of the complex mathematical derivation. A more detailed description of MEFs for attitude estimation, formulated on the Lie group, are presented in Chapters 4 and 5.

CHAPTER 3

Spacecraft Relative Dynamics

All the effects of nature are only the
mathematical consequences of a small number
of immutable laws.
Théorie des probabilités (1847)
PIERRE-SIMON LAPLACE

THE outcome of any state estimation technique strongly depends on the choice of the dynamical model describing the evolution of the state. The choice of the proper dynamical model is always a trade-off between accuracy and computational cost. In the particular case of the estimation of spacecraft relative pose, it is necessary to adopt the appropriate dynamical model depending on the considered application. In this chapter, the main dynamical models for the description of the relative translational and rotational motion between two spacecraft are described.

3.1 Coordinate Systems

In this section, the coordinate systems used in the following descriptions are introduced.

3.1.1 Earth-centered Coordinate Systems

Two spacecraft orbiting the Earth are considered, a *chaser* and a *target* respectively. The reference frames used in this formulation are: an Earth-centered, inertial reference frame, $\mathcal{I}$ as in Fig.3.1, a local-vertical, local-horizontal (LVLH) reference frame fixed to the chaser spacecraft center of mass, $\mathcal{C}$ with $\hat{x}$ being a unit vector directed radially outward, $\hat{z}$ normal to the spacecraft orbital plane, and $\hat{y}$ completes the triad; a Cartesian body-fixed reference frame attached to the target spacecraft center of mass, $\mathcal{T}$.

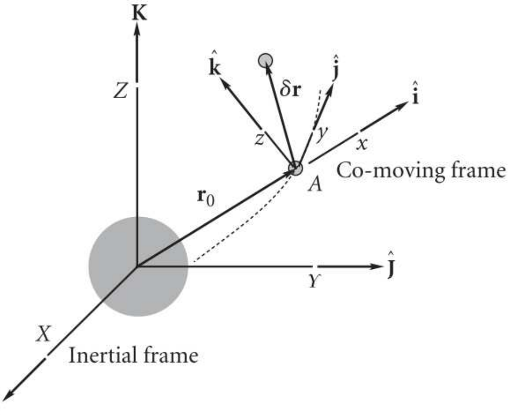


Figure 3.1: Co-moving LVLH frame [46].

3.1.2 Moon-Earth Coordinate Systems

In a cislunar environment, instead, according to [47], the relative dynamics between two bodies of masses m_T and m_C, target and chaser respectively, can be expressed in the inertial reference frame $\mathcal{I}_{Cl}$, as in Figure 3.2, centered at the center of mass O of the primaries m_E (Earth) and m_M (Moon), and defined by the versors $\hat{\mathbf{X}}$, $\hat{\mathbf{Y}}$ and $\hat{\mathbf{Z}}$. Analogously to Section 3.1.1, Cartesian body-fixed reference frames centered in the chaser and target center of mass can be defined $\mathcal{C}_{Cl}$ and $\mathcal{T}_{Cl}$ of versors $\hat{\mathbf{c}}_1$, $\hat{\mathbf{c}}_2$ and $\hat{\mathbf{c}}_3$, $\hat{\mathbf{t}}_1$, $\hat{\mathbf{t}}_2$ and $\hat{\mathbf{t}}_3$ respectively.

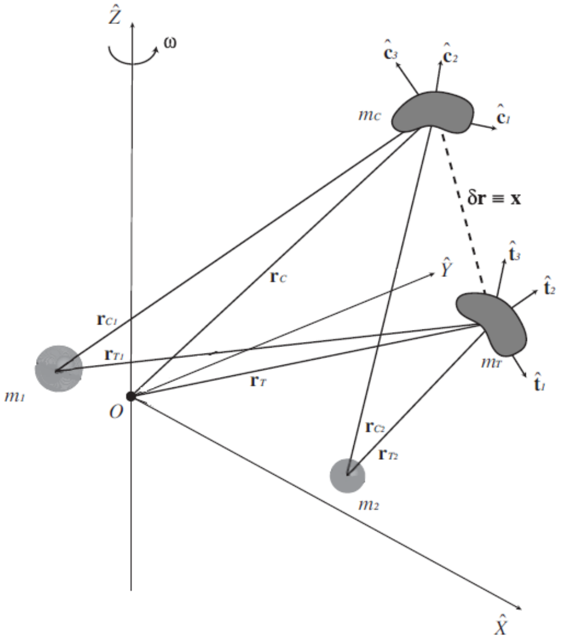


Figure 3.2: Cislunar Relative Dynamics Reference Frames [47]

3.2 Relative Translational Dynamics - Earth

In this section, different alternatives to describe the relative translational motion between two spacecraft orbiting the Earth are presented.

3.2.1 Nonlinear Dynamic Model of Unperturbed Relative Motion

In the most general case, the relative motion in unperturbed orbits can be described by a set of nonlinear differential equations, without any restrictive assumptions or linearization. In particular, the translational dynamics of the target with respect to the chaser, expressed in the chaser frame $\mathcal{C}$, can be written as:

$$\begin{aligned}
\delta\ddot{x} - 2\dot{f}_C\delta\dot{y} - \ddot{f}_C\delta y - \dot{f}_C^2\delta x &= \frac{\mu(r_C+\delta x)}{[(r_C+\delta x)^2+\delta y^2+\delta z^2]^{\frac{3}{2}}} + \frac{\mu}{r_C^2} \\
\delta\ddot{y} + 2\dot{f}_C\delta\dot{x} + \ddot{f}_C\delta x - \dot{f}_C^2\delta y &= \frac{\mu\delta y}{[(r_C+\delta x)^2+\delta y^2+\delta z^2]^{\frac{3}{2}}} \\
\delta\ddot{z} &= \frac{\mu\delta z}{[(r_C+\delta x)^2+\delta y^2+\delta z^2]^{\frac{3}{2}}}
\end{aligned} \tag{3.1}$$

with $\dot{f}_C$ and $\ddot{f}_C$ being the chaser orbital angular velocity and acceleration respectively.

3.2.2 Clohessy-Wiltshire Equations

The most common set of equations to describe the relative dynamics between two spacecraft are the well-known Clohessy-Wiltshire[48]. They hold for nearly circular orbits, small target-chaser distance compared to the orbit radius, and spherical Earth. For a full derivation, the author suggests to refer to [46]. With reference to Figure 3.1, the chaser spacecraft is in a reference orbit at a distance $\mathbf{r_0}$ from the attractor. If the target is in close proximity of the chaser, the orbital radius can be expressed as $\mathbf{r} = \mathbf{r_0} + \delta\mathbf{r}$. The equation of motion of the target spacecraft is:

$$\ddot{\mathbf{r}} = -\mu\frac{\mathbf{r}}{r^3} \tag{3.2}$$

where μ is the gravitational constant of the orbital motion. With some mathematical manipulations, the Hill's equations can be obtained:

$$\begin{aligned} \delta\ddot{x} - 3n^2\delta x - 2n\delta\dot{y} &= 0 \\ \delta\ddot{y} + 2n\delta\dot{x} &= 0 \\ \delta\ddot{z} - n^2\delta z &= 0 \end{aligned} \tag{3.3}$$

where $n = \frac{2\pi}{T}$ is the orbital mean motion.

3.2.3 Yamanaka-Ankersen Model

In 2002, Yamanaka and Ankersen [49] proposed a state transition matrix for relative motion on an arbitrary elliptical orbit. This convenient formulation is very useful when dealing with filtering equations. However, at periapsis it can lose accuracy due to the high speed. The details of the derivation can be found in [49]. In this section, only the implemented state transition matrix is presented. The propagation in time of the relative position and velocity can be expressed as:

$$\begin{bmatrix} \delta x_t \\ \delta y_t \\ \delta\dot{x}_t \\ \delta\dot{y}_t \end{bmatrix} = \begin{bmatrix} 0 & -s & -c & -(2-3esJ) \\ 1 & -c(1+1/\rho) & s(1+1/\rho) & 3\rho^2 J \\ 0 & -s' & -c' & 3e(s'J + s/\rho^2) \\ 0 & 2s & 2c-e & 3(1-2esJ) \end{bmatrix}_\theta \begin{bmatrix} \delta\tilde{x}_0 \\ \delta\tilde{y}_0 \\ \delta\dot{\tilde{x}}_0 \\ \delta\dot{\tilde{y}}_0 \end{bmatrix} \tag{3.4}$$

with

$$\begin{bmatrix} \delta\tilde{x}_0 \\ \delta\tilde{y}_0 \\ \delta\dot{\tilde{x}}_0 \\ \delta\dot{\tilde{y}}_0 \end{bmatrix} = \frac{1}{1-e^2} \begin{bmatrix} 0 & 3s(1/\rho + e^2/\rho^2) & -s(1+1/\rho) & 2e-c \\ 1-e^2 & 3es(1/\rho+1/\rho^2) & -es(1+1/\rho) & -ec+2 \\ 0 & 1-e^2-3\rho & \rho^2 & -es \\ 0 & -3(c/\rho+e) & c(1+1/\rho)+e & -s \end{bmatrix}_{\theta_0} \begin{bmatrix} \delta x_0 \\ \delta y_0 \\ \delta\dot{x}_0 \\ \delta\dot{y}_0 \end{bmatrix}. \tag{3.5}$$

And for the out-of-plane component:

$$\begin{bmatrix} \delta z_t \\ \delta\dot{z}_t \end{bmatrix} = \frac{1}{\rho_{\theta-\theta_0}} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}_{\theta-\theta_0} \begin{bmatrix} \delta z_0 \\ \delta\dot{x}_0 \end{bmatrix}, \tag{3.6}$$

where

$$\begin{aligned} \rho &= 1 + e\cos(\theta) \\ s &= \rho\sin\theta \\ c &= \rho\cos\theta \\ s' &= \cos\theta + e\cos 2\theta \\ c' &= -\left(\sin\theta + e\sin 2\theta\right) \\ J &= k^2(t-t_0) \\ k^2 &= h/p^2 \end{aligned} \tag{3.7}$$

with θ, true anomaly that can be computed using Kepler's equation, p the semilatus rectum and $|_\theta$ meaning evaluated at the true anomaly θ at the current step.

3.2.4 Nonlinear Dynamic Model of J_2-Perturbed Relative Motion

If we want to consider orbital perturbation due to Earth oblateness, we have to adopt the nonlinear dynamical model for J_2-perturbed relative orbit, derived in [50]. Hereby, the fundamental equations are solely reported; for a thorough derivation, please refer to [50]. With reference to Figure 3.1, the relative dynamics in the chaser reference frame $\mathcal{C}$ can be written as:

$$\begin{aligned} \delta\ddot{x} &= 2\delta\dot{y}\omega_z - \delta x(n_j^2 - \omega_z^2) + \delta y\alpha_z - \delta z\omega_x\omega_z - (\zeta_j - \zeta)s_i s_\theta - r(n_j^2 - n^2) + a_x \\ \delta\ddot{y} &= -2\delta\dot{x}\omega_z + 2\delta z\omega_x - \delta x\alpha_z - \delta y(n_j^2 - \omega_z^2 - \omega_x^2) + \delta z\alpha_x - (\zeta_j - \zeta)s_i c_\theta + a_y \\ \delta\ddot{z} &= -2\delta\dot{y}\omega_x - \delta x\omega_x\omega_z - \delta y\alpha_x - \delta z(n_j^2 - \omega_x^2) - (\zeta_j - \zeta)c_i + \alpha_z \end{aligned} \tag{3.8}$$

where the contributing terms are:

$$\begin{aligned}
n^2 &= \frac{\mu}{r_0^3} + \frac{k_{J2}}{r_0^5} - \frac{5k_{J2}s_i^2 s_\theta^2}{r^5} \\
n_j^2 &= \frac{\mu}{r^3} + \frac{k_{J2}}{r^5} - \frac{5k_{J2}r_{JZ}^2}{r^7} \\
r_{JZ} &= (r_0 + \delta x)s_i s_\theta + \delta y s_i c_\theta + \delta z c_i \\
k_{J2} &= \frac{3J_2\mu R_e^2}{2} \\
\omega_x &= -\frac{k_{J2}s_{2i}s_\theta}{hr_0^3} \\
\omega_z &= \frac{h}{r_0^2} \\
\alpha_x &= \dot{\omega}_x = \frac{k_{J2}s_{2i}c_\theta}{r_0^5} + \frac{3\dot{r}_0 k_{J2}s_{2i}s\theta}{r_0^4 h} - \frac{8k_{J2}^2 s_i^3 c_i s_\theta^2 c_\theta}{r^6 h^2} \\
\alpha_z &= \dot{\omega}_z = -\frac{2h\dot{r}_0}{r_0^3} - \frac{k_{J2}s_i^2 s_{2\theta}}{r_0^5} \\
\zeta &= \frac{2k_{J2}s_i s_\theta}{r^4} \\
\zeta_j &= \frac{2k_{J2}r_{JZ}}{r^5}
\end{aligned} \tag{3.9}$$

in which h is the orbital angular momentum, J_2 is the zonal harmonic coefficient ($1.0826 \cdot 10^{-3}$ for Earth), R_e is the Earth radius. The spacecraft relative motion is, therefore, described by 11 first-order differential equations, namely $(\delta x, \delta y, \delta z, \delta\dot{x}, \delta\dot{y}, \delta\dot{z})$ and $(r, \dot{r}, h, i, \theta)$.

3.3 Relative Translational Dynamics - Moon/Earth

In the Earth-Moon space, a different model has to be adopted, considering the different gravitational effects that are involved. In particular, the absolute orbit dynamics is expressed using $\mathbf{r}_T$ and $\mathbf{r}_C$, the position vectors of the centers of mass of the bodies m_T and m_C respectively. The relative translational dynamics in such environment can be obtained from the absolute dynamics expressed in the frame $\mathcal{I}_{Cl}$ [47]. The relative translational dynamics is immediately available from the definition of the relative position vector, $\delta\mathbf{r}$:

$$\mathbf{x} \equiv \delta\mathbf{r} = \mathbf{r}_T - \mathbf{r}_C, \tag{3.10}$$

expressed in the inertial reference frame. Differentiating the expression of the relative position vector twice in time we obtain:

$$\ddot{\mathbf{x}} = \ddot{\mathbf{r}}_T - \ddot{\mathbf{r}}_C, \tag{3.11}$$

where $\ddot{\mathbf{r}}_C$ and $\ddot{\mathbf{r}}_T$ are the absolute acceleration vectors of chaser and target and they can be expressed as [47]:

$$\ddot{\mathbf{r}}_C = -\frac{\mu_E}{r_{C_E}^3}\mathbf{r}_{C_E} - \frac{\mu_M}{r_{C_M}^3}\mathbf{r}_{C_M} + \mathbf{a}_{C_S} + \mathbf{a}_{C_{SRP}}, \tag{3.12}$$

$$\ddot{\mathbf{r}}_T = -\frac{\mu_E}{r_{T_E}^3}\mathbf{r}_{T_E} - \frac{\mu_M}{r_{T_M}^3}\mathbf{r}_{T_M} + \mathbf{a}_{T_S} + \mathbf{a}_{T_{SRP}}, \tag{3.13}$$

where $\mu_E = Gm_E$ and $\mu_M = Gm_M$ are the dimensional mass parameters. The terms $\mathbf{a}_{C_S}$, $\mathbf{a}_{T_S}$ and $\mathbf{a}_{C_{SRP}}\mathbf{a}_{T_{SRP}}$ are the perturbing accelerations in cislunar space acting on the spacecraft due to the gravitational presence of the Sun and due to the solar radiation pressure (SRP). In this work, we considered an ephemeris model. In practice, the position of Earth, Moon and Sun are extracted from numerical ephemerides contained in the SPICE Toolkit by NASA/JPL. A simpler alternative is represented by the Circular Restricted Three Body Problem (CR3BP) model [51] to describe the motion of spacecraft, with negligible mass, under the gravitational attraction of two primaries, with masses m_T and m_C. This model is a valuable alternative for preliminary analyses but it neglects some effect that may be relevant for some orbits (e.g. Near Rectilinear Halo Orbit (NRHO), [52]). The CR3BP is not considered in this work because it does not provide generally valid approximations of the relative dynamics in the Earth-Moon space [53].

Linearized Model For computational and implementation reasons, it can be useful to have a linear model describing the relative dynamics. Considering the equation 3.11, a linearization can be performed by assuming the relative distance between chaser and target to be small compared to the distance between the chaser and the primaries: $\|\mathbf{x}\| \ll r_{C_E}$ and $\|\mathbf{x}\| \ll r_{C_M}$. In this case, performing a first order expansion of Equation (3.11) we obtain [53], [54]:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \ddot{\mathbf{x}} \end{bmatrix} \approx \begin{bmatrix} \mathbf{0} & \mathbf{I}_{3\times3} \\ \boldsymbol{\Xi}(t) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{3\times3} \end{bmatrix} (\delta\mathbf{a}_S + \delta\mathbf{a}_{SRP}), \tag{3.14}$$

where $\Xi(t)$ is a matrix depending on the known absolute orbital state of the chaser:

$$\begin{aligned}\Xi(t) = &-\left(\frac{\mu_E}{r_{C_E}^3} + \frac{\mu_M}{r_{C_M}^3}\right)\mathbf{I}_{3\times3} + 3\frac{\mu_E}{r_{C_E}^3}\left[\hat{r}_{C_E}\hat{r}_{C_E}^{\mathrm{T}}\right] \\ &+ 3\frac{\mu_M}{r_{C_M}^3}\left[\hat{r}_{C_M}\hat{r}_{C_M}^{\mathrm{T}}\right].\end{aligned} \tag{3.15}$$

The effects of the discussed perturbations $\mathbf{a}_S$ and $\mathbf{a}_{SRP}$ are treated as additive perturbative terms also in the linearized dynamic model.

3.4 Relative Rotational Dynamics and Kinematics

In this section, the models describing the rotational motion of the target relative to the chaser are illustrated. Please note that the rotational dynamics and kinematics are described by the same set of equations in both Earth or cislunar environment. The only difference are the perturbations acting on the system. For simplicity, the notation of the Earth reference frames is used in this section. To parametrize the relative attitude, a rotation matrix $\mathbf{R}_{\mathcal{CT}}$ which performs the transformation from the target body fixed frame $\mathcal{T}$ to the chaser body fixed frame $\mathcal{C}$ is considered. The components of this matrix are combinations of relative quaternions q_0, q_1, q_2, q_3. Knowing the rotation matrix $\mathbf{R}_{\mathcal{CT}}$, the relative angular velocity in both target and chaser frames can be calculated.

$$\boldsymbol{\omega}|_{\mathcal{C}} = \mathbf{R}_{\mathcal{CT}}\boldsymbol{\omega}_T|_{\mathcal{T}} - \boldsymbol{\omega}_C|_{\mathcal{C}}. \tag{3.16}$$

Combining the Euler equations for both chaser and target, the relative rotational dynamics can be expressed as [55]:

$$\begin{aligned}\mathbf{I}_C\dot{\boldsymbol{\omega}} =& \mathbf{I}_C\mathbf{R}_{\mathcal{CT}}\mathbf{I}_T^{-1}[\mathbf{N}_T - \mathbf{R}_{\mathcal{CT}}^T(\boldsymbol{\omega}|_{\mathcal{C}} + \boldsymbol{\omega}_C|_{\mathcal{C}}) \times \mathbf{I}_T\mathbf{R}_{\mathcal{CT}}^T(\boldsymbol{\omega}|_{\mathcal{C}} + \boldsymbol{\omega}_C|_{\mathcal{C}})] \\ &- \mathbf{I}_C\boldsymbol{\omega}_C|_{\mathcal{C}} \times \boldsymbol{\omega}|_{\mathcal{C}} - [\mathbf{N}_C - \boldsymbol{\omega}_C|_{\mathcal{C}} \times \mathbf{I}_C\boldsymbol{\omega}_C|_{\mathcal{C}}].\end{aligned} \tag{3.17}$$

Where $\mathbf{N}_T$ and $\mathbf{N}_C$ are the external torques on the target and chaser. In the previous expressions, 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. Using the same quaternion parametrization, the relative attitude kinematics can be described by:

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{Q}\boldsymbol{\omega}|_{\mathcal{T}} \tag{3.18}$$

with

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

In some cases, however, it is more convenient to express the relative attitude kinematics in terms of rotation matrix, without using any additional parametrization:

$$\dot{\mathbf{R}}_{\mathcal{CT}}(t) = \mathbf{R}_{\mathcal{CT}}(\boldsymbol{\omega}(t)_\times), \quad \mathbf{R}_{\mathcal{CT}}(0) = \mathbf{R}_{\mathcal{CT}0}, \tag{3.19}$$

where $\mathbf{R}_{\mathcal{CT}}(t) \in SO(3)$ is the relative rotation matrix expressed in the chaser frame, $\boldsymbol{\omega}$ is the relative angular velocity expressed in the chaser frame and the lower index operator $(.)_\times$ represents the skew-symmetric matrix.

CHAPTER 4

Vision-based Relative Navigation around an Uncooperative-known Space Object

> I roamed the infinite sky, and soared the ideal
> world, and floated through the firmament.
> But here I am, prisoner of measurement.
> *Song of Humanity*
> KAHLIL GIBRAN

IN this chapter the problem of navigating around an uncooperative-known space object is analyzed. This scenario is the most attractive among the ones presented in Table 1.1, In fact, it can be applied to O-OS and ADR missions and, in such cases, at least basic information about target geometry is assumed to be available and can be exploited within the developed relative navigation architecture. Under such condition, loosely-coupled architectures are typically preferred. Therefore, in this work, a loosely-coupled architecture for relative navigation based on monocular images as well as on the knowledge of a simplified target model is presented. Specifically, original contributions are provided with respect to the state of the art in terms of both pose determination and filtering solutions. With regards to the pose determination block, several monocular techniques have been recently proposed [15], and their performance has been tested using synthetic data [56], real images from past missions [57], [58] or experimental tests [59], [60]. These studies have shown that main open challenges are related to the pose initialization, i.e., the initial estimation of the pose parameters carried out when no prior information is available about the target pose [15]. Specifically, the necessity to limit the computational load as well as to enhance robustness against variability of pose conditions has been identified. To this aim, two different strategies for pose determination are presented in this chapter. The Strategy A is based on a customized implementation of the RANdom SAmple Consensus (RANSAC) algorithm which exploits the Principal Component Analysis (PCA) and the knowledge of a simplified target model. Furthermore, pose tracking, i.e., the problem of updating the pose parameters based on new measurements, is carried out exploiting the SoftPOSIT algorithm [61]. The main innovative aspects of the strategy are in the target pose acquisition and in the kinematic filter, on the Lie group, used for relative attitude estimation. The algorithm implemented for Strategy B works detecting salient features from the incoming images, i.e. ORB descriptors [22]; among these, the ones corresponding to the uncooperative object are matched to an already available on-board map. In this way, a set of 3D to 2D correspondences is obtained. The set of correspondences is then used to solve the so-called Perspective-n-Point problem (PnP) [62], which gives as result a first estimate of the relative pose between the two spacecraft. Motion only Bundle Adjustment (BA) [63], is then applied for pose optimization. The main contribution of this second strategy is to proposed a feature-based method for relative pose estimation of an object in space. The presented Strategy A and B have different level of maturity. The first one was developed in collaboration with the University of Naples in the framework of the VINAG project [33]. Strategy B instead, was developed entirely, more recently, at Politecnico di Milano, Aerospace Science and Technology Department (DAER) and it is still under intense improvement. Despite the selected strategy, once a pose estimate is available, this measurement is typically filtered to improve the state estimation. The EKF is the standard approach for space applications. This solution has been used also for several relative navigation techniques with different sensors architectures [25], [64]–[66]. Two different strategies exist for relative navigation filters architectures. Specifically, a single filter for both translation and rotation can be exploited or the dynamics can be decoupled. The first approach is necessary when the measurements are a combination of relative translational position and angular velocity [25], [66] (i.e. measurements are directly the feature points in a tightly-coupled configuration). However, due to the nonlinear nature of the rotational dynamics, the choice of the filter is limited to sub-optimal nonlinear approaches such as EKF or UKF. This implies a higher computational time (with respect to linear filters) and a degradation of the performance with non-nominal tuning and non-Gaussian measurement noise. Instead, if the measurements are directly the relative position and relative attitude, the two filters can be decoupled. In our case, the nature of the problem and the classical limited duration of proximity operations suggest to decouple the relative translational and rotational dynamics by neglecting external disturbances and spacecraft flexibility. Another important point to underline is that, typically, the determination of the relative position is usually more robust with respect to relative attitude estimation [15], which is more prone to errors and ambiguities. For this reason, a decoupled strategy in which the translational and rotational filters work independently, guarantees the control of the proximity phase also in the case of poor relative attitude estimation. The proposed architecture exploits a linear H-∞ Filter for the translational motion and a 2^{nd} Order Non-linear Filter on the Special Orthogonal group ($SO(3)$) for the rotational part. The H-∞ Filter guarantees robustness by minimizing the ∞-$norm$ of the estimation error and it does not make any assumptions about the statistics of the process and measurement noise [39]. The implemented 2^{nd} Order Non-linear Filter is based on a novel formulation on the $SO(3)$. This choice is justified by the fact that filters exploiting a dynamics directly expressed on the $SO(3)$ have shown better performance than filters designed using other attitude parameterizations [67]–[69].

4.1 Monocular Relative Navigation Architecture

The architecture proposed for relative navigation of a chaser satellite with respect to an uncooperative target is presented in the block diagram in Figure 4.1.

This architecture is loosely-coupled since it is composed of two separate blocks. The pose estimator processes the images acquired by a monocular camera to compute the target/chaser position and attitude parameters, which are then used as input for the navigation filter. The pose estimator exploits a simplified model of the target which is built off-line thanks to the knowledge of the target geometry or optical properties (e.g., a CAD model or feature points descriptors). The navigation filter is also divided in two components responsible of the relative rotational and translational dynamics, respectively. A link between the two blocks is ensured in the

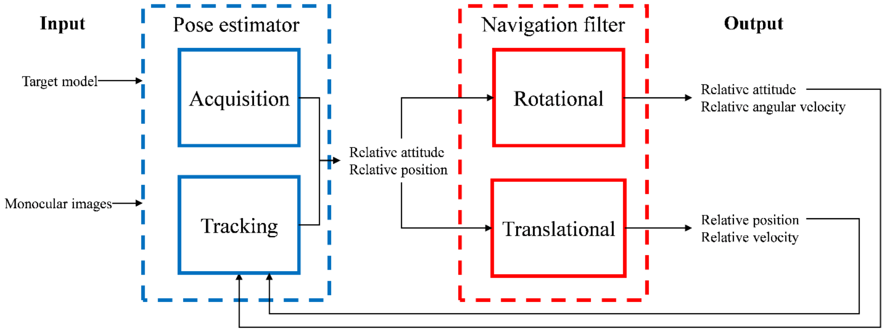


Figure 4.1: Relative Navigation Architecture: Block Diagram

case of Strategy A by the fact that the updated target pose provided by the filter is used to initialize the tracking function of the pose estimator.

4.2 Pose Estimator

In this section, the two adopted strategies for vision-based pose estimation are presented and detailed.

4.2.1 Pose Estimation Algorithm - Strategy A

The output of this block, i.e., the target/chaser pose, is parameterized by a rotation matrix ($\boldsymbol{R}_{\mathcal{CT}}$) which represents the attitude of the Target Reference Frame ($\mathcal{T}$) with respect to the Chaser body Reference Frame ($\mathcal{C}$), and a position vector ($\boldsymbol{t}$) directed from $\mathcal{C}$ to $\mathcal{T}$ (expressed in $\mathcal{C}$). Both the acquisition and tracking functions of the pose estimator are entrusted to feature-based algorithms. Specifically, they allow estimating $\boldsymbol{R}_{\mathcal{CT}}$ and $\boldsymbol{t}$ by trying to optimize the matches between natural features extracted from the acquired images, and the target model. In this work, the attention is focused on point features, such as corners (which can be detected using standard image processing algorithms [19]). Consequently, the target model, generated off-line (i.e., on ground before mission starts) from the knowledge of the target geometry, will be a dataset of M 3D landmarks, i.e., the position vectors of the real corners of the target in $\mathcal{T}$ (generally speaking the landmarks can be 3D points of the target object which are more likely to be identified by image processing algorithms). Clearly, a critical issue for both acquisition and tracking is the identification of correct correspondences between the extracted feature points ($\boldsymbol{p}$) and the model's landmarks ($\boldsymbol{P}$). Indeed, the problems of image-model matching and pose estimation are coupled and can be solved using iterative approaches. As regards the tracking step, the evolution of the pose parameters is followed by implementing the SoftPOSIT algorithm [61]. Specifically, the relative position and attitude parameters are determined by optimizing in a least-squares sense, a cost-function that is derived following the POSIT (Pose from Orthography and Scaling with Iterations) approach [70]. This method returns the pose by approximating the true perspective projection with a scaled orthographic projection but requires the assignment between image and model corners to be known. Therefore, a correspondence matrix is considered as an additional decisional variable, and it is built iteratively through Softassign-based routines [71]. As long as the optimization proceeds, the correspondence matrix tends to a zero-one matrix which univocally discriminates image-model associations, while the relative pose is refined to correct the initial assumption of scaled projection. SoftPOSIT requires an initial guess, which is set as the solution computed at the acquisition step. Hence, an original approach has been developed to derive the initial relative state of the target. In fact, PCA has never been used in the context of spacecraft relative navigation. The coupled problem of feature-matching and pose-estimation is entrusted to a Hypothesize-and-Test approach. First, a set of image-model correspondences (typically indicated as consensus set, S) is selected. These matches are used to compute a tentative pose solving the Perspective-n-Point (PnP) problem [62]. The size of the initial consensus set (n) is determined by the selected PnP solver. The estimated values of $\boldsymbol{t}$ and $\boldsymbol{R}_{\mathcal{CT}}$ are then used to project all the 3D landmarks on the focal plane (thus generating a virtual image). At this point, the re-projected feature points are compared to the ones detected in the original image to try to obtain an enlarged consensus set (S^*). If the hypothesized correspondences are not confirmed, the procedure shall be restarted selecting a new initial consensus set. On the other hand, if the number of matched feature points reaches a limit value (n_{lim}), the matching process is ended, and the pose parameters are refined based on the matches in S^*. The proposed Hypothesize-and-Test approach, summarized by the pseudo-code in Algorithm 1, is now analyzed step by step.

First, image acquisition and processing are carried out to extract a set of N_p 2D feature points (**Step 0**). Then, S must be generated assigning at least n matches between the N_p 2D feature points and the M 3D landmarks (**Step 1**). To this aim, a RANSAC-based approach [62] aided by PCA [72] is applied. Indeed, the purely random nature of RANSAC-based image-model matching strategy is not efficient and may produce an unacceptable runtime. So, the proposed method consists in building S considering only a subset (T) of the M 3D landmarks, i.e. those belonging to spacecraft components which are clearly separated with respect to the target center

Algorithm 1 Hypothesize-and-Test approach

```
Step 0: Image acquisition and processing
Corner Detection
Step 1: Hypothesize
Select the initial consensus set $S$ $(size(S) = n < n_{lim})$
Step 2: Test
Compute $\boldsymbol{t}$ and $\boldsymbol{R}_{\mathcal{TC}}$ (P$n$P solver)
Corner re-projection
Check correspondences
if correspondences=1 (correct initial matches) then
    Creation of $S^*$
else
    Go back to Step 1
end if
Step 3: Pose refinement
if size $S^* < n_{lim}$ then
    Go back to Step 2
else
    Compute $\boldsymbol{t}$ and $\boldsymbol{R}_{\mathcal{CT}}$ (P$n$P solver)
end if
Step 4: Runtime check
Evaluate runtime
if $runtime \geq t_{lim}$ then
    Start again from Step 0
end if
```

of mass (e.g., solar arrays, telescopes, antennas), and the corresponding 2D feature points in the image. The elements of T are selected off-line, depending on the target geometry. On the other hand, the corresponding 2D feature points are identified exploiting information obtained applying the PCA. Indeed, if the target is fully visible in the camera Field of View (FOV) (this condition will always occur at the beginning of the close-proximity maneuver, when pose acquisition is required), the target *appendices* (not moving, such as, solar panels) are very likely to be imaged far from the target centroid on the focal plane. This method requires the *principaldirections* of the target to be as different as possible from one another. PCA is a technique used to analyze multidimensional datasets. Specifically, it allows determining their principal directions by analyzing eigenvectors and eigenvalues of the associated covariance matrix. For instance, if N_p feature points are extracted in the image, the covariance matrix (Q) can be computed as follows:

$$Q = \frac{1}{N_p} \begin{bmatrix} \sum_{i=1}^{N_p} (u_i - u_c)^2 & \sum_{i=1}^{N_p} (u_i - u_c)(v_i - v_c) \\ \sum_{i=1}^{N_p} (u_i - u_c)(v_i - v_c) & \sum_{i=1}^{N_p} (v_i - v_c)^2 \end{bmatrix} \tag{4.1}$$

where (u_i, v_i) and $(u_c\ v_c)$ are the focal-plane coordinates of the i^{th} extracted feature point and image centroid, respectively. The eigenvectors of Q identify two directions on the image plane ($\boldsymbol{d_1}$ and $\boldsymbol{d_2}$). Hence, the extracted 2D feature points can be classified in two datasets (B_1 and B_2) depending on their distances from the PCA axes (D_1 and D_2), computed as follows:

$$\begin{aligned} D_{1,2} &= |\boldsymbol{d_1} \cdot \boldsymbol{p_i}| \\ D_{2,1} &= |\boldsymbol{d_2} \cdot \boldsymbol{p_i}| \end{aligned} \qquad i = 1, ..., N_p \tag{4.2}$$

Specifically, B_1 and B_2 are defined as:

$$B_1 = \{\boldsymbol{p_i} : D_{1,i} > D_{th}\}, \;\; B_2 = \{\boldsymbol{p_i} : D_{2,i} > D_{th}\} \tag{4.3}$$

where D_{th} is a distance-threshold on the image plane. It is defined as the mean Euclidean distance of the features from the image centroid. An example of the results of the off-line classification of the 3D landmarks, and PCA-driven on-line classifications of the feature points is shown in Figure 4.2.

At this point, S is generated by randomly selecting n matches between T and B, where B (i) is the random choice between B_1 and B_2 if both the subsets contain more than n elements, (ii) coincides with B_1 if B_2 contains

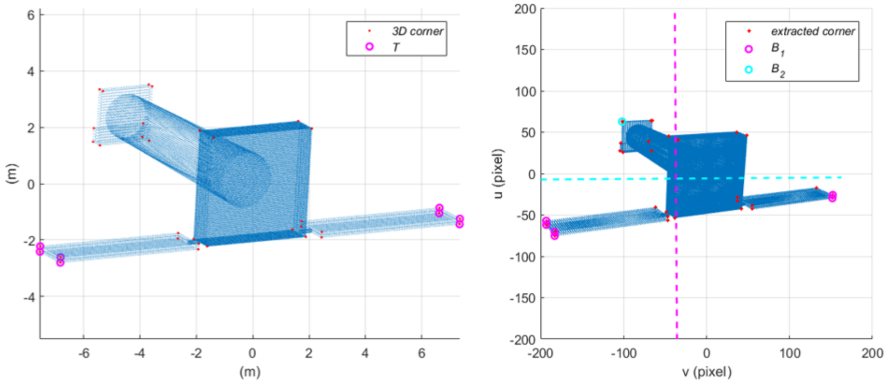


Figure 4.2: Off-line feature point classification for the simplified target model (right). PCA-driver feature point classification for a simulated image (left). The target is the XMM-satellite.

less than n elements, or (iii) coincides with B_2 if B_1 contains less than n elements. Clearly, if both B_1 and B_2 contain less than n feature points, it is highly probable that the camera is observing the target from a not favorable point of view (i.e., the principal directions of the target are not clearly imaged on the focal plane). If this occurs, the algorithm shall return to **Step 0**. This choice allows increasing the robustness of the proposed acquisition strategy since not favorable observation geometries (which are likely to produce wrong matches, and, consequently, inaccurate pose estimates) are discarded. As soon as the initial consensus set is created, a tentative value for the pose parameters is evaluated and the assumed correspondences are verified to try enlarging S (**Step 2**). Concerning the PnP solver, the Efficient Perspective-n-Points (EPnP) method [73] has been selected, since it provides a closed-form solution based on a minimum of $n = 4$ image-model matches. For this reason, the EPnP solver is highly efficient (which is important when several runs are requested as in RANSAC-based iterative approaches [56]). This pose estimate is then used to verify the image-model correspondences. First, the 3D feature points are re-projected on the focal plane ($\boldsymbol{p}^r$), using the classical equation of the perspective projection:

$$\boldsymbol{p}^r = \boldsymbol{K}_I(\boldsymbol{R}_{\mathcal{CT}}\boldsymbol{P} + \boldsymbol{t}) \tag{4.4}$$

where $\boldsymbol{K}_I$ is intrinsic camera calibration matrix. Then, a feature-matching algorithm is applied to verify the correspondences and, if possible, enlarge the consensus set. Specifically, each of the re-projected feature points is associated to the closest image feature according to the Euclidean metric. Hence, the obtained $(\boldsymbol{p}, \boldsymbol{p}^r)$ couples are declared as confirmed matches if their Euclidean distance is below a threshold (D_m). The value of D_m is computed dynamically as the minimum detectable distance between two different bi-dimensional feature points. Indeed, it is reasonable to define this acceptance threshold according to the separation between the satellites: the farther the target from the camera, the smaller the distance between extracted feature points and the harder the image-model association. On the other hand, if the acquisition scenario is assumed to be same, the closer the target, the higher the separation among the extracted feature points, hence the D_m value. If one of the elements of S is not declared as a confirmed match, a new initial consensus set must be generated. Instead, if additional matches are confirmed, they are added to S thus creating S^*. This process is iterated until the size of S^* is lower than an acceptance threshold (n_{lim}), selected as the closest integer to $0.6N_p$. This means that the iteration process is ended when 60% of the detected feature points are associated to the target model (this value has to be tuned beforehand to guarantee successful pose acquisition). As soon as this condition is satisfied, the pose solution is refined exploiting the optimized EP*n*P routine (Efficient Gauss Newton Optimization) proposed in [73]. Finally, a check on the algorithm runtime is carried out (**Step 4**). Specifically, if it exceeds a threshold (t_{lim}), typically of the order of a few seconds, the acquisition process is restarted acquiring a new image. This latter functionality is necessary to avoid that the target/chaser pose evolves too quickly during the runtime (which can occur if the relative rotational dynamics is particularly fast). Indeed, if the run-time threshold is exceeded, the initial pose solution could fall outside the field of convergence of the subsequent tracking algorithm. Once the initial pose is derived, as previously explained, the SoftPOSIT algorithm is run to track its evolution in time.

4.2.2 Pose Estimation Algorithm - Strategy B

The second proposed pose estimation technique, implements a Visual Odometry like routine [74], [75] and works detecting the target object features from the incoming images, given by the mono-camera; these are then matched to an already available on-board map (constituted by a mesh of 3D points, each one correlated to a descriptor) and, in this way, a set of 3D to 2D correspondences is built. From the set of correspondences the P*n*P problem is built and solved within a RANSAC routine in order to delete incoming outliers (wrong match between target image and on-board map) and obtain a first estimate of the relative pose. BA, an optimization technique widely diffused in computer vision [63], is then applied on the map and on the 2D features (which constitute the measurements) in order to optimize the obtained pose. The on-board 3D sparse map of the target object is built on-ground with a dedicated algorithm, by correlations of a 3D model of the target uncooperative object with descriptors extracted from multiple images.

Features detection Features are salient points that can be localized on each image coming from the camera, and different algorithms for their detection exist [21], [22], [76], [77]. Features detection is a fundamental step in the routine of the proposed algorithm because it allows to build the correspondences with the available target spacecraft map that are then used to retrieve the relative motion. In fact, the accuracy of the pose estimation depends on the precision of the matching and, therefore, on the robustness of the extracted features (i.e. noise free and repeatably observable in multiple images). In a general in-orbit relative chaser-target spacecraft trajectory, modest scale variations between different frames along with multiple rotations are expected; moreover, different light conditions will be met moving along the orbit. This is a quite challenging scenario for any computer vision algorithm. The extracted features, to be robust, have to be invariant to each of these parameters. Furthermore, computational cost has to be considered and has to be as low as possible for real-time hardware implementation. Considering these requirements, ORB detector [22] has been selected to be exploited by the tracking algorithm. ORB is a feature detector and fast binary descriptor based on BRIEF [78] and FAST [77] whose features are extremely fast to compute, have good invariance to viewpoint, scale and are resilient to different light conditions, as also proven by its implementation on many state of the art computer vision algorithm [79]. At the current developed stage ORB detector has been implemented in the algorithm to work building an $8-$Level gray scale pyramid with a scale factor of 1.2 for each incoming image. An upper bound of 300 features along with their descriptors is set to be extracted to guarantee a low computational cost. Figure 4.3 shows an example of extracted features on an image of the NASA/ESA SOHO (Solar and Heliospheric Observatory) satellite. Green markers show features location along with their pyramid level.

Routine enhancement with background subtraction methods and selective surface detection will be carefully evaluated in the future both in terms of performances of the tracking algorithm and computational costs in order to prevent key-points detection from the background and extract a higher number of features on the spacecraft body, avoiding unnecessary zones.

Features matching with 3D-model Features matching with the 3D-model of the target spacecraft represents the most delicate and challenging step of

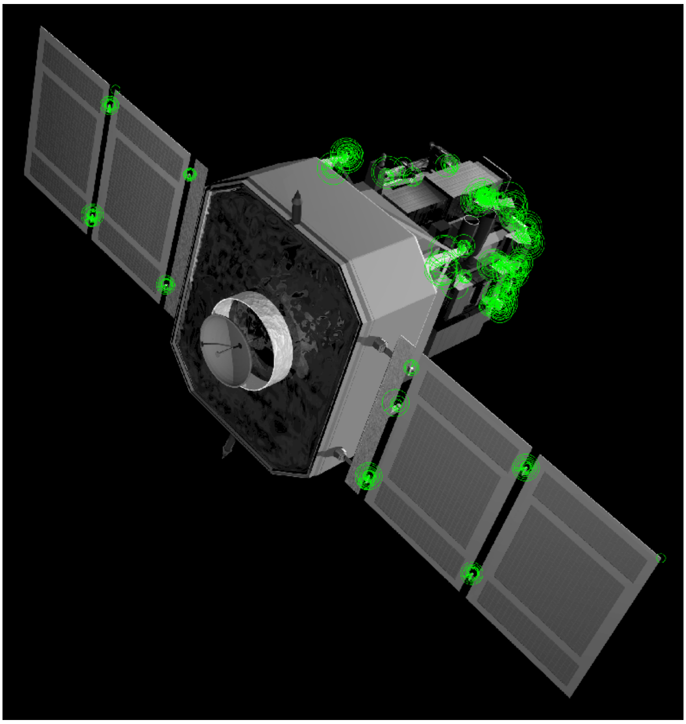


Figure 4.3: Example Image with ORB Features Extracted

the algorithm and consists in building a set of correspondences between the detected 2D features in the current frame and the points of the available target spacecraft 3D map, given their descriptors. This step is crucial because only a good number of correct matches with few outliers allows good performance of the motion estimation step. Moreover, matching with this kind of images is intrinsically complicated due to the uniformity of the texture of the satellite surfaces and the challenging light and orientation conditions constantly changing between each frame. In order to be robust, matching is performed with a symmetry double ratio test. *Hamming distance* is exploited as linearity measure between ORB descriptors extracted in the image and descriptors of the 3D-model. Hamming distance can be computed very efficiently between corresponding binary descriptor strings and makes the process very fast. For each descriptor from the frame features, the matcher finds the two closest descriptors (in order of score) in the 3D-model by trying one by one and, then, it makes the same inverse process for the 3D-model descriptors. Once the two set of matches is obtained, a ratio test according to [21] is applied: the distance between the closest descriptor and the second-closest descriptor on each set is compared; correct matches need to have the closest descriptor significantly closer than the closest incorrect match to achieve reliable matching. Only matches with a distance ratio lower than a fixed threshold are therefore retained. Ratios between 0.7 and 0.8 ensure the best performances. A symmetry test is in the end performed: only correspondent features matched in both the sets are considered. This is an heuristic way to immediately discard clear outliers and retain only good matches.

Motion estimation with P*n*P The 2D features from frame to 3D-model correspondences obtained from the features matching step are used to solve the P*n*P problem. The EP*n*P algorithm [73] is exploited by the tracking algorithm at this step. It is an efficient solution to the problem, being non-iterative and applicable for both planar and non-planar 3D clouds configurations. Given the set of 2D-3D correspondences, the algorithm retrieves the unknown orientation and translation of the camera as the Euclidean motion which aligns both sets of coordinates.

Motion only Bundle Adjustment Motion estimation by itself may not give a sufficient accuracy in trajectory reconstruction. Uncertainty in extracted features location due to noise, presence of outliers between the 2D-3D correspondences and uncertainty in the 3D-model all result in a pose estimation from the EP*n*P algorithm which drifts from the true trajectory. BA [63] is therefore implemented to counteract these effects and to correct the trajectory estimation. BA is a widely diffused technique in computer vision and it is designed to optimize camera pose minimizing the re-projection error of the 3D points on the 2D image plane, taking as constraint the scene geometry. *g2o*, a library specifically developed for pose graph optimization, is exploited as tool for the BA implementation in the tracking algorithm. Pose graph is a way of formulating Simultaneous Localitazion And Mapping (SLAM) problem in robotics. Specifically, the nodes of the graph represent map points and poses of the robot in time and the edges represent constraints between the poses (measurements). Once such a graph is constructed, BA is applied to find a configuration of the nodes which is maximally consistent with the measurements. At the current stage, this approach has been implemented for the tracking algorithm in a simplified *'Motion only BA'* version: at the end of each motion estimation step, retrieved camera pose and matched 3D-model points, seen from the current frame, are set as vertices of the graph, while observed features, representing the measurements, define the edges. The obtained graph is optimized keeping fixed the map points and improving the camera pose estimation only.

Target spacecraft 3D-Model The map construction represents a critical step for the good functioning of the system. At the current stage the map-construction algorithm is under heavy development and is based on Structure From Motion (SfM) concepts merged with informations already obtainable from a CAD model of the target satellite. Being the final objective to have a 3D sparse map of precisely triangulated points, correlated with significant descriptors, the following procedure is exploited: first, a set of high quality images of the target body is acquired. ORB features and descriptors are extracted from each image. Given an available mesh of the target, this is manually matched and projected on the images to retrieve target pose in the image with respect to the mesh reference frame. Given this pose, triangulation of the extracted features is possible, with the mesh also exploited together with the Möller-Trumbore intersection algorithm to compute the 3D coordinates of the features. This procedure works only for planar surfaces, therefore needs to be performed iteratively for each target satellite surface, with one image per surface theoretically sufficient. This means that only few views, covering the whole satellite, are needed. At the end of the process, the list of 3D features with correlated descriptors is stored in a file which is easily accessible by the tracking algorithm.

4.3 Navigation Filter

In this section, the navigation filter used to process the measurements available from the pose determination block is described. The output of the tracking algorithm is the relative position vector between the target and chaser center of mass and the relative rotation matrix between the target and the chaser spacecraft. These quantities are fed to the navigation filter. As already mentioned, the translational and rotational dynamics are treated separately. Hence, the adopted filtering strategy exploits two separated filters, one for the relative translation and another for the relative rotation.

4.3.1 Translation Filter

In this subsection, the H-∞ Filter adopted for the translational motion estimation is presented. The choice of such filter is driven by the fact that, dealing with optical measurement, the assumptions of the KF are usually not satisfied. Moreover, both process and measurement noise are usually time-varying, depending on the position of the spacecraft and on the illumination conditions respectively. All these considerations lead to a robust approach rather than an optimal one. The formulation of the H-∞ Filter is presented in Section 2.1.2. The possibility to use a linear filtering is justified by the fact that the relative translational dynamics can be described by a set of linear equations. The relative translational motion between two objects in space can be described in different ways as detailed in Chapter 3. The most general approach, exploiting nonlinear differential equations implies the use of nonlinear filtering techniques, such as the EKF. In the following, the Yamanaka and Ankersen model [49] is considered. Such a linearized formulation accounts for arbitrary elliptical orbits and leads to a state transition matrix that is advantageous when implementing filtering techniques. In fact, an expression for the $\mathbf{F_k}$ matrix of Equation (2.2) is directly derived. The details of the implementation can be found in [49]. The choice of adopting a linearized model, without considering any external disturbances if justified by the limited duration of proximity operations for the selected scenario. In this case, the difference between the nonlinear and linearized model is not very significant. Furthermore, the adopted model can be used also for elliptical orbits and, therefore, it does not limit the applicability of the proposed approach. Finally, the linearized formulation allows to limit the computational cost of the overall filtering algorithm by preserving accuracy and robustness.

4.3.2 Rotation Filter

For the rotation part, a second-order minimum energy filter on the Lie group is implemented. A filter derived directly on the rotation matrix is used, despite the more complicated formulation. This is done to avoid the non-uniqueness of the classical quaternion attitude parametrization and the undesired unwinding phenomenon [27] as underlined in Chapter 1. Minimum energy filtering was introduced by Mortensen [45], and has been specialized to attitude estimation on the Special Orthogonal Group SO(3) by Zamani et al. [80]. In [81], the authors showed that such minimum energy filter on SO(3), thanks to the Lie group structure of SO(3), outperforms the industry standard Multiplicative Extended Kalman Filter (MEKF) for absolute attitude estimation. A further development of the presented minimum energy filter on $SO(3)$ was introduced by Saccon *et al* [82]. They present an explicit formula for a second-order optimal nonlinear filter. A slightly different formulation of the second-order minimum energy filter is hereby proposed. First, the filter in [82] is adapted to relative state estimation. Despite the kinematic equations do not change in a relative framework, the dynamics must include both the target and chaser contributions. In our formulation the derivative of the angular acceleration is forced to zero to avoid the explicit dependence on the inertia matrix of the target spacecraft. This is somehow similar to what it is usually done for α-β-γ filters [39]. Despite the possibility of achieving slightly worse performance that the one provided by the filter scheme in [82], the proposed formulation is more promising due to its capability to deal also with unknown or partially unknown objects. In particular, the rotational dynamics exploited by the filter is:

$$\dot{\boldsymbol{R}}_{\mathcal{CT}} = \boldsymbol{R}_{\mathcal{CT}} \left(\boldsymbol{\omega}(t)\right)_{\times}, \tag{4.5}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{B}\boldsymbol{\delta}. \tag{4.6}$$

with $(\boldsymbol{R}_{\mathcal{CT}}, \boldsymbol{\omega}) \in SO(3) \times \mathbb{R}^3$, $\boldsymbol{\omega}$ representing the relative angular velocity and $\boldsymbol{B} \in \mathbb{R}^{3x3}$ indicating a coefficient matrix that allows for different weightings of the components of the unknown process noise $\boldsymbol{\delta}$.

Equations (4.5) and (4.6) are directly expressed in relative terms and in the chaser frame $\mathcal{C}$, without an explicit dependence on the inertia matrix of the target spacecraft. The formulation of the proposed relative filter is summarized in Table 4.1.

4.4 Simulation Environment and Results - Strategy A

In this section, the numerical simulation environment to validate the proposed relative navigation algorithm is described. The measurement generation process is illustrated and the simulation scenarios are presented. Finally, an overview of the results for different orbital conditions, target spacecraft and noise level is provided.

4.4.1 Numerical Simulation Environment Description

The proposed relative navigation architecture is preliminarily validated through numerical simulations, carried out in MATLAB environment. In particular, the relative translational and rotational orbital dynamics between two spacecraft are reproduced by using nonlinear dynamical models that are detailed in Chapter 3, in particular Equations (3.1) and (3.17). An unperturbed, unforced motion is considered mainly because of the short duration of vision-based relative proximity operations and of the limited chaser-target separation. Moreover, the two spacecraft are assumed to do not perform any maneuver during the simulation. This is a common assumption for the validation of similar vision-based relative navigation algorithms [25], [66], [83]. The generated relative trajectory is used as a reference to evaluate the relative navigation algorithm performance,

Table 4.1: The 2^{nd} Order Minimum Energy Filter on $SO(3)$

Initialization:

$$\hat{\boldsymbol{R}}_{\mathcal{CT}}(t_0) = \hat{\boldsymbol{R}}_{\mathcal{CT}0},\ \hat{\boldsymbol{\omega}}(\mathbf{t_0}) = \hat{\boldsymbol{\omega}}_{\mathbf{0}},\ \boldsymbol{K}(t_0) = \boldsymbol{K}_0,$$

Filtering:

$$\dot{\hat{\boldsymbol{R}}}_{\mathcal{CT}} = \hat{\boldsymbol{R}}_{\mathcal{CT}}\left(\hat{\boldsymbol{\omega}}(t) + K_{11}\boldsymbol{r}^R + K_{12}\boldsymbol{r}^\omega\right)_\times,$$

$$\dot{\hat{\boldsymbol{\omega}}} = K_{21}\boldsymbol{r}^R + K_{22}\boldsymbol{r}^\omega,$$

$$\dot{\boldsymbol{K}}(t) = -\alpha\boldsymbol{K} + \boldsymbol{A}\boldsymbol{K} + \boldsymbol{K}\boldsymbol{A}^{\boldsymbol{T}} - \boldsymbol{K}\boldsymbol{E}\boldsymbol{K} + \boldsymbol{B}\boldsymbol{R}^{\boldsymbol{-1}}\boldsymbol{B}^{\boldsymbol{T}} - \boldsymbol{W}\boldsymbol{K} - \boldsymbol{K}\boldsymbol{W}^{\boldsymbol{T}},$$

where

$$\boldsymbol{r}_t = \begin{bmatrix} \boldsymbol{r}^R \\ \boldsymbol{r}^\omega \end{bmatrix} = \begin{bmatrix} -\boldsymbol{u}_1(\hat{\boldsymbol{r}}_1 \times \boldsymbol{r}_1) - \boldsymbol{u}_2(\hat{\boldsymbol{r}}_2 \times \boldsymbol{r}_2) \\ \mathbf{0} \end{bmatrix},$$

$$u_i = \frac{b^2}{d_i^2},$$

$$\hat{\boldsymbol{r}}_i = \hat{\boldsymbol{R}}_{\mathcal{CT}}^T\bar{\boldsymbol{r}}_i, \quad \boldsymbol{r}_i = \hat{\boldsymbol{R}}_{\mathcal{CT}}^T\bar{\boldsymbol{r}}_i + d_i\boldsymbol{\epsilon},$$

$$\boldsymbol{A} = \begin{bmatrix} -\hat{\boldsymbol{\omega}}_\times & \mathbf{I} \\ 0 & 0 \end{bmatrix},$$

$$\boldsymbol{E} = \begin{bmatrix} \sum_{i=1}^2 \boldsymbol{u}_i \left((\hat{\boldsymbol{r}}_i)_\times (\boldsymbol{r}_i)_\times + (\boldsymbol{r}_i)_\times (\hat{\boldsymbol{r}}_i)_\times\right)/2 & 0 \\ 0 & 0 \end{bmatrix},$$

$$\boldsymbol{B}\boldsymbol{R}^{\boldsymbol{-1}}\boldsymbol{B}^T = \begin{bmatrix} 0_{3\times 3} & 0 \\ 0 & \mathbf{B}\mathbf{R}^{\mathbf{-1}}\mathbf{B}^T \end{bmatrix}$$

$$\boldsymbol{W} = \begin{bmatrix} \frac{1}{2}\left(K_{11}\boldsymbol{r}^R + K_{12}\boldsymbol{r}^\omega\right)_\times & 0 \\ 0 & 0 \end{bmatrix}.$$

---

but also to provide input measurements to the pose determination block (as detailed in section 4.4.1.1). It is worth underlying that the ground-truth and the dynamics implemented in the filters are distinct to generate representative results.

4.4.1.1 Measurement Generation

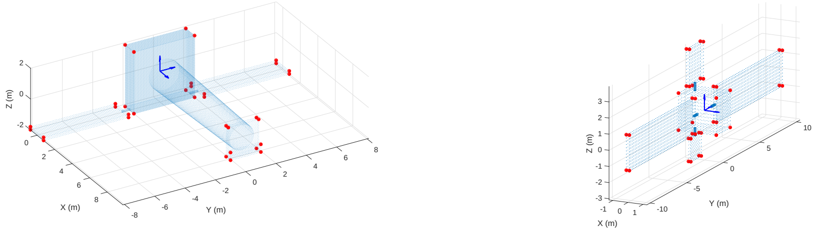
In order to preliminary assess the performance of the proposed approaches for monocular pose determination, simulated sets of 2D point features (e.g., corners) are generated. At this step of the process, image processing is not considered. Specifically, 3D landmarks composing the target model are projected on the image according to the true relative attitude and position parameters. Hence, the position of each detected feature point on the image plane is modified to account for potential errors introduced by the image processing. Specifically, a Gaussian white noise is added to the pixel coordinates of each point feature (whose standard deviation (σ_{pix}) is expressed in terms of a certain number of pixel). Moreover, a number of outliers (n_{out}) is randomly located in the region of the image plane occupied by the target (according to a uniform distribution) to take into account either false alarms from the image processing algorithms or potential deviation of the simplified model from the actual target geometry. A camera with a resolution of 400 x 400 and a focal length equal to $f = 7mm$ is used.

4.4.2 Simulation Scenario

In this section, the scenarios considered to validate the proposed algorithm, are presented, both in terms of target and orbit characteristics.

4.4.2.1 Target

XMM-Newton The first selected target is the ESA X-ray space observatory, XMM-Newton. This is an example of very large spacecraft for which servicing or removal operations may be necessary. The inertia matrix of the object has been computed considering available information about its size, shape and weight [84], [85], and assuming a simplified geometry (i.e., the real geometry of the target is modeled as a combination of elementary volume elements, such as cuboids, cones, cylinders etc.), and uniform mass distribution. Clearly, the estimated values do not represent with high accuracy the real inertia of the two considered targets. This is due to the



(a) XMM-Newton **(b)** Hispasat

Figure 4.4: Targets Point Cloud and Reference Frame

lack of information about spacecraft components available in the open literature. The resulting inertia matrix:

$$\mathbf{I_{XMM}} = \begin{bmatrix} 7052 & -820 & -3991 \\ -820 & 129129 & 0 \\ -3991 & 0 & 131196 \end{bmatrix} kgm^2 \tag{4.7}$$

is computed considering the body fixed reference frame as in Figure 4.4a.

Hispasat The second target is one of the satellites of the Hispasat family. It represents the next generation of SmallGEO platforms for telecommunication. This lower mass class of telecom satellites is the ideal target for future on-orbit servicing missions. Similarly to XMM, the inertia matrix is computed by exploiting the information in [86], and assuming the reference frame as in Figure 4.4b:

$$\mathbf{I_{hispa}} = \begin{bmatrix} 6396 & 0 & -46 \\ 0 & 3703 & 0 \\ -46 & 0 & 5918 \end{bmatrix} kgm^2 \tag{4.8}$$

4.4.2.2 Orbital Scenario

Absolute Orbital Parameters Testing the filter and its dynamics over different possible absolute orbits is important to verify its robustness. For this reason, three different orbital regimes are considered: Low Earth Orbit (LEO), Highly Elliptical Orbit (HEO) and GEO. These alternatives represent the most important classes of Earth orbits with a variety of altitudes and eccentricities. Three real satellites are considered, Envisat (LEO), XMM (HEO), Hispasat (GEO). The associated classic orbital parameters are reported in Table 4.2.

Table 4.2: Scenarios Orbital Parameters

	LEO	**HEO**	**GEO**
a [km]	7143.1	66931.6	42164.1
e [-]	$1.4 \cdot 10^{-4}$	0.808	$1.1 \cdot 10^{-4}$
i [deg]	98.2	69.9	0.005
ω [deg]	85.9	96	41.6
Ω [deg]	79.2	352.5	274.9

Relative Motion Relative translational trajectories and rotational motion are defined depending on the absolute reference orbit of the chaser spacecraft and on the relative angular velocity. Five different scenarios are considered, as summarized in Table 4.3.

Table 4.3: Cases Definition

	Case 1 LEO Low	**Case 2 LEO High**	**Case 3 LEO Mix**	**Case 4 HEO**	**Case 5 GEO**
Chaser Orbit	LEO	LEO	LEO	HEO	GEO
Target	XMM	XMM	XMM	XMM	Hispasat
ρ_0 [m]	[0 30 0]	[0 30 0]	[0 30 0]	[0 30 0]	[0 40 0]
$\dot{\rho}_0$ [m/s]	$[0\ 1\ 0] \cdot 10^{-4}$	$[0\ 1\ 0] \cdot 10^{-4}$	$[0\ 1\ 0] \cdot 10^{-4}$	$[3\ {-1}\ 0] \cdot 10^{-3}$	$[0\ {-1}\ 0] \cdot 10^{-4}$
$\omega_{t0}\|_{\mathcal{T}}$ [deg/s]	[0.1 0 0]	[1 0 0]	[0.5 -0.3 0.2]	[0.1 0 0]	[0.1 0 0]

where ρ_0 and $\dot{\rho}_0$ are the initial relative position and translational velocity expressed in the chaser frame $\mathcal{C}$ and $\omega_{t0}|_{\mathcal{T}}$ is the initial target angular velocity expressed in the target reference frame $\mathcal{T}$. These quantities are the initial conditions for Equations (3.8) and (3.17). For sake of completeness, it is worth noting that no control is assumed on the chaser spacecraft orientation and that it is always aligned with its LVLH reference frame. This entails that the relative translational velocity has to be kept very small in order to avoid a rapid drift of the target spacecraft. For this reason, the initial conditions for all the different cases are not representative of any particular operative scenario but they are selected to guarantee a limited evolution of the relative position, in a leader-follower configuration, for the complete duration of the simulation. Different cases of relative angular velocity are considered to test its effect on performance of the proposed architecture.

4.4.2.3 Noise Level

As previously explained in section 4.4.1.1, the noise is added while generating the simulated images according to a Gaussian distribution with standard deviation equal to σ_{pix}. An additional source of error is represented by the presence of n_{out} outliers. Another relevant feature concerns the precise determination of the chaser true anomaly, that is used to propagate the relative translational dynamics equations (see Equation (3.1)). In real applications, this value is computed by means of the spacecraft position and velocity coming from the absolute navigation filter. In literature, similar works [25], [66], [83] neglect the possible noise associated to this quantity. However, a noisy fluctuation of the chaser true anomaly can strongly affect the filter robustness and stability. For this reason, the true anomaly of the chaser spacecraft is corrupted by noise in this work. In particular, the associated noise level is described by a Gaussian distribution with standard deviation respectively $\sigma_{pos} = 10^{-2} km$ and $\sigma_{vel} = 10^{-4} km/s$.

4.4.3 Results

In this section, the results of the algorithm validation are presented. As in Figure 4.1, the pose estimator is divided in two blocks: *Acquisition* and *Tracking*. First, an analysis of the *Acquisition* process is carried out. In particular, the robustness to different relative angular velocities is tested. This analysis is critical since the speed of the relative rotational motion strongly affects the performance during *Acquisition* Algorithm 1. With this study, statistical quantities of the acquired pose and acquisition time are obtained (i.e. mean and standard deviation of the distribution). These statistical parameters are then used to generate random initial conditions to test the overall architecture proposed in this manuscript (including the pose tracking algorithm and the filtering techniques) over a relatively long time interval. Before presenting the results, the definitions of the estimation errors, used for performance assessment, are introduced.

The relative position error is defined as:

$$e_\rho = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2} \tag{4.9}$$

where $\hat{x}, \hat{y}, \hat{z}$ are the position components estimates. Similarly, the relative attitude error is computed as in [87]:

$$e_R = \arccos\left(1 - \frac{tr(\mathbf{I} - \boldsymbol{R}_{\mathcal{CT}i}{}^T \hat{\boldsymbol{R}}_{\mathcal{CT}i})}{2}\right) \tag{4.10}$$

with $\hat{\boldsymbol{R}}_{\mathcal{CT}}$ being the estimated rotation matrix.

Acquisition Results The content of this section is focused on the performance achievable by the proposed acquisition algorithm. A deep analysis

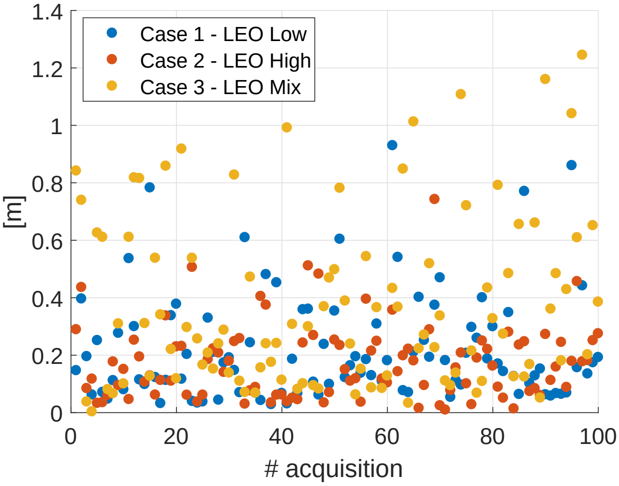


Figure 4.5: Acquired Position Error

of the *Acquisition* process is presented in this paragraph. A statistical analysis of the results obtained over 100 runs is performed considering three scenarios: Case 1 - LEO Low, Case 2 - LEO High, Case 3 - LEO Mix (see Section 4.4.2.2). For each of the analyzed cases, the noise level associated to the result of the image processing step is simulated as a Gaussian white noise on the location of the detected point features with standard deviation $\sigma_{pix} = 1$ as in [58]. As for the number of outliers, it is fixed to $n_{out} = 1$. The algorithm is run (according to Algorithm 1) until the initial acquisition of the pose parameters is declared. Its performance is evaluated in terms of pose estimation error processing time. In this regard, it is worth outlining that the value of the elapsed time is not intended as an absolute measurement of the algorithm computational efficiency (since the simulations are run in MATLAB environment). However, the results can help to assess the effect on the processing burden of the increase in the target/chaser relative rotational dynamics. Figures 4.5 to 4.7 show the position, rotation error and runtime at the end of the *Acquisition*. Figures 4.5 and 4.6 show that a difference in the initial relative angular velocity does not drastically affect the error in the acquired pose. In fact, despite the error of Case 3 is often higher than in the other two cases, the order of magnitude is comparable. This is expected since the algorithm processes one image at the time during the acquisition step. However, a significant difference is present in Figure 4.7. In fact, as expected, a higher relative angular velocity implies a difficulty in the acquisition convergence and, therefore, a higher time to complete the acquisition. This is certainly an aspect to consider while approaching fast-spinning objects.

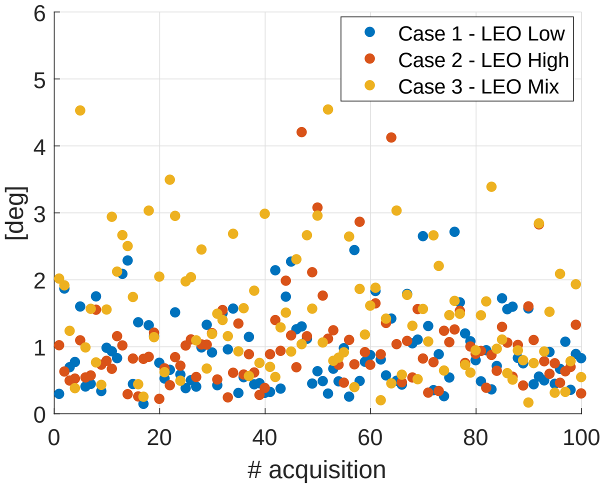


Figure 4.6: Acquired Attitude Error

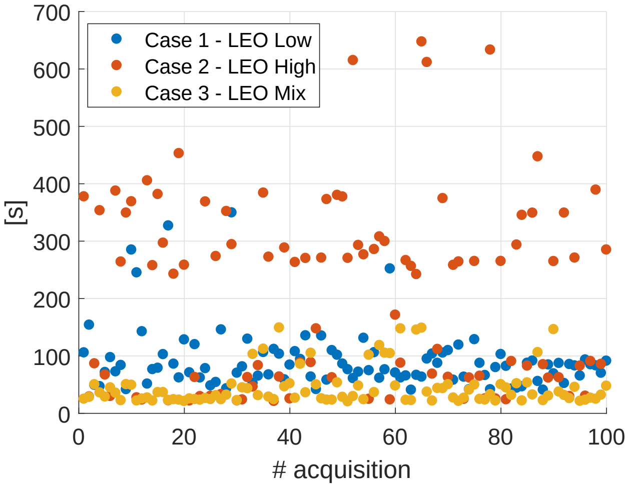


Figure 4.7: Time to Complete the Acquisition

For a more quantitative description of the results, the main statistics of the obtained distributions are reported in Table 4.4.

Table 4.4: Acquisition Statistical Results

	Case 1 LEO Low	**Case 2 LEO High**	**Case 3 LEO Mix**
e_ρ - Mean [m]	0.2196	0.1747	0.3767
e_ρ - STD [m]	0.1879	0.1314	0.3003
e_R - Mean [deg]	0.9502	1.007	1.474
e_R - STD [deg]	0.5939	0.692	0.9339
Time - Mean [s]	91.48	206.6	44.97
Time - STD [s]	54.02	160.7	32.91

It is worth underlying that similar results are obtained also considering the Hispasat target geometry.

Tracking and Filter Results The statistics in Table 4.4, obtained with the *Acquisition* analysis, are used to generate the random input for the *Tracking* and subsequently the filtering step. Please note that for Case 4 - HEO and Case 5 - GEO the statistics of the Case 1 - LEO are used, having the three cases the same initial relative angular velocity.

The first presented analysis wants to show the beneficial effect of having a filter downstream of the vision-based pose determination algorithm. In Figures 4.8 and 4.9 the relative position and attitude estimation errors of the *Tracking* process only are shown, together with the results from the approach combining filter and monocular navigation. The presented simulation is run on the Case 1 - LEO Low scenario with a noise of $\sigma_{pix} = 2$ and $n_{out} = 1$. In this case, a 1Hz frame rate is considered.

The filter benefits are especially evident for the position error. In fact, the estimation error with the filter in the loop is reduced by more than one order of magnitude in Figure 4.8. In Figure 4.9, the improvement is still evident but less remarkable. Indeed, the overall estimation error is reduced by roughly the 20%. In Figures 4.8 and 4.9 a fluctuation in the estimation error is observed. This effect can be due to the relative rotational dynamics that causes a change in the observation conditions of the target. Periodically, the camera observes the target from an unfavorable point of view; this affects the accuracy of the pose determination and, consequently, of the navigation filter estimates. The robustness of the proposed approach is further proved by means of a statistical analysis. A set of 50 runs, considering the first 30 minutes, are performed for all the five different test scenarios. It is worth underlying that, for each simulation, the initial conditions of the tracking are randomly selected from the statistics

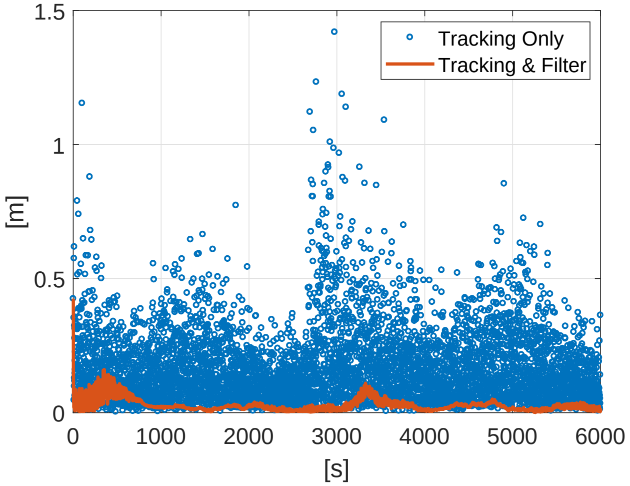


Figure 4.8: Relative Position Error e_ρ

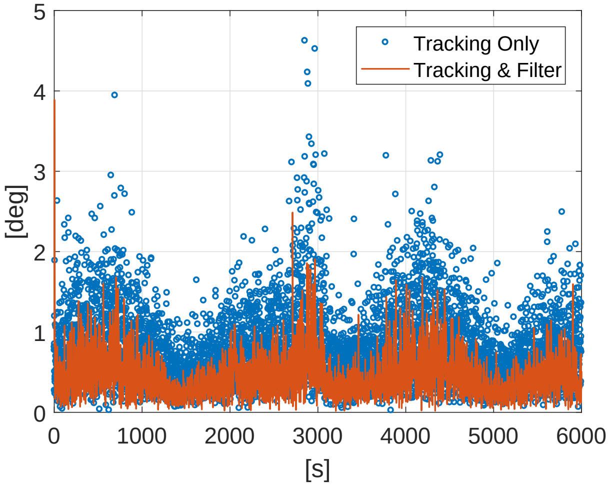


Figure 4.9: Relative Attitude Error e_R

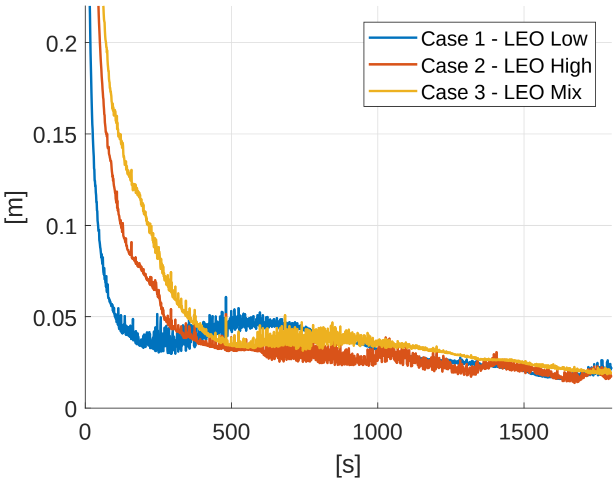


Figure 4.10: Average Relative Position Error e_ρ

derived from the analysis on the *Acquisition* step. Moreover, the noise associated to the point features extracted from the simulated images is randomly picked from a zero-mean Gaussian distribution with standard deviation of $\sigma_{pix} = 2$. A number of outliers of $n_{out} = 1$ is used. Figure 4.10 shows mean relative position and attitude errors, computed according to Equations (4.9) and (4.10) and averaged for each time step over the 50 runs for the three different LEO cases.

As expected, a higher relative angular velocity implies a higher error and a slower convergence, for both the LEO Mix and LEO High cases, visible in Figure 4.11. This is justified by the fact that the *Tracking* process is more problematic while experiencing higher angular velocity. Moreover, recalling Equation (4.6), the rotational filter is built assuming the relative angular acceleration equal zero and considering only white noise components. This particular formulation strongly affects the attitude estimation in the case of higher relative angular velocities. Concerning the relative position error, the first three LEO cases converge to almost the same value with a slightly different convergence time. This is in line with the expected results being the translational motion equal in all the three cases. The different convergence rate is due to the coupling between the rotation matrix and position vector while solving the SoftPOSIT optimization. For a more quantitative analysis, in Table 4.5 the steady state RMSE for all the cases, considering the last 1000 seconds, are reported.

Table 4.5: Tracking & Filter RMSE Results

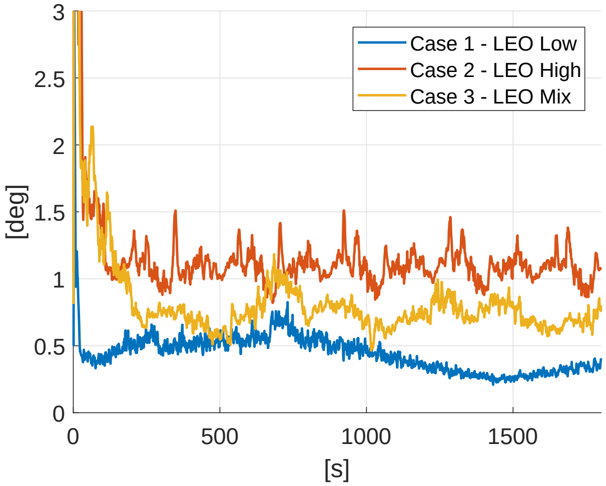


Figure 4.11: Average Relative Attitude Error e_R

	RMSE - Position [m]	**RMSE** - Attitude [deg]
Case 1 - LEO Low	0.0109	0.3259
Case 2 - LEO High	0.0225	1.0834
Case 3 - LEO Mix	0.0269	0.7205
Case 4 - HEO	0.0673	0.6711
Case 5 - GEO	0.0264	0.4183

The Case 4 - HEO is substantially different from the previous ones. In fact, the high value of the chaser orbit eccentricity implies a higher error between the linearized relative dynamics equations, used in the filter, and the reality. This is evidently remarkable from Table 4.5 where the error of the HEO case is much higher with respect to all the others. However, this does not have any strong influence in the relative attitude estimate error that is comparable to the other test cases. Finally, despite the different target, the GEO case RMSEs are very similar to LEO cases.

Noise Sensitivity Analysis In order to evaluate the robustness of the proposed method, a sensitivity analysis over σ_{pix} is performed. The orbital scenario of Case 1 is propagated for 1800 seconds with a noise standard deviation ranging from 2 to 6 pixels. The tuning of the filter is kept equal for all the different test cases. Given the estimation error in time for the different simulations, the RMSE from 50 seconds to the end is computed and the change in the steady state error is thus evaluated. Table 4.6 reports the RMSE for different noise levels, for both relative position and attitude estimation errors.

Table 4.6: RMSE for Different Noise Levels

$\sigma_{\mathbf{pix}}$	**RMSE** - Position [m]	**RMSE** - Attitude [deg]
2	0.0312	0.4221
3	0.0599	0.6671
4	0.0735	0.9430
5	0.11	1.1323
6	0.1455	1.4902

As expected, the errors increase as long as the noise level increases. However, both the positional and angular accuracy are acceptable even considering very high levels in the measurement noise (in reference [58] the maximum value considered for σ_{pix} for numerical simulation is equal to 3 pixels).

Outliers Sensitivity Analysis A similar sensitivity analysis is also carried out to evaluate robustness against increased number of outliers. Also in this case the simulations are run on the Case 1 scenario with fixed initial conditions and variable number of outliers n_{out}. The noise level is kept at $\sigma_{pix} = 2$. Analogously, the RMSE is computed from 50 seconds to the end of the simulation. Table 4.6 reports the RMSE for different numbers of outliers, for both relative position and attitude estimation errors.

Table 4.7: RMSE for Different Number of Outliers

$\mathbf{n_{out}}$	**RMSE** - Position [m]	**RMSE** - Attitude [deg]
1	0.0312	0.4221
2	0.0341	0.4314
3	0.0363	0.4371
4	0.0444	0.4479
5	0.0479	0.4648

Table 4.7 shows that despite a very slow increase in the estimation errors, the proposed algorithm is able to cope also with a relatively large number of outliers (it works also if the outliers are the 10% of the extracted features).

4.4.4 Preliminary Experimental Validation

The pose estimation technique described in Section 4.2 was also validated through a preliminary experimental campaign.

4.4.4.1 Experimental Facility

The algorithm was validated at the new experimental facility dedicated to vision-based autonomous GNC algorithm at Politecnico di Milano, Aerospace Science and Technology Department (DAER) [32]. The system is composed by a 7-DoF robotic arm with a navigation camera mounted on the tip, reproducing the two spacecraft relative dynamics. A realistic satellite mock-up is used as target. Controlled, realistic lighting condition is reproduced by using a dedicated illumination system.

Robotic arm The robotic arm represents the most important element of the facility. In fact, it allows to properly simulate the relative motion between two spacecraft for different mission scenarios. A 7-DoF Mitsubishi PA10-7c is used: it is servo-actuated and sensorized with brush-less resolvers. The robot operative envelope has 1m radius approximately and allows for spacecraft motion simulation under different scaled trajectories.

Navigation camera A PointGrey Chameleon-3 is adopted as navigation camera for the facility. Its characteristics are shown in Table 4.8.

Table 4.8: Navigation Camera Technical Specification

	Technical specification
Resolution	1280x1024
Frame rate	149
megapixels	1.3
Chroma	color/gray-scale
ADC	10 Bit
Sensor format	1/2
Focal length	6 mm
Field of view	43.5 ^{o}

Illumination system Ensuring the proper illumination and environmental conditions is a fundamental task to obtain realistic images. To this aim, a dedicated dark room has been built around the facility exploiting black curtains and dark floor cover which prevent from light reflection. A proper lighting is used to simulate the Sun illumination: a dedicated LED array with narrow beam angle and 5700K light temperature is exploited.

Satellite mock-up A scaled satellite mock-up has been realized at PoliMI-DAER premises for in-orbit relative navigation around uncooperative target simulations. The model of the Solar and Heliospheric Observatory (SOHO) spacecraft was 3D printed at PoliMI-DAER Laboratory. A spray-plaster of a gray color was applied twice on the model and we sanded it down with sandpaper with different grit size. A spray acrylic black paint, usually used for modeling, was applied twice on the surface to reproduce the black surface of the spacecraft. The details were adjusted by a thin brush. Gold aluminum foils were used to reproduce the classical thermal protection of spacecraft, made of Multi-Layer Insulation (MLI). Finally, we used some leftover of real spacecraft solar cells from Azurespace. The final spacecraft mock-up is shown in Figure 4.12.

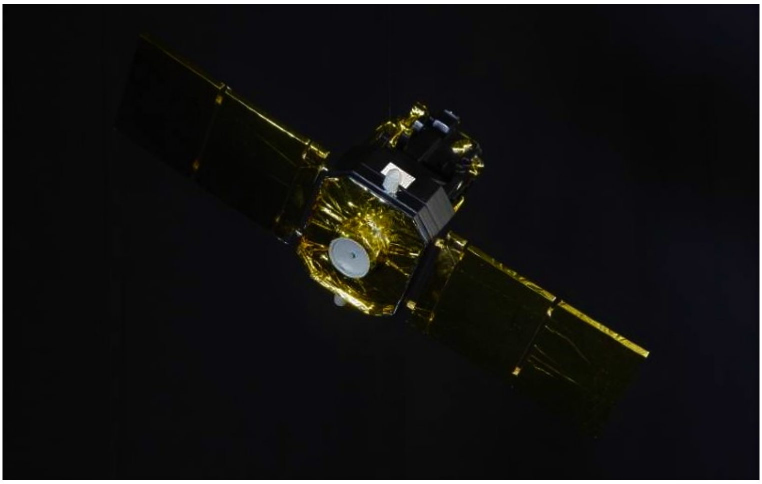


Figure 4.12: Satellite Mock-up

4.4.4.2 Experimental Test

The test was conducted by imposing a controlled motion to the robotic arm and keeping the spacecraft mock-up anchored to a fixed support. An approach trajectory was imposed to the arm with also a rotational component. Some snapshots of the simulated trajectory are reported in Figure 4.13.

4.4.4.3 Experimental Test Results

The acquired images were processed off-line by the proposed pose determination algorithm and a pose estimate was produced. It is worth underlying that the filter was not inserted in the loop and only the vision-based algorithms (acquisition and processing) were tested. In order to evaluate

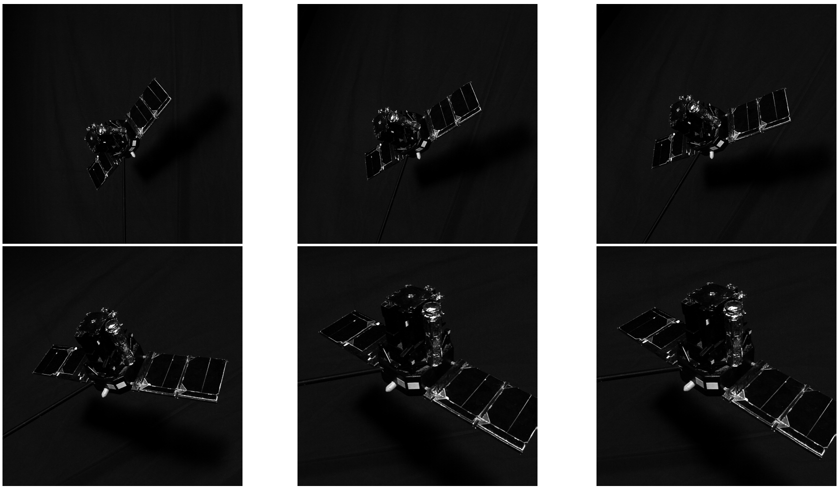


Figure 4.13: Acquired Images Sequence Example

the accuracy of the proposed approach, a ground truth of the actual pose was necessary. This quantity was extracted at each time step of the simulation by enforcing, by hand, the known correspondences between the 3D feature points of the model and the 2D points on the camera plane. This step has some intrinsic, unavoidable errors. For this reason, an accurate calibration of the facility is currently ongoing. For the acquired trajectory we evaluated the relative position for each axis and attitude errors as in Equation (4.10). The results of the preliminary experimental validation are shown in Figures 4.14 and 4.15.

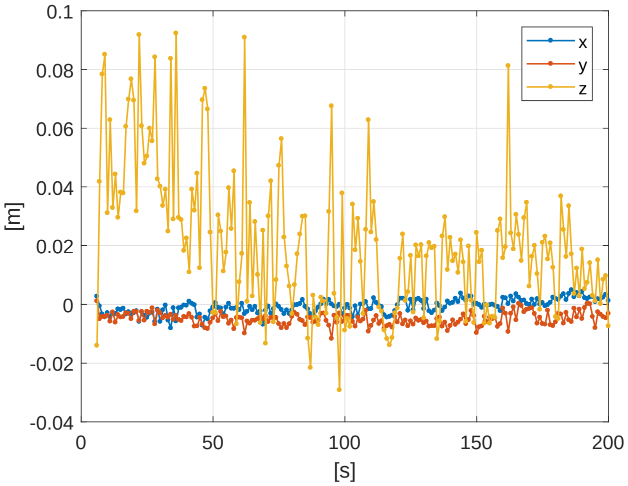


Figure 4.14: Relative Position Error

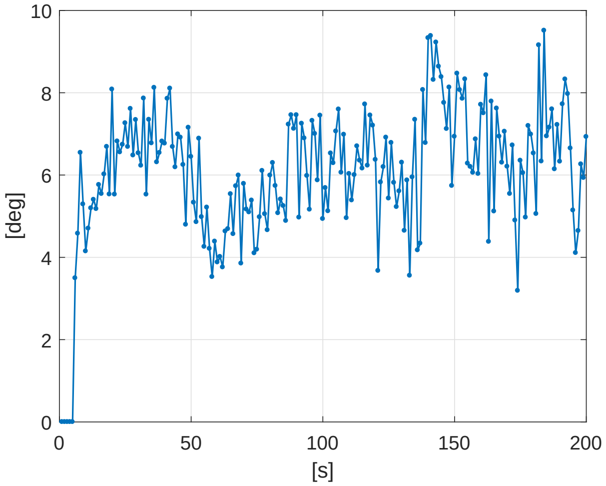


Figure 4.15: Relative Attitude Error

Please note that the position error reported in Figure 4.14 is not scaled. Considering the relative distance between the mock-up and the arm tip, this error has to be scaled by a factor of roughly 30 (if we consider the orbital scenarios taken into account for the numerical validation and the true relative distance between the robotic arm and the target mock-up). However, the magnified error is in line with what we expected from numerical simulation, working with the pose determination algorithm only.

4.5 Simulation Environment and Results - Strategy B

Strategy B involves an algorithm that has been recently developed. For this reason, only a preliminary numerical validation is presented in this thesis. Similarly to Strategy A, a reference scenario is defined. A Medium Earth Orbit (MEO) orbit is selected for the chaser spacecraft with an eccentricity of 0.17 and semi-major axis of 8790 km. The relative reference dynamic is simulated, as for Strategy A, directly integrating the nonlinear equations for the relative motion without considering any orbital perturbation (Equation (3.1)). The assumed initial conditions are:

$$\begin{aligned} \rho_0 &= [50, 0, 0] \qquad m; \\ \dot{\rho}_0 &= [0, -0.1, 0] \quad m/s; \end{aligned} \tag{4.11}$$

expressed in the local-vertical, local-horizontal (LVLH) reference frame fixed to the chaser spacecraft center of mass $\mathcal{C}$. This initial conditions have been chosen to have an in-plane elliptical motion of one spacecraft with respect to the other. This trajectory can be representative of a monitoring or close approach phase. Also in this case the non perfect absolute state determination of the chaser spacecraft is taken into account. For the relative dynamics, a torque-free tumbling motion is imposed to the simulated target spacecraft. The motion is simulated as done for Strategy A (Equation (3.17)), imposing the following initial conditions: $\omega_0 = [0.5, -0.2, 0.3] deg/s$. The target spacecraft is modeled with a simple shape taking as reference the NASA Suomi-NPP satellite. A set of 38 uniform distributed points is considered and constitutes the 3D-model used by the tracking algorithm. The set of images to be provided to the navigation algorithm is instead obtained as explained in Section 4.4.1.1. To take into account for the uncertainty in features detection that arise due to the presence of noise for an application on real images, a value of the noise extracted from a Gaussian distribution with $\sigma_{pix} = 2$ is added to the 2D features location. Figure 4.16 shows the true 3D built model of the spacecraft along with the selected features, both with zero noise. Figure 4.17 shows instead a short sequence of the generated extracted features, from which it is possible to observe a portion of the spacecraft tumbling motion.

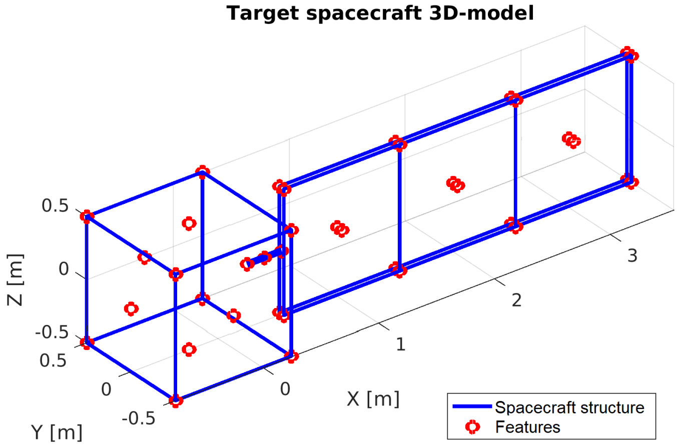


Figure 4.16: *3D* Spacecraft Model

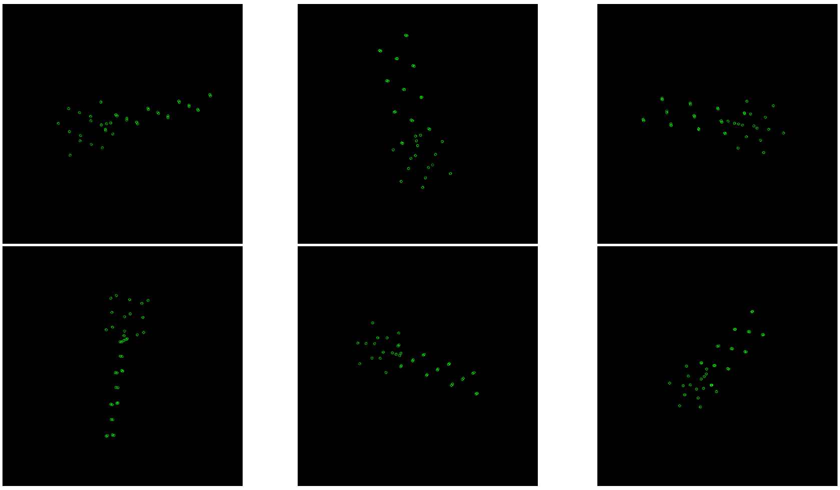


Figure 4.17: Simulated Target Spacecraft Measurements

4.5.1 Results

In this section, the results of the pose determination algorithm and of the overall relative state determination are presented. The estimation errors are computed as in Equations (4.9) and (4.10). The results of the translational H-∞ Filter are presented in this paragraph along with the pose determination results. In particular, the estimation errors corresponding to the $\sigma_{pix} = 2$ noise case with measurement frequency of 1 Hz are considered. Figure 4.18 shows the estimation error of the position.

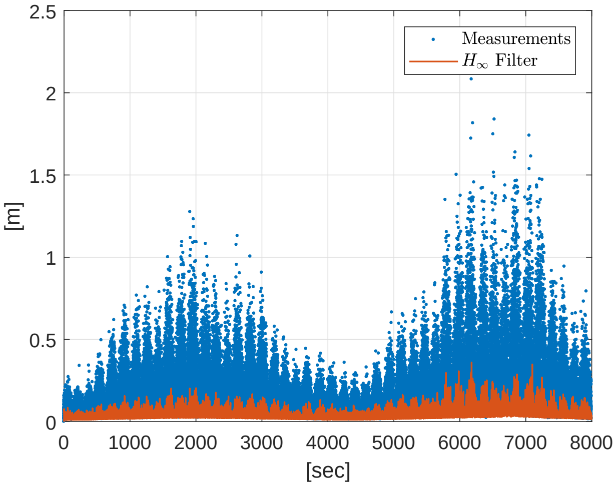


Figure 4.18: Relative Position Error

The advantage of having a filter downstream of the image processing is evident also with Strategy B. The overall position error is strongly reduced and it always stays beyond 0.4 m. Similarly to what has been done for the translational filter, the results of the rotational filter are be presented. Figure 4.19 show the rotation filtering results. Once again the filter has improved quite significantly the measurements from the vision-based tracking, The overall rotation error is always beyond 0.1 deg. Strategy B shows more promising results if compared to the numerical analysis of Strategy A. However, the implementability of such approach presents some challenges. In particular, the construction of the target spacecraft 3D model, with the associated descriptors, is complicated; especially if it has to take into account possible variation of the illumination conditions experienced by the target spacecraft. In this sense, Strategy A offers a simpler and more robust alternative, relying only on the geometry of the target spacecraft during the pose initialization. On the other hand, Strategy B could potentially be more accurate and independent of the geometry of the spacecraft, at the cost of increased complexity.

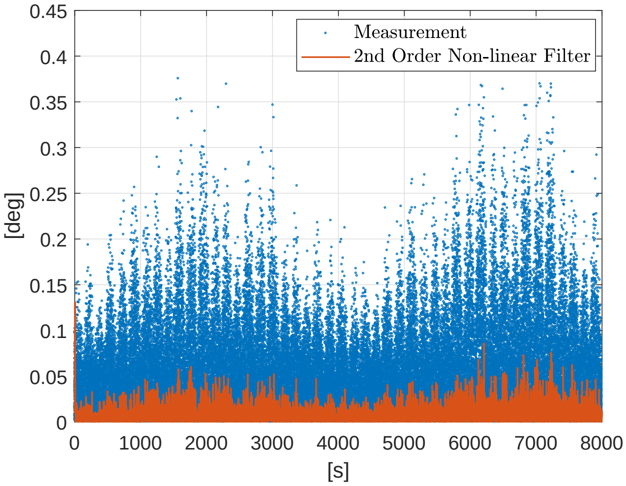


Figure 4.19: Relative Attitude Error

CHAPTER 5

Comparison of Filtering Techniques For Relative Attitude Estimation of Uncooperative Space Objects

> In theory, there is no difference between
> theory and practice. But, in practice, there is.
> Jan L. A. van de Snepscheut

The development of a peculiar filtering technique for relative attitude estimation, introduced in Chapter 4, has entailed an extensive comparative analysis with classical filters. In this chapter we review and compare the conventional and novel methods for attitude estimation, and tailor these formulations to the problem of uncooperative attitude estimation. Absolute attitude estimation has been extensively studied in the past decades and several algorithms have been proposed [88], [89]. These techniques usually rely on accurate measurements of angular rate and line of sight vector. However, the accessibility of these measurements is not guaranteed when trying to estimate the relative state of an uncooperative object, especially in space. The main difference between absolute and relative attitude estimation is that, when dealing with uncooperative objects, measurements of the target angular velocity cannot be provided to the chaser. In fact, classical sensors exploited for relative proximity operations, such as lidar or cameras, do not provide reliable information of the angular velocity of the target body. In principle, only a rough estimate of the target angular velocity can be obtained by using optical-flow techniques. Moreover, the measurements update frequency is usually lower due to the image-processing computational time. This lack of information inevitably affects the accuracy of the estimates and, therefore, some alternative filtering formulations have to be adopted. This chapter offers a comparison of different filtering techniques for relative state estimation. We start by introducing different filtering formulations, then, these alternatives are tested and compared considering a representative simulation scenario. Starting from the case in which reliable measurements of the angular velocity are available, several cases are analyzed and discussed. Convergence, steady-state error and robustness are considered as performance indexes.

5.1 Filtering Algorithms

In this section we present all the filters that have been implemented for the comparison.

5.1.1 Multiplicative Extended Kalman Filter (MEKF)

The common form of the Multiplicative Extended Kalman Filter (MEKF), utilizing the unit quaternion parametrization for the global attitude, is outlined in this subsection. It is worth noting that the MEKF commonly relies on a *linear* measurement update based on Euler angles hence limiting the covariance matrix to 3 × 3.

The measured relative angular velocity $\tilde{\boldsymbol{\omega}}$ approximates the true relative angular velocity $\boldsymbol{\omega}$ in (Equation (3.17)) up to a process disturbance $B\boldsymbol{w}$ as

$$\boldsymbol{\omega} = \tilde{\boldsymbol{\omega}} + B\boldsymbol{w}, \tag{5.1}$$

where $\boldsymbol{w}$ is a zero-mean Gaussian noise process with covariance given by

$$E\left(\boldsymbol{w}(t)\boldsymbol{w}(\tau)^T\right) = \mathbf{Q}(t)\delta(t-\tau).$$

For the case of uncooperative attitude estimation, please note that $\boldsymbol{\omega}$ and $\boldsymbol{q}$ represent the relative angular velocity and the relative quaternion expressed in the chaser frame. It is important to underline that, since the propagation (Equation (5.1)) is based only on the attitude kinematics, this formulation holds if expressed in any generic reference frame.

For the case of uncooperative attitude estimation, $\tilde{\boldsymbol{\omega}}$ is unavailable and therefore $\boldsymbol{\omega}$ is assumed to be a random process defined solely by $\mathbf{B}\boldsymbol{w}$. Hence (Equation (5.1)) becomes

$$\boldsymbol{\omega} = \mathbf{B}\boldsymbol{w}. \tag{5.2}$$

This seems to be a reasonable assumption if no prior information about the angular velocities of the target object and chaser spacecraft is available.

The vector measurements, typically output of a star camera, approximate the line of sight up to a noise process $\mathbf{D}\boldsymbol{v}$. The measurement equation is thus given, as in [89], by

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{q}) + \mathbf{D}\boldsymbol{v}; \qquad \boldsymbol{h}(\boldsymbol{q}) = \begin{bmatrix} \mathbf{A}(\boldsymbol{q})\boldsymbol{r}_1 \\ \mathbf{A}(\boldsymbol{q})\boldsymbol{r}_2 \end{bmatrix} \tag{5.3}$$

where $\boldsymbol{h}(\boldsymbol{q})$ is the concatenated vector of the noise-free vector measurements, $\mathbf{A}$ is the relative Direction Cosine Matrix (DCM), $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are two mutually perpendicular directions and $\boldsymbol{v}$ is a vector of zero-mean Gaussian noise processes with covariance given by

$$E\left(\boldsymbol{v}(t)\boldsymbol{v}(\tau)^T\right) = \mathbf{R}(t)\delta(t-\tau).$$

Please notice that although all other filters directly utilize the attitude measurement provided by the vision system on-board the chaser, standard MEKF formulation uses the line of sight measurements. Therefore, in this work, the vector measurements for the MEKF are obtained by projecting the measured attitude matrix on two mutually orthogonal unit vectors, and then fed to the MEKF, since the main motivation of the analysis is to compare the geometric filters with the usual form of the MEKF.

The Kalman filtering problem can be cast into minimization of the cost function,

$$J = \frac{1}{2}E\left[\left\|\xi^T(\boldsymbol{q}(t))\hat{\boldsymbol{q}}(t)\right\|^2\right], \tag{5.4}$$

where the term $\xi^T(\boldsymbol{q}(t))\hat{\boldsymbol{q}}(t)$ represents the error between estimated and true quaternions. To solve the optimal filtering problem, the cost J is to be optimized subject to the dynamic constraints (Equations (3.18) and (5.3)). The optimization problem admits an analytical solution for the special case of linear systems. Therefore, assuming the linear propagation for the covariance matrix, one can obtain the filtering equation together

Table 5.1: Multiplicative Extended Kalman Filter (MEKF)

Initialization:

$$\hat{\boldsymbol{q}}(t_0) = \hat{\boldsymbol{q}}_0,\ \mathbf{P}(t_0) = \mathbf{P}_0,$$

Prediction:

$$\dot{\hat{\boldsymbol{q}}} = \begin{cases} \frac{1}{2}\xi(\hat{\boldsymbol{q}})\hat{\boldsymbol{\omega}} & \text{with } \omega \text{ measurement,} \\ \mathbf{0} & \text{without } \omega \text{ measurement,} \end{cases}$$

$$\dot{\mathbf{P}}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T(t) + \mathbf{B}\mathbf{Q}\mathbf{B}^T$$

$$\mathbf{F}(t) = \begin{cases} -\hat{\boldsymbol{\omega}}(t)_\times & \text{with } \omega \text{ measurement} \\ \mathbf{0} & \text{without } \omega \text{ measurement} \end{cases}$$

$$\mathbf{B} = \mathbf{I}_{3\times3},\ \ \mathbf{Q} = \sigma_w^2\mathbf{I}_{3\times3}$$

Correction:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T\left[\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}\right]^{-1}$$

$$\mathbf{R} = \sigma_v^2\mathbf{I}_{3\times3}$$

$$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k(\hat{\boldsymbol{q}}_k^-)]\mathbf{P}_k^-$$

$$\mathbf{P}_k(+) = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\,\mathbf{P}_k(-)\,[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T$$

$$\boldsymbol{h}(\hat{\boldsymbol{q}}_k^-) = \begin{bmatrix} \mathbf{A}(\hat{\boldsymbol{q}}_k^-)\boldsymbol{r}_1 \\ \mathbf{A}(\hat{\boldsymbol{q}}_k^-)\boldsymbol{r}_2 \end{bmatrix}$$

$$\hat{\boldsymbol{q}}_k^+ = \hat{\boldsymbol{q}}_k^- + \frac{1}{2}\Xi(\hat{\boldsymbol{q}}_k^-)K_k\left[\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{q}}_k^-)\right]$$

---

with the matrix differential equation (Riccati equation) for the covariance propagation, as well as the expression for the filter gain. The complete MEKF formulation for the kinematic system defined by Equation (3.18) and the line of sight measurements (Equation (5.3)), is given in Table 5.1, see, e.g., [89], [90].

The $\mathbf{H}_k(\hat{\boldsymbol{q}}_k^-)$ term in Table 5.1 is the linearization of $\boldsymbol{h}(\boldsymbol{q})$ based on the estimated attitude and the reference directions (of the target object measured in chaser frame). Therefore, $\mathbf{H}_k(\hat{\boldsymbol{q}}_k^-)$ is calculated as

$$\mathbf{H}_k(\hat{\boldsymbol{q}}_k^-) = \begin{bmatrix} \left[\mathbf{A}(\hat{\boldsymbol{q}}_k^-)\boldsymbol{r}_1\right]_\times \\ \left[\mathbf{A}(\hat{\boldsymbol{q}}_k^-)\boldsymbol{r}_2\right]_\times \end{bmatrix},$$

where the estimated relative attitude, *i.e.*, the relative DCM, is constructed based on the estimated quaternions as

$$\mathbf{A}(\boldsymbol{q}) = \|\boldsymbol{q}\|^{-2}\left((q_4^2 - \|\boldsymbol{\varrho}\|^2)\mathbf{I}_3 + 2\boldsymbol{\varrho}\boldsymbol{\varrho}^T - 2q_4\boldsymbol{\varrho}_\times\right).$$

The attitude measurement typically provided by the vision based solutions is a rotation matrix and it can be right-multiplied by two mutually perpendicular reference directions (namely, $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$) to provide two line of sight (vector) measurements, consistent with Equation (5.3). These vector measurements can then be fused with the relative attitude estimates using the optimal gain, as outlined in Table 5.1. It is to be pointed out that the assumption made in Equation (5.2) results in trivialization of the attitude prediction step, *i.e.*, attitude prediction is switched off and only the correction step is responsible for tracking the noisy output, see Table 5.1.

5.1.2 Minimum Energy Filter on $SO(3)$

The formulation of the minimum energy filter on $SO(3)$ is based on the perturbed kinematic model:

$$\dot{\mathbf{R}}_{\mathcal{CT}}(t) = \mathbf{R}_{\mathcal{CT}}(\tilde{\boldsymbol{\omega}}(t)_\times + g\delta(t)), \quad \mathbf{R}_{\mathcal{CT}}(0) = \mathbf{R}_{\mathcal{CT}0}, \tag{5.5}$$

where $\mathbf{R}_{\mathcal{CT}}(t) \in SO(3)$ is the relative rotation matrix expressed in the chaser frame, and $g\delta(t) \in so(3)$ is the process disturbance. Since the measured relative angular velocity $\tilde{\boldsymbol{\omega}}$ is not available in an uncooperative attitude estimation setup, $\dot{\mathbf{R}}_{\mathcal{CT}}(t)$ will be assumed to be driven only by $g\delta(t)$. The measurement $\mathbf{Y}(t) \in SO(3)$, which is again a relative rotation matrix expressed in the chaser frame, is typically obtained through a vision-based solution. The measurement $\mathbf{Y}(t)$ can be modeled as

$$\mathbf{Y}(t) = \mathbf{R}_{\mathcal{CT}}(t)\epsilon(t), \tag{5.6}$$

where $\epsilon(t) \in SO(3)$ is the measurement error. The minimum-energy filtering approach aims at obtaining a state estimate $\hat{\mathbf{R}}_{\mathcal{CT}}$ by minimizing a cost function J at each time step t, given the actual measurements (Equation (5.6)). To obtain the state estimate $\hat{\mathbf{R}}_{\mathcal{CT}}$, one seeks a combination of the unknowns (*i.e.,* the initial state $\mathbf{R}_{\mathcal{CT}0}$ and the process and measurement disturbances δ_0, ϵ_0) that is compatible with the actual measurements (Equation (5.6)) and system model (Equation (3.19)). The resulting cost function is:

$$\begin{aligned} J = {} & \frac{1}{4}trace\left[(\mathbf{R}_{\mathcal{CT}0} - \mathbf{I})^T\,\mathbf{K}_0^{-1}\,(\mathbf{R}_{\mathcal{CT}0} - \mathbf{I})\right] \\ & + \int_0^T \left(\frac{1}{2}trace\left[\delta^T(\tau)\delta(\tau)\right] + \frac{1}{4}trace\left[(\epsilon(\tau) - \mathbf{I})^T\,(\epsilon(\tau) - \mathbf{I})\right]\right) d\tau, \end{aligned} \tag{5.7}$$

Table 5.2: The Minimum Energy Filter on $SO(3)$

Initialization:

$$\hat{\mathbf{R}}_{\mathcal{CT}}(t_0) = \hat{\mathbf{R}}_{\mathcal{CT}0},\ \mathbf{K}(t_0) = \mathbf{K}_0,$$

Filtering:

$$\dot{\hat{\mathbf{R}}}_{\mathcal{CT}} = \begin{cases} \hat{\mathbf{R}}_{\mathcal{CT}}(\hat{\boldsymbol{\omega}}(t)_\times - \mathbb{P}_a(\mathbf{K}\mathbf{Y}^T\hat{\mathbf{R}}_{\mathcal{CT}})) & \text{with } \omega \text{ measurement,} \\ -\hat{\mathbf{R}}_{\mathcal{CT}}\mathbb{P}_a(\mathbf{K}\mathbf{Y}^T\hat{\mathbf{R}}_{\mathcal{CT}}) & \text{without } \omega \text{ measurement,} \end{cases}$$

$$\dot{\mathbf{K}}(t) = \frac{1}{2}\mathbf{Q} - \mathbf{K}(\mathbf{Y}^T\hat{\mathbf{R}}_{\mathcal{CT}} + \hat{\mathbf{R}}_{\mathcal{CT}}^T\mathbf{Y})\mathbf{K} + \mathbf{K}(t)\hat{\boldsymbol{\omega}}(t)_\times - \hat{\boldsymbol{\omega}}(t)_\times\mathbf{K}(t),$$

$$\mathbb{P}_a(\mathbf{X}) = \frac{1}{2}(\mathbf{X} - \mathbf{X}^T).$$

---

where $\mathbf{K}_0$ is a symmetric positive definite matrix. If the signals associated with a given hypothesis minimize the cost over all possible choices of unknown signals, then the hypothesis is termed optimal. The value of the associated state trajectory is considered as the optimal (minimum-energy) state estimate at time T. As it is evident from the cost function, the new measurements arriving at every sample time may yield a different optimal trajectory of the state $\mathbf{R}_{\mathcal{CT}}^*(t)$ thus making the problem infinite dimensional. Using Mortensen's approach [45], the optimal filtering problem is broken into a recursive filtering equation and a gain update equation. A second-order minimum energy filter for attitude estimation has been provided in [91]. The optimality gap between the full order minimum energy filter and its second-order approximation is small and an analytical upper-bound on this gap has been provided in [92]. Alternatively, in [91], [93] the upper bound on the optimality gap $\mathbf{W}(t)$, has also been calculated using a candidate Lyapunov function, and rearranging the terms after differentiating it along the trajectories of the system. The relationship between the cost J incurred by the second-order approximation of minimum energy filter, the optimal cost J^* of the optimal minimum energy filter, and the upper bound $\mathbf{W}(t)$ on the optimality gap, is given by:

$$J - J^* = J - \frac{1}{4}\int_0^T trace\left[\left(\mathbf{Y}^T\mathbf{R}_{\mathcal{CT}} - \mathbf{I}\right)^T\left(\mathbf{Y}^T\mathbf{R}_{\mathcal{CT}} - \mathbf{I}\right)\right]d\tau \leq \mathbf{W}(t). \tag{5.8}$$

For the system described by Equation (3.19), the second-order minimum energy filter is given in Table 5.2, see [91], [93] for details.

5.1.3 Attitude Observer on $SO(3)$

The attitude observer on $SO(3)$ consists in the same filtering equation as that of a minimum energy filter, given in Table 5.2. The gain of the observer, instead of evolving as a solution of the Riccati equation as for the minimum energy filter, is set to a constant matrix which can be found in a number of ways. In the present case it was observed that a favorable trade-off between convergence and steady-state performance can be obtained if the observer gain is set equal to the steady-state gain matrix of the minimum energy filter. It is noted that although the fixed-gain observer is not an optimal estimator, it is free from the instabilities which may arise from the numerical integration of the Riccati equation in an optimal or near-optimal filter.

5.1.4 2^{nd} Order Minimum Energy Filter

A further development of the presented minimum energy filter on $SO(3)$ was introduced by Saccon *et al* [82]. In the cited paper an explicit formula for a second-order optimal nonlinear filter on general Lie groups was developed. As an example, the authors develop a second-order filter on $SO(3)$ which depends on the choice of affine connection which encodes the nonlinear geometry of the state space. When the symmetric Cartan-Shouten (0)-connection is chosen, the filter has the familiar form of a gradient estimator along with a perturbed Riccati-type matrix differential equation which describes the evolution of the filter gain. The second-order minimum energy filter [82] is based on the choice of an affine connection, and it is not straightforward to extend the proposed filter to relative attitude dynamics, except for the special case of non-rotating chaser spacecraft, *i.e.*,

$$\dot{\boldsymbol{\omega}} = \mathbf{R}_{\mathcal{CT}}\mathbf{I}_t^{-1}\left[-\boldsymbol{\omega}_t|_{\mathcal{T}} \times \mathbf{I}_t\boldsymbol{\omega}_t|_{\mathcal{T}}\right] + \mathbf{B}\boldsymbol{\delta}. \tag{5.9}$$

Since the chaser spacecraft cannot always be expected to be non-rotating in real applications, this assumption of a non-rotating chaser spacecraft limits the practical use of this formulation of second-order minimum energy filter for a generic relative attitude estimation problem. To generalize this method to the case of an arbitrarily rotating chaser spacecraft, a different affine connection has to be developed by incorporating the relative attitude dynamics (Equation (3.18)) and that will result into a slightly different filtering formulation. The measurements are assumed to be the noisy line of sight vectors $\boldsymbol{r}_i$, as:

$$\boldsymbol{r}_i = \mathbf{R}_{\mathcal{CT}}^T\bar{\boldsymbol{r}}_i + d_i\boldsymbol{\epsilon}, \tag{5.10}$$

where $\boldsymbol{\epsilon}$ is the measurement noise vector and d_i is the scaling factor.

Table 5.3: The 2^{nd} Order Minimum Energy Filter on $SO(3)$

Initialization:

$$\hat{\mathbf{R}}_{\mathcal{CT}}(t_0) = \hat{\mathbf{R}}_{\mathcal{CT}0},\ \hat{\boldsymbol{\omega}}(t_0) = \hat{\boldsymbol{\omega}}_0,\ \mathbf{K}(t_0) = \mathbf{K}_0,$$

Filtering:

$$\dot{\hat{\mathbf{R}}}_{\mathcal{CT}} = \hat{\mathbf{R}}_{\mathcal{CT}}\left(\hat{\boldsymbol{\omega}}(t) + K_{11}\boldsymbol{r}^{\mathbf{R}} + K_{12}\boldsymbol{r}^{\omega}\right)_\times ,$$

$$\dot{\hat{\boldsymbol{\omega}}} = \mathbf{I}_t^{-1}\left((\mathbf{I}_t\hat{\boldsymbol{\omega}})_\times \hat{\boldsymbol{\omega}} + \boldsymbol{\tau}\right) + K_{21}\boldsymbol{r}^{\mathbf{R}} + K_{22}\boldsymbol{r}^{\omega},$$

$$\dot{\mathbf{K}}(t) = -\alpha\mathbf{K} + \mathbf{AK} + \mathbf{KA}^T - \mathbf{KEK} + \mathbf{BR}^{-1}\mathbf{B}^T - \mathbf{WK} - \mathbf{KW}^T,$$

where

$$\begin{bmatrix} \boldsymbol{r}^{\mathbf{R}} \\ \boldsymbol{r}^{\omega} \end{bmatrix} = \begin{bmatrix} -u_1(\hat{\boldsymbol{r}}_1 \times \boldsymbol{r}_1) - u_2(\hat{\boldsymbol{r}}_2 \times \boldsymbol{r}_2) \\ \mathbf{0} \end{bmatrix},$$

$$u_i = \frac{b^2}{d_i^2},$$

$$\hat{\boldsymbol{r}}_i = \hat{\mathbf{R}}_{\mathcal{CT}}^T\bar{\boldsymbol{r}}_i, \quad \boldsymbol{r}_i = \hat{\mathbf{R}}_{\mathcal{CT}}^T\bar{\boldsymbol{r}}_i + d_i\boldsymbol{\epsilon},$$

$$\mathbf{A} = \begin{bmatrix} -\hat{\boldsymbol{\omega}}_\times & \mathbf{I} \\ 0 & \mathbf{R}_{\mathcal{CT}}\mathbf{I}_t^{-1}\left[(\mathbf{I}_t\hat{\boldsymbol{\omega}}) - \hat{\boldsymbol{\omega}}_\times\mathbf{I}_t\right] \end{bmatrix},$$

$$\mathbf{E} = \begin{bmatrix} \sum_{i=1}^{2}\boldsymbol{u}_i\left((\hat{\boldsymbol{r}}_i)_\times(\boldsymbol{r}_i)_\times + (\boldsymbol{r}_i)_\times(\hat{\boldsymbol{r}}_i)_\times\right)/2 & 0 \\ 0 & 0 \end{bmatrix},$$

$$\mathbf{BR}^{-1}\mathbf{B}^T = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{BR}^{-1}\mathbf{B}^T \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} \frac{1}{2}\left(K_{11}\boldsymbol{r}^{R} + K_{12}\boldsymbol{r}^{\omega}\right)_\times & 0 \\ 0 & 0 \end{bmatrix}.$$

The complete formulation of the second-order filter on $SO(3)$ for Equation (5.9) and the symmetric Cartan-Shouten (0)-connection is provided in Table 5.3.

5.1.5 2^{nd} Order Minimum Energy Filter without Dynamics

As highlighted in Section 5.1.4, the use of an affine connection in estimating the angular velocity limits the applicability of a second-order minimum energy filter for generic relative attitude estimation problems beyond the special case of a non-rotating chaser spacecraft. Furthermore, for the cases of uncooperative objects in space (*e.g.*, derelict satellites, space debris), the inertia matrix of the target is usually uncertain. Therefore, a different formulation of the second-order minimum energy filter is presented in Section 4.3.2. In fact, a simplification to the filter proposed in [82] is introduced without considering the dynamics of the system. Equations (4.5) and (4.6) are directly expressed in relative terms and in the chaser frame, not relying on the dynamics equation. It is worth underlying that this formulation does not require the knowledge of the inertia matrix. This is very important when the inertia matrix of the body is uncertain or completely unknown, as in the case of derelict satellites or asteroids. The formulation of the kinematic second-order filter (*i.e.*, without the dynamics terms) is summarized in Table 4.1.

5.2 Simulation Scenarios and Results

Since the objective of this study is to compare the performance of the filtering algorithms described above for relative attitude estimation of uncooperative space objects, a torque-free tumbling motion has been simulated for the target spacecraft. In particular, the relative rotational motion has been simulated according to Equation (3.17). In our simulation cases, we assumed, for simplicity, that the chaser spacecraft is not rotating and therefore $\omega_c = [0\ 0\ 0]^T$, for which Equation (3.17) reduces to Equation (5.9). The angular velocity of the target depends on the simulation case and it is reported in Table 5.4.

The motion has been simulated using the Euler equation for rigid body, using the inertia matrix of the Envisat spacecraft [94] as

$$I = \begin{bmatrix} 16979.74 & 0 & 0 \\ 0 & 124801.21 & 0 \\ 0 & 0 & 129180.25 \end{bmatrix} kgm^2. \tag{5.11}$$

The measurements are generated at 10 Hz (similarly to [95]) transforming the relative rotation matrix to Euler angles and adding noise to each of the three angles according to a normal distribution with zero mean and a standard deviation of $6 \cdot 10^{-2}$ rad. This noise level is representative of commonly used vision-based attitude determination techniques [57]. Moreover, for a more complete comparison, the MEKF and the minimum energy filter are also tested for the hypothetical case when angular velocity measurements of the target body are also available. The process noise for this hypothetical case, which reflects the mismatch between measured and actual angular velocity, was modeled according to a normal distribution with zero mean and a standard deviation of $1 \cdot 10^{-3}$ rad/s. It is worth emphasizing that this measurement is not available when dealing with uncooperative objects unless the target body is cooperative and its gyro measurements are accessible from the chaser spacecraft. Since all the filters being considered assume different process and measurement models, all of these filters are individually tuned for a good trade-off between convergence, steady-state performance and robustness. The MEKF, with and without angular velocity measurement, is tuned with $\mathbf{Q} = (1 \cdot 10^{-3})^2 \mathbf{I}_3$, $\mathbf{R} = (1 \cdot 10^{-3})^2 \mathbf{I}_3$ and $\mathbf{Q} = (6 \cdot 10^{-2})^2 \mathbf{I}_3$, $\mathbf{R} = (1 \cdot 10^{-3})^2 \mathbf{I}_3$, respectively. The minimum energy filter with angular velocity measurement is tuned with $\mathbf{Q} = (1 \cdot 10^{-3}/6 \cdot 10^{-2})^2 \mathbf{I}_3$, while for the case without angular velocity measurement we set $\mathbf{Q} = 3\mathbf{I}_3$. The observer gain is fixed equal to the steady-state values of the gain of the minimum energy filter, *i.e.*, $\mathbf{K}_{observer} \equiv 2.1\mathbf{I}_3$. Finally, we take $\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T = 5 \cdot 10^{-2}\mathbf{I}_3$, $u_i = 6 \cdot 10^{-2}$ and $\alpha = 1 \cdot 10^{-3}$ for the second-order minimum energy filter, while $\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T = 1 \cdot 10^{-2}\mathbf{I}_3$ is used for the second-order minimum energy filter without the dynamics. It is to be reminded that these tuning settings are not claimed to be the best possible settings, rather these settings were found to offer reasonable trade-off between convergence/stability and the steady-state performance. By testing and comparing the filters in various off-design conditions, we make an attempt to decouple the effects of tuning parameters on the performance of the filters. We are interested in studying the estimation errors, especially their sensitivity to the variations in measurement noise intensity, initial conditions, and angular velocities of the non-cooperative object in space. Therefore, we consider the cases outlined in Table 5.4, which are used for trajectory generation. The filters, however, are not "*aware*" of the configurations of any test case and are used with the nominal tuning. To cope with initial estimation errors all the first order filters are initialized with

$$\mathbf{K}_0 = 5\mathbf{I}_3,$$

Table 5.4: Considered Simulation Cases

Cases	Number of Runs	Initial Estimation Error (deg)	Measurement Noise Std. (deg)	Object's Angular Velocity (deg/s)	Moment of Inertia
A1	100	$\phi, \theta, \psi \in [-28.6, +28.6]$	$\sigma = 3.4$	$\begin{bmatrix} 1 \\ 0.1 \\ 0.3 \end{bmatrix}$	I
A2	100	$\phi, \theta, \psi \in [-28.6, +28.6]$	$\sigma = 3.4$	$\begin{bmatrix} 5 \\ 0.1 \\ 0.3 \end{bmatrix}$	I
B1	100	$\begin{bmatrix} 10 \\ -10 \\ 10 \end{bmatrix}$	$\sigma \sim WN(3.4, 0.3 \times 3.4)$	$\begin{bmatrix} 1 \\ 0.1 \\ 0.3 \end{bmatrix}$	I
B2	100	$\begin{bmatrix} 10 \\ -10 \\ 10 \end{bmatrix}$	$\sigma \sim WN(3.4, 0.3 \times 3.4)$	$\begin{bmatrix} 5 \\ 0.1 \\ 0.3 \end{bmatrix}$	I
C1	100	$\phi, \theta, \psi \in [-28.6, +28.6]$	$\sigma \sim WN(3.4, 0.3 \times 3.4)$	$\begin{bmatrix} 1 \\ 0.1 \\ 0.3 \end{bmatrix}$	I
C2	100	$\phi, \theta, \psi \in [-28.6, +28.6]$	$\sigma \sim WN(3.4, 0.3 \times 3.4)$	$\begin{bmatrix} 5 \\ 0.1 \\ 0.3 \end{bmatrix}$	I
D	100	$\phi, \theta, \psi \in [-28.6, +28.6]$	$\sigma \sim WN(3.4, 0.3 \times 3.4)$	$\begin{bmatrix} 5 \\ 0.1 \\ 0.3 \end{bmatrix}$	$I \in [I - 0.45I, I + 0.45I]$

while the second-order filters are initialized with

$$\mathbf{K}_0 = \begin{bmatrix} 42.5\mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & 0.909\mathbf{I}_3 \end{bmatrix}.$$

All the filters in this study are run with an update rate of 10 Hz.

The cases A1 and A2 are designed to investigate the effect of variations in initial estimation error and in angular velocity of the uncooperative space object. For A1 and A2, the measurements of the true trajectory were simulated with a constant zero-mean measurement noise of standard deviation $6 \cdot 10^{-2}$ rad (or 3.4 deg) added to the samples of true trajectory (for each Euler angle), and the initial attitude was constructed from a set of Euler angles each of which were uniformly drawn from $\left[-0.5, +0.5\right]$ rad (or $\left[-28.6, +28.6\right]$ deg). A1 and A2 were simulated with the nominal angular velocity $\boldsymbol{\omega}_1$ (*i.e.*, equal to that of Envisat) and $\boldsymbol{\omega}_2$, respectively, given by:

$$\boldsymbol{\omega}_1 = \begin{bmatrix} 1 \\ 0.1 \\ 0.3 \end{bmatrix} deg/s, \quad \boldsymbol{\omega}_2 = \begin{bmatrix} 5 \\ 0.1 \\ 0.3 \end{bmatrix} deg/s. \tag{5.12}$$

Both A1 and A2 are run 100 times, using fixed inertia matrix which equals that of Envisat (Equation (5.11)).

For cases B1 and B2, the initial conditions are fixed to $\begin{bmatrix}10 & -10 & 10\end{bmatrix}^T$ *deg*, and the initial angular velocity is $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$, respectively. The standard deviation of actual measurement noise which is used in the measurement generation is set to be a random variable. The random variable has a mean equal to the nominal standard deviation σ, and a standard deviation 30% of the nominal standard deviation. Both cases are run 100 times each, but the filters are unaware of the actual measurement noise and therefore are still tuned for the nominal standard deviation. The inertia matrix is kept fixed to the value given in Equation (5.11) in these simulations and is assumed to be perfectly known.

The cases C1 and C2 are run for 100 times each, and combine the uncertain initial conditions from cases A1 and A2, with the uncertain measurement noise properties from B1 and B2. The inertia matrix is still assumed to be fixed (*i.e.*, Envisat inertia matrix in these cases) and perfectly known a priori.

Finally the case D is run for 100 times, with all the same parameters as that of C2 but with the uncertain inertia matrix this time. The diagonal terms of the inertia matrix are now assumed to be uniformly drawn from a set of $[-45, +45]$ % of their nominal values. This has been done to evaluate how an inaccurate knowledge of the inertia properties of the body affects the estimation.

5.2.1 Single Run (Case C2)

To visualize the convergence and steady-state performance of various filters, results from a single run (under simulation parameters of case C2) are reported in Figure 5.1 and Figure 5.2. The metric adopted for the estimation error is the axis-angle representation which yields a scalar measure of distance between two reference frames.

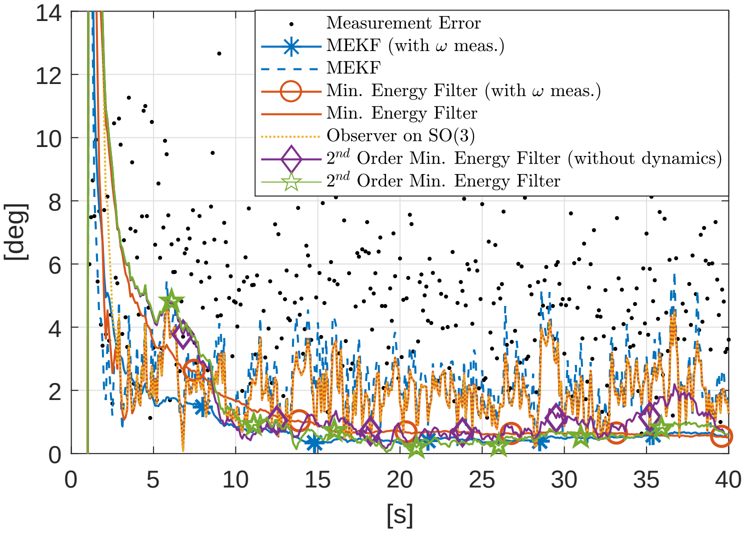


Figure 5.1: Single Run Estimation Errors - Convergence

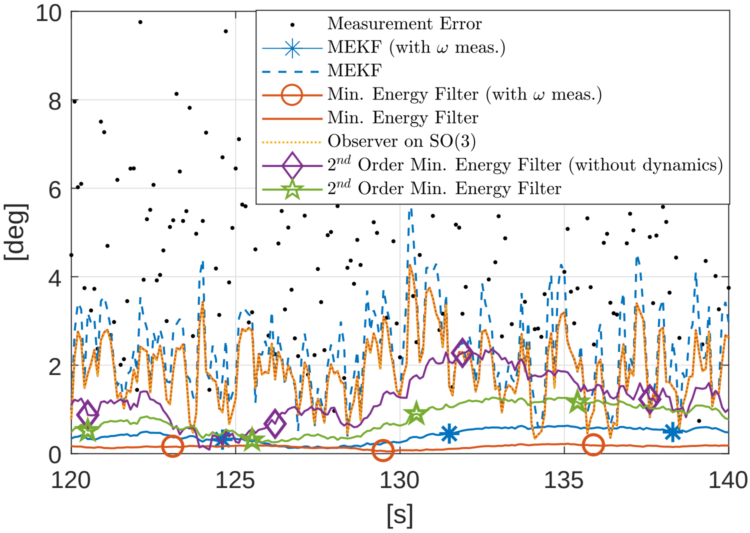


Figure 5.2: Single Run Estimation Errors - Steady-State

Figure 5.1 depicts the convergence properties of estimation errors of all the filters under a single run, and therefore only initial 40 seconds are shown. Both variants of MEKF (*i.e.*, with and without angular velocity measurement) are the quickest to converge. This stems from the fact that measurement update step in the MEKF (Table 5.1, see also [89], [90]) is based on first order approximation of quaternion update and therefore the measurement step is *linear*. The minimum energy filter is slower to converge than the MEKF, due to a fully nonlinear *geometric* measurement update. The minimum energy filter and the $SO(3)$ observer are still quicker to converge as compared to the second-order minimum energy filters, while the minimum energy filter with angular velocity measurement is the slowest to converge. The second-order filters are prone to convergence problems if initialized with a large estimation error, which stems from the fact that second-order filters estimate not only the attitude but also the angular velocity. If initialized with large estimation error, during the convergence the second-order filters *build up* an angular velocity in the direction of convergence and this creates undesired oscillations after the filter has reached steady-state. To circumvent this problem, the angular velocity of second-order filters is not updated for initial 5.5 seconds, and hence the angular velocity is only estimated after 5.5 seconds when the initial transients have vanished. This transient time of 5.5 was chosen because it provided a reasonable trade-off between speed of convergence and the steady-state error.

The slow convergence of the minimum energy filter in comparison to MEKF is in contradiction to [81], but the possible explanation could be that the MEKF adopted in [81] is not the usual MEKF we have considered in this analysis, see *e.g.*, [89], [90]. The MEKF used in [81] contains the same nonlinear measurement update equation as the minimum energy filter, and differs from the minimum energy filter only in its gain update equation.

The steady-state performance of the filters, in a single run, is shown in Figure 5.2. The MEKF turns out to be the worst performer at steady-state, which only slightly improves the measurements. Since the minimum energy filter and the observer designed on $SO(3)$ are equivalent at steady-state, we see that their estimation errors are identical at steady-state. The second-order minimum energy filter (without dynamics) performs better than the minimum energy filter but worse than the second-order minimum energy filter, whose performance nears the performance of the MEKF (with angular measurement) and the minimum energy filter (with angular measurement).

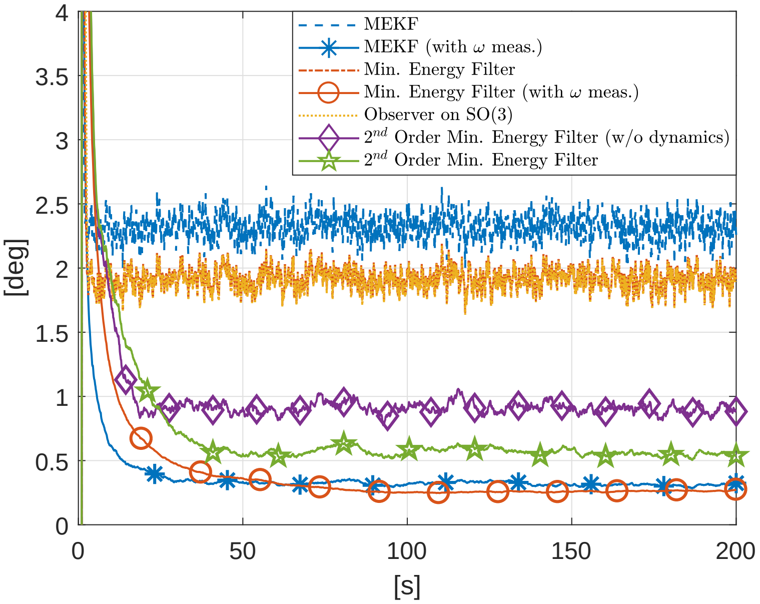


Figure 5.3: Average Estimation Errors - Case A1

5.2.2 Cases A1, A2 [Random Initial Conditions, Known Measurement Noise]

As outlined in Table 5.4, the cases A1 and A2 are designed to investigate the effect of variations in initial estimation error and in angular velocity of the uncooperative space object. The estimation errors are averaged over 100 Monte-Carlo runs, and presented in Figure 5.3 and Figure 5.4 for case A1 and Case A2, respectively. A more quantitative measure of filters performance in transient (i.e., initial 60 seconds) and steady-state (i.e., 60-200 seconds) is the Root Mean Squared (RMS) estimation error which is reported in Table 5.5, for both cases.

For the case A1, it is evident from Figure 5.3 and Table 5.5 that both variants of MEKF are quickest to converge, in comparison with all other filters. The observer on $SO(3)$ and the minimum energy filter also exhibit good convergence properties, while the second-order minimum energy filters are slower than the minimum energy filter. The minimum energy filter with angular velocity measurements is the slowest to reach its steady-state error. At the steady-state, the second-order minimum energy filter exhibits the best performance, after the filters which utilize the angular velocity measurements. The second-order minimum energy filter without the dynamics terms containing inertia matrix is slightly worse than the

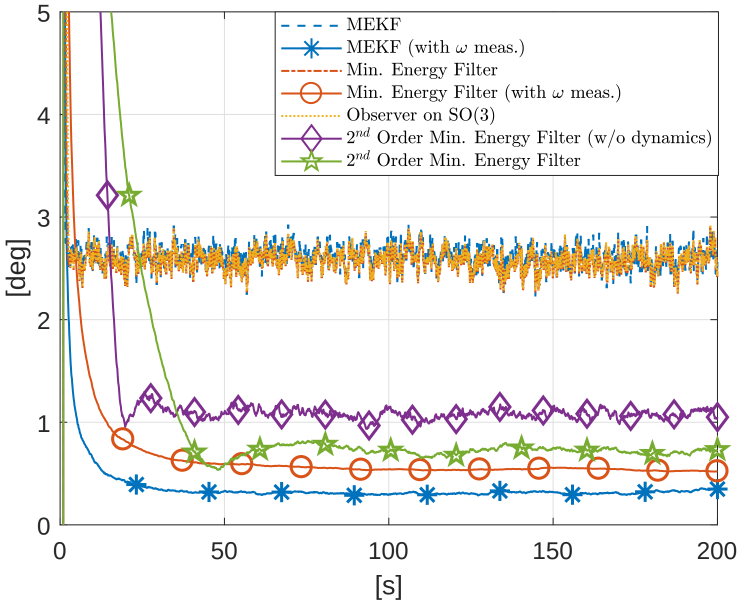


Figure 5.4: Average Estimation Errors - Case A2

Table 5.5: RMSE (deg): Cases A1 and A2

Filters	**Transient Error**		**Steady-State Error**	
	Case A1	Case A2	Case A1	Case A2
MEKF (with ω measurement)	1.29	1.19	0.34	0.33
Minimum energy filter (with ω measurement)	2.59	2.57	0.28	0.56
MEKF	2.71	2.97	2.52	2.83
Minimum energy filter	2.88	3.35	2.08	2.76
Observer on $SO(3)$	3.90	4.19	2.06	2.77
2^{nd} order Minimum energy filter	3.37	4.63	0.99	1.15
2^{nd} order Minimum energy filter (with dynamics)	3.35	5.09	0.61	0.78

the second-order minimum energy filter, but it is free from the need of accurate absolute attitude estimates as well as from the need to have any idea about the moment of inertia matrix of the target. The minimum energy filter is further significantly worse, but is quite better than the MEKF which is the worst at steady-state. The observer on $SO(3)$ also performs better than MEKF and its steady-state performance is almost identical to the minimum energy filter.

For the case A2, we note from Figure 5.4 and Table 5.5 that the performance of almost all the filters deteriorates when tracking the attitude of a body rotating at a higher angular velocity. However, the performance of MEKF, with and without the angular velocity measurement, is less affected and we conjecture that this characteristic of the MEKF also owes its existence to the *linear* measurement update step. The convergence properties of all the filters are similar to case A1, except second-order filters which take quite longer to converge in this case. At the steady-state, second-order filters are still the best performers which tend to achieve the estimation errors closer to filters operating on the angular velocity measurement. The minimum energy filter and the observer on $SO(3)$ perform slightly better than the MEKF, but their filter is significantly worse than the second-order filters.

5.2.3 Cases B1,B2 [Known Initial Conditions, Uncertain Measurement Noise]

The cases B1 and B2 are mainly selected to probe the sensitivity of filters' performance under random variations in the measurement noise intensity. The standard deviation of actual measurement noise is assumed to be a random variable in this case. This random variable has a mean equal to the nominal standard deviation σ, and a standard deviation equal to 30% of the nominal standard deviation. Both cases are run 100 times each, but the filters are unaware of the actual measurement noise and therefore are used with the nominal tuning.

The estimation errors averaged over 100 Monte-Carlo runs are provided in Figure 5.5 and Figure 5.6 , for case B1 and B2, respectively. Table 5.6 contains the Mean Square Error (MSE) of different filters for a direct comparison. The results are mainly consistent with the results presented in the previous section. The MEKF is the quickest to converge but the worst performer at the steady-state, although with a much lower margin for the case B2. The observer is slower than the minimum energy filter in terms of convergence but, as can be expected, its performance is almost indistinguishable from the minimum energy filter at the steady-state. The

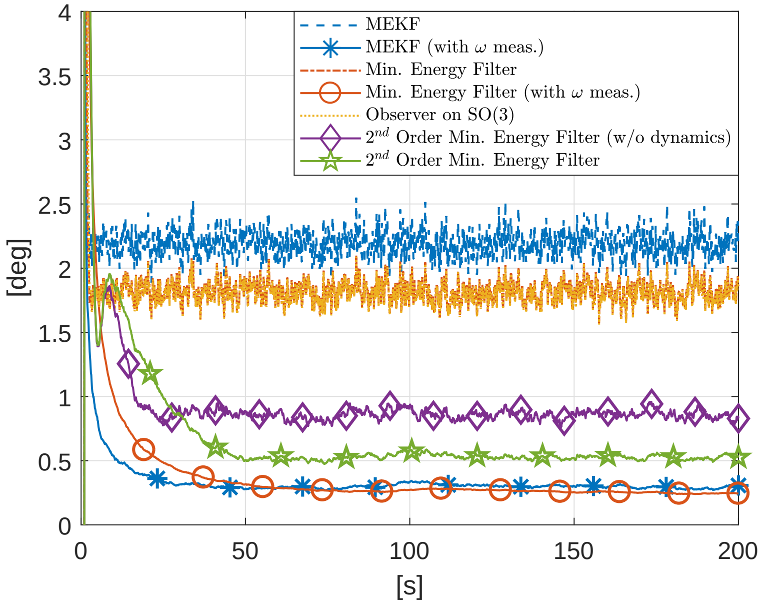


Figure 5.5: Average Estimation Errors - Case B1

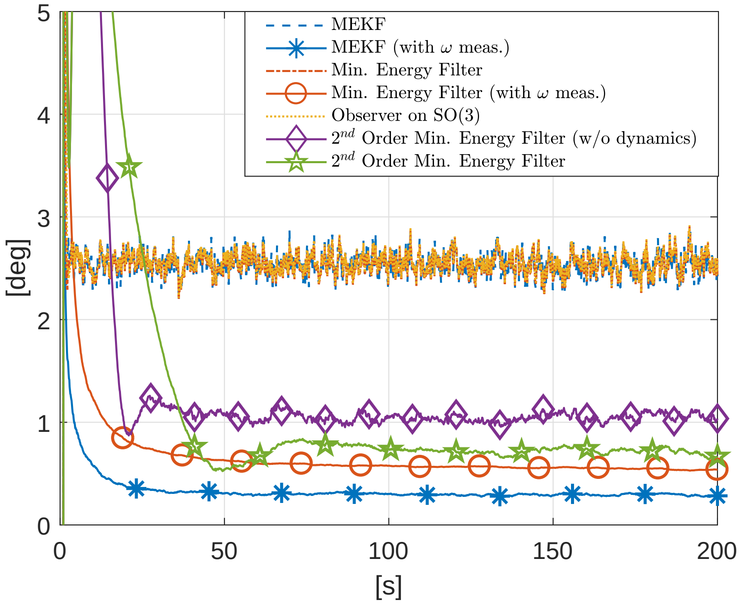


Figure 5.6: Average Estimation Errors - Case B2

Table 5.6: RMSE (deg): Cases B1 and B2

Filters	**Transient Error**		**Steady-State Error**	
	Case B1	Case B2	Case B1	Case B2
MEKF (with ω measurement)	0.70	0.72	0.33	0.33
Minimum energy filter (with ω measurement)	1.63	1.75	0.28	0.57
MEKF	2.53	2.83	2.49	2.79
Minimum energy filter	2.35	2.95	2.05	2.73
Observer on $SO(3)$	2.80	3.31	2.04	2.74
2^{nd} order Minimum energy filter	2.23	3.74	0.97	1.14
2^{nd} order Minimum energy filter (with dynamics)	2.21	4.33	0.59	0.78

minimum energy filter is faster to converge as compared to the second-order minimum energy filters, but at the expense of considerable degradation in performance at the steady-state. The second-order minimum energy filter without dynamics remarkably improves the performance in comparison with the minimum energy filter, and the second-order minimum energy filter slightly improves it further close to the order of filters which utilize angular velocity measurement. The convergence of second-order filters is, however, worse than the minimum energy filter and they are prone to oscillations if initialized with a large estimation error especially for the higher angular velocity case (*i.e.*, case B2).

Furthermore, we also notice the same trends of degradation in steady-state performance in higher angular velocity case, and the performance of the MEKF suffers the least also in this case.

5.2.4 Cases C1, C2 [Random Initial Conditions, Uncertain Measurement Noise]

The case C1 and C2 combine the uncertain initial conditions from Case A1 and A2, with the uncertain measurement noise properties from B1 and B2. The inertia matrix is still assumed to be fixed and perfectly known. The estimation errors averaged over 100 Monte-Carlo runs are presented in Figure 5.7 and Figure 5.8 , for case C1 and D2, respectively. While the MSE achieved of different filters are illustrated in Table 5.7.

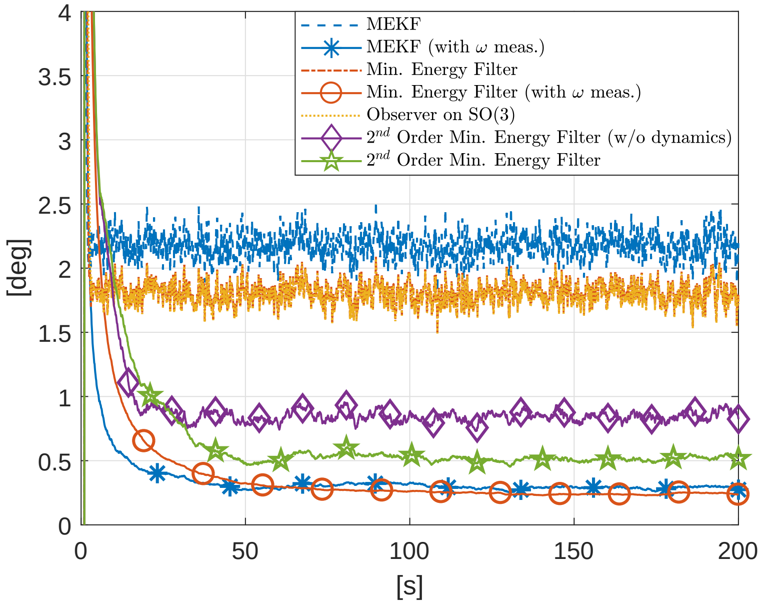


Figure 5.7: Average Estimation Errors - Case C1

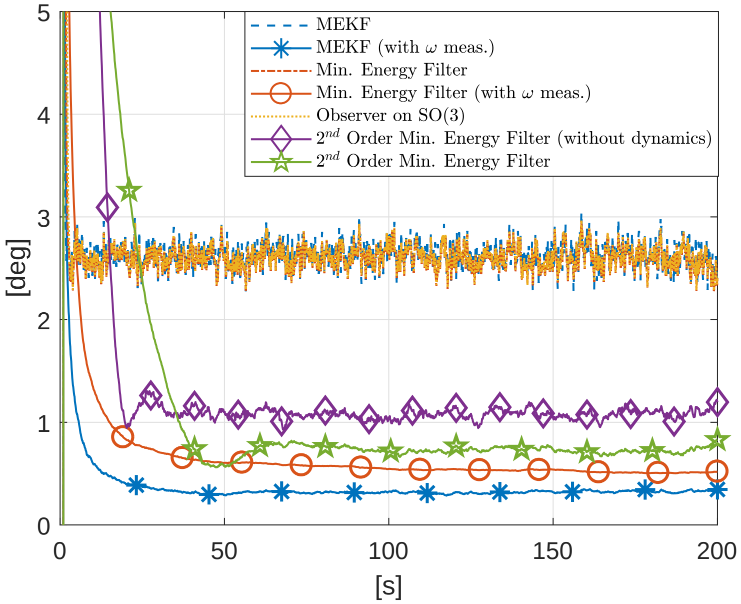


Figure 5.8: Average Estimation Errors - Case C2

Table 5.7: RMSE (deg): Cases C1 and C2

Filters	**Transient Error**		**Steady-State Error**	
	Case C1	Case C2	Case C1	Case C2
MEKF (with ω measurement)	1.31	1.21	0.33	0.35
Minimum energy filter (with ω measurement)	2.58	2.64	0.27	0.55
MEKF	2.67	3.03	2.47	2.91
Minimum energy filter	2.86	3.41	2.04	2.82
Observer on $SO(3)$	3.85	4.26	2.03	2.83
2^{nd} order Minimum energy filter	3.37	4.65	0.96	1.18
2^{nd} order Minimum energy filter (with dynamics)	3.36	5.10	0.59	0.80

The MEKF is the fastest to converge, to be followed by the observer on $SO(3)$ and the minimum energy filter. The second-order minimum energy filters are slowest to converge, especially in the higher angular velocity case. The minimum energy filter with angular velocity measurement is the slowest to converge, due to the fully nonlinear measurement update. The steady-state properties of the observer on $SO(3)$ and the minimum energy filter are much better than the MEKF in the case C1, but only slightly better in the case C2. The second-order minimum energy filter, as in previous test cases, tends to approach the performance of filters utilizing the angular velocity measurement but it requires the knowledge of the inertia matrix as well as the reliable absolute attitude estimates. The steady-state performance of the second-order minimum energy filter without the dynamics term is very close to the performance of the second-order minimum energy filter in both cases, but with the added advantage that it does not require any knowledge about the inertia matrix or the absolute attitude of the chaser spacecraft.

5.2.5 Case D [Random Initial Conditions, Uncertain Measurement Noise, Uncertain Inertia of the Target]

Finally with the parameters of case D, simulations are run for 100 times where we take the same parameters as that of C2 but with the uncertain inertia matrix this time. The diagonal terms of the Inertia matrix are assumed to be uniformly drawn from a set of $[-45, +45]\,\%$ of their nominal values.

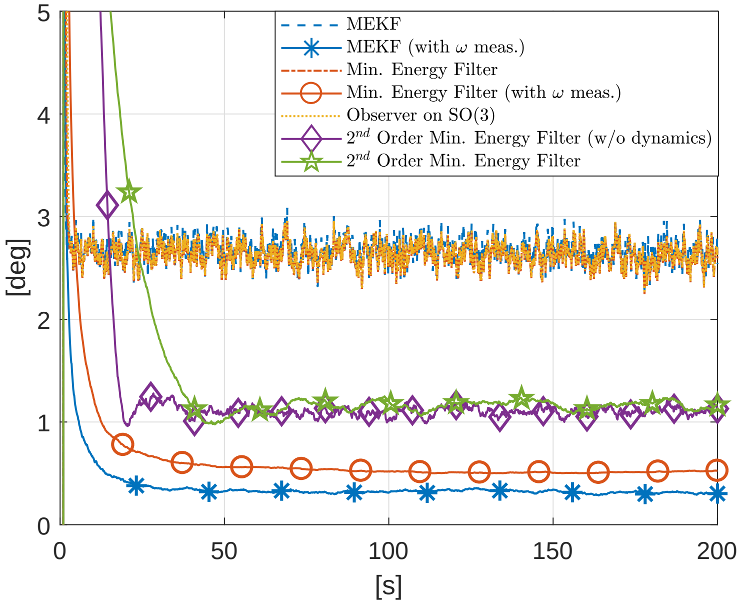


Figure 5.9: Average Estimation Errors - Case D

Table 5.8: RMSE (deg): Case D

Filters	Transient Error	Steady-State Error
MEKF (with ω measurement)	1.22	0.35
Minimum energy filter (with ω measurement)	2.60	0.55
MEKF	3.02	2.88
Minimum energy filter	3.39	2.80
Observer on $SO(3)$	4.26	2.81
2^{nd} order Minimum energy filter	4.70	1.18
2^{nd} order Minimum energy filter (with dynamics, uncertain inertia matrix)	5.21	1.28

The Figure 5.9 and Table 5.8 contain the results of case D. Since it is exactly the same case as C2 except for the uncertain variations in the inertia matrix, the results are similar to the ones presented in the previous subsection, except for the second-order minimum energy filter which depends on the inertia matrix. With this large uncertainty in the inertia matrix, which is not unusual for uncooperative objects in the space, the performance of the second-order minimum energy filter slightly degrades to a level comparable to the performance of the second-order minimum energy filter without dynamics.

The results in this case suggest that the usage of a second-order minimum energy filter without dynamics term may actually yield better performance than the standard second-order minimum energy filter, both during transients and at the steady-state, when the accurate knowledge of inertia matrix of the target object and the absolute attitude estimates of the chaser spacecraft are not available.

5.3 Remarks

In this chapter, the results of a simulation campaign, performed evaluating the performance of each filter under different sources of uncertainty, are presented. Even if the results change, depending on the simulation case, a general trend can be observed. In fact, it turns out that for this application, although MEKF is the quickest to converge, it loses its edge at the steady-state and has inferior performance to the minimum energy filters or even to an observer at $SO(3)$. The observer on $SO(3)$ has an additional advantage of posing much less computational burden. There is also an added benefit that an observer is not prone to the numerical instabilities which may arise from the numerical integration of Riccati equation. It is to be noted that although the second-order minimum energy filters offer the best performance, the minimum energy filter provides improved performance as compared to MEKF and is significantly simpler in modeling and implementation than the second-order minimum energy filter. For a more demanding application, the second-order minimum energy filter without dynamics has been proposed in this thesis and seems to be the best option, since it requires neither an estimate of the absolute attitude nor any knowledge of the inertia matrix. Although the second-order minimum energy filter is the best performer at the steady-state, it may be outperformed by the second-order minimum energy filter without dynamics, when accurate estimates of the target's inertia matrix and absolute attitude of the chaser spacecraft are not available. This leads to the conclusion that for high image acquisition rate a kinematic filter outperforms a reduced-dynamics filter. This is generally not true, instead, for sparse measurements and known dynamics. It should also be pointed out that during the extensive sensitivity analysis of the filter performance with varying update rates, the performance of the minimum energy filter was found to be improving with the increase in update rate of the filter.

CHAPTER 6

Vision-based Relative Navigation for Non-Keplerian Orbits

> [...] to myself I seem to have been only like a boy playing on the seashore, and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary, whilst the great ocean of truth lay all undiscovered before me.
>
> Sir Isaac Newton

This chapter deals with the estimation of relative pose of a passively cooperative space object, in a non-Keplerian environment. Earth-Moon space is very attractive for future space missions as possible outpost to enable human presence beyond Earth [96]. In particular, the idea is to place modular large space infrastructures in non-Keplerian orbits, precisely in the Moon proximity. This ambitious concept will require accurate autonomous guidance, navigation and control algorithms to enable safe close proximity operations and docking. This work wants to investigate the possibility to navigate in such environment, in the vicinity of a passively cooperative target, exploiting precise relative dynamical models. Relative navigation approaches in cislunar space have been proposed in the past [97], [98]. However, these studies focus their attention on the relative navigation between two actively cooperative spacecraft, using crosslink range measurements, without any Earth-based tracking. Moreover, they consider the navigation of a two-spacecraft constellation with one spacecraft in a halo orbit and the other in lunar orbit. In this work, a vision-based *close approach* with a passively cooperative space object is considered, but the proposed navigation algorithm can be easily extended to different navigation sensors. To the author's knowledge, no previous works exists considering this kind of application scenario in a Earth-Moon environment. This chapter presents the detailed formulation of the proposed navigation algorithm and its numerical validation considering a representative scenario. Statistical analysis are performed to verify the robustness of the algorithm.

6.1 Mathematical Formulation

6.1.1 Assumptions and Notation

The inspector spacecraft is defined as *chaser* and the other spacecraft is the *target*. The only available data are provided by two cameras placed on the chaser and by markers located on the target. In principle, a known pattern of markers can provide closed-form solution of the P-n-P problem. However, the architecture with a stereo camera, allows us to easily adapt the algorithm and extend it to approaches around unknown objects. The chaser is assumed to collect and track N known feature points on the target. The adopted reference frames for both target and chaser are introduced in Section 3.1.2.

6.1.2 Filter Architecture

The proposed architecture for relative navigation of a chaser satellite with respect to a passively cooperative target is summarized in the block diagram in Figure 6.1.

This architecture is tightly-coupled. In fact, the measurements are directly processed by the navigation filter. The filter processes the features extracted by the two cameras to compute the relative target/chaser position and attitude. In the observation model of the camera, the knowledge of feature points on the target is assumed. Since the observation model depends on both position and attitude of the target spacecraft, the navigation filter

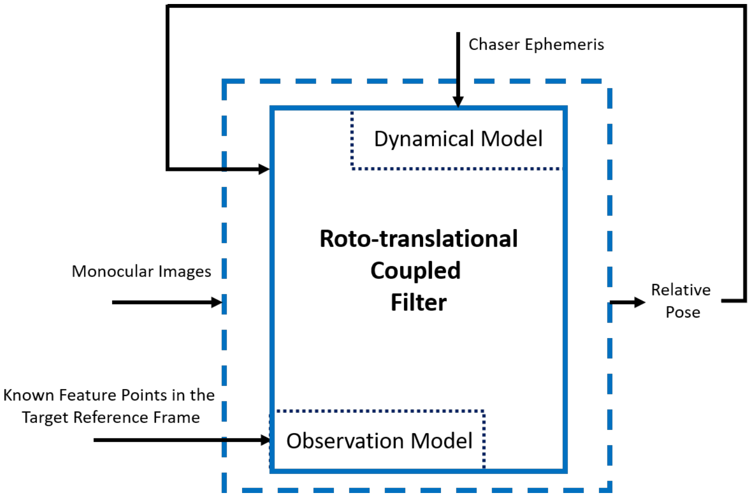


Figure 6.1: Relative Navigation Architecture: Block Diagram

has to be coupled and non-linear. The state vector of the filter is defined as:

$$\mathbf{x} = [\boldsymbol{\rho}_0^T,\ \dot{\boldsymbol{\rho}}_0^T,\ \mathbf{q}^T,\ \boldsymbol{\omega}^T]^T \tag{6.1}$$

being $\boldsymbol{\rho}_0$ and $\dot{\boldsymbol{\rho}}_0$ the relative position and translational velocity between the spacecraft centers of mass, $\mathbf{q}$ the relative quaternion and $\boldsymbol{\omega}$ the relative angular velocity.

6.1.3 Dynamical Model

The state dynamics inside the filter is assumed to evolve according to the models presented in Section 3.3 for the translational motion and Section 3.4 for the rotational dynamics. In particular, for the relative translational dynamics, the linearized model, as in Equation (3.14) but without considering the perturbations, is adopted. This is done to strongly limit the computational burden of the navigation filter. In fact, adopting this formulation, an analytical expression of the Jacobian is available. Also for the rotational dynamics, the equations presented in Section 3.4 are used assuming a perturbation-free motion.

6.1.4 Observation Model

A simple pinhole camera model [99] has been used as camera model. It assumes that all the light rays travel through an infinitely small hole and are projected onto an image frame. No lenses are used to focus the light and, therefore, distortion is not considered. Using prospective geometry rules, it is possible to mathematically describe the relationship between the 3D coordinates of a generic point and the 2D coordinates of its projection onto the image plane of the camera.

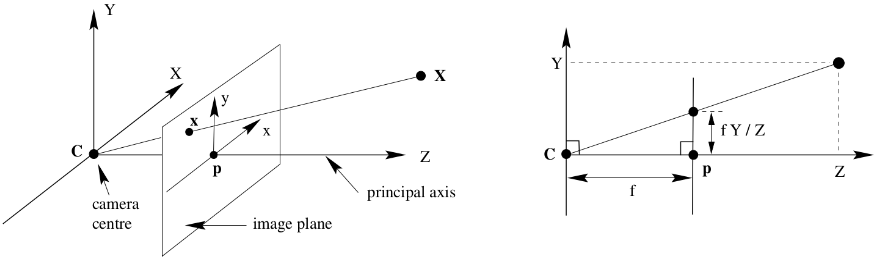


Figure 6.2: Pinhole Camera Model

Looking at Figure 6.2, a derivation of how the coordinates of the point **x** = [u,v] in the image plane, depend on the coordinates of **X** = [X,Y,Z] is provided. Exploiting a simple rule of similar triangles, the following expressions are obtained:

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

Assuming the chaser body-fixed reference frame $\mathcal{C}_{Cl}$ aligned with the left camera center of projection, and to have a stereo rig composed by two cameras with focal length f and separated by a given baseline b, the following expressions are derived:

$$u_L(i) = f\frac{x_i}{y_i} \tag{6.2}$$

$$v_L(i) = f\frac{z_i}{y_i} \tag{6.3}$$

where $u_L(i)$ and $v_L(i)$ are the coordinates in the image plane of the left camera, with $\rho_i = [x_i, y_i, z_i]$ being a generic feature point in the chaser frame $\mathcal{C}_{Cl}$. Similarly, for the right camera:

$$u_R(i) = f\frac{x_i + b}{y_i} \tag{6.4}$$

$$v_R(i) = f\frac{z_i}{y_i} \tag{6.5}$$

Finally, we can also define $\mathbf{l}_L = [u_L,\, v_L]$ and $\mathbf{l}_R = [u_R,\, v_R]$. Given this camera model, at each time step, the discrete measurement vector provided by the cameras is:

$$\mathbf{z}_i = [\mathbf{l}_{Li},\ \mathbf{l}_{Ri}] \tag{6.6}$$

Please, keep in mind that $\mathbf{z}_i$ is function of the state being $\rho_i = \rho_0 + \mathbf{R}_{\mathcal{CT}}(\boldsymbol{q}_i)\boldsymbol{P}_i$.

6.1.5 Estimation Procedure

The assumed observation model is nonlinear. For this reason, it is necessary to adopt a nonlinear filtering technique. The more common techniques, i.e. EKF and UKF were taken into account. One of the main drivers for navigation filters is their computational cost. As already briefly discussed before, adopting a linear dynamical model to describe the translational motion, allow us to have an analytical expression for the translational part of Jacobian. This is extremely important and computational cost-effective in real-time implementations. In fact, using an EKF, only one run of the dynamical model is necessary at each time step, strongly limiting the computational burden. For this reason, we adopted the linearized dynamical model within an EKF instead of an UKF. In fact, for the latter filter, the state dynamics have to be propagated at each time step for $2n$ sigma points, where n is the number of states ($n = 13$ in this case).

Knowing the process (Equations (3.14), (3.17) and (3.18)) and measurement (6.6) equations and using the EKF equations presented in Section 2.2.1, the relative pose can be estimated.

6.2 Simulation Environment and Results

Also for this application, the simulation environment to numerically validate the proposed relative navigation algorithm is presented. In particular, the measurement generation process is described along with the simulation scenarios. Finally, a sensitivity analysis campaign is presented.

6.2.1 Numerical Simulation Environment Description

The proposed relative navigation architecture is preliminarily validated through numerical simulations, carried out in MATLAB environment.

The relative translational dynamics between the two spacecraft is reproduced by integrating the full nonlinear equations presented in Section 3.3, including perturbations effect. For the relative rotational dynamics, equations introduced in Section 3.4 are used considering, also in this case, the perturbations acting on the spacecraft. Using the complete models, a relative trajectory is generated and used as reference to evaluate the relative navigation algorithm performance.

6.2.1.1 Measurement Generation

In order to evaluate the performance of the proposed navigation filter, simulated sets of 2D point features are generated. Specifically, N 3D feature points are randomly generated on the target according to a uniform distribution along each of the three axes, considering the target dimensions. The evolution of these points in the chaser reference frame $\mathcal{C}_{Cl}$ is then computed according to the true relative position and attitude. Hence, the 3D position of each detected feature point is projected on the image plane of the right and left camera. The obtained 2D coordinates are modified to account for potential errors introduced by the image processing. Also in this case, a Gaussian white noise is added to the pixel coordinates of each point feature (whose standard deviation (σ_{pix}) is expressed in terms of a certain number of pixel). In this work we assumed to have a camera similar to the one of the Prisma mission [100]. In particular, the focal length is assumed equal to $f = 30mm$ and a camera resolution of 2048 x 2048.

6.2.2 Simulation Scenario

In this section, the scenarios considered for the algorithm numerical validation are presented. The relative orbital trajectory and the feature points generation are discussed.

6.2.2.1 Feature Points

A set of M feature points on the target spacecraft are assumed to be known. These points can be representative of LEDs or visual markers placed on the target spacecraft. This set of points is extracted uniformly at each simulation to test the filter robustness for random configurations of feature points. For the numerical validation, we assumed to extract uniformly the feature points considering the following boundaries $b_x = [-1.2; 1.2]$,

$b_y = [-2; 2]$, $b_z = [-1.2; 1.2]$, having a symmetric target and the target body-fixed reference frame centered in the geometric center of the satellite.

6.2.2.2 Orbital Scenario

The filter is tested considering relative orbits in a cislunar space. In particular, the target is assumed to orbit in a L1 NRHO while the relative initial conditions are selected in order to have a quasi-bounded motion of the chaser with respect to the target. The corresponding relative translational state initial conditions are:

$$\begin{aligned}\rho_0 =&[24.57; -26.04; -34.90]m\\ \dot{\rho}_0 =&[-7.6 \cdot 10^{-6}; -2.8 \cdot 10^{-4}; 4.3 \cdot 10^{-4}]m/s\\ \mathbf{q}_0 =&[0; 0; 0; 1]\\ \omega_0 =&[1.2; -1.3; 2]deg/s\end{aligned} \tag{6.7}$$

where ρ_0 and $\dot{\rho}_0$ are the initial relative position and translational velocity expressed in the inertial frame $\mathcal{I}_{Cl}$, $\mathbf{q}_0$ is the initial relative quaternion and ω_0 is the initial relative angular velocity expressed in the chaser reference frame $\mathcal{C}_{Cl}$.

6.2.3 Results

In this section, the performance of the algorithm are evaluated. The definitions of the estimation errors are equal to the ones used for the Keplerian case but they are recalled for sake of clarity. The relative position error is defined as:

$$e_\rho = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2} \tag{6.8}$$

where $\hat{x}, \hat{y}, \hat{z}$ are the position components estimates.

And the relative attitude error is computed as:

$$e_\theta = 2\cos^{-1}(q_{e_0}) \tag{6.9}$$

where in our notation, q_{e_0} is the scalar part of the error quaternion $\mathbf{q}_e = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1}$.

Nominal Scenario For the nominal scenario, we run a statistical analysis of 50 runs. The initial conditions are extracted from a Gaussian distribution

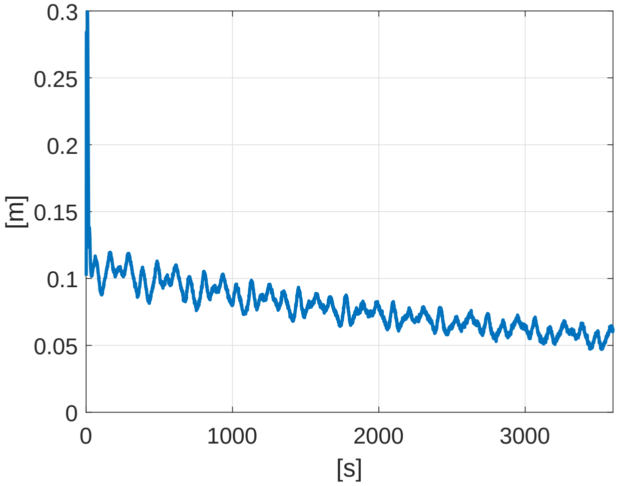


Figure 6.3: Average Relative Position Error - Non-Keplerian Nominal Scenario

with mean equal to the true state and covariance given by the initial state covariance matrix **P**. The initial state covariance matrix **P** is selected as:

$$\mathbf{P} = \begin{bmatrix} {\sigma_\rho}^2 & 0 & 0 & 0 \\ 0 & {\sigma_{\dot{\rho}}}^2 & 0 & 0 \\ 0 & 0 & {\sigma_q}^2 & 0 \\ 0 & 0 & 0 & {\sigma_\omega}^2 \end{bmatrix} \tag{6.10}$$

with:

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

The filter is run at 1Hz with a noise associated to the feature extraction of $\sigma_{pix} = 1$. In this first simulation, a set of 25 feature points is considered. The simulation time is set to the first 1 hour, enough to evaluate the steady state performance of the algorithm. Figures 6.3 and 6.4 show mean relative position and attitude errors, averaged for each time step over the 50 runs.

Figures 6.3 and 6.4 show a fast convergence of the filter and acceptable errors if considering close approach or monitoring scenarios. In particular, the error in the estimation of the relative position is lower than $0.1m$ and the relative attitude error is always lower than $0.2°$ at steady-state.

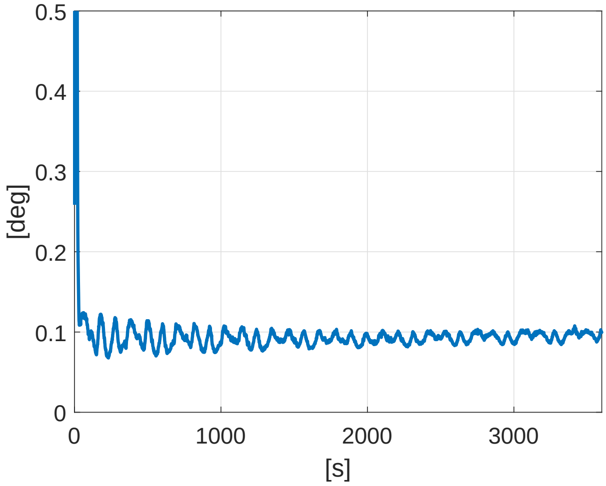


Figure 6.4: Average Relative Attitude Error - Non-Keplerian Nominal Scenario

Noise Sensitivity Analysis Also for this scenario, the robustness of the proposed algorithm over the noise value is analyzed. A sensitivity analysis over σ_{pix} is performed. The nominal scenario is propagated for 3600 seconds with a noise standard deviation ranging from 0.5 to 2.5 pixels. The initial conditions and the tuning of the filter are kept constant for all the different test cases. The results of such analysis are shown in Figures 6.5 and 6.6.

As expected, both relative position and attitude errors increase with increasing noise level. Also in the most pessimistic considered case, the filter is able to correctly estimate the relative spacecraft pose.

Number of Feature Points Sensitivity Analysis Another important aspect to take into account is the number of considered feature points. In this sensitivity analysis we run the nominal scenario with fixed initial conditions and tuning, varying the number of known feature points. In particular, we considered a number of feature points M swinging from 10 to 30. The noise level is kept at $\sigma_{pix} = 1$. Analogously, the trends of the different estimation errors are reported in Figures 6.7 and 6.8.

These plots show how, increasing the number of feature points, the estimate of both relative position and attitude improves. However, for the presented simulation, considering more than 25 feature points, the benefit of adding more features is reduced and, therefore, it may only represent an additional, ineffective computational cost.

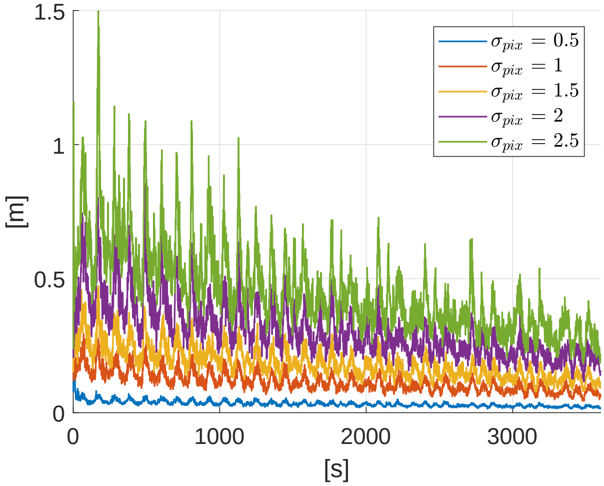


Figure 6.5: Average Relative Position Error - Noise Sensitivity Analysis

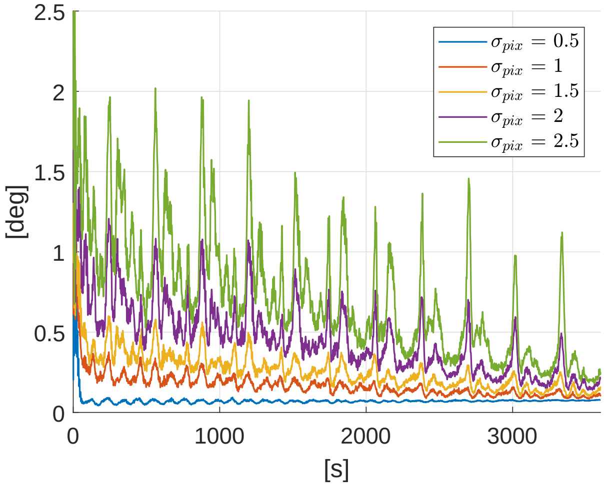


Figure 6.6: Average Relative Attitude Error - Noise Sensitivity Analysis

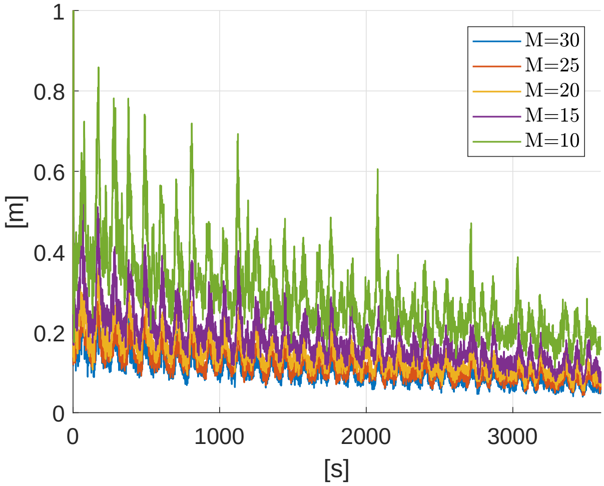


Figure 6.7: Average Relative Position Error - Feature Points Sensitivity Analysis

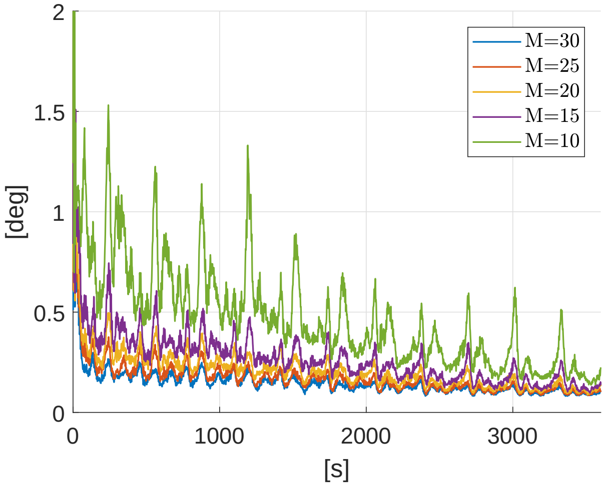


Figure 6.8: Average Relative Attitude Error - Feature Points Sensitivity Analysis

CHAPTER 7

Radial Basis Function Neural Network - Adaptive Extended Kalman Filter

> Don't waste any neurons on what doesn't work. Devote those neurons to new ideas that better explain the data.
>
> *Wonder and Skepticism (1995)*
> Carl Sagan

Current navigation algorithms rely on the accurate knowledge of the system dynamics. In particular, we already discussed the importance of proper dynamical modeling in Section 1.1.1. Adopting a precise dynamical model is not always possible; this is desirable whenever spacecraft orbiting the Earth are considered, where the environment can be accurately modeled to a great extent. Nevertheless, when dealing with relative approach with unknown bodies or interplanetary missions, the modeling of the system dynamics yields inevitable unmodeled uncertainties. This is mainly due to partial knowledge of the operative scenario, e.g. orbital disturbances acting on the target spacecraft. Furthermore, the growing interest towards micro-platforms, both for Earth and interplanetary missions, has significantly reduced the spacecraft available computational power; hence, very sophisticated models cannot be anymore handled on-board. Such limitation leads to a degradation of performance of the Guidance Navigation & Control (GNC) subsystem [4]. In this framework, on one hand, the dynamical model employed in the on-board algorithms needs to be simplified, on the other hand, the accuracy of such model significantly deteriorates the GNC performance [4], due to the absence of nonlinear terms as well as disturbances. The Artificial Neural Networks (ANNs) are a powerful tool to bridge this gap. ANNs are becoming increasingly important when dealing with uncertain processes. In particular, their approximation capability of unknown functions can be employed to reconstruct system nonlinearities as well as unmodeled environmental disturbances. The advantage of estimating such uncertainties benefits the whole GNC process chain. In this framework, Gurfil et al. [101] presented a nonlinear adaptive neural control method applicable to deep space formation flying. Bae and Kim [102] developed a neural network aided sliding mode control scheme for spacecraft formation flying. Recently Zhou [103] proposed a neural-network based reconfiguration control for spacecraft formation in obstacle environments. Finally, in recent past, some strategies on pose estimation using ANN has been proposed by Sharma [104]. Traditionally, the ANNs are solely employed for disturbance estimation, yet the aim of the navigation filter is to estimate the system state. In past years, there have been attempts to couple ANNs with EKF. In particular, the most common approach is to employ EKFs to train the ANNs [105]. In this configuration, the state of the KF is augmented with the ANN weights. For a large network this can be a problem, being the computational effort burdensome. Furthermore, the resulting coupled structure cannot provide an estimate of the uncertainties, unless the disturbance vector is added to the state vector and estimated as a constant parameter. An alternative solution is to use the estimated disturbance term, output of the ANN, directly in the dynamical propagation of the filter [106]. In this way, instead of the state vector, the dynamics of the EKF is augmented by an ANN that captures the unmodeled dynamics. The ANN learns online the function describing the disturbance, i.e. the mismatch between the measurement and the a-priori guess given by the model selected for the EKF. However, in this case, the *augmented* dynamical model accuracy changes in time and therefore, its covariance matrix has to be adapted at each step to capture this variation. In the past years, few solutions have been proposed to derive an efficient formulation for neural network aided filters. Gao et al. [107] derived a RBFNN - KF to improve the estimation accuracy for seam tracking during high-power fiber laser welding. They proposed a coupled formulation where the RBFNN is used to compensate for the model and

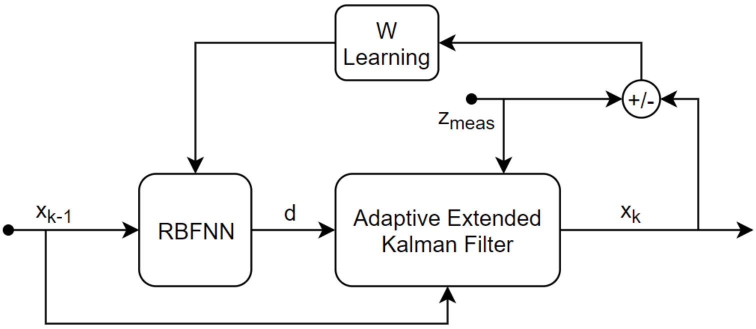


Figure 7.1: Proposed Architecture for the RBFNN-AEKF

noise uncertainties. However, they do not consider any online adaptation of the filter covariances. Similarly, Stubberud et al. [106] developed a neuro-observer based on an EKF and a multilayer feed-forward neural network. Their formulation involves two coupled KFs, one to estimate the state and the other to tune the neural network. Other authors also proposed neural network for system identification based on offline training [108], [109]. Recently Harl et al. [110] developed a reduced-order modified state observer for uncertainties estimation in nonlinear systems. They also applied the proposed technique to estimate the uncertain disturbances caused by J_2 perturbation around the Earth. Also in this case, the gain of the observer is user-selected and there is not any kind of adaptation depending on the experienced scenario. In this chapter, we propose and describe a Radial Basis Function Neural Network aided Adaptive Extended Kalman Filter (RBFNN-AEKF) for state and disturbance estimation. RBFNN are selected for their simple structure and suitability for fast online training [111]. The neural network estimates the unmodeled terms which are fed to the EKF as an additional term to the state and covariance prediction step. Finally, a recursive form of the adaptive EKF is employed to limit the overall computational cost.

7.1 Algorithm Architecture

The filter architecture is sketched in Figure 7.1. The neural network estimates the disturbances acting on the system, which are then added to the prediction step of the filter. The innovation term is used to carry out the adaptivity task. Whereas, the residual term, taking into account the estimation state at step k, is fed into the online learning algorithm of the network's weights. Each block of the Radial Basis Function Neural

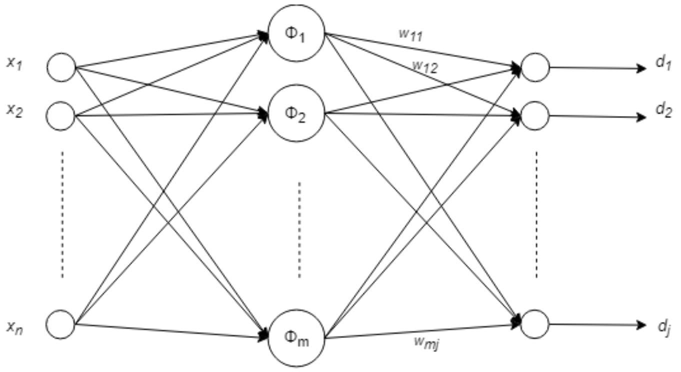


Figure 7.2: Architecture of the RBF neural network. The input, hidden, and output layers have n, m, and j neurons, respectively. $\Phi_i(\mathbf{x})$ denotes the nonlinearity at the hidden node i.

Network Adaptive Extended Kalman Filter (RBFNN-AEKF) is detailed in the following subsections.

7.1.1 RBFNN

The RBFNN is a popular network typology, which has the capability of universal approximation [111] [112]. Due to its simple structure and much quicker learning process, it stands out compared to the classic Multi-Layer Perceptron (MLP), especially for function approximation applications [111]. The RBFNN is a three-layer feedforward network, as seen in Figure 7.2. The RBFNN does not need multiple layers to obtain nonlinear behaviour, as in MLP, given that the Gaussian neurons inherently implements a nonlinear function. Hence, the lightest network has been chosen, consisting of one input layer, one hidden and one output layer. The hidden layer performs a nonlinear mapping of the input, whereas, the output layer is a linear combination of the nonlinear hidden neurons transformed into the resultant output space.

7.1.1.1 Neural Network Structure

A RBFNN is used to estimate the unmodeled disturbances, as well as the nonlinearities present in the system dynamics. The generic layout of the network is sketched in Figure 7.2. The network has a 3-layers structure, comprising an input, output and hidden layer. For the sake of derivation we call $\mathbf{x} \in \mathbb{R}^n$ the input vector. It is hereby remarked that the vector $\mathbf{x}$ is employed to derive the network structure: in the following sections the distinction between state vector and estimated state will be described and treated accordingly. Similarly to the input vector, $\mathbf{\Phi} \in \mathbb{R}^m$ is the hidden layer vector and i the associated index, $\mathbf{d} \in \mathbb{R}^j$ is the output vector and l the associated index. In this derivation we assume that $n \equiv j$. Essentially, the hidden layer evaluates a set of m radial basis functions $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$, where n is the number of states, which are chosen as centered-Gaussian expression:

$$\Phi_i(\mathbf{x}) = e^{-\eta(||\mathbf{x}-\mathbf{c}_i||)^2} \tag{7.1}$$

for $i = 1 : m$, where m is the number of neurons and $\mathbf{c}_i$ is the randomly selected center for neuron i. The output of the neural network hidden layer, namely the radial functions evaluation, is normalized:

$$\mathbf{\Phi}_{\mathbf{norm}}(\mathbf{x}) = \frac{\mathbf{\Phi}(\mathbf{x})}{\sum_{i=1}^{m} \Phi_i(\mathbf{x})} \tag{7.2}$$

The classic RBF network presents an inherent localized characteristic; whereas, the normalized RBF network exhibits good generalization properties, which decreases the curse of dimensionality that occurs with classic RBFNN [111]. In the following derivation, the output vector of the hidden layer is simply called $\mathbf{\Phi}(\mathbf{x})$ without the subscript *norm* for the sake of simplicity. For a generic input $\mathbf{x} \in \mathbb{R}^n$, the components of the output vector $\mathbf{d} \in \mathbb{R}^j$ of the network is:

$$d_l(\mathbf{x}) = \sum_{i=1}^{m} w_{il} \Phi_i(\mathbf{x}) \tag{7.3}$$

In a compact form, the output of the network can be expressed as:

$$\mathbf{d}(\mathbf{x}) = \mathbf{W}^T \mathbf{\Phi}(\mathbf{x}) \tag{7.4}$$

where $\mathbf{W} = [w_{il}]$ for $i = 1, ..., m$ and $l = 1, ..., j$ is the trained weight matrix and $\mathbf{\Phi}(\mathbf{x}) = [\Phi_1(\mathbf{x}) \ \Phi_2(\mathbf{x}) \ \cdots \ \Phi_m(\mathbf{x})]^T$ is the vector containing the output of the radial basis functions, evaluated at the current system state.

7.1.1.2 Online Learning Algorithm

The environmental perturbations, nonlinearities and unmodeled uncertainties need to be estimated online. Hence, an online learning algorithm, which drives the update of the weights, is required. The weights update law is derived to guarantee the stability of the feedback system. In the following mathematical derivation we make use of the universal approximation theorem for neural networks that guarantees the existence of a set of weights $\mathbf{W}$ that approximates a function with a bounded arbitrary approximation error [111]. Such weights are unknown, hence the algorithm is designed to obtain an estimate $\hat{\mathbf{W}}$ of the ideal weights. The neural network learning algorithm relies on the estimation error dynamics, targeting convergence and stability of the estimated weights matrix $\hat{\mathbf{W}}$ and the error $\mathbf{e}$. The symbol $\hat{(\cdot)}$ is used to refer to *estimated* quantities.

To derive the error dynamics, let us assume the actual system dynamics is described by the following set of nonlinear differential equations:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{d_{ext}} \tag{7.5}$$

where $\mathbf{d_{ext}}$ is the external disturbance term. The actual system dynamics can be rewritten as the following equation, assuming to isolate all the nonlinear terms into $d(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^j, j \equiv n$, which is the vector-valued function equivalent to the RBFNN output vector:

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} + d(\mathbf{x}) \tag{7.6}$$

where the term $d(\mathbf{x})$ captures all the nonlinearities together with the disturbances external to the system, namely $d(\mathbf{x}) = f(\mathbf{x}) - \mathbf{A}\mathbf{x} + \mathbf{d_{ext}}$. The expression of the continuous single-step KF is written as, see Section 7.1.2:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A} \cdot \hat{\mathbf{x}} + \hat{d}(\hat{\mathbf{x}}) + \mathbf{K_k}\mathbf{H}(\mathbf{x} - \hat{\mathbf{x}}) \tag{7.7}$$

where $\hat{d}$ is estimated using the radial-basis function neural network, $\mathbf{K_k}$ is the time-varying gain matrix of the KF and $\mathbf{H}$ is the observation matrix. Consider that the continuous form is employed for the sake of derivation, indeed the learning rule is then discretized for the actual implementation. The error dynamics can be derived as:

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}} \tag{7.8}$$

$$\dot{\mathbf{e}} = \dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}} = d(\mathbf{x}) - \hat{d}(\hat{\mathbf{x}}) + (\mathbf{A} - \mathbf{K_k}\mathbf{H})\mathbf{e} \tag{7.9}$$

Invoking the universal approximation theorem for neural networks [111], we can assume there exists an ideal approximation of the disturbance term $d(\mathbf{x})$:

$$d(\mathbf{x}) = \mathbf{W}^T \mathbf{\Phi}(\mathbf{x}) + \epsilon \tag{7.10}$$

where ϵ is a bounded arbitrary approximation error. Consequently, the error in estimation can be written as:

$$d(\mathbf{x}) - \hat{d}(\hat{\mathbf{x}}) = \mathbf{W}^T \mathbf{\Phi}(\mathbf{x}) + \epsilon - \hat{\mathbf{W}}^T \mathbf{\Phi}(\hat{\mathbf{x}}) \tag{7.11}$$

by adding and subtracting the term $\mathbf{W} \cdot \mathbf{\Phi}(\hat{\mathbf{x}})$ and performing few mathematical manipulations, Equation (7.11) can be expressed as:

$$\tilde{d} = \tilde{\mathbf{W}}^T \Phi(\hat{\mathbf{x}}) + \epsilon' \tag{7.12}$$

where $\tilde{d} = d - \hat{d}$, $\tilde{\mathbf{W}} = \mathbf{W} - \hat{\mathbf{W}}$ and the bounded term $\epsilon' = \epsilon + \mathbf{W} \cdot [\boldsymbol{\Phi}(\mathbf{x}) - \boldsymbol{\Phi}(\hat{\mathbf{x}})]$. The aim of the learning rule is to drive the dynamics error to zero, as well as forcing the weights to converge to the ideal ones. Namely:

$$\mathbf{e} \to \mathbf{0}, \;\; \tilde{\mathbf{W}} \to [\mathbf{0}]$$

Similarly to [110], introducing the following scalar Lyapunov function for the feedback system, including the network weights and the estimation error, the weights update rule $\dot{\hat{\mathbf{W}}}$ is derived to guarantee the stability of the system:

$$V = \frac{1}{2} tr(\xi \tilde{\mathbf{W}}^T \tilde{\mathbf{W}}) + \frac{\eta}{2} \mathbf{e}^T \mathbf{e} \tag{7.13}$$

where $tr(\cdot)$ is the trace operator, ξ, $\eta > 0$ are user-defined coefficients. Recalling Equations (7.9) and (7.12), the derivative of the Lyapunov function can be written as:

$$\begin{aligned}
\dot{V} &= tr(\xi \tilde{\mathbf{W}}^T \dot{\tilde{\mathbf{W}}}) + \eta \mathbf{e}^T \dot{\mathbf{e}} \\
&= tr(\xi \tilde{\mathbf{W}}^T \dot{\tilde{\mathbf{W}}}) + \eta \mathbf{e}^T (\tilde{\mathbf{W}}^T \boldsymbol{\Phi}(\hat{\mathbf{x}}) + \epsilon' + (\mathbf{A} - \mathbf{K_k H})\mathbf{e}) \\
&= tr(\xi \tilde{\mathbf{W}}^T \dot{\tilde{\mathbf{W}}}) + \eta \mathbf{e}^T \tilde{\mathbf{W}}^T \boldsymbol{\Phi}(\hat{\mathbf{x}}) + \eta \mathbf{e}^T \epsilon' + \eta \mathbf{e}^T (\mathbf{A} - \mathbf{K_k H})\mathbf{e} \\
&= tr(\xi \tilde{\mathbf{W}}^T \dot{\tilde{\mathbf{W}}} + \eta \tilde{\mathbf{W}}^T \boldsymbol{\Phi}(\hat{\mathbf{x}}) \mathbf{e}^T) + \eta \mathbf{e}^T \epsilon' + \eta \mathbf{e}^T (\mathbf{A} - \mathbf{K_k H})\mathbf{e} \\
&= tr(\tilde{\mathbf{W}}^T (\xi \dot{\tilde{\mathbf{W}}} + \eta \boldsymbol{\Phi}(\hat{\mathbf{x}}) \mathbf{e}^T)) + \eta \mathbf{e}^T \epsilon' + \eta \mathbf{e}^T (\mathbf{A} - \mathbf{K_k H})\mathbf{e} < 0
\end{aligned} \tag{7.14}$$

Recalling that $\dot{\tilde{\mathbf{W}}} = -\dot{\hat{\mathbf{W}}}$, the expression for the weights update rule that guarantees stability and convergence of the feedback system to ideal weights and state is:

$$\dot{\hat{\mathbf{W}}} = \frac{\eta}{\xi} \boldsymbol{\Phi}(\hat{\mathbf{x}}) \mathbf{e}^T \tag{7.15}$$

Indeed, by inserting Equation (7.15) into Equation (7.14), the expression for the derivative of the Lyapunov function reduces to the stability of the error estimation of the EKF. In the case of linear systems, the term $(\mathbf{A} - \mathbf{K_k H})$ grants asymptotic stability of the KF if $\mathbf{A}$ is reachable and $\mathbf{H}$ is observable. In the case of nonlinear systems, this is not always true. However, it has been proved [113] that the estimation error of an EKF is exponentially bounded if:

- $\mathbf{A}$ is nonsingular for every $k \geq 0$;
- there exist real constants $p_1, p_2 > 0$ such that $p_1 \cdot \mathbf{I} \leq \mathbf{P_k} \leq p_2 \cdot \mathbf{I}$, where $\mathbf{P_k}$ is the state covariance matrix;
- the initial estimation error satisfies $||\hat{\mathbf{x}}_\mathbf{0} - \mathbf{x_0}|| \leq \epsilon$ and the covariance matrices are bounded

where $\hat{\mathbf{x}}_\mathbf{0}$ and $\mathbf{x_0}$ are the estimated and true state vector at the initial step.

7.1.2 Adaptive Extended Kalman Filter

The EKF is one of the most common approach for nonlinear state estimation and the standard approach for relative navigation filters [25]. Let's consider the system and measurement models:

$$\mathbf{x_k} = f(\mathbf{x_{k-1}}, \mathbf{u_{k-1}}) + \mathbf{w_{k-1}} \tag{7.16}$$

$$\mathbf{z_k} = h(\mathbf{x_k}) + \mathbf{v_k} \tag{7.17}$$

with $\mathbf{x}$ being the state vector, $\mathbf{u}$ the control input, $\mathbf{w}$ and $\mathbf{v}$ measurement and process noise, described by zero-mean white noise uncorrelated distributions with covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ respectively. The standard EKF formulation is given in Section 2.2.1. In the proposed filter, however, the dynamical model includes the nonlinear contribution of the RBFNN $d(\mathbf{x})$. This term directly affects the state estimation and, moreover, it implies an adaptive online tuning of the EKF. In fact, the accuracy of the known dynamical model plus the estimated disturbance term is very difficult to be established a-priori and, furthermore, it is likely to be time-varying. Although this aspect is often neglected [107], an online tuning of the process covariance matrix $\mathbf{Q}$ is fundamental to ensure filter accuracy and robustness. The formulation of the RBFNN-AEKF is given by:

$$\hat{\mathbf{x}}_\mathbf{k}^- = f(\hat{\mathbf{x}}_\mathbf{k-1}^+, \mathbf{u_{k-1}}) + d(\hat{\mathbf{x}}_\mathbf{k-1}^-, \mathbf{u_{k-1}}) \tag{7.18}$$

$$\mathbf{P}_\mathbf{k}^- = \tilde{\mathbf{F}}_\mathbf{k-1}\mathbf{P}_\mathbf{k-1}^+\tilde{\mathbf{F}}_\mathbf{k-1}^\mathbf{T} + \mathbf{Q_{k-1}} \tag{7.19}$$

$$\mathbf{K_k} = \mathbf{P}_\mathbf{k}^-\mathbf{H}_\mathbf{k}^\mathbf{T}(\mathbf{H_k}\mathbf{P}_\mathbf{k}^-\mathbf{H}_\mathbf{k}^\mathbf{T} + \mathbf{R_k})^{-1} \tag{7.20}$$

$$\mathbf{P}_\mathbf{k}^+ = (\mathbf{I} - \mathbf{K_k}\mathbf{H_k})\mathbf{P}_\mathbf{k}^- \tag{7.21}$$

$$\hat{\mathbf{x}}_\mathbf{k}^+ = \hat{\mathbf{x}}_\mathbf{k}^- + \mathbf{K_k}(\mathbf{z_k} - h(\hat{\mathbf{x}}_\mathbf{k}^-)) \tag{7.22}$$

with

$$\tilde{\mathbf{F}} = \frac{\partial f}{\partial \mathbf{x}} + \frac{\partial d}{\partial \mathbf{x}}; \qquad \mathbf{H} = \frac{\partial h}{\partial \mathbf{x}} \tag{7.23}$$

and

$$\mathbf{Q_k} = \alpha\mathbf{Q_{k-1}} + (1-\alpha)(\mathbf{K_k}\delta_\mathbf{k}\delta_\mathbf{k}^\mathbf{T}\mathbf{K}_\mathbf{k}^\mathbf{T}) \tag{7.24}$$

where α is a forgetting factor and $\delta_\mathbf{k} = \mathbf{z_k} - h(\hat{\mathbf{x}}_\mathbf{k}^-)$ is the filter innovation. The adaptation of $\mathbf{Q}$ is performed according to Equation (7.24) as in [114] to limit the computational effort. Please note that the adaptive formulation guarantees that, even in the unfortunate event when the neural network produces a completely wrong estimates of the disturbances, the filter, at least, follows the available measurements.

7.2 Application to Spacecraft Relative Navigation

In this section, we introduce one of the possible applications of the proposed RBFNN-AEKF. The relative navigation between two spacecraft orbiting the Earth is considered for a dual motivation: it is a well-known scenario, hence sophisticated model can be employed to simulate the reality to evaluate the filter performances; also, there are many available dynamical models with increasing levels of accuracy that can be used for comparison. It is worth remarking that this is not the only application nor the most appealing one, since more uncommon scenarios are expected to emphasize the benefit of the proposed filter, such as interplanetary mission or non-keplerian orbits. Hereby, the different dynamical models for filter propagation are presented as well as the filter alternatives used for comparison.

7.2.1 Tested Filters

In this subsection, we present the filters used for the comparison. Besides the new filter proposed in this chapter RBFNN-AEKF, other filters will be tested under the same simulation scenario:

- a state observer based on the formulation in [110];
- a standard, non-adaptive EKF aided with a RBFNN;
- an EKF exploiting a more accurate, nonlinear dynamical model.

7.2.1.1 Observer

The dynamics of the relative motion between the spacecraft is reconstructed using a modified full-state observer [110]. In the same fashion as Section 7.1.1, the state observer can be constructed as follows [115]:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}_{cw} \cdot \hat{\mathbf{x}} + \hat{d}(\hat{\mathbf{x}}) + \mathbf{K}_{\mathbf{h}}(\mathbf{z} - \hat{\mathbf{x}}) \tag{7.25}$$

where $\mathbf{A}_{cw}$ is the linear time invariant matrix of the Clohessy-Wiltshire dynamics presented in Section 3.2.2, $\hat{d}$ is estimated using the radial-basis function neural network in Section 7.1.1, $\mathbf{K}_{\mathbf{h}}$ is the user-defined observer gain matrix.

7.2.1.2 RBFNN-EKF

This filter is a standard EKF aided with RBFNN as presented in Section 7.1. The only difference with respect to the proposed RBFNN-AEKF is that the value of the process covariance $\mathbf{Q}$ is fixed in time. It is worth underlying that this can be a very weak point because it is hard to a-priori establish the accuracy of the RBFNN-based disturbance estimation, especially for very uncertain dynamics. The RBFNN-EKF formulation is based on Equations (7.18) to (7.22).

7.2.1.3 EKF - nonlinear

The last filter that we want to test is an EKF with a different dynamical model. There is not any coupling with the neural network but the used dynamical model is nonlinear and accounting for J_2 perturbations. In particular, a standard EKF is employed (Equations (7.18) to (7.22)) where the evolution of the state vector is described by the nonlinear model introduced in Section 3.2.1.

7.3 Scenarios & Results

In this section, the numerical simulation environment to evaluate the filter performance is described. First, the selected scenario is presented. Subsequently, the capability of disturbance reconstruction of the RBFNN-AEKF is tested. Then, the filters presented in Section 7.2.1 are compared using a realistic orbital environment. At this point, the definition of measurement noise levels and filters tuning are introduced. Finally, the same simulation is performed using off-nominal conditions to test the robustness of the navigation filters.

7.3.1 Orbital Scenario

The reference relative orbital motion is generated considering two spacecraft with the same initial orbital parameters except for the true anomaly. In particular, Table 7.1 reports the chaser and target initial orbital parameters along with the cross-sectional area, important for disturbances evaluation.

Table 7.1: Chaser-Target Orbital Parameters

	Chaser	**Target**
a [km]	8143.1	8143.1
e [-]	$1.4{\cdot}10^{-1}$	$1.4{\cdot}10^{-1}$
i [°]	98.2	98.2
ω [°]	85.9	85.9
Ω [°]	79.2	79.2
θ [°]	0	$1{\cdot}10^{-4}$
A_{sp} $[m^2]$	1.2	0.2

These orbital parameters result in the following relative initial conditions:

$$\rho_0 = [-0.0017,\ -12.2042,\ \ 4.7 \cdot 10^{-4}]\ m \tag{7.26}$$

$$\dot{\rho}_0 = [-0.0017,\ \ 3.9 \cdot 10^{-6},\ -5.6 \cdot 10^{-10}]\ m/s \tag{7.27}$$

Please note that the reference orbits are eccentric and that the cross sectional areas are different, resulting in a strongly different perturbation effect. The presented scenario has been selected as representative of a leader-follower formation, separated along the orbit by a difference in the true anomaly.

7.3.2 Disturbance Reconstruction

The RBFNN disturbance approximation capability is assessed through the simulation of the scenario presented in Section 7.3.1. In order to have a quantitative disturbance term, which can be compared to the ANN estimation, the actual relative motion is propagated using the J_2-perturbed relative motion in Section 3.2.1. Instead, the filter exploits a simple Clohessy-Wiltshire linearized model described in Section 3.2.2. The normalized ANN consists of 60 hidden neurons with Gaussian-basis radial functions; the function centers are generated randomly. The number of neurons has been selected by trading-off the reconstruction accuracy and the computational time. In such framework, the disturbances that need to be estimated are caused by the following elements:

- J_2 zonal gravity perturbation
- $e \neq 0$, elliptical orbits;

together with the nonlinearities neglected in the derivation in Section 3.2.2. Using the J_2-perturbed nonlinear model in Section 3.2.1, the disturbance term is explicit in the form of $\mathbf{d} = [d_x\ d_y\ d_z]^T$ acceleration term. The reference orbit is a LEO, which is incidentally assumed to be the orbit of the target spacecraft. Table 4.2 reports the orbital parameters of the two spacecraft. The estimation of the disturbance acceleration term converges

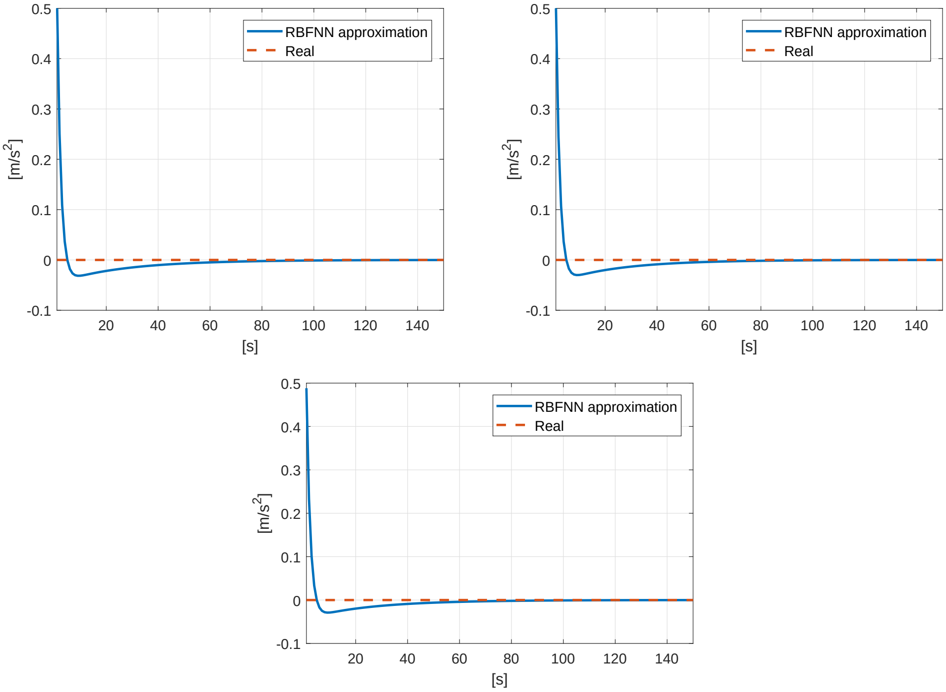


Figure 7.3: Estimation of the disturbance acceleration term for LEO reference orbit. From left to right: d_x, d_y and d_z.

after a transient time of nearly 350 s: this represents the main learning process of the randomly initialized network. Figure 7.3 shows the learning curve of the network during the early phase of the orbital motion. The disturbance acceleration components are shown in Figure 7.4. Despite the measurement noise, the estimation yields a RMSE reported in Table 7.2.

Table 7.2: RMSE of the Disturbance Estimation Term - LEO Orbit

	Value
$\sigma_x \left[\frac{m}{s^2}\right]$	$7.2 \cdot 10^{-7}$
$\sigma_y \left[\frac{m}{s^2}\right]$	$7.6 \cdot 10^{-7}$
$\sigma_z \left[\frac{m}{s^2}\right]$	$5.9 \cdot 10^{-7}$

7.3.3 Relative Navigation - Nominal Case

An accurate orbital simulator is used to test the filters in a realistic environment. In fact, the relative motion between target and chaser is obtained by integrating separately the chaser and the target orbital dynamics considering the perturbations acting on each spacecraft. In particular, the model considers irregularities in the gravitational potential

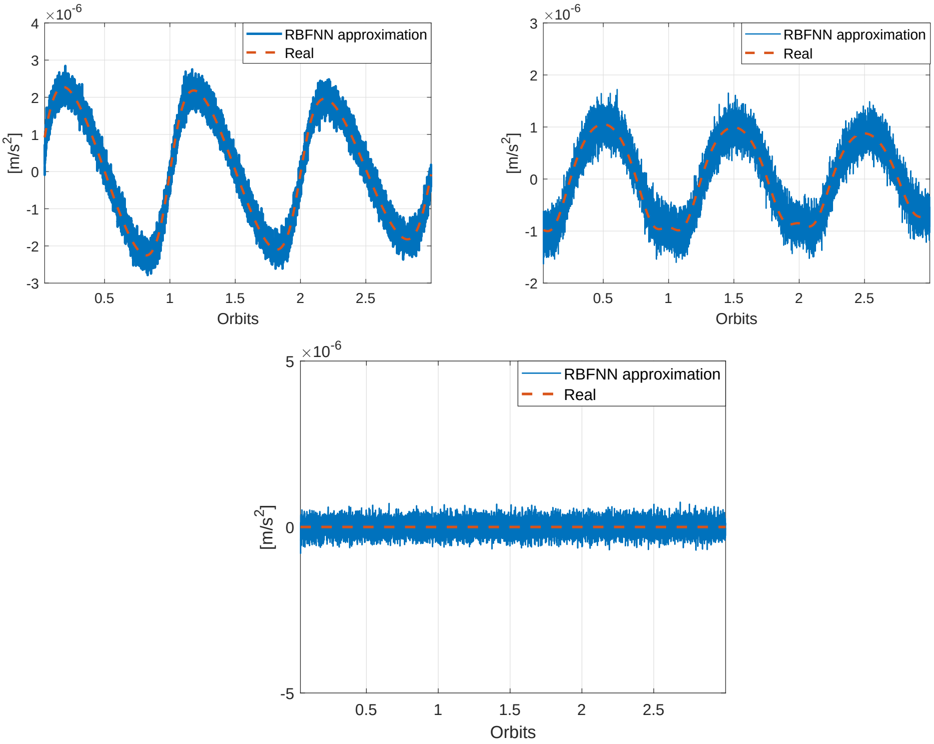


Figure 7.4: Estimation of the disturbance acceleration term for LEO reference orbit. From left to right: d_x, d_y and d_z.

due to non-spherical distribution of Earth's mass, the presence of the Moon and the Sun as third and fourth bodies, the effect of the SRP and the atmospheric drag. The adopted Earth gravitational model is the EGM96 with harmonics up to the third degree and order. On the other hand, the atmospheric drag force is computed by using the Jacchia Reference Atmosphere model. The normalized neural network consists of 60 hidden neurons with Gaussian-basis radial functions; the function centers are generated randomly. This reference orbits are also used to generate relative measurements by adding a fictitious noise, representative of realistic sensors uncertainty. In particular, the noise level associated to relative position and velocity measurement respectively, is described by a Gaussian distribution with standard deviation $\sigma_{pos} = 10^{-2} m$ and $\sigma_{vel} = 10^{-4} m/s$. It is important to remark that the orbits are eccentric and the cross sectional areas of the two spacecraft are significantly different, yielding a strong differential perturbation effect. The estimation errors, used for performance assessment, are recalled. The relative position error is defined as:

$$e_\rho = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2} \tag{7.28}$$

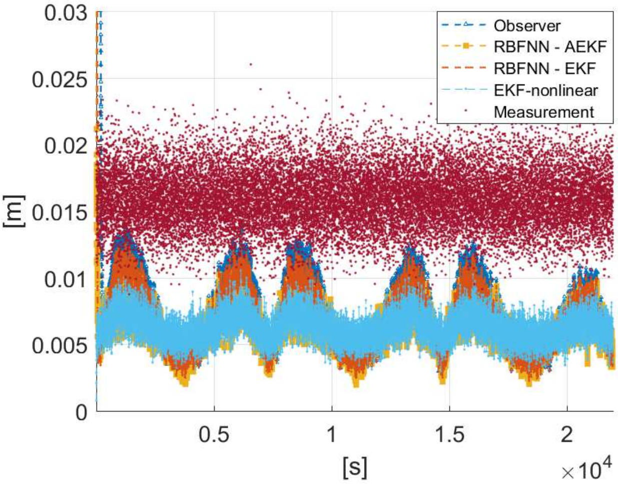


Figure 7.5: Relative Position Error

where $\hat{x}, \hat{y}, \hat{z}$ are the position components estimates. Similarly, the relative velocity error is:

$$e_{\dot{\rho}} = \sqrt{(\dot{x}_i - \hat{\dot{x}}_i)^2 + (\dot{y}_i - \hat{\dot{y}}_i)^2 + (\dot{z}_i - \hat{\dot{z}}_i)^2} \tag{7.29}$$

with $\hat{\dot{x}}, \hat{\dot{y}}, \hat{\dot{z}}$ are the velocity components estimates.

The measurement covariance matrix $\mathbf{R}$ for all the filters is tuned according to the imposed measurement noise level. The same process covariance matrix $\mathbf{Q}$ is used for the RBFNN-AEKF and RBFNN-EKF and, for the nonlinear EKF, it is properly selected to guarantee the best steady state error performance. Similarly, the observer gain $\mathbf{K_h}$ is tuned to guarantee the minimum steady state error. A statistical analysis of the filters has been performed over 100 runs for the described scenario. The filters run with a frequency of 1Hz and the simulation duration is set to three chaser orbits to appreciate the disturbances effect. Figures 7.5 and 7.6 show the relative position and velocity error averaged over 100 runs. For a more quantitative analysis of the results, the RMSE starting from time step 300 are computed and reported in Table 7.3 to evaluate the steady state performance of the filters.

Figures 7.5 and 7.6 show the beneficial effect of the filters compared to the measurements error. The RBFNN-AEKF and the EKF-nonlinear show a similar behavior for the relative position error (Figure 7.5) and, as in Table 7.3, they outperform the other alternatives. On the other hand, for what concerns the velocity estimation, the Observer, with this tuning, has better performance than the other filters. Despite these small differences,

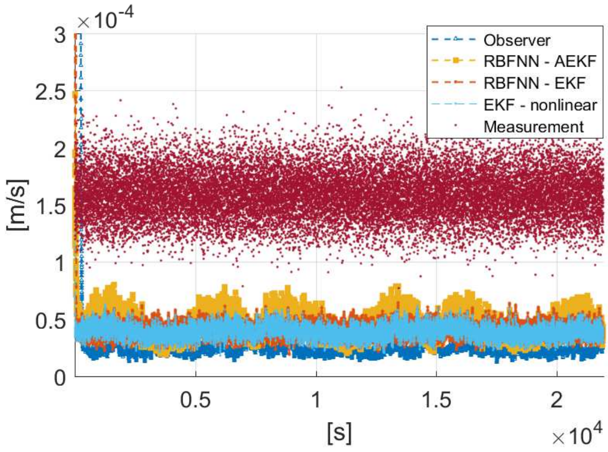


Figure 7.6: Relative Velocity Error

the compared filters show similar performance and the order of magnitude of the RMSE, reported in Table 7.3 is the same.

Table 7.3: Filters RMSE Results

	RMSE - Position [m]	**RMSE** - Velocity [m/s]
Observer	0.0079	$2.39 \cdot 10^{-5}$
RBFNN - AEKF	0.0063	$4.49 \cdot 10^{-5}$
RBFNN - EKF	0.0074	$4.03 \cdot 10^{-5}$
EKF - nonlinear	0.0064	$3.92 \cdot 10^{-5}$

7.3.4 Relative Navigation - Non-nominal Case

A proper tuning of the filter, however, is difficult to achieve when the process dynamics is not well known and time-varying. Moreover, it is very hard to a-priori determine the accuracy in the estimation that the RBFNN can achieve for that particular case. For this reason, we tested all the filters with off-nominal conditions. In particular, for each simulation, the value of $\mathbf{Q}$ and $\mathbf{K_h}$ were randomly selected according to a uniform distribution centered in the nominal value and spanning two order of magnitudes. This can be a very high uncertainty value for some applications, but we wanted to show how the tuning strongly affects the filter performance. Table 7.4 shows the relative position and velocity RMSE computed over 100 runs.

Table 7.4: Filters RMSE Results - Non-Nominal

	RMSE - Position [m]	**RMSE** - Velocity [m/s]
Observer	0.0149	$5.98 \cdot 10^{-5}$
RBFNN - AEKF	0.0064	$4.79 \cdot 10^{-5}$
RBFNN - EKF	0.0090	$9.34 \cdot 10^{-5}$
EKF - nonlinear	0.0110	$9.55 \cdot 10^{-5}$

It is possible to appreciate how the estimation error of the RBFNN-AEKF is very similar to the nominal case. This is an evidence of high robustness of the proposed solution. On the contrary, all the other filters are badly affected from the wrong selection of $\mathbf{Q}$ or $\mathbf{K_h}$ respectively.

CHAPTER *8*

Autonomous Navigation & Mapping of Small Bodies

> We need the tonic of wildness...At the same time that we are earnest to explore and learn all things, we require that all things be mysterious and unexplorable, that land and sea be indefinitely wild, unsurveyed and unfathomed by us because unfathomable. We can never have enough of nature.
>
> *Walden: Or, Life in the Woods (1854)*
> HENRY DAVID THOREAU

THIS chapter deals with the approach to an uncooperative unknown space object. We discuss the challenges in adopting an autonomous approach for missions designed to operate in proximity and map the surface of unknown objects (e.g., asteroids or comets). Nowadays, dealing with such challenging environment with high levels of uncertainty requires human intervention across different phases of the mission. In particular, the current state-of-the-art techniques heavily rely on human intervention in the first part of the proximity operations phase (Figures 8.1 and 8.3). Moreover, for an

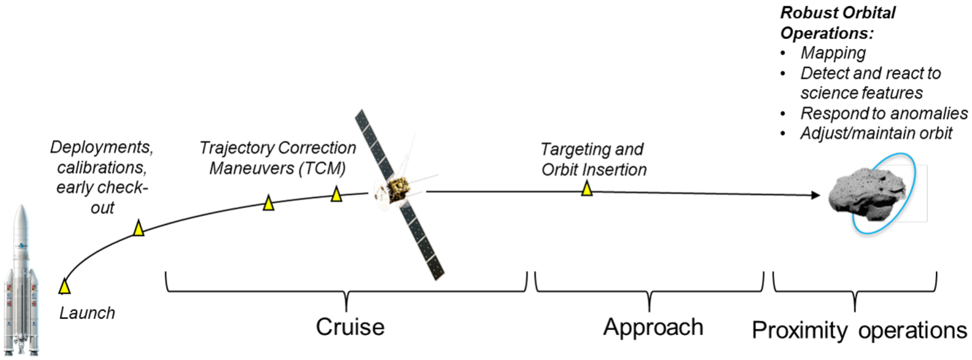


Figure 8.1: Classical Small-Body Mission Architecture

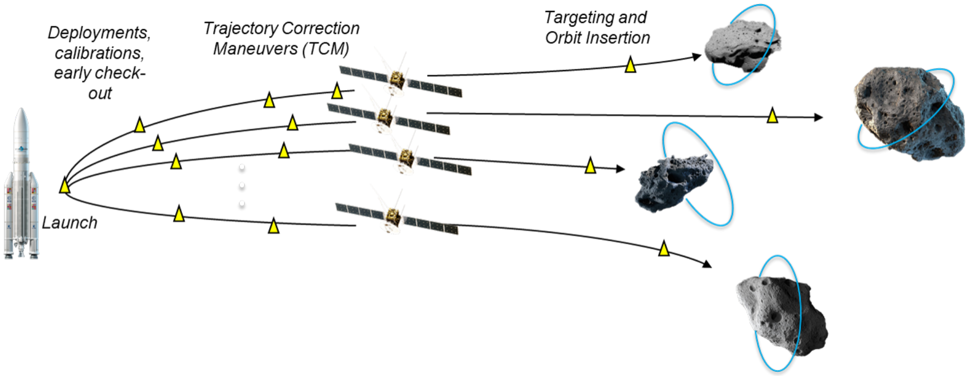


Figure 8.2: Futuristic Small-Body Mission Architecture

adequate mapping and acquisition of scientific data, proper maneuvers have to be designed. Currently, trained analysts play a crucial role in this process by designing trajectories and assisting the mapping of small bodies. This human-in-the-loop process is both time consuming and limits the application of the methods to many future missions, e.g., sending multiple spacecraft to explore several small bodies simultaneously, see Figure 8.2. The aim of the work presented in this chapter is to take a step toward full spacecraft autonomy. In particular, we focus on the orbit selection process to enhance mapping accuracy and coverage of small bodies.

Following the literature in robotic autonomy, this work is most related to literature on *Active* SLAM [116]. The general problem is to *localize* the spacecraft with respect to the small body and *plan* the best trajectory that would improve the *mapping* of the small body. In this chapter we will focus on a reduced version of the Active SLAM problem. In particular, we will present a path planning algorithm that maximizes the information gain in mapping accuracy. The main goal of this work is to enable the spacecraft to autonomously take actions to improve the knowledge of the environment (i.e., small body shape).

The most common techniques for small body mapping are *Stereophotogrammetry (SPG)*[117] and *Stereophotoclinometry (SPC)* [118]. In this chapter, we focus on the SPC approach, and aim to build an autonomous orbit selection framework to enhance the SPC performance. The main idea of SPC is to build small digital topography and albedo maps (L-map) from images of the same area under different illumination conditions. Once a map is acquired, the rendered visual appearance of the map is predicted and compared to actual images. Then, the center of the map (landmark) and the spacecraft pose can be estimated by iteratively minimizing the difference between expected map appearance and actual image. Existing small body mapping techniques (e.g., [118], [117], [119]) require human supervision and they are typically decoupled from the trajectory design process. Regarding the localization methods for space applications, one the most common methods is the AutoNav approach [120], which uses a batch filter, where data are stored on-board, edited, and then input to the filter. Other example localization systems include landmark-based filtering techniques using EKF [121], higher order Additive Divided difference sigma points Filter [122], UKF [123] and PF [124]. However, the problem of joint estimation and autonomous decision making (i.e., active SLAM) is a much less explored area in the space literature. Most related work to active SLAM is focused on landing application (e.g., [125]). On the other hand, active SLAM has received a lot of attention in robotics community. Feder et al. [116] propose an algorithm to determine the optimal action to maximize the next step information gain given the current knowledge of the map. Bourgault [126] improved [116] by including a global term in the utility function that takes into account, for each possible action, the reduction of the uncertainty of the map, represented by an occupancy grid. More recent works have proposed belief space path planner with minimum uncertainty by utilizing filters and pose graph representations [127], [128], [129]. In this work, we propose a framework for simultaneous mapping and planning around small bodies. Due to the high complexity of the environment and continuous solution space, the exact problem formulation is computationally intractable. In this work, we propose an approach to reduce the problem to a computationally tractable one. In particular, we propose an algorithm that enables the spacecraft to autonomously and adaptively select the best orbit, given a reduced set of possible future orbits. We incorporate the mapping accuracy as the driver for the optimization process. We start by deriving formal quantitative measures that characterize the accuracy of the small body map, and use these measures

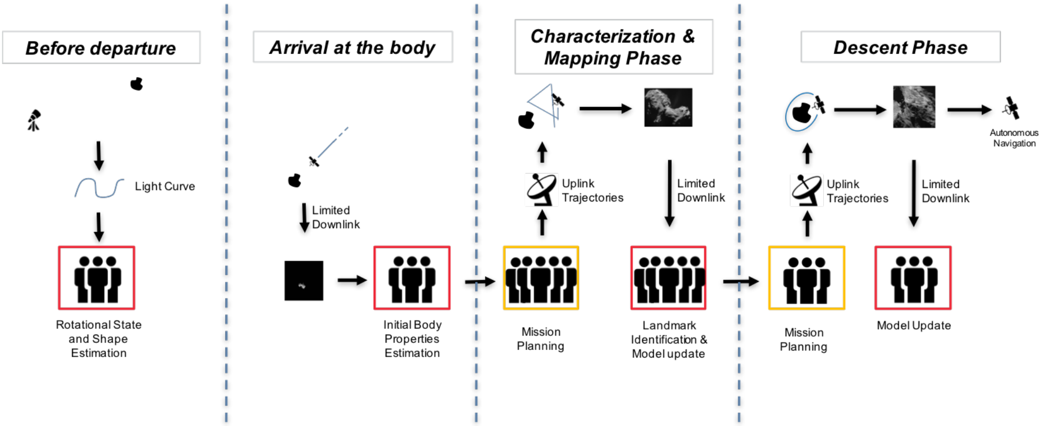


Figure 8.3: Operations Timeline

in an optimization process to compute the next best orbit that maximally contributes to the map enhancement. The highlights and contributions of the proposed method can be summarized as: *(i)* Developing formal definitions of quantitative measures that characterize the accuracy of the small body map. *(ii)* Proposing an autonomous orbit selection method to maximize the small body coverage. The proposed approach take an step toward enhancing the autonomy capabilities of the spacecraft during proximity operations with a small body. It also reduces the human effort during trajectory planning and mapping phases.

8.1 Mission Architecture

A mission to a small body can be divided into a few main operational phases based on the distance to the body (see Figure 8.4). In particular, after the deep space phase, the spacecraft *arrives at the body*. From previous and future missions (i.e., Rosetta, OSIRIS-REx), the arrival phase starts when the ratio between the distance and the object's maximum dimension (also referred to as the "interest ratio") becomes larger than 25. The exact value of this threshold depends on the mission and it is slightly different for any mission/body. After arrival, an initial *characterization phase* starts. For this stage, the interest ratio is about 14-18. The main objective of the characterization phase is to identify landmarks on the small body surface and estimate their position to determine the rotation state of the body and construct an initial shape model. For each part of the body surface, several images need to be taken to construct an accurate map.

Table 8.1: OSIRIS Camera Specifications

Field of view	12°(y) x 12°(x)
Detector size (pixels)	2048 x 2048
Focal length	140 mm

Figure 8.4 shows the operational sequence and phases for the Rosetta mission, as an example. In the Rosetta mission, in the characterization phase images were acquired every hour, and during the initial days of the characterization phase, a first set of landmarks were manually identified by the operators. The next phase is the *global mapping* phase (where interest ratio is roughly between 4 and 7). At this stage, initial estimates of the gravity potential and rotational state of the body, landmark positions and shape and topological model are being enhanced and refined. Finally a *close observation orbit* (interest ratio is roughly 2-3) is needed for the selection of possible landing sites.

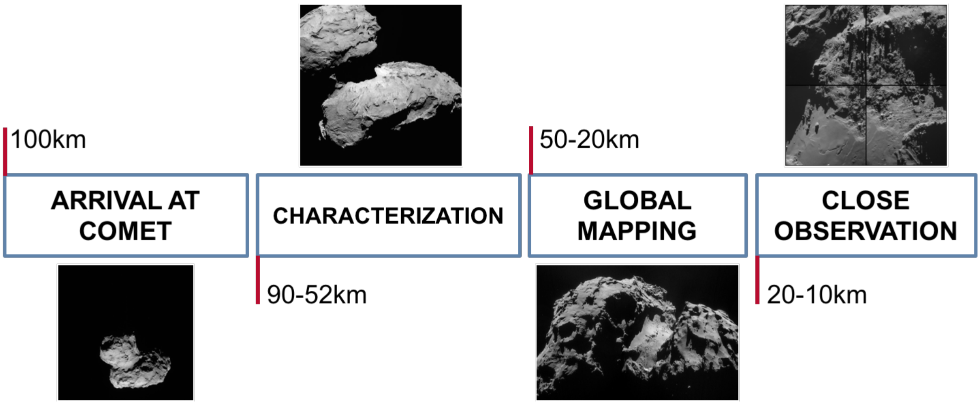


Figure 8.4: Rosetta Operational Phases

The relative distance and the dimensions of the small body can be not perfectly known during these phases. However, the projection of the object on the camera plane, measured in pixel, can be a good indicator of the corresponding phase. Figure 8.5, reports the size of the object maximum dimension on the sensor in pixel, in the different phases of the Rosetta mission. A simplified 2D analysis has been carried out. The plot is obtained considering the OSIRIS wide angle camera used in the Rosetta mission and a constant 2D object projection of 4km in its maximum dimension. The OSIRIS cam specifications are reported in Table 8.1.

Mission scenario & Assumptions Traditionally, the deep space trajectory and the arrival at the comet are designed on the ground before the launch.

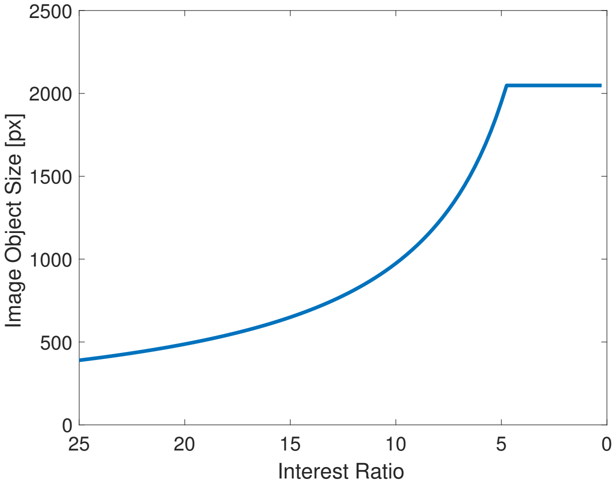


Figure 8.5: Small Body Dimension

However, for future missions, with more complex, uncertain, and far bodies, autonomy is crucial in the mission success. In particular, automating the characterization phase can significantly help with reducing the human intervention and enabling complex future missions. Thus, in this work, we will mainly focus on the characterization and global mapping phases. At this phase, there exists a very rough estimate of the body shape and rotational state. This information is resulted from the arrival phase and pre-launch calculations based on light-curve analysis (see Figure 8.3).

8.2 Planning Under Uncertainty

This work is concerned with the problem of autonomous mapping with planning around small bodies. This is an instance of the general problem of planning under motion, sensing, and map uncertainty, also known as *active* SLAM [116]. In its most general form, active SLAM can be cast as a Partially Observable Markov Decision Process (POMDP) problem [130] [131] [132].

POMDP is one of the most principled ways of modeling the process of sequential decision making under uncertainty. To describe an abstract POMDP problem, let us denote the system state, action, and observation at the k-th time-step by x_k, u_k, z_k, respectively. Let $x_{k+1} = f(x_k, u_k, w_k)$ and $z_k = h(x_k, v_k)$ denote the system dynamics and measurement models, where $w_k \sim p_w(\cdot|x_k)$ and $v_k \sim p_v(\cdot|x_k)$ denote the state-dependent process and observation noise. Due to the observation noise, the best one can infer about the system state is a probability distribution over all possible states, referred to as belief $b_k = p(x_k|z_{0:k})$. Belief space (i.e., the set of all beliefs) is denoted by $\mathbb{B}$. Belief is typically evolved using a recursive filter denoted by τ as $b_{k+1} = \tau(b_k, u_k, z_{k+1})$. The control policy π in a partially observable setting is a mapping from belief space to the action space, i.e., $u_k = \pi(b_k)$. Defining the one-step reward of taking action u at belief b by $\mathcal{R}(b, u)$, we can formally define the T-step POMDP problem as:

$$\pi^*_{0:T} = \arg\max_{\Pi_{0:T}} \mathbb{E} \sum_{k=0}^{T} \mathcal{R}(b_k, \pi_k(b_k)) \tag{8.1}$$

Π is the space of all possible policies. Often, in solving POMDPs, in order to reduce complexity, the set of admissible policies Π is reduced to a simpler subset over which the optimization can be carried out. In the next section, we will describe concrete realization of these terms and the POMDP problem associated with the small body mapping. Then, we aim to approximate and solve this problem to compute a policy for spacecraft motion.

8.3 Small Body Mapping and Spacecraft Localization

In this section, we briefly review the mapping and state estimation around small bodies and discuss how its elements correspond to the abstract POMDP problem described in Section 8.2.

8.3.1 System definition

In the small body mapping and planning problem, the overall system state x_k is represented by the spacecraft state and the map state $x_k = (s_k, m_k)$, where $s_k = (\rho_k, \dot{\rho}_k) \in \mathbb{R}^6$ being the spacecraft position and translational velocity relative to the body. m_k represents the state of the map of the body. In this work, we assume that the spacecraft is always pointing towards the center of the small body, thus, we do not include the spacecraft attitude in the state definition.

8.3.2 Map representation

In this work, we rely on the SPC technique [118] for mapping. SPC is the main method utilized in small body missions to generate the digital topography/albedo map of the body from multiple images. In SPC, the small body surface map m is represented by a set of maplets with their associated digital topology and albedo. In this work, we only work with the topology and not albedo. Formally, the map m is then defined as a mesh with n facets, i.e., $m = (f^1, f^2, \cdots, f^n)$. The i-th facet $f^i = (v^{i1}, v^{i2}, v^{i3}, \hat{n}^i)$ is a triangle characterized by three vertices v^{i1}, v^{i2}, and v^{i3}. $\hat{n}^i$ denotes the normal vector of the facet, which (is a dependent variable and) can be computed based on the three vertices.

8.3.3 System Belief

Since the spacecraft observations z_k (i.e., images) are noisy, one can only create a probability distribution function (pdf) over all possible system states. We refer to this pdf as the joint belief over the spacecraft state and the map state, denoted by $b^x = p(x|z_{0:k}, u_{0:k})$.

8.3.4 Joint State Estimation

Typically, the joint belief is evolved recursively using a filter. We represent the recursive update of the belief under filter τ as

$$b^x_{k+1} = \tau(b^x_k, u_k, z_{k+1}) \tag{8.2}$$

where, u_k is the control command sent to the spacecraft and z_k is the observation (image) received from spacecraft cameras. For this joint state estimation, we follow some common framework in space missions. In particular, we rely on AutoNav [120] for the spacecraft state estimation, and SPC [118], for the map estimation and update. The AutoNav uses a batch filter, where data are stored, edited, and then input to the filter. The details of AutoNav and SPC are beyond the scope of this thesis. The reader is encouraged to see [120] and [118] for further details.

8.4 Autonomous Planning for Small Body Mapping

Leveraging small body mapping and spacecraft localization from Section 8.3, in this section, we discuss the problem of trajectory planning around small bodies for efficient mapping. Planning in an environment that is strongly perturbed, unknown and subject to very complicated dynamics, as the one around a small body, is noticeably different from the classical deterministic planning problems. A wrong choice of an action, due to unmodeled dynamics or unknown environment, could suddenly cause a collision with the body or generate an escape trajectory. In its most general form the problem is a POMDP (Equation (8.1)), whose solution is computationally intractable. To reduce the problem to a computationally tractable one, we start by approximating the policy space Π in Equation (8.1) with a smaller set $\bar{\Pi}$, as described below.

Reduced policy space

The primitive spacecraft actions $u \in \mathbb{U}$ are thruster commands, and hence the full policy $\pi : \mathbb{B} \to \mathbb{U}$ is a mapping from the joint belief space to the thruster commands $u = \pi(b)$. However, planning in the continuous space $\mathbb{U}$ of low-level thruster commands is a challenge. Thus, in this work, we first reduced the command space to $\bar{\mathbb{U}}$, which includes a set of desirable orbits to choose from. In other words, instead of selecting a low-level thruster command, we select a target orbit and rely on an existing controller to generate thrust commands that take us to the target orbit. We refer to thruster commands as actions and refer to orbit commands as macro-actions (since each macro-action is composed of a sequence of low-level actions). In the following, we discuss how each macro-action is parametrized.

Trajectory definition A spacecraft trajectory can be defined in several different ways. Following the classical planning formulation, we define the trajectory via a time-stamped sequence of spacecraft state and actions.

$$\mathcal{T} = (s(t), u(t)) \tag{8.3}$$

Orbit parameters An orbit is a certain trajectory, which can be parametrized in different ways. We represent the orbit with the following parameters:

$$\omega = (s_0, t_\omega, t_z) \in \Omega \tag{8.4}$$

where, s_0 is the initial state (position and velocity) of the spacecraft at the beginning of the orbit. $t_\omega \in \mathbb{R}_{\geq 0}$ represents how long the spacecraft need to stay on the orbit ω. Vector $t_z \in \mathbb{R}^{\eta}_{\geq 0}$ is the sequence of image acquisition time along the orbit. Variable $\eta \in \mathbb{N}$, dictated by the mission constraints, is the upper limit on the number of images that can be taken during time t_ω. Set Ω is the space of all possible ω, which is the reduced action space, i.e., $\bar{\mathbb{U}} = \Omega$. Please note that to recover $\mathcal{T}$ from ω it is necessary to rely on a dynamical model of the motion of the spacecraft that takes into account also the gravity field of the small body.

Reduced Policy Consequently, the reduced policy is defined as:

$$\bar{\pi} : \mathbb{B} \to \Omega, \qquad \omega = \bar{\pi}(b) \tag{8.5}$$

$\bar{\Pi}$ os the set of all possible $\bar{\pi}$. To compute the best policy, we first define the reward and value functions.

One-orbit reward Similar to the one-step reward, we define one-orbit reward $\bar{\mathcal{R}}(b_k, \omega_k)$ for macro-actions. The reward $\bar{\mathcal{R}}$ represents the gain in map accuracy for a given orbit, and will be discussed in-depth in Section 8.5.

Value function Accordingly, the value function $\bar{J}$ is defined as the running-sum of all the rewards:

$$\bar{J}(b_0; \bar{\pi}) = \sum_{k=0}^{T} \bar{\mathcal{R}}_k(b_k, \bar{\pi}(b_k)) = \sum_{k=0}^{T} \bar{\mathcal{R}}_k(b_k, \omega_k)$$

Reduced planning problem Optimizing over the reduced policy space, the planning problem in Equation (8.1) is reduced to finding the optimal $\bar{\pi}^*$ given a defined subset of possible orbits Ω:

$$\bar{\pi}^* = \arg\max_{\bar{\Pi}} J(b_0; \bar{\pi}) \tag{8.6}$$

The block diagram in Figure 8.6 represents the autonomous planning architecture.

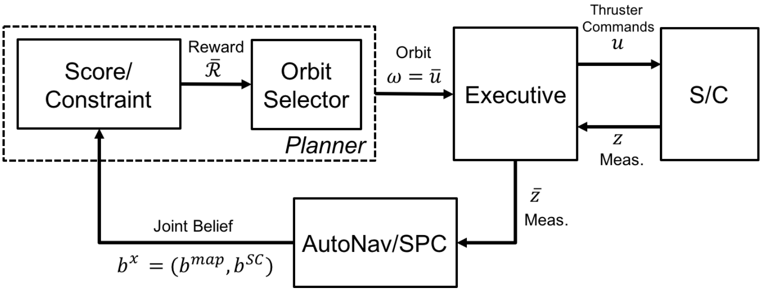


Figure 8.6: Autonomous Planning Architecture. The visual representation shows how the *planner* selects the best orbit based on the reward $\bar{\mathcal{R}}$.

8.5 Mapping accuracy

For a given orbit, we derive a quantitative measure to capture the information gain in map accuracy. This measure will drive the planning framework. We denote this measure (or reward) by $\bar{\mathcal{R}}$. The reward is a function of the system belief b_k and the selected orbit ω_k, i.e, $\bar{\mathcal{R}}(b_k, \omega_k)$. The system belief b_k encodes the accuracy of map representation as well as the spacecraft state. ω_k is the granularity of the decision variable. The reward over belief can be written as:

$$\bar{\mathcal{R}}(b_k, \omega_k) = \mathbb{E}\bar{\mathcal{R}}_x(x_k, \omega_k), \quad x_k \sim b_k(x) \tag{8.7}$$

where, $\bar{\mathcal{R}}_x(x_k, \omega_k)$ is the information gain measure for a given state of the system (i.e., map and spacecraft) x_k. $\mathbb{E}[\cdot]$ represents the expectation operator.

Computing the exact value of the expectation in Equation (8.7) for a general belief is computationally intractable. The expectation can be approximated via a finite sum or via most likely state assumption:

$$\bar{\mathcal{R}}(b_k, \omega_k) \approx \sum_{i=1}^{n} \bar{\mathcal{R}}_x(x_k^{(i)}, \omega_k), \quad x_k^{(i)} \sim b_k \tag{8.8}$$

$$\bar{\mathcal{R}}(b_k, \omega_k) \approx \bar{\mathcal{R}}_x(\grave{x}_k, \omega_k) = \bar{\mathcal{R}}_x(\arg\max_x b_k(x), \omega_k) \tag{8.9}$$

where, $x_k^{(1)}, \cdots, x_k^{(n)}$ are n samples drawn from the belief distribution, and $\grave{x}_k$ is the most likely state under b_k.

The definition of the reward $\bar{\mathcal{R}}_x(x_k, \omega_k)$ depends on the selected mapping technique. While the framework can use different measures and mapping methods, in this work, we focus on a measure that suits the SPC mapping method. Translating all the SPC mapping performance indices into a single quantitative measure is a challenge. Focusing on information gain and coverage, the measure is derived from mapping requirements in terms of emission, incidence, solar azimuth, and spacecraft azimuth angles. We detail this derivation in the following paragraphs. We start by defining some concepts for the photometry purposes.

Angles definition The definition of three angles is necessary to describe the reflectance properties of a surface: incidence angle ι, emission angle e, and phase angle ϕ. As it is seen in Figure 8.7, ι defines the angle between Sun incidence and surface normal $\hat{n}$. The emission angle e is the angle between emission ray and surface normal. The phase angle ϕ is the angle between the vector connecting the Sun to the center of the body and the vector connecting the spacecraft to the center of the body. Moreover, solar and spacecraft azimuth angles can be defined and they will be used as indicators of the goodness of the mapping phase. The solar azimuth angle (α) is the angle between local north and the projection of the vector from the surface to the Sun onto the local surface. The spacecraft azimuth angle (β) is the angle between local north and the projection of the vector from the surface to the spacecraft onto the local surface.

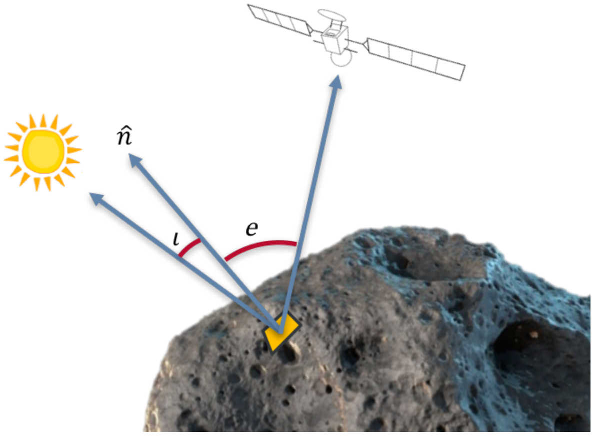


Figure 8.7: Angles Definition

Emission Angle Requirements In small body mapping, the core of the algorithm is a least square solver. The variation in emission angle increases the robustness of the solver since it minimizes the weighted sum of square-residuals between the extracted (from real images) and predicted brightness (with rendered images). In fact, the brightness of a point on the surface is dependent on the emission and incidence angles. In particular, large variation in emission angle is desirable because, in this case, the orbit offers a great variety of observation geometry of the same portion of the body. Beneficial conditions are achieved when images of each facet are taken at, at least, 3 distinct emission angle bins in intervals 20-30°, 30-40°, 40-50°, and 50-60°. Moreover, at least, one image should have an emission angle less than 10°. Mathematically, the above-mentioned requirements can be translated into:

$$S^i_{e,k} = f^i_{e,k} + l^i_{e,k} \tag{8.10}$$

where: Function $U(e, a, b)$ returns one if $a \leq e \leq b$ and returns zero otherwise. $l^i_{e,k} = U(e^i_k, 0°, 10°)$.

Incidence Angle Requirements The incidence angle is an indicator of the illumination condition of the body. In general, for SPC, the best condition for incidence angle is when images are taken with an angle between 20°and 60°. The reason is that the small angles result in images with excessive brightness that can saturate the image and large angles produce images with excessive shadows and dark areas. Mathematically, the desired score is represented as:

$$S^i_{\iota,k} = U(\iota^i_k, 20°, 60°) \tag{8.11}$$

Solar Azimuth Angle Requirements The solar azimuth angle is another indicator of the variation of the illumination conditions. As previously explained, for SPC, it is advantageous to have a great variety of illumination levels. In other words, for a given orbit, it is desirable that the maximum variation of the solar azimuth is greater than 120°. This can be expressed as:

$$S^i_{\alpha,k} = U(|max(\alpha^i_k) - min(\alpha^i_k)|, 120°, 360°) \tag{8.12}$$

Spacecraft Azimuth Angle Requirements On the contrary, the spacecraft azimuth angle indicates how the spacecraft position changes with respect to the small body. It is desirable to ensure robust image matching across different views of the same facet. Therefore, we limit the maximum variation in the viewing angle to 100°. Therefore:

$$S^i_{\beta,k} = U(|max(\beta^i_k) - min(\beta^i_k)|, 0°, 100°) \tag{8.13}$$

Overall single-facet score function The overall score function for the i-th facet is the combination of the previous contributions:

$$S^i_k = S^i_{e,k} + S^i_{\iota,k} + S^i_{\alpha,k} + S^i_{\beta,k} \tag{8.14}$$

Total reward The score function defined in Equation (8.14) captures the value of information in the observation. One needs to quantify how much this information contributes to the map information. For example, if a facet has been already observed many times in favorable angles, there will be limited gain in observing it again. In general, the gain in map accuracy for a facet depends on the previous observations from that facet. The information stored in the i-th facet can be expressed as:

$$B^i_k = \sum_{k=0}^{T} S^i_k \tag{8.15}$$

The contribution of the measured score S^i_k (of the i-th facet at the k-th time step) is weighted by $w^i_k = \xi(|\xi + \beta^{L^i_k}|)^{-1}$ where $L^i_k = 1 - e^{-\psi K^i_k}$ is a quantity, bounded between 0 and 1, representing the map accuracy. β, ξ and ψ are tuning parameters. The trend of the weight, for different values of the tuning parameter ξ, and for a fixed $\beta = 500$ is shown in Figure 8.8.

Figure 8.8 confirms that the facets that have already been observed well are not affected much by new observations, and the facets that are poorly observed, are highly influenced by addition of the new information. Finally, the reward function can be defined as the information gain at every time step for every facet:

$$\bar{\mathcal{R}}_k = \sum_{i=1}^{n} w_k^i S_k^i \tag{8.16}$$

and therefore, the corresponding planning problem is:

$$\bar{\pi}^* = \arg\max_{\bar{\Pi}} \sum_{k=0}^{T} \bar{\mathcal{R}}_k(x_k, \omega_k) \tag{8.17}$$

This results in selecting the orbit that maximizes the information gain of the poorly observed regions increasing their map accuracy.

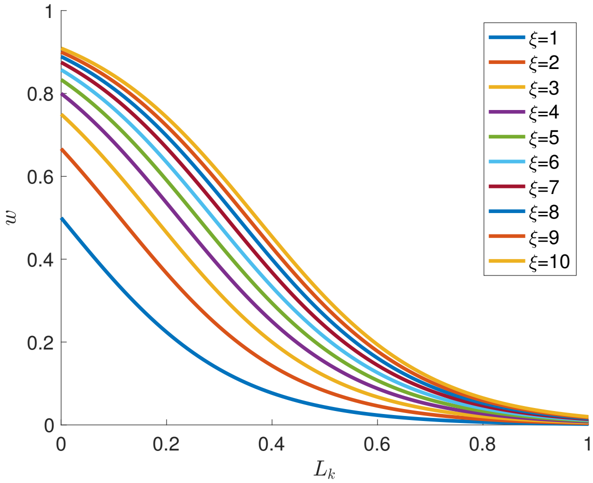


Figure 8.8: Weight Function Trend

8.6 Results

In this section, we present numerical results of the proposed algorithm. In our simulations, to create a high-fidelity model of the small body shape and gravity, we rely on JPL's Small Body Dynamics Toolkit (SBDT) [133] and the Primitive Body Coverage and Geometry Evaluator (PB-CAGE) library [134].

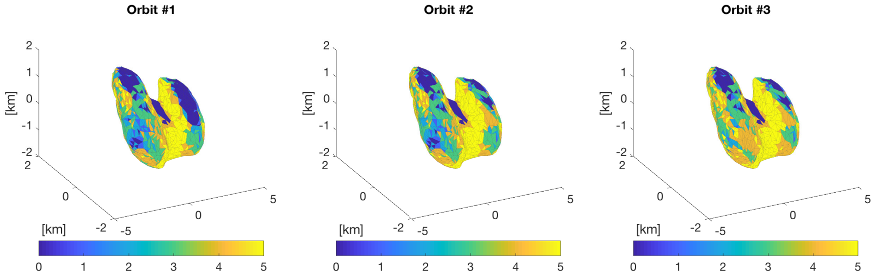


Figure 8.9: Maximum Facet Score Evolution - Proposed Solution - OOS

For this simulation, we adopt a model of the comet CG67-P where its surface is represented by a mesh with 1146 vertices and 2288 facets. For a given orbit, we evaluate the reward, introduced in Section 8.4, for each facet while taking into account the position of spacecraft, small body and Sun. We evaluate the score for a set of circular orbits of radius 10km, centered in the small body. Each orbit is characterized by a different inclination angle, ranging from 0° to 170° with a delta-angle of 10°.

We compare the proposed method for autonomous selection of optimal orbit sequence (OOS) with two baseline approaches: fixed inclination orbit sequence (FIOS), and random sequence of orbits (RSO). In the FIOS baseline, user selects a suitable orbit at the beginning of the mapping and the orbit does not change over time. In our simulation, we selected a polar orbit for the FIOS case. In RSO, at each step, we pick a random orbit from the set of candidate orbits.

The performance and comparison metric among these three different alternatives is the overall coverage score. A higher score is translated into a better coverage, resulting in a better map. In fact, acquiring *good* images is mandatory for the correct behavior of the SPC mapping algorithm. *Bad* images can lead the process to diverge and, therefore, this implies a heavier human effort. Designing a good trajectory is the first step to make the mapping procedure easier and, in principle, automatic.

A sequence of three orbits is considered for each case. Figures 8.9 to 8.11 show the evolution in time of the maximum score for each facet.

The yellow areas represent the part of the surface that have an higher coverage score after one orbit. As it can be seen in Figure 8.9, the OOS guarantees an increase of the coverage score in time. In Figure 8.10, RSO shows an overall lower score compared to OOS, with some poor areas. Finally, in Figure 8.11, as expected, the coverage score for the FIOS case is constant in time. To have a more quantitative idea of the results, Table 8.2

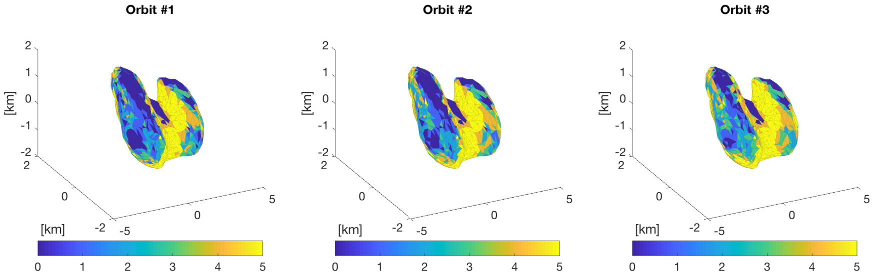


Figure 8.10: Maximum Facet Score Evolution - RSO

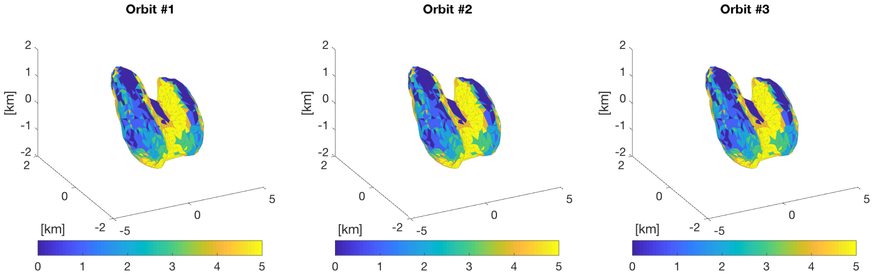


Figure 8.11: Maximum Facet Score Evolution - FIOS

reports the total coverage score at the end of the third orbit. This value is computed by summing the maximum score for each facet at the end of the mapping phase.

Table 8.2: Total Coverage Score

	Total Coverage Score
Proposed Solution - OOS	8956
RSO	7992
FIOS	6875

Moreover, Table 8.3 and Figure 8.12 report the number of facets with global score <3 at the end of the mapping phase. This is an indication of the number of poor covered regions at the end of the mapping phase.

Figure 8.12 shows how the selection of the optimal sequence offers a significant improvement in the global coverage of the body. Finally, Figure 8.13 shows the time evolution of the facets with score equal to 5, the ideal condition in terms of coverage. According to the measures in Figure 8.13, OOS supersedes other methods in the number of facets with maximum score.

Table 8.3: Poor Coverage Regions

	Poor Score Facets
Proposed Solution - OOS	283
RSO	576
FIOS	867

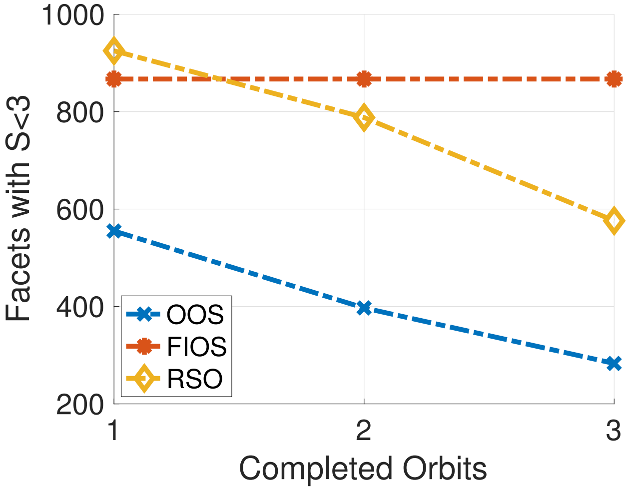


Figure 8.12: Facets with Score < 3

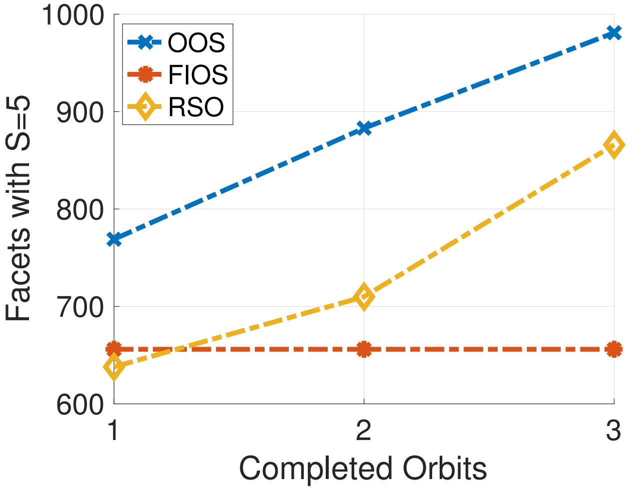


Figure 8.13: Facets with Score $= 5$

CHAPTER 9

Conclusion

> The important thing is to search, and not
> whether you find something or not
> *Per Isabel (2015)*
> ANTONIO TABUCCHI

THIS thesis offers a comprehensive study about relative navigation. Innovative concepts and algorithms have been proposed for vision-based relative navigation for approach to uncooperative space objects. Different target classes and orbital scenarios are considered; pose determination and filtering techniques, and planning strategies are developed and tested. In particular, in Chapter 4 an original approach for relative navigation between two uncooperative but known spacecraft is presented. The proposed algorithm relies on a loosely coupled approach, involving a vision-based pose determination technique and a navigation filter. Two different strategies are detailed and results with different level of maturity are presented. The algorithm implemented for Strategy A involves two main steps, i.e. pose acquisition and tracking. The acquisition phase is managed by an innovative methodology conceived to compute the target initial state when no a-priori information is available about this satellite, except that concerning its geometry. A RANSAC & PCA based procedure is proposed to solve the 2D-3D feature matching problem. Hence, the spacecraft initial pose is derived through the EPnP solver. Once the target state is acquired, it is tracked over time by filtering the monocular pose determined through SoftPOSIT. The navigation filter is conceived to have a decoupled structure. A robust H-∞ filtering strategy is exploited for the translational motion. On the other hand, an original formulation on Lie groups is developed for relative attitude estimation. A numerical validation campaign and performance assessment are carried considering different simulation scenarios. Representative target/chaser relative dynamics, target geometries and vision-based measurement are reproduced. Simulation results show the capability of the proposed pose acquisition algorithm to provide an accurate initialization for the tracking step (position and attitude errors lower than $0.5m$ and 2.5° in most of the cases). The sensitivity analysis on the acquisition step highlights a dependence between relative angular velocity and acquisition time. Specifically, higher relative angular velocities imply longer time to initialize the pose parameters. Furthermore, the pose tracking functionality and the navigation filter are validated through statistical simulations, considering different orbital scenarios. Satisfactory results are obtained for all the presented cases for both position and attitude estimation. In fact, steady state relative position and attitude RMSE are lower than $3cm$ (except in the HEO case) and 1° respectively. Finally, sensitivity analyses are performed to demonstrate the algorithm robustness against measurement noise and error sources. A preliminary experimental validation campaign is also presented to test the image processing algorithm considering realistic images. Exploiting the facility for vision-based autonomous GNC validation at PoliMi-DAER, the algorithm is tested under controlled, realistic illumination conditions and relative approach trajectory. A representative mock-up is manufactured to reproduce the optical properties of a real spacecraft. The results of the preliminary experimental validation campaign show that the image processing algorithm is able to correctly perform pose acquisition and tracking even with real images. The accuracy of the proposed algorithm with real images is in line with the values obtained with simulated, numerical acquisitions. On the other hand, Strategy B consists of an algorithm that works detecting ORB descriptors in the incoming images. These descriptors are then matched with an already available on-board map of the target satellite. In this way, a set of 3D to 2D correspondences is obtained. The set of correspondences is then used to solve the PnP problem to estimate the relative pose between the two spacecraft. Finally, Motion only BA, is applied to optimize the relative pose estimate. Preliminary numerical validation results, carried out on a realistic relative motion and target geometry, are satisfactory. A deeper analysis is currently on-going to test the algorithm robustness, also considering real images. Moreover, regarding the specific topic of attitude estimation, in Chapter 5 a comparative analysis of different filtering techniques for relative attitude estimation is presented. In particular, their sensitivity to the variations in measurement noise intensity, initial conditions, and angular velocities of the non-cooperative object in space is tested. This analysis confirmed that the second-order minimum energy filter without dynamics, proposed in this thesis, seems to be the best option for relative attitude estimation dealing with partially-known space objects. In fact, it requires neither an estimate of the absolute attitude nor any knowledge of the inertia matrix. Although the second-order minimum energy filter is the best performer at the steady-state, it may be outperformed by the second-order minimum energy filter without dynamics, when accurate estimates of the target's inertia matrix and absolute attitude of the chaser spacecraft are not available. A slightly different scenario is considered in Chapter 6. In this chapter the performance of a vision-based relative navigation algorithm in cislunar space is investigated. A realistic orbital scenario is adopted, considering a chaser equipped with two cameras and a passively cooperative target. In particular, we assumed to know the position of feature points on the target, representative of optical markers. An EKF is developed, using a linearized, ephemeris-based model to describe the relative translational dynamics and a classical combination of the Euler equations for target and chaser for the rotational dynamics. A proper observation model, considering a stereo camera, is adopted. A preliminary numerical validation is presented and promising results are obtained. In fact, assuming realistic noise values, position and attitude errors are lower than $0.1m$ and $0.2°$ respectively, in the nominal case. The algorithm robustness is also tested over different noise levels and number of feature points. The presented analyses show that a vision-based approach for relative navigation can be adopted for cislunar relative operations, if moderate level of accuracy is needed. In Chapter 7 an original approach for relative state estimation and uncertainties estimation is presented. The proposed algorithm relies on a RBFNN coupled with an Adaptive Extended Kalman Filter (AEKF). The proposed neural-network performs an on-line estimation of the disturbances acting on the spacecraft, which are included in the prediction step of the filter. The on-line learning algorithm exploits the state estimation worked out by the filter itself to update the neural network weights. Moreover, an innovation-based recursive filter architecture is employed. Preliminary numerical validation, performance assessment and comparison are carried considering a spacecraft relative navigation scenario. Representative target/chaser relative dynamics are reproduced. Simulation results show the capability of the proposed solution to reconstruct the dynamics of a spacecraft in elliptic orbits with the J_2 perturbation in an Earth orbit environment. Furthermore, the filter performance is compared to more classical approaches and tested on realistic scenarios, through statistical simulations. Finally, the robustness over very poor tuning of the state covariance matrix is considered. Satisfactory results are obtained for the proposed solution in all the presented cases and the sensitivity analysis demonstrated the algorithm robustness in non-ideal situations. Chapter 7 has presented the application to a Earth-bounded motion. Nevertheless, the proposed method finds applications, definitely more attractive, to those environment which are not as characterized as Earth orbits, such as small bodies, Non-Keplerian dynamics and, in general, proximity maneuvers with uncooperative targets. Finally, in Chapter 8 the problem of navigating, mapping and planning around an uncooperative, unknown object is presented. In this chapter, we introduce an algorithm for autonomous orbit selection and adaptation around small bodies while mapping its surface. Focusing on the SPC mapping method, we develop cost functions that quantify the 'orbit goodness' in the sense of map improvement. These mapping accuracy measures are then used to guide the orbit selection process. We rely on a reduced policy space where orbits are selected from a representative family of common orbits around small bodies. Via numerical simulations, we show that the optimal sequence of orbits, resulting from the optimization process, offers a substantial improvement of the mapping performance compared to the fixed orbit case. The proposed algorithm is a step toward a higher level of autonomy and on-board trajectory planning around small bodies and reduces the human involvement in this process. Concluding, all the different approaches and algorithms presented in this dissertation provide possible answers to the main challenges of vision-based proximity relative navigation with uncooperative objects.

Future Work

The work presented in this dissertation can be further validated and extended to different mission scenarios. In particular, the algorithm including the Strategy A pose estimation and the navigation filter, presented in Chapter 3, needs to be validated through a more extensive experimental campaign. In the framework of the ASI-funded VINAG project, an experimental test is scheduled to be performed by the end of 2018. During this test, both pose determination algorithm and filters are planned to work in real-time in a Hardware In the Loop (HIL) simulation. The algorithm will run in real-time on the VINAG Central Unit, based on a Microsemi RTG4

Flash FPGA. For a more detailed description of the hardware, please refer to [31]. For Strategy B, instead, an extended numerical validation campaign is currently on-going, along with the generation of the real 3D map of the target satellite for a preliminary experimental validation campaign. The algorithm presented in Chapter 6 has a different level of maturity and, even if a similar experimental validation setup can be used also in this case, some aspects still have to investigated numerically. In particular, the influence of the position of the feature points and the rotational dynamics of the target spacecraft has to be further examined. Different relative orbital scenarios have to be considered and a validity domain defined depending on the camera parameters. Moreover, for the RBFNN-AEKF described in Chapter 7, future developments aim at implementing and testing a recurrent structure for the neural network, which is expected to be significantly more performing when secular disturbance terms become predominant. Finally, the natural future step for the planning solution proposed in Chapter 8 is to incorporate the small body gravity and high-fidelity orbital dynamics into the formulation, considering also the observation time along the orbit.

Bibliography

[1] J. Miller, *Planetary Spacecraft Navigation.* Springer International Publishing, 2019. DOI: `10.1007/978-3-319-78916-3`.

[2] R Ambrose, I. Nesnas, F Chandler, B. Allen, T Fong, L Matthies, and R Mueller, "NASA technology roadmaps: TA 4: Robotics and autonomous systems", Technical report, NASA, Tech. Rep., 2015.

[3] J. A. Starek, B. Açıkmeşe, I. A. Nesnas, and M. Pavone, "Spacecraft autonomy challenges for next-generation space missions", in *Advances in Control System Technology for Aerospace Applications*, Springer, 2016, pp. 1–48. DOI: `10.1007/978-3-662-47694-9_1`.

[4] G. Di Mauro, M. Lawn, and R. Bevilacqua, "Survey on Guidance Navigation and Control Requirements for Spacecraft Formation-Flying Missions", *Journal of Guidance, Control, and Dynamics*, no. December 2017, pp. 1–22, 2017. DOI: `10.2514/1.G002868`.

[5] X. Wang, W. Qin, Y. Bai, and N. Cui, "A novel decentralized relative navigation algorithm for spacecraft formation flying", *Aerospace Science and Technology*, vol. 48, pp. 28–36, 2016. DOI: `10.1016/j.ast.2015.10.014`.

[6] J. P. Alepuz, M. R. Emami, and J. Pomares, "Direct image-based visual servoing of free-floating space manipulators", *Aerospace Science and Technology*, vol. 55, pp. 1–9, 2016. DOI: `10.1016/j.ast.2016.05.012`.

[7] D. Pinard, S. Reynaud, P. Delpy, and S. E. Strandmoe, "Accurate and autonomous navigation for the ATV", *Aerospace Science and Technology*, vol. 11, no. 6, pp. 490–498, 2007. DOI: `10.1016/j.ast.2007.02.009`.

[8] L. Zhang, S. Zhang, H. Yang, H. Cai, and S. Qian, "Relative attitude and position estimation for a tumbling spacecraft", *Aerospace Science and Technology*, vol. 42, pp. 97–105, 2015. DOI: `10.1016/j.ast.2014.12.025`.

[9] L. Felicetti and M. R. Emami, "Image-based attitude maneuvers for space debris tracking", *Aerospace Science and Technology*, vol. 76, pp. 58–71, 2018. DOI: `10.1016/j.ast.2018.02.002`.

[10] 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 *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, IEEE. DOI: `10.1109/robot.2000.844757`.

[11] S Croomes, "Overview of the DART mishap investigation results", *NASA Report*, pp. 1–10, 2006.

[12] R. B. Friend, "Orbital Express program summary and mission overview", in *Sensors and Systems for Space Applications II*, R. T. Howard and P. Motaghedi, Eds., SPIE, 2008. DOI: `10.1117/12.783792`.

[13] P. Bodin, R. Noteborn, R. Larsson, T. Karlsson, S. D'Amico, J. S. Ardaens, M. Delpech, and J.-C. Berges, "The prisma formation flying demonstrator: Overview and conclusions from the nominal mission", *Advances in the Astronautical Sciences*, vol. 144, no. 2012, pp. 441–460, 2012.

[14] S. D'Amico, J.-S. Ardaens, and R. Larsson, "Spaceborne Autonomous Formation Flying Experiment on the PRISMA Mission", in *AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, 2011. DOI: `10.2514/6.2011-6232`.

[15] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations", *Progress in Aerospace Sciences*, vol. 93, pp. 53–72, 2017. DOI: `10.1016/j.paerosci.2017.07.001`.

[16] R. Kroes, O. Montenbruck, W. Bertiger, and P. Visser, "Precise GRACE baseline determination using GPS", *GPS Solutions*, vol. 9, no. 1, pp. 21–31, 2005. DOI: `10.1007/s10291-004-0123-5`.

[17] X. Clerc and I. Retat, "Astrium vision on space debris removal", in *Proceeding of the 63rd International Astronautical Congress (IAC 2012), Napoli, Italy*, vol. 15, 2012.

[18] R. Volpe, G. B. Palmerini, and C. Circi, "Preliminary analysis of visual navigation performance in close formation flying", in *Aerospace Conference, 2017 IEEE*, IEEE, 2017, pp. 1–12. DOI: `10.1109/aero.2017.7943759`.

[19] J. Shi and Tomasi, "Good features to track", in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*, IEEE, IEEE Comput. Soc. Press, 1994, pp. 593–600. DOI: `10.1109/cvpr.1994.323794`.

[20] J. Canny, "A computational approach to edge detection", in *Readings in Computer Vision*, Elsevier, 1987, pp. 184–203. DOI: `10.1016/b978-0-08-051581-6.50024-6`.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004. DOI: `10.1023/b:visi.0000029664.99615.94`.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF", in *Computer Vision (ICCV), 2011 IEEE international conference on*, IEEE, 2011, pp. 2564–2571. DOI: `10.1109/iccv.2011.6126544`.

[23] 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. DOI: `10.1109/icra.2011.5980232`.

[24] R. Volpe, M. Sabatini, and G. B. Palmerini, "Pose and Shape Reconstruction of a Noncooperative Spacecraft Using Camera and Range Measurements", *Interna-*

tional Journal of Aerospace Engineering, vol. 2017, 2017. DOI: `10.1155/2017/4535316`.

[25] V. Pesce, M. Lavagna, and R. Bevilacqua, “Stereovision-based pose and inertia estimation of unknown and uncooperative space objects”, *Advances in Space Research*, vol. 59, no. 1, pp. 236–251, 2017. DOI: `10.1016/j.asr.2016.10.002`.

[26] J. Yuan, X. Hou, C. Sun, and Y. Cheng, “Fault-tolerant pose and inertial parameters estimation of an uncooperative spacecraft based on dual vector quaternions”, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 2018. DOI: `10.1177/0954410017751766`.

[27] N. Chaturvedi, A. K. Sanyal, N. H. McClamroch, *et al.*, “Rigid-body attitude control”, *IEEE control systems magazine*, vol. 31, no. 3, pp. 30–51, 2011. DOI: `10.1109/mcs.2011.940459`.

[28] V. Pesce, S. Silvestrini, and M. Lavagna, “Radial Basis Function Neural Network aided Adaptive Extended Kalman Filter”, *Journal of Guidance, Control and Dynamics [under review]*, 2018.

[29] V. Pesce, M. F. Haydar, M. Lavagna, and M. Lovera, “Comparison of Filtering Techniques For Relative Attitude Estimation of Uncooperative Space Objects”, *Aerospace Science and Technology*, 2018. DOI: `10.1016/j.ast.2018.10.031`.

[30] V. Pesce, R. Opromolla, S. Sarno, M. Lavagna, and M. Grassi, “Autonomous Relative Navigation Around Uncooperative Spacecraft Based on a Single Camera”, *Aerospace Science and Technology [under review]*, 2018.

[31] V. Capuano, V. Pesce, R. Opromolla, G. Cuciniello, S. Sarno, G. Capuano, M. Lavagna, M. Grassi, F. Corraro, P. Tabacco, *et al.*, “A Highly Integrated Navigation Unit for On-Orbit Servicing Missions”, in *69th International Astronautical Congress (IAC 2018)*, 2018, pp. 1–13.

[32] P. Lunghi, V. Pesce, L. Losi, and M. Lavagna, “Ground testing of vision-based GNC systems by means of a new experimental facility”, in *Proc. 69th International Astronautical Congress*, 2018.

[33] V. Capuano, V. Pesce, R. Opromolla, S. Sarno, M. Lavagna, M. Grassi, F. Corraro, G. Capuano, P. Tabacco, *et al.*, “VINAG: A Highly integrated system for autonomous on-board absolute and relative spacecraft navigation”, in *Small Satellites Systems and Services - The 4S Symposium*, 2018.

[34] V. Pesce, A.-a. Agha-mohammadi, and M. Lavagna, “Autonomous navigation & mapping of small bodies”, in *2018 IEEE Aerospace Conference*, IEEE, 2018, pp. 1–10. DOI: `10.1109/aero.2018.8396797`.

[35] V. Pesce, R. Opromolla, S. Sarno, M. Lavagna, and M. Grassi, “Vision-Based Pose Estimation and Relative Navigation Around Uncooperative Space Objects”, in *10th International ESA Conference on Guidance, Navigation & Control Systems (GNC 2017)*, 2017, pp. 1–16.

[36] V. Pesce, L. Losi, and M. Lavagna, “Vision-Based State Estimation of an Uncooperative Space Object”, in *68th International Astronautical Congress (IAC 2017)*, International Astronautical Federation, IAF, 2017, pp. 7018–7026.

[37] R. E. Kalman, “A new approach to linear filtering and prediction problems”, *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960. DOI: `10.1115/1.3662552`.

[38] R. E. Kalman and R. S. Bucy, “New results in linear filtering and prediction theory”, *Journal of basic engineering*, vol. 83, no. 1, pp. 95–108, 1961. DOI: `10.1115/1.3658902`.

[39] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006. DOI: `10.5860/choice.44-3334`.

[40] J. Bellantoni and K. Dodge, "A square root formulation of the Kalman-Schmidt filter.", *AIAA journal*, vol. 5, no. 7, pp. 1309–1314, 1967. DOI: `10.2514/6.1967-90`.

[41] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation", in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, Ieee, 2000, pp. 153–158.

[42] N. Stacey and S. D'Amico, "Autonomous Swarming for Simultaneous Navigation and Asteroid Characterization", in *AAS/AIAA Astrodynamics Specialist Conference*, 2018.

[43] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", in *IEE Proceedings F (Radar and Signal Processing)*, IET, vol. 140, 1993, pp. 107–113. DOI: `10.1049/ip-f-2.1993.0015`.

[44] F. Gustafsson, "Particle filter theory and practice with positioning applications", *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, 2010. DOI: `10.1109/maes.2010.5546308`.

[45] R. E. Mortensen, "Maximum-likelihood recursive nonlinear filtering", *Journal of Optimization Theory and Applications*, vol. 2, no. 6, pp. 386–394, 1968. DOI: `10.1007/bf00925744`.

[46] H. Curtis, *Orbital Mechanics for Engineering Students*. Elsevier, 2005. DOI: `10.1016/c2011-0-69685-1`.

[47] A. Colagrossi and M. Lavagna, "Preliminary results on the dynamics of large and flexible space structures in Halo orbits", *Acta Astronautica*, vol. 134, pp. 355–367, 2017. DOI: `10.1016/j.actaastro.2017.02.020`.

[48] W. H. Clohessy and R. S. Wiltshire, "Terminal Guidance System for Satellite Rendezvous", *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960. DOI: `10.2514/8.8704`.

[49] K. Yamanaka and F. Ankersen, "New state transition matrix for relative motion on an arbitrary elliptical orbit", *Journal of guidance, control, and dynamics*, vol. 25, no. 1, pp. 60–66, 2002. DOI: `10.2514/2.4875`.

[50] D. Wang, B. Wu, and E. K. Poh, *Satellite Formation Flying*. 2017, vol. 87. DOI: `10.1007/978-981-10-2383-5`.

[51] V. Szebehely and F. T. Geyling, "Theory of Orbits: The Restricted Problem of Three Bodies", *Journal of Applied Mechanics*, vol. 35, no. 3, p. 624, 1968. DOI: `10.1115/1.3601280`.

[52] L Bucci, M Lavagna, R Jehn, *et al.*, "Station Keeping Techniques for Near Rectilinear Orbits in the Earth-Moon System", in *10th International ESA Conference on Guidance, Navigation & Control Systems (GNC 2017)*, 2017, pp. 1–14.

[53] A. Colagrossi and M. Lavagna, "Cislunar non-Keplerian orbits rendezvous and docking: 6DOF Guidance and Control", in *Proc. 69th International Astronautical Congress*, 2018.

[54] R. J. Luquette, "Nonlinear control design techniques for precision formation flying at lagrange points", PhD thesis, 2006.

[55] 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. DOI: `10.2514/1.39320`.

[56] S. Sharma *et al.*, “Comparative assessment of techniques for initial pose estimation using monocular vision”, *Acta Astronautica*, vol. 123, pp. 435–445, 2016. DOI: `10.1016/j.actaastro.2015.12.032`.

[57] 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 5*, vol. 2, no. 2, pp. 171–189, 2014. DOI: `10.1504/ijspacese.2014.060600`.

[58] S. Sharma, J. Ventura, and S. D’Amico, “Robust Model-Based Monocular Pose Initialization for Noncooperative Spacecraft Rendezvous”, *Journal of Spacecraft and Rockets*, pp. 1–16, 2018. DOI: `10.2514/1.a34124`.

[59] B. J. Naasz, R. D. Burns, S. Z. Queen, J. Van Eepoel, J. Hannah, and E. Skelton, “The HST SM4 Relative Navigation Sensor System: Overview and Preliminary Testing Results from the Flight Robotics Lab”, *The Journal of the Astronautical Sciences*, vol. 57, no. 1-2, pp. 457–483, 2009. DOI: `10.1007/bf03321512`.

[60] J.-F. Shi, S. Ulrich, and S. Ruel, “Spacecraft Pose Estimation using Principal Component Analysis and a Monocular Camera”, in *AIAA Guidance, Navigation, and Control Conference*, 2017, p. 1034. DOI: `10.2514/6.2017-1034`.

[61] P. David, D. Dementhon, R. Duraiswami, and H. Samet, “SoftPOSIT: Simultaneous pose and correspondence determination”, *International Journal of Computer Vision*, vol. 59, no. 3, pp. 259–284, 2004. DOI: `10.1007/3-540-47977-5_46`.

[62] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”, *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981. DOI: `10.1145/358669.358692`.

[63] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment - a modern synthesis”, in *Vision algorithms: theory and practice*, Springer, 1999, pp. 298–372. DOI: `10.1007/3-540-44480-7_21`.

[64] A. Rhodes, E. Kim, J. A. Christian, and T. Evans, “LIDAR-based relative navigation of non-cooperative objects using point Cloud Descriptors”, in *AIAA/AAS Astrodynamics Specialist Conference*, 2016, p. 5517. DOI: `10.2514/6.2016-5517`.

[65] 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, pp. 2974–2979. DOI: `10.1109/robot.2004.1307513`.

[66] S. Segal, A. Carmi, and P. Gurfil, “Stereovision-based estimation of relative dynamics between noncooperative satellites: Theory and experiments”, *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 568–584, 2014. DOI: `10.1109/tcst.2013.2255288`.

[67] S. Berkane and A. Tayebi, “On the design of synergistic potential functions on SO (3)”, in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, IEEE, 2015, pp. 270–275. DOI: `10.1109/cdc.2015.7402120`.

[68] D Choukroun, H Weiss, I. Bar-Itzhack, and Y Oshman, “Direction cosine matrix estimation from vector observations using a matrix Kalman filter”, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 1, 2010. DOI: `10.1109/taes.2010.5417148`.

[69] C. Lageman, J. Trumpf, and R. Mahony, “Gradient-like observers for invariant dynamics on a Lie group”, *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 367–377, 2010. DOI: `10.1109/tac.2009.2034937`.

[70] D. F. Dementhon and L. S. Davis, “Model-based object pose in 25 lines of code”, *International journal of computer vision*, vol. 15, no. 1, pp. 123–141, 1995. DOI: `10.1007/3-540-55426-2_38`.

[71] S. Gold and A. Rangarajan, “A graduated assignment algorithm for graph matching”, *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 4, pp. 377–388, 1996. DOI: `10.1109/icnn.1996.549117`.

[72] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis”, *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987. DOI: `10.1007/0-387-22440-8_7`.

[73] V. Lepetit, F. Moreno-Noguer, and P. Fua, “EPnP: An Accurate O(n) Solution to the PnP Problem”, *International journal of computer vision*, vol. 81, no. 2, pp. 155–166, 2009. DOI: `10.1007/s11263-008-0152-6`.

[74] D. Scaramuzza and F. Fraundorfer, “Visual odometry [tutorial]”, *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 80–92, 2011. DOI: `10.1109/mra.2011.943233`.

[75] F. Fraundorfer and D. Scaramuzza, “Visual odometry: Part II: Matching, robustness, optimization, and applications”, *Robotics & Automation Magazine, IEEE*, vol. 19, no. 2, pp. 78–90, 2012. DOI: `10.1109/mra.2012.2182810`.

[76] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features”, in *Computer vision–ECCV 2006*, Springer, 2006, pp. 404–417. DOI: `10.1007/11744023_32`.

[77] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection”, in *Computer Vision–ECCV 2006*, Springer, 2006, pp. 430–443. DOI: `10.1007/11744023_34`.

[78] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features”, *Computer Vision–ECCV 2010*, pp. 778–792, 2010. DOI: `10.1007/978-3-642-15561-1_56`.

[79] R. Mur-Artal, J. Montiel, and J. D. Tardos, “ORB-SLAM: a versatile and accurate monocular SLAM system”, *Robotics, IEEE Transactions on*, vol. 31, no. 5, pp. 1147–1163, 2015. DOI: `10.1109/tro.2015.2463671`.

[80] M. Zamani, J. Trumpf, and R. Mahony, “Minimum-energy filtering for attitude estimation”, *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2917–2921, 2013. DOI: `10.1109/tac.2013.2259092`.

[81] M. Zamani, J. Trumpf, and R Mahony, “Nonlinear attitude filtering: a comparison study”, *arXiv preprint arXiv:1502.03990*, 2015.

[82] A. Saccon, J. Trumpf, R. Mahony, and A. P. Aguiar, “Second-order-optimal minimum-energy filters on lie groups”, *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2906–2919, 2016. DOI: `10.1109/tac.2015.2506662`.

[83] B. E. Tweddle and A. Saenz-Otero, “Relative computer vision-based navigation for small inspection spacecraft”, *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 5, pp. 969–978, 2014. DOI: `10.2514/6.2011-6296`.

[84] H Barré, H Nye, and G Janin, “An overview of the XMM observatory system.”, *ESA Bulletin*, vol. 100, pp. 15–20, 1999.

[85] K Van Katwijk, T Van Der Laan, and D Stramaccioni, “Mechanical and Thermal Design of XMM”, *ESA Bulletin*, vol. 100, pp. 44–49, 1999.

[86] SpaceFlight101. (2017). Hispasat 30W-6, [Online]. Available: `https://spaceflight101.com/falcon-9-hispasat-30w-6/hispasat-30w-6/`.

[87] F. L. Markley and D. Mortari, “Quaternion attitude estimation using vector observations”, *Journal of the Astronautical Sciences*, vol. 48, no. 2, pp. 359–380, 2000.

[88] F. L. Markley, J. L. Crassidis, and Y. Cheng, “Nonlinear attitude filtering methods”, in *AIAA Guidance, Navigation, and Control Conference*, 2005, pp. 15–18. DOI: `10.2514/6.2005-5927`.

[89] F. L. Markley and J. L. Crassidis, *Fundamentals of spacecraft attitude determination and control*. Springer, 2014, vol. 33. DOI: `10.1007/978-1-4939-0802-8`.

[90] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*. CRC press, 2011. DOI: `10.1201/b11154`.

[91] M. Zamani, J. Trumpf, and R. Mahony, “Near-optimal deterministic filtering on the rotation group”, *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1411–1414, 2011. DOI: `10.1109/tac.2011.2109436`.

[92] ——, “On the distance to optimality of the geometric approximate minimum-energy attitude filter”, in *American Control Conference (ACC), 2014*, 2014, pp. 4943–4948. DOI: `10.1109/acc.2014.6858915`.

[93] ——, “Near-optimal deterministic attitude filtering”, in *49th IEEE Conference on Decision and Control (CDC), 2010*, IEEE, 2010, pp. 6511–6516. DOI: `10.1109/cdc.2010.5717043`.

[94] B. B. Virgili, S Lemmens, and H Krag, “Investigation on Envisat attitude motion”, in *e. Deorbit Workshop*, 2014.

[95] F. Schnitzer, A. Sonnenburg, J. Klaus, and M. Sanchez Gestido, “Lessons-learned from on-ground testing of image-based non-cooperative rendezvous navigation with visible-spectrum and thermal infrared cameras”, in *10th International ESA Conference on Guidance, Navigation & Control Systems (GNC 2017)*, 2017.

[96] J. C. Crusan, R. M. Smith, D. A. Craig, J. M. Caram, J. Guidi, M. Gates, J. M. Krezel, and N. B. Herrmann, “Deep space gateway concept: Extending human presence into cislunar space”, in *2018 IEEE Aerospace Conference*, IEEE, 2018. DOI: `10.1109/aero.2018.8396541`.

[97] K. Hill, J. Parker, G. Born, and N. Demandante, “A Lunar L2 Navigation, Communication, and Gravity Mission”, in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, American Institute of Aeronautics and Astronautics, 2006. DOI: `10.2514/6.2006-6662`.

[98] P. Liu, X.-Y. Hou, J.-S. Tang, and L. Liu, “Application of two special orbits in the orbit determination of lunar satellites”, *Research in Astronomy and Astrophysics*, vol. 14, no. 10, pp. 1307–1328, 2014. DOI: `10.1088/1674-4527/14/10/010`.

[99] T. Svoboda, T. Pajdla, and V. Hlaváč, “Epipolar geometry for panoramic cameras”, in *European Conference on Computer Vision*, Springer, 1998, pp. 218–231.

[100] S. Persson, P. Bodin, E. Gill, J. Harr, and J. Jörgensen, “PRISMA–an autonomous formation flying mission”, in *ESA Small Satellite Systems and Services Symposium (4S), Sardinia, Italy*, 2006, pp. 25–29.

[101] P Gurfil, M Idan, and N. J. Kasdin, “Adaptive neural control of deep-space formation flying”, *Journal of Guidance Control and Dynamics*, vol. 26, no. 3, pp. 491–501, 2003. DOI: `10.2514/2.5072`.

[102] J. Bae and Y. Kim, “Adaptive controller design for spacecraft formation flying using sliding mode controller and neural networks”, *Journal of the Franklin Institute*, vol. 349, no. 2, pp. 578–603, 2012. DOI: `10.1016/j.jfranklin.2011.08.009`.

[103] N. Zhou, R. Chen, Y. Xia, J. Huang, and G. Wen, “Neural network-based reconfiguration control for spacecraft formation in obstacle environments”, *International Journal of Robust and Nonlinear Control*, vol. 28, no. 6, pp. 2442–2456, 2018. DOI: `10.1002/rnc.4025`.

[104] S. Sharma, C. Beierle, and S. D’Amico, “Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks”, in *2018 IEEE Aerospace Conference*, IEEE, 2018, pp. 1–12.

[105] D. Simon, “Training radial basis neural networks with the extended Kalman filter”, *Neurocomputing*, vol. 48, no. 1-4, pp. 455–475, 2002. DOI: `10.1016/s0925-2312(01)00611-7`.

[106] S. C. Stubberud, R. N. Lobbia, and M. Owen, “An adaptive extended Kalman filter using artificial neural networks”, in *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, IEEE, vol. 2, 1995, pp. 1852–1856. DOI: `10.1109/cdc.1995.480611`.

[107] X. Gao, X. Zhong, D. You, and S. Katayama, “Kalman Filtering Compensated by Radial Basis Function Neural Network for Seam Tracking of Laser Welding”, *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1916–1923, 2013. DOI: `10.1109/tcst.2012.2219861`.

[108] A Stubberud, H Wabgaonkar, and S Stubberud, “A neural-network-based system identification technique”, in *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on*, IEEE, 1991, pp. 869–870. DOI: `10.1109/cdc.1991.261441`.

[109] J. Dah-Jing and J.-J. Chen, “Neural network aided adaptive Kalman filter for GPS/INS navigation system design”, in *Proc. 9th IFAC Workshop*, 2011, pp. 1–7.

[110] N. Harl, K. Rajagopal, and S. N. Balakrishnan, “Neural Network Based Modified State Observer for Orbit Uncertainty Estimation”, *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 4, pp. 1194–1209, 2013. DOI: `10.2514/1.55711`.

[111] Y. Wu, H. Wang, B. Zhang, and K.-L. Du, “Using Radial Basis Function Networks for Function Approximation and Classification”, *ISRN Applied Mathematics*, vol. 2012, no. March, pp. 1–34, 2012. DOI: `10.5402/2012/324194`.

[112] J. Park and I. W. Sandberg, “Universal Approximation Using Radial-Basis-Function Networks”, *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991. DOI: `10.1162/neco.1991.3.2.246`.

[113] K. Reif and R. Unbehauen, “The extended Kalman filter as an exponential observer for nonlinear systems”, *IEEE Transactions on Signal processing*, vol. 47, no. 8, pp. 2324–2328, 1999.

[114] S. Akhlaghi, N. Zhou, and Z. Huang, “Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation”, *arXiv preprint arXiv:1702.00884*, 2017. DOI: `10.1109/pesgm.2017.8273755`.

[115] R. Burns, *Advanced control engineering.* Elsevier, 2001. DOI: `10.1515/9783110306637`.

[116] H. J. S. Feder, J. J. Leonard, and C. M. Smith, “Adaptive mobile robot navigation and mapping”, *The International Journal of Robotics Research*, vol. 18, no. 7, pp. 650–668, 1999. DOI: `10.1177/02783649922066484`.

[117] F. Preusker, F Scholten, K.-D. Matz, T. Roatsch, K. Willner, S. Hviid, J Knollenberg, L Jorda, P. J. Gutiérrez, E. Kührt, *et al.*, “Shape model, reference system definition, and cartographic mapping standards for comet 67P/Churyumov-Gerasimenko–Stereo-photogrammetric analysis of Rosetta/OSIRIS image data”, *Astronomy & Astrophysics*, vol. 583, A33, 2015. DOI: `10.1051/0004-6361/201526349`.

[118] R. W. Gaskell, O. S. Barnouin-jha, D. J. Scheeres, A. S. Konopliv, T Mukai, S Abe, J Saito, M Ishiguro, T Kubota, T Hashimoto, *et al.*, "Characterizing and navigating small bodies with imaging data", *Meteoritics & Planetary Science*, vol. 43, no. 6, pp. 1049–1061, 2008. DOI: `10.1111/j.1945-5100.2008.tb00692.x`.

[119] C. Capanna, G. Gesquière, L. Jorda, P. Lamy, and D. Vibert, "Three-dimensional reconstruction using multiresolution photoclinometry by deformation", *The Visual Computer*, vol. 29, no. 6-8, pp. 825–835, 2013. DOI: `10.1007/s00371-013-0821-5`.

[120] S. Bhaskaran, J. E. Riedel, and S. P. Synnott, "Autonomous target tracking of small bodies during flybys", *Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2004.*, 2004.

[121] C. Olson, R. P. Russell, and S. Bhaskaran, "Spin State Estimation of Tumbling Small Bodies", *The Journal of the Astronautical Sciences*, vol. 63, no. 2, pp. 124–157, 2016. DOI: `10.1007/s40295-015-0080-y`.

[122] C. Olson, "Sequential estimation methods for small body optical navigation", PhD thesis, 2016.

[123] H. Xiangyu, C. Hutao, and C. Pingyuan, "An autonomous optical navigation and guidance for soft landing on asteroids", *Acta Astronautica*, vol. 54, no. 10, pp. 763–771, 2004. DOI: `10.1016/j.actaastro.2003.09.001`.

[124] C. Cocaud and T. Kubota, "Autonomous navigation near asteroids based on visual SLAM", in *Proceedings of the 23rd International Symposium on Space Flight Dynamics, Pasadena, California*, 2012.

[125] S. Li, P. Cui, and H. Cui, "Autonomous navigation and guidance for landing on asteroids", *Aerospace science and technology*, vol. 10, no. 3, pp. 239–247, 2006. DOI: `10.1016/j.ast.2005.12.003`.

[126] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration", in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, IEEE, vol. 1, 2002, pp. 540–545. DOI: `10.1109/irds.2002.1041446`.

[127] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato, "FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements", *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 268–304, 2014. DOI: `10.1177/0278364913501564`.

[128] R. Valencia and J. Andrade-Cetto, "Active pose SLAM", in *Mapping, Planning and Exploration with Pose SLAM*, Springer, 2018, pp. 89–108. DOI: `10.1007/978-3-319-60603-3_5`.

[129] H. Carrillo and J. A. Castellanos, "Navigation Under Uncertainty Based on Active SLAM Concepts", in *Handling Uncertainty and Networked Structure in Robot Control*, Springer, 2015, pp. 209–235. DOI: `10.1007/978-3-319-26327-4_9`.

[130] K. J. Åström, "Optimal control of Markov processes with incomplete state information", *Journal of Mathematical Analysis and Applications*, vol. 10, no. 1, pp. 174–205, 1965. DOI: `10.1016/0022-247x(65)90154-x`.

[131] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable Markov processes over a finite horizon", *Operations research*, vol. 21, no. 5, pp. 1071–1088, 1973. DOI: `10.1287/opre.21.5.1071`.

[132] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains", *Artificial intelligence*, vol. 101, no. 1, pp. 99–134, 1998. DOI: `10.1016/s0004-3702(98)00023-x`.

[133] S. B. Broschart, M. Abrahamson, S. Bhaskaran, E. G. Fahnestock, R. R. Karimi, G. Lantoine, T. A. Pavlak, and L. Chappaz, "The small-body dynamics toolkit and associated close-proximity navigation analysis tools at JPL", in *2015 AAS Guidance and Control Conference, Breckenridge, Colorado*, 2015.

[134] T. A. Pavlak, S. B. Broschart, and G. Lantoine, "Quantifying mapping orbit performance in the vicinity of primitive bodies", 2015.

Colophon

This thesis was typeset with LaTeX and BibTeX, using a typographical look-and-feel created by Vincenzo Pesce. The style was inspired by D.A. Dei Tos `PhD_Dis` and by J. Stevens, L. Fossati `phdthesis` styles.